TU WIEN Informatics

Refining Labelled Systems for Modal and Constructive Logics with Applications

DISSERTATION

zur Erlangung des akademischen Grades

Doktor der Technischen Wissenschaften

eingereicht von

Timothy Stephen Lyon
Matrikelnummer 11701257

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Prof. Dr. Agata Ciabattoni
Zweitbetreuung: Dr. Revantha Ramanayake

Diese Dissertation haben begutachtet:

Alwen Tiu

Lutz Straßburger

Wien, 30. Juni 2021

Timothy Stephen Lyon

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

TU WIEN Informatics

Refining Labelled Systems for Modal and Constructive Logics with Applications

DISSERTATION

submitted in partial fulfillment of the requirements for the degree of

Doktor der Technischen Wissenschaften

by

Timothy Stephen Lyon
Registration Number 11701257

to the Faculty of Informatics

at the TU Wien

Advisor: Prof. Dr. Agata Ciabattoni
Second advisor: Dr. Revantha Ramanayake

The dissertation has been reviewed by:

Alwen Tiu

Lutz Straßburger

Vienna, 30th June, 2021

Timothy Stephen Lyon

Erklärung zur Verfassung der Arbeit

Timothy Stephen Lyon

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 30. Juni 2021

Timothy Stephen Lyon

Danksagung

Ich möchte meinen Eltern Lisa C. Lyon und Stephen W. Lyon, sowie meinen Großeltern Charlotte Chesnutt und Lee Chesnutt und dem Rest meiner Familie für ihre kontinuierliche Unterstützung und Beratung danken, ohne die diese Arbeit niemals möglich gewesen wäre. Des Weiteren bin ich Mr. William Ashby und Mrs. Vaughnene Ashby für ihre Großzügigkeit über all die Jahre sehr dankbar.

Meiner Betreuerin, Frau Prof. Agata Ciabattoni möchte ich von ganzem Herzen für ihre Beratung und Freundlichkeit danken, ohne die ich niemals so viel hätte erreichen können. Ich bin sehr dankbar für die Begleitung und das Wissen von Dr. Revantha Ramanayake. Ich danke Ihnen beiden, dass Sie sich die Zeit genommen haben, meine Arbeit zu überprüfen und mir wertvolles Feedback zu geben.

Ich bin dankbar für die sorgfältige Durchsicht meiner Arbeit und die hilfreichen Kommentare von Dr. Alwen Tiu und Dr. Lutz Straßburger. Des Weiteren möchte ich allen Teilnehmern des TICAMORE-Projekts meinen aufrichtigen Dank aussprechen.

Danke an Dr. Anna Prianichnikova und die gesamte Fakultät der DK LogiCS für die Leitung eines Doktorandenprogramms, das für uns Studenten von großem Nutzen ist. Ein besonderer Dank geht auch an Doris Hotz, die mir bei bürokratischen Angelegenheiten immer zur Seite stand.

Außerdem hatte ich das große Vergnügen, mit Prof. Nick Galatos an der University of Denver sowie mit Dr. Alwen Tiu und Prof. Rajeev Goré an der Australian National University zusammenzuarbeiten. Ich danke Ihnen für Ihre Gastfreundschaft und dafür, dass Sie Ihr Fachwissen mit mir geteilt haben.

Acknowledgements

I would like to thank my parents Lisa C. Lyon and Stephen W. Lyon, my grandparents Charlotte Chesnutt and Lee Chesnutt, and the rest of my family, for their continuous support and guidance, without whom this thesis would never be possible. Also, I will forever be indebted to Mr. William Ashby and Mrs. Vaughnene Ashby for their generosity throughout the years.

I would also like to whole-heatedly thank my supervisor Prof. Agata Ciabattoni for her mentorship and kindness, without whom I could have never achieved so much. I am also very thankful for the guidance and knowledge of Dr. Revantha Ramanayake. Thank you both for taking the time to review my thesis and provide me with valuable feedback.

I am grateful for the careful reviews of my thesis and useful comments given by Dr. Alwen Tiu and Dr. Lutz Straßburger. Furthermore, I would like to express my sincere thanks to all of the participants of the TICAMORE project.

Thank you to Dr. Anna Prianichnikova and all of the faculty of DK LogiCS for managing a PhD program that greatly benefits the lives of us students. Also, special thanks to Doris Hotz for always assisting me with bureaucratic matters.

I also had the great pleasure of working with Prof. Nick Galatos at the University of Denver, and Dr. Alwen Tiu and Prof. Rajeev Goré at the Australian National University. Thank you for your hospitality and for sharing your expertise with me.

Kurzfassung

In dieser Arbeit wird die *Methode der strukturellen Verfeinerung* (im Folgenden *Verfeinerung* genannt) vorgestellt, die dazu dient, die relationale (Kripke-)Semantik einer modalen und/oder konstruktiven Logik in ein "sparsames" Beweissystem zu transformieren, indem zwei beweistheoretische Paradigmen verbunden werden: gelabelte Sequenzen und verschachtelte Sequenzkalküle. Der Formalismus der gelabelten Sequenzen hat sich insofern bewährt, als dass schnittfreie Kalküle im Besitz wünschenswerter beweistheoretischer Eigenschaften (z. B. Zulässigkeit von Strukturregeln, Invertierbarkeit von Regeln, etc.) für große Klassen von Logiken automatisch generiert werden können. Trotz dieser Eigenschaften verwenden gelabelte Systeme eine komplizierte Syntax, die die Semantik der zugehörigen Logik explizit einbezieht, und solche Systeme verletzen normalerweise die Subformel-Eigenschaft in hohem Maße. Im Gegensatz dazu verwenden verschachtelte Sequenzkalküle eine einfachere Syntax und halten sich an eine strenge Lesart der Subformel-Eigenschaft, welche solche Systeme für den Entwurf von automatischen Schlussfolgerungsalgorithmen sinnvoll macht. Der Nachteil des Paradigmas der verschachtelten Sequenzen ist jedoch, dass eine allgemeine Theorie zur automatischen Konstruktion solcher Kalküle (wie im gelabelten Formalismus) im Wesentlichen fehlt, was bedeutet, dass die Konstruktion von verschachtelten Systemen und die Bestätigung ihrer Eigenschaften in der Regel auf einer Fall-zu-Fall-Basis erfolgt. Die Verfeinerungsmethode verbindet beide Paradigmen erfolgreich, indem sie gelabelte Systeme in geschachtelte (oder verfeinerte gelabelte) Systeme transformiert, wobei die Eigenschaften der Ersteren während des Transformationsprozesses erhalten bleiben. Die Eigenschaften von verschachtelten und verfeinerten gelabelten Systeme erleichtern die Arbeit mit ihnen, können zu einer Platzersparnis führen und eine gesteigerte Effizienz beim Automatisieren und Lösen von Argumentationsaufgaben bewirken (z. B. Beweissuche, effektive Interpolation, etc.).

Um die Methode der Verfeinerung und einige ihrer Anwendungen zu demonstrieren, betrachten wir eine vielfältige Gruppe von modalen und konstruktiven Logiken: kontextfreie Grammatiklogiken mit Konversen, intuitionistische Logiken erster Ordnung und deontische STIT -Logiken. Die vorgestellten verfeinerten gelabelten Kalküle werden verwendet, um die ersten Algorithmen zur Beweissuche und zur automatischen Extraktion von Gegenmodellen für deontische STIT -Logiken bereitzustellen und damit Entscheidungsprozeduren für die Logiken zu erhalten. Darüber hinaus verwenden wir unsere verfeinerten gelabelten Kalküle für kontextfreie Grammatiklogiken mit Konversen, um zu zeigen, dass jede Logik in der Klasse die effektive Lyndon-Interpolationseigenschaft besitzt.

Um dieses Ergebnis zu realisieren, verwenden wir ein syntaktisches, beweistheoretisches Verfahren der Lyndon-Interpolation.

Abstract

This thesis introduces the *method of structural refinement* (that will be referred to more simply as *refinement*), which serves as a means of transforming the relational (Kripke) semantics of a modal and/or constructive logic into an 'economical' proof system by connecting two proof-theoretic paradigms: labelled sequent and nested sequent calculi. The formalism of labelled sequents has been successful in that cut-free calculi in possession of desirable proof-theoretic properties (e.g. admissibility of structural rules, invertibility of rules, etc.) can be automatically generated for large classes of logics. Despite these qualities, labelled systems make use of a complicated syntax that explicitly incorporates the semantics of the associated logic, and such systems typically violate the subformula property to a high degree. By contrast, nested sequent calculi employ a simpler syntax and adhere to a strict reading of the subformula property, making such systems useful in the design of automated reasoning algorithms. However, the downside of the nested sequent paradigm is that a general theory concerning the automated construction of such calculi (as in the labelled setting) is essentially absent, meaning that the construction of nested systems and the confirmation of their properties is usually done on a case-by-case basis. The refinement method connects both paradigms in a fruitful way, by transforming labelled systems into nested (or, refined labelled) systems with the properties of the former preserved throughout the transformation process. The qualities of nested and refined labelled systems makes them easier to work with, can lead to a savings in space, and can bring about an increased efficiency in automating and solving reasoning tasks (e.g. proof-search, effective interpolation, etc.).

To demonstrate the method of refinement and some of its applications, we consider a varied group of modal and constructive logics: context-free grammar logics with converse, first-order intuitionistic logics, and deontic STIT logics. The introduced refined labelled calculi will be used to provide the first proof-search and automated counter-model extraction algorithms for deontic STIT logics, thus yielding decision procedures for the logics. Furthermore, we employ our refined labelled calculi for context-free grammar logics with converse to show that every logic in the class possesses the effective Lyndon interpolation property. In order to carry out this result, we make use of a syntactic, proof-theoretic method of Lyndon interpolation.

Contents

List of the Logics and Proof Systems Discussed

Logic	**Description**	**Page**
Km	Base grammar logic	p. 18
$\mathsf{Km}(S)$	The grammar logic of the CFCST system S	p. 18
IntQ	First-order intuitionistic logic with non-constant domains	p. 23
IntQC	First-order intuitionistic logic with constant domains	p. 23
DS_n^k	Deontic STIT logic for n agents with a maximum of k choices	p. 28

Calculus	**Description**	**Page**
$\mathsf{G3Km}(S)$	Labelled calculus for $\mathsf{Km}(S)$	p. 45
$\mathsf{G3IntQ}$	Labelled calculus for IntQ	p. 56
$\mathsf{G3IntQC}$	Labelled calculus for IntQC	p. 56
$\mathsf{G3DS}_n^k$	Labelled calculus for DS_n^k	p. 69
$\mathsf{Km}(S)\mathsf{L}$	Refined labelled calculus for $\mathsf{Km}(S)$	p. 98
IntQL	Refined labelled calculus for IntQ	p. 159
IntQCL	Refined labelled calculus for IntQC	p. 159
$\mathsf{DS}_n^k\mathsf{L}$	Refined labelled calculus for DS_n^k	p. 120
$\mathsf{DKm}(S)$	Nested calculus for $\mathsf{Km}(S)$	p. 110
DIntQ	Nested calculus for IntQ	p. 167
DIntQC	Nested calculus for IntQC	p. 167

CHAPTER 1

Introduction

Between 1932 and 1935, Gerhard Gentzen introduced the *natural deduction* and *sequent calculus* frameworks for classical and intuitionistic logic, which broke with the proof-theoretic paradigm of the time [Gen35a, Gen35b, vP18]. Up until that point, proof systems—based on the work of Frege, Peano, and Russell [Fre79, Pea89, Rus06]—consisted primarily of *axioms* and few inference rules (e.g. modus ponens and universal generalization). By contrast, Gentzen's formalisms consisted primarily of *rules*, and explicitly defined proofs as trees with assumptions, or trivial logical truths, as the leaves, and inference rules as the edges synthesizing the information via repeated applications into a concluding formula (i.e. the root of the tree)—taken to be the theorem derived [NVP11].

Both of Gentzen's formalisms possessed a significant advantage over the proof systems of the antecedent paradigm, namely, proofs constructed within Gentzen's natural deduction and sequent calculus frameworks were found to enjoy the so-called *subformula property* [Gen35a, Gen35b, Pra65, Rag65] (for a historical discussion, see [vP18]). The subformula property states that any theorem derivable within the deductive system is derivable with a proof consisting solely of subformulae of the derived theorem, that is to say, every formula used to reach the conclusion, occurs as a subformula of the conclusion. Typically, proofs and proof systems that possess this property (or a variation thereof) are qualified as *analytic*. This property is not only useful in establishing the consistency of a system's associated logic, but can be harnessed for other applications (e.g. decidability).

The subformula property, however, does not come for free, as the natural deduction systems contain elimination rules and the sequent calculi contain the *cut* rule, which delete formulae when going from the premises to the conclusion; thus, both frameworks *prima facie* break the subformula property. To overcome this obstacle in the sequent calculus framework, Gentzen proved the celebrated *Hauptsatz* (now called the *cut elimination theorem*) showing that any formula derivable in the sequent calculus (for classical and intuitionistic logic) is derivable *without the cut rule* [Gen35a, Gen35b]. The subformula property of the calculi followed as a corollary. Later work by Prawitz [Pra65] and

Raggio [Rag65] showed that natural deduction proofs could be *normalized*, i.e. put into a form where no formula occurs as the major premise of an elimination rule and as the conclusion of an introduction rule.

Due to its utility and simplicity, Gentzen's sequent formalism has continued to be implemented, leading to the creation (or, discovery) of analytic calculi for a wide variety of logics (e.g. [Cor89, Cur52, Gir87, Rau80, Ten87]), and leading even to the discovery of new logics (e.g. [Gir87]). Such calculi have found a range of applications: from establishing logical properties (e.g. consistency, decidability, interpolation), to automating reasoning with the associated logic (e.g. [Dyc92, Sla97]). In spite of these advantages, Gentzen's sequent formalism does have its drawbacks; namely, the formalism appears to be *too simple* to provide proof systems for the vast classes of non-classical logics studied by contemporary logicians. For example, despite effort to the contrary, a perspicuous and modular framework—in the style of Gentzen—that uniformly covers normal modal and tense logics (e.g. Kt, KB, and $\mathsf{S5}$) has proven to be elusive (see [Wan02]). The inability of the Gentzen sequent formalism to uniformly provide analytic calculi for large classes of non-classical logics is a serious limitation, as the study and application of such logics has grown substantially over the last few decades (see [Avr96]), implying the need for a suitable proof theory.

In this thesis, we will focus on refining labelled systems within the context of a particular group of non-classical logics, that is, *modal and constructive logics*. We will consider grammar logics [dCP88, DdN05], first-order intuitionistic logics [Kle52, Grz64], and deontic STIT logics [vBL19b, vBL21, Hor01, Mur04]; we now briefly describe these logics and their utility. Grammar logics are multi-modal logics that were introduced in [dCP88], and were used—in that paper—to establish an equivalence between the problem of checking if a grammar generates a word and the problem of checking if a formula is a logical theorem. The strong relationship between grammar logics and formal language theory has allowed for (un)decidability results to be transferred between the two settings [Bal00, dCP88, DdN05]. Moreover, the class of grammar logics is interesting in that it contains a large assortment of modal logics which can be viewed as epistemic logics [Bal00, HM92], temporal logics [dCH95], and even description logics (used in knowledge representation) with inverse roles and complex role inclusions axioms [HS04, TIG12]. First-order (and propositional) intuitionistic logics have been employed in the study of constructive reasoning and mathematics [TvD88], and are highly relevant in computer science; e.g. the Curry-Howard correspondence [How80], and answer-set programming [ONA04]. A set of logics that will be of particular interest in this work is the logic of STIT (an acronym for 'Seeing To It That') introduced by Belnap and Perloff in [BP90] to study and clarify agentive sentences. Since then, their formalism has been extended and applied to model agent-based choice making and interaction in the domain of epistemic reasoning [Bro11a], deontic reasoning [vBL19b, vBL21, Hor01, Mur04], the formal analysis of legal reasoning [Bro11a, LS15], and has been used to model and verify autonomous systems [SBA20]. Thus, the logics we consider are not only diverse in their nature, but in their applications.

In search of a suitable and uniform proof-theoretic framework for modal and constructive logics, a diverse number of formalisms have been assembled and proposed—examples include (prefixed) tableaux [Fit72, Fit14], display calculi [Bel82, Wan94], hypersequents [Pot83, Avr96], labelled sequents [Gab96, Neg05, Vig00], 2-sequents [Mas92, Mas93], nested sequents [Bul92, Brü09, Kas94, Pog09], and linear nested sequents [Lel15, LP15]. Each formalism extends or reworks the structure of the Gentzen-style sequent in a distinct way, incorporating additional bureaucracy that allows for proof-calculi to be constructed for certain classes of logics within the formalism, and which tend to uniformly possess fundamental proof-theoretic properties such as cut-admissibility, invertibility of rules, etc. (see, e.g. [GPT11, Neg05]). (NB. We will discuss the proof-theoretic formalisms relevant for, and used within, this thesis in more detail below.) As the number and diversity of proof-theoretic formalisms and systems increased, logicians began to ponder which criteria should be used to distinguish 'nice' proof systems from less desirable ones, leading to some notable proposals [Avr96, Doš88, Wan94].

One significant proposal, which we will consider, comes from Wansing [Wan94], who argues on philosophical and functional grounds, that proof systems ought to satisfy the following desiderata:

($\mathbf{W_1}$) *Separation*: Each logical rule exhibits no other logical connectives than the one to be introduced.

($\mathbf{W_2}$) *(Weak) Symmetry*: Each logical rule should be a left or right introduction rule, and symmetry stipulates that that the calculus is weakly symmetric and each logical connective has a left and right introduction rule.

($\mathbf{W_3}$) *(Weak) Explicitness*: Each introduced logical connective appears only in the conclusion of its corresponding logical rule, and explicitness holds if the calculus is weakly explicit and there is only one occurrence of the logical connective in the conclusion of each corresponding logical rule.

($\mathbf{W_4}$) *Unique Characterization*: Each logical connective should be uniquely characterized by its corresponding logical rules.

($\mathbf{W_5}$) *Došen's Principle*: "[T]he rules for the logical operations are never changed: all changes are made in the structural rules" [Doš88, p. 352].

($\mathbf{W_6}$) *Subformula property and cut-freedom*: The cut rule should be admissible and the system should possess the subformula property.

Desiderata ($\mathbf{W_1}$)–($\mathbf{W_3}$) fix the meaning of a logical connective independent of the other logical connectives present in the language, and ($\mathbf{W_4}$) ensures that two logical connectives collapse to the same connective if their logical rules are essentially identical (with the only distinguishing feature being that each set of logical rules introduces their own version of the connective). In addition, Wansing remarks ([Wan94, p. 129]) that if

($\mathbf{W_1}$)–($\mathbf{W_3}$) hold of a system and the cut rule is *admissible* (meaning that anything provable with cut, is provable without cut), then the subformula property holds for the system as a consequence. Došen's principle ($\mathbf{W_5}$) stipulates that the logical rules should fix a base system for a base logic (e.g. the modal logic K) with structural rule extensions yielding new systems for extensions of the base logic (e.g. the modal logics $\mathsf{S4}$ and $\mathsf{S5}$). If a formalism satisfies such a principle, then a high degree of modularity is obtained, allowing for a uniform proof-theoretic presentation for the corresponding set of logics. The last desideratum ($\mathbf{W_6}$) is desirable as the subformula property is useful in automated reasoning, and cut-freedom/elimination has relevance to the Curry-Howard correspondence [How80], in proving completeness, and in proving interpolation [Mae60], among other things.

Wansing's desiderata are mentioned not only due to their historical relevance in characterizing desirable proof systems, but due to their relevance in judging the systems obtained via the *method of structural refinement* (the central topic of this thesis), which we refer to more simply as *refinement* and which begets calculi that satisfy Wansing's proof-theoretic design principles to a high degree (though, we will argue below that Došen's principle is too strong and ought to be weakened). Refinement is intimately connected with the formalism of labelled sequents, and is a strategy for transforming such systems into systems of a 'simpler' formalism. Before making this strategy precise, we discuss the two main proof-theoretic formalisms that refinement concerns, namely, the labelled sequent and nested sequent formalisms.

The formalism of labelled sequents is rooted in the work of Kanger [Kan57], who introduced sequent calculi for the modal logics T, $\mathsf{S4}$, and $\mathsf{S5}$ that made use of *spotted formulae* (i.e. formulae annotated with natural numbers). This method of annotating, or prefixing, formulae with *labels* became the characteristic feature of the labelled paradigm, and is commonly used to incorporate semantic information into the syntax of the associated proof systems. There are numerous examples of calculi that fall within the labelled formalism, such as tableaux [BGM98, BG97, Gab96, Ner91], natural deduction systems [BMV97, Sim94, Vig00], and sequent-style systems [DN12, Min97, Neg05, NVP11, Sim94, Vig00]. We will be chiefly concerned with the latter—*labelled sequent systems*.

The labelled sequent formalism offers many advantages: first, for sizable classes of modal and constructive logics, it has been shown that the relational semantics of each logic can be straightforwardly transformed into an associated labelled sequent calculus [DN12, Neg05, NVP11, Sim94, Vig00]; in fact, it was shown that the process of constructing labelled sequent calculi can be automated [CMS13]. Second, calculi built within the labelled paradigm tend to be exceptionally modular, meaning that such systems allow for the addition or deletion of rules to obtain calculi for logics with desired properties. Third, general results have been given, which show that for a wide array of modal and constructive logics, their associated labelled calculi uniformly possess proof-theoretic properties such as height-preserving admissibility of structural rules, height-preserving invertibility of rules, and syntactic cut-elimination [DN12, Neg05, NVP11, Vig00]. Although such characteristics are certainly desirable, the labelled formalism does have its shortcomings.

For example, labelled calculi commonly involve a complicated syntax (due to the explicit incorporation of semantic information), labelled sequents encode general graphs (which contrasts with other formalisms that employ simpler data structures such as linear graphs [Lel15], or trees [Fit14, GR12]), and labelled systems usually contain larger sets of rules compared to other calculi (within different formalisms) for the same logics. The more complex data structures encoded in labelled sequents and the larger number of inference rules (which increases the size of proofs) can cause proof-search algorithms based on such systems to consume more spatial and temporal resources than is optimally required. Not only this, their complex nature can make labelled proof systems unwieldy both in use and in applications. Last, labelled calculi typically contain inference rules that delete formulae from premise to conclusion, thus violating the subformula property to a high degree—in contradiction with Wansing's desideratum ($\mathbf{W_6}$).

While labelled calculi customarily employ sequents that encode a general graph structure, nested sequents encode a *tree* of formulae, and therefore, employ a simpler data structure. The origin of the nested sequent formalism is often attributed to Bull [Bul92] and Kashima [Kas94], though it should be noted that the prefixed tableaux of Fitting [Fit72] can be viewed as 'upside-down' versions of nested sequent calculi. (NB. For a discussion on the relationship between prefixed tableaux and nested sequent calculi, see [Fit14]). In recent times, the use of nested sequents has become more widespread, with nested calculi constructed for normal modal logics [Brü09, Pog09], tense logics [GPT11], grammar logics [TIG12], intuitionistic modal logics [Str13], and (first-order) intuitionistic logics [Fit14]. Moreover, such calculi have been used in applications such as providing proof-search and counter-model construction algorithms for logics (e.g. [GPT11, TIG12]), and confirming interpolation as well as automating the extraction of interpolants (e.g. [FK15, LTGC20]). A strength of such calculi is that they employ a relatively simple data structure, minimizing the bureaucracy occurring in proofs and making the calculi better suited for applications (relative to labelled calculi). Despite these advantages, a major drawback of the nested formalism is that the construction of such calculi, and the confirmation of their proof-theoretic properties (e.g. cut-admissibility), is often done on a case-by-case basis. That is to say, the nested paradigm does not currently possess the same generality of results as those enjoyed in the labelled setting concerning the automatic construction of calculi in possession of fundamental properties.

Since the labelled formalism is well-suited for constructing calculi in possession of favorable proof-theoretic properties, and the nested formalism is simpler and better-suited for applications, a method of transforming calculi of the former formalism into the latter formalism—with proof-theoretic properties sufficiently preserved—is highly desirable. By connecting the two formalisms, we effectively obtain the best of both worlds: we may invoke the general results of the labelled paradigm to construct large classes of calculi for modal and constructive logics, and then transform these calculi into more *refined* versions of themselves (e.g. nested calculi) that are suitable for applications. We will see that such calculi utilize simpler structures, leading to a compression in proof size and allowing for uniform presentations of terminating proof-search. Similar relationships between labelled and ‘more refined’ systems have been discussed in the literature [GR12, Lyo20a, Lyo20b, LvB19, Pim18], where nested, tree-hypersequent, and ‘forestlike’ labelled calculi were derived from proper labelled calculi for modal and constructive logics.

The **first** component of the *method of refinement* consists of extracting labelled sequent calculi from the semantics of a class of logics for which one is interested. Such methods of extraction have been known for some time, with large classes of labelled sequent calculi being produced for large classes of modal and constructive logics [vBL19a, CS02, DN12, KO03, Neg05, Sim94, Vig00]. If the labelled calculi are of a certain form (as will be discussed in Ch. 4 and 5), then through the elimination of structural rules (encoding properties such as reflexivity, transitivity, etc.), we obtain classes of refined labelled calculi complete relative to their respective classes of logics. (NB. It will be seen that refined labelled calculi are ordinarily notational variants of nested sequent systems, though not necessarily so, as discussed in Sect. 4.2.)

The **second** step in the refinement procedure—*structural rule elimination*—proceeds by considering how logical rules in the labelled calculi ought to be strengthened to ensure the elimination of structural rules. The strengthened logical rules required to complete the second step are frequently found to be *propagation rules.* Such rules have been employed in prefixed tableaux [CdCGH97, Fit72], nested [TIG12], and labelled sequent systems [CLRT20, LvB19] in the literature, and acquire their name on the basis of their functionality—when applied bottom-up, the rules propagate formulae to the end of ‘paths’ occurring within sequents or tableaux. What is interesting however, is that the investigations offered in this thesis will also identify a new class of rules—*reachability rules.* These rules generalize the behavior of propagation rules by not only propagating formulae along paths, but additionally checking if data occurs along (potentially alternative) paths within a sequent. Reachability rules will be discussed in Ch. 5 on refining labelled calculi for first-order intuitionistic logics.

It should be noted that in the first-order setting structural rule elimination is composed with a further step in the refinement process—*domain atom removal.* Although structural rule elimination does yield ‘simpler’ proof calculi in the first-order setting, the syntax of sequents can be simplified further by showing the superfluity of certain syntactic structures that encode information about domains (hence, this step is unnecessary in the propositional setting where quantification over domains is absent).

The **last** component of refinement regards the establishment of proof-theoretic properties for the refined labelled calculi. This step can be carried out by either showing that the refined labelled calculi inherit the properties of their ‘parent’ labelled calculi, or by directly showing that each refined calculus possesses desirable properties without making reference to the original labelled systems that begat them. The former approach proceeds by reversing the structural rule elimination process to show that not only can the derivations in the parental labelled systems be transformed into derivations in the refined labelled systems, but the reverse transformation is possible as well. We will make use of both approaches in this thesis.

Furthermore, the calculi obtained via the method of refinement satisfy Wansing's desiderata completely (or to a high degree) with the only exception being Došen's principle ($\mathbf{W_5}$). Došen's principle implicitly equates modularity and uniform coverage of logics with the addition and subtraction of structural rules. Nevertheless, our refined labelled calculi will be highly modular and provide uniform coverage over large classes of logics by making use of *grammar theoretic machinery* within propagation and reachability rules, as opposed to using structural rules. This idea is motivated by and based upon the work in [CLRT20, GPT11, LvB19, TIG12]. As explained above, propagation and reachability rules operate by considering 'paths' within sequents, and then propagate formulae accordingly. Since we will encode these paths as strings generated by formal grammars, we can change the operation of our propagation and reachability rules by simply changing the formal grammar considered by the rule. This offers an alternative approach to the type of modularity expressed in Došen's principle, which has become widespread (for good reason) in the construction of proof systems. Still, the benefits of this alternative approach to modularity and uniform coverage are many: first, propagation and reachability rules are *formula driven*, meaning that bottom-up applications of rules only rely on logical formulae occurring within sequents (as opposed to other structures), typically making proof-search procedures easier to write. Second, systems built with grammar theoretic machinery tend to have fewer rules since certain structural rules are rendered superfluous—the omission of such structural rules brings about a compression in proof size and minimization of sequential structure. Third, exploiting formal grammars provides information about the complexity and (un)decidability of the associated logics by recognizing which formal grammars are required in a logic's propagation and reachability rules. Therefore, although the refined calculi we will obtain are in violation of Došen's principle, we argue that our calculi capture the essence of the principle, namely, modularity and uniform coverage of logics.

Beyond the construction of 'nice' proof systems, refining labelled systems also brings about an intriguing observation: many of the refined labelled calculi happen to be notational variants of existing nested systems. For example, in [CLRT20] it was found that refining labelled sequent systems for tense logics produced the nested sequent systems of [GPT11]; in [Lyo20a, Lyo20b] refining labelled systems for propositional and first-order intuitionistic logics yielded the nested systems of Fitting [Fit14]; and, as will be shown in Sect. 4.1, refining labelled calculi for grammar logics gives the nested calculi of [TIG12]. These observations suggest a naturalness to the method of refinement, which can be seen as an underlying procedure unifying various nested sequent systems within a single theoretical framework.

Although the thesis will primarily focus on the method of refinement, we will also put our refined systems to work, showing that the class of grammar logics considered admit effective Lyndon interpolation, and providing proof-search and counter-model extraction algorithms for a class of deontic STIT logics. The interpolation method is an additional, novel feature of this thesis and is based on the author's joint work in [LTGC20]. The interpolation method provides a purely syntactic and uniform approach to proving that logics possess the Lyndon (and Craig) interpolation property by harnessing their nested sequent systems, i.e. such systems can be used to show that for any valid implication $\phi \rightarrow \psi$, there exists a formula χ built with propositional atoms from ϕ and ψ (in Craig interpolation) and propositional atoms from ϕ and ψ with the same polarity (in Lyndon interpolation) such that $\phi \rightarrow \chi$ and $\chi \rightarrow \psi$ are valid; such properties have been used in verification [McM18], to establish Beth definability [KO10], and to conceal or forget information in ontology querying [LW11]. This method is both a generalization of Maehara's method [Mae60] and a variant of the successful semantic method of interpolation, initially provided by Fitting and Kuznets in [FK15], and expanded upon by the latter author in a sequence of papers [Kuz16b, Kuz16a, Kuz18, KL18].

The main contributions of this thesis are as follows: first, the refinement method is put forth, which transforms relational semantics into a nested or refined labelled proof system for a diverse class of logics. To illustrate this method, labelled calculi are constructed for grammar logics, first-order intuitionistic logics, and deontic STIT logics. The thesis dedicates many pages to explaining *how* propagation and reachability rules are discovered through the process of structural rule elimination, giving the thesis added explanatory value. Furthermore, comparisons are provided for labelled and refined labelled calculi, showing and justifying why the latter require less structure in their sequents compared to the former. Also, although the nested calculi for grammar logics (obtained via refinement) already exist in Tiu *et al.*'s paper [TIG12], the nested and refined labelled calculi for first-order intuitionistic logics and deontic STIT logics are new. Even though the nested calculi for first-order intuitionistic logics resemble Fitting's nested calculi to a degree (cf. [Fit14]), it will be argued that the use of propagation and reachability rules permit the calculi to be easily transformed into calculi for alternative logics. Last, the first proof-search and counter-model extraction algorithms for (single-agent) deontic STIT logics are provided, along with an application of a syntactic method of interpolation to uniformly prove that context-free grammar logics with converse have the effective Lyndon interpolation property, thus generalizing the results of [GN05], which proved effective Craig interpolation for regular grammar logics.

1.1 Outline of Dissertation

The dissertation is structured as follows:

In Ch. 2, we introduce the semantics and axiomatizations of context-free grammar logics with converse, first-order intuitionistic logics, and deontic STIT logics. Ch. 3 presents labelled calculi for the three classes of logics and argues that the calculi possess proof-theoretic properties such as hp-admissibility of structural rules, hp-invertibility of rules, and syntactic cut-elimination. In Ch. 4, the method of refinement is introduced and applied to the labelled calculi for grammar logics and deontic STIT logics. The first section of Ch. 4 will also show that all refined labelled calculi for grammar logics are notational variants of (slight reformulations of) the existing nested systems provided in [TIG12], and will touch on the relationship between such systems and display calculi. The second section of Ch. 4 will show how to refine the labelled calculi for deontic STIT logics. In Ch. 5, we will cover refinement in the first-order setting and will show that the refined labelled calculi obtained are labelled versions of nested systems. Ch. 6 discusses applications of refined labelled calculi, giving proof-search and counter-model extraction procedures for deontic STIT logics in the first section, followed by a uniform proof of effective Lyndon interpolation for grammar logics in the second section; also, we will briefly compare the syntactic method of interpolation from [LTGC20] and the semantic method from [FK15]. The last chapter (Ch. 7) concludes and discusses future work. Also, we note that all logics and proof systems discussed within the thesis are presented in tables prior to this introduction (p. xvii) along with short descriptions of each and the number of the page where the logic or proof system is introduced.

1.2 Publications

This dissertation is based on work from the following papers:

1. Ciabattoni, A., Lyon, T., & Ramanayake, R. (2018). From Display to Labelled Proofs for Tense Logics. In International Symposium on Logical Foundations of Computer Science (pp. 120-139). Springer, Cham.

2. Berkel, K., & Lyon, T. (2019). Cut-Free Calculi and Relational Semantics for Temporal STIT Logics. In European Conference on Logics in Artificial Intelligence (pp. 803-819). Springer, Cham.

3. Lyon, T., & Berkel, K. (2019). Automating Agential Reasoning: Proof-Calculi and Syntactic Decidability for STIT Logics. In International Conference on Principles and Practice of Multi-Agent Systems (pp. 202-218). Springer, Cham.

4. Lyon, T., Tiu, A., Goré, R., & Clouston, R. (2020). Syntactic Interpolation for Tense Logics and Bi-Intuitionistic Logic via Nested Sequents. In 28th EACSL Annual Conference on Computer Science Logic (CSL 2020). Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

5. Lyon, T. (2020). On Deriving Nested Calculi for Intuitionistic Logics from Semantic Systems. In International Symposium on Logical Foundations of Computer Science (pp. 177-194). Springer, Cham.

6. Lyon, T. (2021). On the Correspondence between Nested Calculi and Semantic Systems for Intuitionistic Logics. Journal of Logic and Computation. Oxford University Press.

7. Berkel, K. & Lyon, T. (2021). The Varieties of Ought-Implies-Can and Deontic STIT Logic. In: Fenrong Liu, Alessandra Marra, Paul Portner, and Frederik Van De Putte (eds.). Deontic Logic and Normative Systems: 15th International Conference (DEON2020/2021, Munich). London: College Publications.

8. Ciabattoni, A., Lyon, T., Ramanayake, R., & Tiu, A. (2021). Display to Labelled Proofs and Back Again for Tense Logics. ACM Transactions on Computational Logic (TOCL).

CHAPTER 2

Preliminaries for Modal and Constructive Logics

Contemporary modal logic is often traced back to the work of C.I. Lewis who attempted to resolve paradoxes of material implication via the formulation of strict implication [Lew18]. Languages of modal logics are characterized by their incorporation of *modalities*—expressions that qualify the truth of a proposition. Common examples of modalities (see [Gar13]) include *alethic* modalities such as "It is necessary that" (often denoted with $\Box$) and "It is possible that" (often denoted with $\Diamond$), *temporal* modalities such as "It will always be the case that" (often denoted by G) and "It has been the case that" (often denoted by P), and *deontic* modalities such as "It is obligatory that" (often denoted by $\mathcal{O}$) and "It is permissible that" (often denoted by $\mathcal{P}$). The extension of a propositional language with modalities makes for a more expressive language for modeling, though, a benefit of modal logics is that despite their increased expressivity over classical propositional logic, decidability commonly holds [Var97]. Moreover, such logics are usually equipped with a relational semantics, which evaluates formulae at points in a relational structure. It can be seen that modal logics possess many advantages then: such logics allow for the qualification of truth, permitting one to model phenomena whose logical consequences depend on such qualifications, such formalisms increase the expressivity of a language while retaining decidability, and their evaluation over relational structures (ubiquitous in mathematics and computer science) is of practical consequence [BdRV01].

Related to modal logics, intuitionistic logics employ a version of implication that is stronger than its classical counterpart, as well as a stronger version of universal quantification (in the first-order setting), both of which can be viewed as modalities. The advent of intuitionistic reasoning came in 1907/08 with the work of L.E.J. Brouwer, who put forth a philosophy of mathematics arguing that the truth of a mathematical statement rests upon a mental construction demonstrating its truth [BH75]. Motivated by the work of

Brouwer, axiomatic systems for propositional intuitionistic logic (based on Brouwer's intuitionism) were provided by Kolmogorov [Kol67], Orlov [Orl28], and Glivenko [Gli29], and a first-order system was given by Heyting [Hey30]. Such logics are essential in the field of constructive mathematics [TvD88], and have important applications in computer science [How80, ONA04].

In this thesis, we study the the refinement of labelled proof calculi for grammar logics [dCP88], first-order intuitionistic logics [Grz64, Hey30], and deontic STIT logics [Hor01, Mur04, vBL21]. These logics are similar enough to permit a parallel and uniform investigation of their proof theory while also possessing enough distinguishing features from each other to make the investigation interesting and to justify the generality of the refinement method. In addition, as mentioned in the introduction (Ch. 1) such logics have a wide variety of applications, thus allowing for the work in this thesis to have potential practical effect.

Each class of logics will be introduced accordingly in the three subsequent sections: we introduce context-free grammar logics with converse in Sect. 2.1, first-order intuitionistic logics in Sect. 2.2, and deontic STIT logics in Sect. 2.3. In each section, we define the language of each logic, supply a semantics, introduce fundamental concepts, and confirm soundness and completeness of each logic's axiomatization. We also adapt the method of canonical models (see [BdRV01]) in Sect. 2.3 to prove a new strong completeness result for our deontic STIT axiom systems.

2.1 Grammar Logics

Grammar logics are multi-modal logics that were introduced in 1988 by Fariñas del Cerro and Penttonen [dCP88]. In that paper, the authors established equivalences between the validity problem for certain classes of grammar logics and the problem of checking if a word (or, string) is generated from a formal grammar. This relationship between modal logic and formal language theory has allowed for (un)decidability results to be transferred between the two settings [BGM98, dCP88, DdN05]. Grammar logics have been widely studied [BGM98, dCP88, Dem01, DdN05, GN05, HS04, NS11] and are significant as they cover many well-known logics such as: description logics with complex role inclusion axioms and inverse roles [HS04], epistemic logics [FMHV95], information logics [Vak87], temporal logics [dCH95], and standard modal logics (e.g. K, $\mathsf{K4}$, T, B, $\mathsf{S4}$, and $\mathsf{S5}$ [DdN05]).

The language of grammar logics is defined relative to an *alphabet* $\sum$ consisting of a (non-empty) countable set of characters. Following [DdN05], we assume that $\sum$ can be partitioned into a *forward* part $\sum^{+} = \{a, b, c, \ldots\}$ and *backward* part $\sum^{-} = \{\overline{a}, \overline{b}, \overline{c}, \ldots\}$, of the same cardinality, satisfying the following:

$$\sum := \sum^{+} \cup \sum^{-} \text{ where } \sum^{+} \cap \sum^{-} = \emptyset \text{ , and } a \in \sum^{+} \text{ iff } \overline{a} \in \sum^{-} .$$

We use a, b, c, … (occasionally with subscripts) to denote the *forward characters* of $\sum^{+}$, $\overline{a}$, $\overline{b}$, $\overline{c}$, … . (occasionally with subscripts) to denote the *backward characters* of $\sum^{-}$, and refer to both forward and backward characters as *characters* more generally, using x, y, z, … to range over such characters in $\sum^{+} \cup \sum^{-} = \sum$. In other words, our alphabet $\sum$ consists of the concrete characters a, $\overline{a}$, b, $\overline{b}$, … but we use x, y, z, … as 'meta-characters' ranging over such concrete characters. This notation will be convenient in the following definitions and results, as it lets us reference characters in $\sum$ without specifying if they are forward or backward characters.

We define an involutory *converse* operation $\bar{\cdot}$ on characters that maps each forward character $a \in \sum^{+}$ to its *converse* $\overline{a} \in \sum^{-}$, and maps each backward character $\overline{a} \in \sum^{-}$ to its *converse* $a \in \sum^{+}$ (cf. [DdN05]). Notice that this operation is in fact an involution, since it satisfies the equation $\overline{\overline{x}} = x$. We stipulate that the symbol $\sum$ will be reserved to denote an alphabet for the remainder of the document. Using such an alphabet, we may define our language, which employs two modalities: $\langle x \rangle$ and $[x]$.[1] Similar to the interpretation of standard modalities (e.g. $\Diamond$ and $\Box$ [BdRV01]), $\langle x \rangle \phi$ is interpreted as saying that there exists an x successor state where ϕ holds, and $[x]\phi$ is interpreted as saying that ϕ holds in all x successor states; these interpretations are made explicit in Def. 5.

Definition 1 (The Language $\mathcal{L}_{\mathsf{Km}}(\sum)$)**.** *The language $\mathcal{L}_{\mathsf{Km}}(\sum)$ for our grammar logics is defined via the following grammar in BNF:*

$$\phi ::= p \mid \neg p \mid (\phi \vee \phi) \mid (\phi \wedge \phi) \mid \langle x \rangle \phi \mid [x]\phi$$

where p is among a denumerable set of propositional variables Prop $= \{p, q, r, \ldots\}$*, and $x \in \sum$. We use ϕ, ψ, χ, … (occasionally with subscripts) to denote formulae in $\mathcal{L}_{\mathsf{Km}}(\sum)$, and refer to formulae of the form p and $\neg p$ (with $p \in$* Prop*) as* literals*.*

The formulae in our language $\mathcal{L}_{\mathsf{Km}}(\sum)$ are given in negation normal form, meaning that applications of negations are restricted to propositional variables. This will simplify the structure of the sequents employed in our calculi as well as reduce the number of cases we need to consider when proving certain proof-theoretic results (see Sect. 3.2). Furthermore, it will be helpful to define the *complexity* of a formula ϕ from $\mathcal{L}_{\mathsf{Km}}(\sum)$, which corresponds to the number of binary connectives and modalities present in ϕ. This measure will occasionally be employed as a parameter in proofs by induction.

Definition 2 (Complexity of an $\mathcal{L}_{\mathsf{Km}}(\sum)$ Formula)**.** *We define the* complexity $|\phi|$ *of a formula $\phi \in \mathcal{L}_{\mathsf{Km}}(\sum)$ inductively as follows:*

- $|p| = |\neg p| = 0$
- $|\psi \wedge \chi| = |\psi \vee \chi| = max\{|\psi|, |\chi|\} + 1$

[1] We note that the language of common modal logics (e.g. K, $\mathsf{S4}$, and $\mathsf{S5}$) may be obtained by restricting our alphabet to a single character without its corresponding converse.

- $|[x]\psi| = |\langle x\rangle\psi| = |\psi| + 1$

We take the literals p and $\neg p$ (for each $p \in \mathrm{Prop}$), the binary operators $\wedge$ and $\vee$, and the modalities $[x]$ and $\langle x\rangle$ (for each $x \in \sum$) to be *duals* of each other. Using the notion of duality, we may define the negation $\neg\phi$ of a formula ϕ as the replacement of each literal, binary connective, and modality with its corresponding dual. The formal definition of negation is given below:

Definition 3 (Negation of a $\mathcal{L}_{\mathsf{Km}}(\sum)$ Formula)**.** *We define the* negation $\neg\phi$ *of a formula* $\phi \in \mathcal{L}_{\mathsf{Km}}(\sum)$ *in the usual way, inductively as follows:*

- *If* $\phi = p$*, then* $\neg\phi := \neg p$
- *If* $\phi = \neg p$*, then* $\neg\phi := p$
- *If* $\phi = \psi \wedge \chi$*, then* $\neg\phi := \neg\psi \vee \neg\chi$
- *If* $\phi = \psi \vee \chi$*, then* $\neg\phi := \neg\psi \wedge \neg\chi$
- *If* $\phi = [x]\psi$*, then* $\neg\phi := \langle x\rangle\neg\psi$
- *If* $\phi = \langle x\rangle\psi$*, then* $\neg\phi := [x]\neg\psi$

For example, if $\phi = [a]\neg p \wedge q$, then $\neg\phi = \langle a\rangle p \vee \neg q$. Negation lets us further define the logical constants ($\top$ and $\bot$) as well as classical implication and bi-implication ($\rightarrow$ and $\leftrightarrow$, resp.), which we will use later on (e.g. Def. 12). For the definition of $\top$ and $\bot$, we fix an arbitrary propositional variable $p \in \mathrm{Prop}$; the definitions are as follows:

$$\top := p \vee \neg p, \quad \bot := p \wedge \neg p, \quad \phi \rightarrow \psi := \neg\phi \vee \psi, \quad \text{and} \quad \phi \leftrightarrow \psi := (\phi \rightarrow \psi) \wedge (\psi \rightarrow \phi).$$

Due to our partitioning of $\sum$, our language consists of *forward modalities* $[a]$ and $\langle a\rangle$ where $a \in \sum^{+}$, and *backward modalities* $[\overline{a}]$ and $\langle\overline{a}\rangle$ where $\overline{a} \in \sum^{-}$. We say that these modalities are *converse* to one another since, as will be seen via the semantics of our language (Def. 5), each class of modalities is interpreted relative to accessibility relations that are converse to one another. We formally define, and make use of, the relational semantics for grammar logics provided in [DdN05] below. Note that these semantics are an extension and slight reformulation of the semantics for grammar logics that were given in Fariñas del Cerro and Penttonen's seminal paper [dCP88].

Definition 4 (Frames and Models for Grammar Logics [DdN05])**.** *A* $\sum$-frame *is an ordered pair* $F = (W, \{R_x \mid x \in \sum\})$ *such that (i)* W *is a non-empty set of worlds* w*,* u*,* v*, … and (ii) for each* $x \in \sum$*,* $R_x \subseteq W \times W$ *is a binary relation on* W *satisfying the* converse condition*:*

$$(\mathbf{C_1}) \qquad (w,u) \in R_x \textit{ iff } (u,w) \in R_{\overline{x}}.$$

A $\sum$-model *is a tuple* $M = (F, V)$ *such that* F *is a* $\sum$*-frame and* $V : \mathrm{Prop} \mapsto 2^{W}$ *is a* valuation function *mapping propositional variables from* Prop *to subsets of* W*.*

The converse condition $\mathbf{(C_1)}$ ensures that the backward modalities behave as expected and are truly converse to the forward modalities. That is to say, if we think of the 'forward' relations R_a (indexed with a forward character $a \in \sum^{+}$) as relating states to successor states, then the converse condition $\mathbf{(C_1)}$ ensures that the 'backward' relations $R_{\overline{a}}$ (indexed with a backward character $\overline{a} \in \sum^{-}$) relate states to predecessor states. This has the consequence that forward modalities such as $[a]$ and $\langle a \rangle$ reference truth in successor states, whereas the backward modalities $[\overline{a}]$ and $\langle \overline{a} \rangle$ reference truth in predecessor states. This semantic relationship between the forward and backward modalities is made explicit via the following definition:

Definition 5 (Satisfaction, Global Truth [DdN05])**.** *Let $M = (W, \{R_x \mid x \in \sum\}, V)$ be a $\sum$-model with $w \in W$, and define $R_x(w) := \{u \mid (w, u) \in R_x\}$ for $x \in \sum$. We define the* satisfaction *of a formula $\phi \in \mathcal{L}_{\mathsf{Km}}(\sum)$ on M at w (written $M, w \Vdash \phi$) inductively as follows:*

- *$M, w \Vdash p$ iff $w \in V(p)$;*
- *$M, w \Vdash \neg p$ iff $w \notin V(p)$;*
- *$M, w \Vdash \phi \wedge \psi$ iff $M, w \Vdash \phi$ and $M, w \Vdash \psi$;*
- *$M, w \Vdash \phi \vee \psi$ iff $M, w \Vdash \phi$ or $M, w \Vdash \psi$;*
- *$M, w \Vdash [x]\phi$ iff for all $u \in R_x(w)$, $M, u \Vdash \phi$;*
- *$M, w \Vdash \langle x \rangle \phi$ iff for some $u \in R_x(w)$, $M, u \Vdash \phi$.*

We say that a formula $\phi \in \mathcal{L}_{\mathsf{Km}}(\sum)$ is globally true *on M (written $M \Vdash \phi$) iff $M, w \Vdash \phi$ for all $w \in W$.*

As mentioned previously (and as is implied by the name), grammar logics connect concepts concerning formal grammars to logical concepts. Therefore, due to the intimate connection between formal grammars and grammar logics, we will introduce additional formal language theoretic concepts below. Such concepts will allow us to establish a correspondence between formal grammars—in particular, specific types of *Semi-Thue Systems* (Def. 8)—with properties imposed on $\sum$-frames (Def. 2.1), and to define new classes of grammar logics (Def. 12). Most definitions are taken from [DdN05].

Definition 6 (Strings over $\sum$)**.** *We let $\cdot$ represent the usual* concatenation *operation and let ε be the* empty string*. The set $\sum^*$ of* strings over $\sum$ *is defined to be the smallest set of strings such that:*

- $\sum \cup \{\varepsilon\} \subseteq \sum^*$
- *If $s, t \in \sum^*$, then $s \cdot t \in \sum^*$*

As usual, we define $s \cdot \varepsilon = \varepsilon \cdot s = s$ for $s \in \sum^*$, showing that the empty string ε is an identity element for the concatenation $\cdot$ operation. Furthermore, for strings $s, t \in \sum^*$, we let $s \cdot t := st$, that is, we will omit explicit mention of the concatenation operation, and simply stick strings together when performing concatenation, as is typically done.

We use s, t, r, ... (possibly annotated) to denote strings in $\sum^*$. Moreover, we extend the converse operation to strings as follows:

- $\overline{\varepsilon} := \varepsilon$
- *If $s = x_0 x_1 \cdots x_{n-1} x_n$, then $\overline{s} := \overline{x}_n \overline{x}_{n-1} \cdots \overline{x}_1 \overline{x}_0$*

Last, we define the length *of a string in $s \in \sum^*$ inductively as follows:*

- $|s| := 0$, *if $s = \varepsilon$*
- $|s| = |tx| := |t| + 1$, *if $s = tx$ with $t \in \sum^*$ and $x \in \sum$*

The context will easily determine if the notation $|\cdot|$ is being used to denote the complexity of a formula or the length of a string.

Since characters from $\sum$ are used to index the accessibility relations in a $\sum$-frame or $\sum$-model, strings from $\sum^*$ denote paths. We extend our relations R_x to relations R_s defined relative to strings below. Correspondingly, we define strings of modalities as follows: if $s = x_0 x_1 \cdots x_{n-1} x_n$, then $[s] = [x_0][x_1] \cdots [x_{n-1}][x_n]$ and $\langle s \rangle = \langle x_0 \rangle \langle x_1 \rangle \cdots \langle x_{n-1} \rangle \langle x_n \rangle$, and if $s = \varepsilon$, then $[s]\phi = \langle s \rangle \phi = \phi$.

Definition 7 (Generalized Relations)**.** *Let $F = (W, \{R_x \mid x \in \sum\})$ be a $\sum$-frame, $x \in \sum$, and $s \in \sum^*$. We extend the definition of an accessibility relation (defined in Def. 5) to strings and define R_s inductively as follows:*

- $R_\varepsilon := \{(w, w) \mid w \in W\}$;
- $R_{sx} := \{(w, u) \mid \exists v \in W, (w, v) \in R_s$ *and* $(v, u) \in R_x\}$.

Remark 1. *The above definition immediately implies that $(w, u) \in R_s$ iff $(u, w) \in R_{\overline{s}}$.*

As in [DdN05], we utilize a language-theoretic framework to define the supplementary frame conditions we will impose on $\sum$-frames. For example, we can encode the well-known symmetry frame condition $\forall w, u (R_a wu \rightarrow R_{\overline{a}} wu)$ by the production rule $\overline{a} \longrightarrow a$, or a version of three-to-one transitivity $\forall w, u, v, z (R_a wu \land R_b uv \land R_c vz \rightarrow R_d wz)$ (equivalently, represented as $\forall w, u (R_{abc} wu \rightarrow R_d wu)$) by the production rule $d \longrightarrow abc$. (NB. See Fig. 2.1 for a collection of common frame conditions and their associated production rules.) In general, a production rule of the form $x \longrightarrow s$ corresponds to a frame condition of the form $\forall w, u (R_s wu \rightarrow R_x wu)$.

Name	Frame Property	Production Rule
Reflexivity	$\forall w R_a ww$	$a \longrightarrow \varepsilon$
Symmetry	$\forall w, u(R_a wu \rightarrow R_{\overline{a}} wu)$	$\overline{a} \longrightarrow a$
Transitivity	$\forall w, v, u(R_a wv \land R_a vu \rightarrow R_a wu)$	$a \longrightarrow aa$
Euclideanity	$\forall w, v, u(R_{\overline{a}} vw \land R_a wu \rightarrow R_a vu)$	$a \longrightarrow \overline{a}a$

Figure 2.1: Common frame properties and their associated production rules.

To make such frame conditions and their consequences precise, we define a restricted version of a *Semi-Thue System* (cf. [Pos47]), i.e. we define a string rewriting system that is *context-free*—meaning the head of each production rule is a single character—and which is *closed*—meaning that the production rules are closed under converses (cf. [DdN05, TIG12]).

Definition 8 (CFCST System)**.** *We let $x \longrightarrow s$ represent a* production rule *that rewrites the character $x \in \sum$ to the string $s \in \sum^*$, and refer to x as the* head *and s as the* tail *of the production rule. A* context-free, closed, Semi-Thue (CFCST) System *is a set S of production rules satisfying the following* closure condition*:*

(C₂) $x \longrightarrow s \in S$ *iff* $\overline{x} \longrightarrow \overline{s} \in S.$

A CFCST system S re-writes strings in the following manner: if $x \longrightarrow s \in S$, then the string txr may be re-written as tsr. For example, if our CFCST system is $S = \{a \longrightarrow bb, \overline{a} \longrightarrow \overline{bb}, b \longrightarrow \overline{b}, \overline{b} \longrightarrow b\}$, then ab can be re-written as bbb or $a\overline{b}$ by applying the first and third production rule in S, respectively; that is, the strings bbb and $a\overline{b}$ can be derived from ab in one-step. As usual, through successive applications of the production rules, one can derive all possible strings produced by taking a string $s \in \sum^*$ as initial, thus generating a set of strings (i.e. language) determined on the basis of the CFCST system and s. Continuing the above example, we could derive the strings $bb\overline{b}$, abb, or $ab\overline{b}b$ by taking ab as initial. We make the above notions formally precise with the following definition:

Definition 9 (Derivation Relation, Language)**.** *Let S be a CFCST system. We define the* one-step derivation relation $\twoheadrightarrow_S$ *relative to S as follows: $s \twoheadrightarrow_S t$ iff there exist $s_0, s_1 \in \sum^*$, and $x \longrightarrow r \in S$, such that $s = s_0 x s_1$ and $t = s_0 r s_1$.*

We define the derivation relation $\twoheadrightarrow^*_S$ *to be the reflexive and transitive closure of the one-step derivation relation $\twoheadrightarrow_S$.*

The length of a derivation $s \twoheadrightarrow^*_S t$ *is defined to be equal to the minimal number of one-step derivations needed to derive t from s in S.*

Last, for each $s \in \sum^*$, we define the language of s relative to S *to be $L_S(s) := \{t \mid s \twoheadrightarrow^*_S t\}$.*

The definition below explains how CFCST systems are exploited to impose certain conditions on $\sum$-frames and models. This leads to a notion of S-validity for formulae (with S a CFCST system) that is provided in the ensuing definition.

Definition 10 (Production Rule Satisfaction). *Let the following ordered pair $F = (W, \{R_x \mid x \in \sum\})$ be a $\sum$-frame. A $\sum$-frame* satisfies a production rule *$a \longrightarrow s$ iff $R_s \subseteq R_a$.*

A $\sum$-frame satisfies a CFCST system *S iff it satisfies all production rules in S.*

Last, a $\sum$-model satisfies a production rule or CFCST system iff its underlying $\sum$-frame does.

Definition 11 (S-validity). *Let S be a CFCST system. We say that a formula $\phi \in \mathcal{L}_{\mathsf{Km}}(\sum)$ is S*-valid *iff for every $\sum$-model M, if M satisfies S, then $M \Vdash \phi$.*

A favorable feature of the grammar logics we are considering is that a set of S-validities (with S a CFCST system) is always axiomatizable. Below, we present an axiomatization and definition for the grammar logic $\mathsf{Km}(S)$, defined relative to a CFCST system S. Similar axiomatizations for grammar logics have been provided in the literature [dCP88, DdN05, TIG12], albeit, such axiom systems were provided for languages based on a different signature, whereas our axiomatization is suited for formulae in negation normal form.

Definition 12 (Axiomatization $\mathsf{HKm}(S)$). *Let S be a CFCST system with alphabet $\sum$. The axiomatization $\mathsf{HKm}(S)$ for the logic $\mathsf{Km}(S)$ is as follows:*

A0 All instances of (classical) propositional tautologies

A1 For all $x \in \sum$, $[x](\phi \rightarrow \psi) \rightarrow ([x]\phi \rightarrow [x]\psi)$

A2 For all $x \in \sum$, $\phi \rightarrow [x]\langle \overline{x} \rangle \phi$

A3 For all $x \longrightarrow s \in S$, $\langle s \rangle \phi \rightarrow \langle x \rangle \phi$

R0 $\dfrac{\phi \qquad \phi \rightarrow \psi}{\psi}$

R1 For all $x \in \sum$, $\dfrac{\phi}{[x]\phi}$

We define the logic $\mathsf{Km}(S)$ *to be the smallest set of formulae from $\mathcal{L}_{\mathsf{Km}}(\sum)$ that is closed under substitutions of the axioms and applications of the inference rules. Also, we refer to a logic $\mathsf{Km}(\emptyset)$ (relative to an alphabet $\sum$) as a* minimal grammar logic *and let $\mathsf{Km} := \mathsf{Km}(\emptyset)$. Last, we say that a formula $\phi \in \mathcal{L}_{\mathsf{Km}}(\sum)$ is a $\mathsf{Km}(S)$*-theorem *(written $\vdash_{\mathsf{Km}(S)} \phi$) iff $\phi \in \mathsf{Km}(S)$.*

The axioms A0 ensure that the logic $\mathsf{Km}(S)$ is an extension of classical propositional logic, and R0 is the well-known rule *modus ponens.* The A1 axioms are the typical *K axioms*, which, together with the R1 *necessitation* rules, ensure that all modalities are *normal.*[2] The A2 axioms encode the fact that the $[a]$ and $\langle a \rangle$ modalities are converses of the $[\overline{a}]$ and $\langle \overline{a} \rangle$ modalities, respectively. Last, the A3 axioms are determined by the CFCST system S and add an axiom—called a *path axiom* (cf. [CLRT20, TIG12])—for each production rule present in S. By standard methods for normal modal logics [BdRV01], the following soundness and completeness theorem for $\mathsf{Km}(S)$ can be shown:

Theorem 1 (Soundness and Completeness)**.** *For all $\phi \in \mathcal{L}_{\mathsf{Km}}(\sum)$, ϕ is S-valid iff $\vdash_{\mathsf{Km}(S)} \phi$.*

2.2 First-Order Intuitionistic Logics

First-order intuitionistic logic *proper* (also called *first-order intuitionistic logic with non-constant domains*) was axiomatized by Heyting early in the development of intuitionistic logic [Hey30]. Over three decades later, a variation of the logic, called *first-order intuitionistic logic with constant domains* was introduced by Grzgorczyk [Grz64], and axiomatized independently by Klemke [Kle69] and Görneman [Gör71]. Both logics admit a relational semantics, with Kripke providing the first relational semantics for first-order (and propositional) intuitionistic logic in 1965 [Kri65]. In this section, we will introduce the semantics and axiomatizations for both first-order intuitionistic logic with constant (IntQ) and non-constant domains (IntQC), and when reference is made to *first-order intuitionistic logics*, these two logics are taken to be the referents. Such logics will be of interest when refining labelled calculi since they will demonstrate how the method is applied in the first-order setting.

Definition 13 (The Language $\mathcal{L}_{\mathsf{Int}}$)**.** *The language $\mathcal{L}_{\mathsf{Int}}$ for first-order intuitionistic logics is defined via the following grammar in BNF:*

$$\phi ::= p(x_1, \ldots, x_n) \mid \bot \mid (\phi \vee \phi) \mid (\phi \wedge \phi) \mid (\phi \supset \phi) \mid (\exists x)\phi \mid (\forall x)\phi$$

where p is among a denumerable set of n-ary predicate symbols $\mathrm{Pred} = \{p, q, r, \ldots\}$ *and $x_1, \ldots, x_n, x$ (with $n \in \mathbb{N}$) are among a denumerable set of* variables $\mathrm{Var} = \{x, y, z, \ldots\}$.

We refer to formulae of the form $p(x_1, \ldots, x_n)$ as atomic formulae*, and also refer to formulae of the form p as* propositional variables *when $n = 0$, that is to say, a 0-ary predicate p is a propositional variable. We will use ϕ, ψ, χ, ... to denote formulae from $\mathcal{L}_{\mathsf{Int}}$, and define intuitionistic negation in the usual way as $\neg\phi := \phi \supset \bot$.*

As usual, we impose an operator precedence on our logical operators occurring in formulae of $\mathcal{L}_{\mathsf{Int}}$ as follows: $\exists$ and $\forall$ bind tighter than $\vee$ and $\wedge$, which bind tighter than $\supset$. For example, the formula $\forall x p(x) \vee q \supset r(x) \wedge q$ would be disambiguated as

[2] See [BdRV01] for a discussion and definition of *normal modal logics*.

$((\forall x p(x)) \vee q) \supset (r(x) \wedge q)$. Imposing a precedence lets us omit parentheses to improve the readability of formulae.

Another concept that will come in handy when studying and leveraging the proof theory of first-order intuitionistic logics is the *complexity* of a formula in $\mathcal{L}_{\mathsf{Int}}$. The complexity of a formula in $\mathcal{L}_{\mathsf{Int}}$ is equal to the number of unary and binary logical operators occurring in a formula, and is useful as it provides a well-founded measure that may be invoked to prove certain results by induction (e.g. the cut-elimination theorem for first-order intuitionistic logics, Thm. 10). The formal definition is given below:

Definition 14 (Complexity of an $\mathcal{L}_{\mathsf{Int}}$ Formula)**.** *We define the* complexity $|\phi|$ *of a formula* $\phi \in \mathcal{L}_{\mathsf{Int}}$ *inductively as follows:*

- $|p(x_1, \dots, x_n)| = |\bot| = 0$
- $|\psi \vee \chi| = |\psi \wedge \chi| = |\psi \supset \chi| = max\{|\psi|, |\chi|\} + 1$
- $|\exists x \psi| = |\forall x \psi| = |\psi| + 1$

A useful concept will be the notion of *free* and *bound variables* occurring in a formula ϕ. Intuitively, a variable x in ϕ is *free* iff it is not within the scope of a quantifier, and it is *bound* iff it is within the scope of a quantifier (cf. [GSS09]). The formal definition of free and bound variables is given below:

Definition 15 (Free and Bound Variables)**.** *We define the set $FV(\phi)$ of free variables in a formula $\phi \in \mathcal{L}_{\mathsf{Int}}$ inductively as follows:*

- $FV(p(x_1, \dots, x_n)) = \{x_1, \dots, x_n\}$
- $FV(\bot) = \emptyset$
- $FV(\psi \vee \chi) = FV(\psi \wedge \chi) = FV(\psi \supset \chi) = FV(\psi) \cup FV(\chi)$
- $FV(\forall x \psi) = FV(\exists x \psi) = FV(\psi) - \{x\}$

We say that a variable x is free *in ϕ iff $x \in FV(\phi)$. Similarly, we define the set $BV(\phi)$ of bound variables in a formula $\phi \in \mathcal{L}_{\mathsf{Int}}$ inductively as follows:*

- $BV(p(x_1, \dots, x_n)) = \emptyset$
- $BV(\bot) = \emptyset$
- $BV(\psi \vee \chi) = BV(\psi \wedge \chi) = BV(\psi \supset \chi) = BV(\psi) \cup BV(\chi)$
- $BV(\forall x \psi) = BV(\exists x \psi) = BV(\psi) \cup \{x\}$

We say that a variable x is bound *in ϕ iff $x \in BV(\phi)$.*

Note that the above definition allows for $FV(\phi) \cap BV(\phi) \neq \emptyset$, that is, variables are allowed to be both free and bound. For instance, the variable x is both free and bound in the formula $\forall x p(x) \wedge q(x)$. Occasionally, we will refer to the *occurrence* of a variable x as being *free* or *bound* in a formula ϕ, in which case, the occurrence of the variable will be exclusively free or bound.

As explained previously, Kripke provided a relational semantics for first-order intuitionistic logic with non-constant domains [Kri65]. Here, we put forth a variant of Kripke's semantics provided by Gabbay *et al.* in [GSS09] for both first-order intuitionistic logic with non-constant and constant domains (see Def. 20 for a definition of these logics).

Definition 16 (First-Order Intuitionistic Frames and Models [GSS09])**.** *We define an* IntQ-frame *to be a tuple $F = (W, \leq, D)$ such that:*

- *W is a non-empty set of worlds w, u, v, ...*
- *$\leq \subseteq W \times W$ is a reflexive and transitive binary relation on W.*[3]
- *D is a* domain function *mapping a world $w \in W$ to a non-empty set of objects $D_w = \{a, b, c, \dots\}$ satisfying the* nested domain condition *shown below:*

(ND) *If $a \in D_w$ and $w \leq v$, then $a \in D_v$.*

A IntQC-frame *is an* IntQ-frame *that additionally satisfies the following* constant domain condition *shown below:*

(CD) *If $a \in D_v$ and $w \leq v$, then $a \in D_w$.*

An IntQ-model *(*IntQC-model*) M is an ordered pair (F, V) where F is an* IntQ*-frame (*IntQC*-frame) and V is a* valuation function *such that $V(p, w) \subseteq (D_w)^n$ (with $n \in \mathbb{N}$) satisfying the following* monotonicity condition*:*

(M) *If $w \leq u$, then $V(p, w) \subseteq V(p, u)$.*

We uphold the convention in [GSS09] and assume that for each world $w \in W$, $(D_w)^0 = \{w\}$, so $V(p, w) = \{w\}$ or $V(p, w) = \emptyset$, for a propositional variable p.

As in [GSS09], we forgo the direct interpretation of formulae from $\mathcal{L}_{\mathsf{Int}}$ on IntQ- and IntQC-models, and instead, introduce D_w-sentences (see Def. 17 below). Defining satisfaction relative to D_w-sentences gives rise to notions of validity for formulae in $\mathcal{L}_{\mathsf{Int}}$ (Def. 18). Additionally, this notion depends on the *universal closure* of a formula, which is defined as follows: For $\phi \in \mathcal{L}_{\mathsf{Int}}$ such that $FV(\phi) = \{x_1, \dots, x_n\}$, the universal closure $\forall \vec{x} \phi$ is taken to be the formula $\forall x_1 \dots \forall x_n \phi$.

[3] The properties imposed on $\leq$ are defined as follows: (reflexivity) for all $w \in W$, $w \leq w$, and (transitivity) for all $w, u, v \in W$, if $w \leq v$ and $v \leq u$, then $w \leq u$.

Definition 17 (D_w-Sentence)**.** *Let $M = (W, \leq, D, V)$ be an* IntQ- *or* IntQC-*model with $w \in W$. We define the set of* parameters $\mathrm{Par}_w := \{\underline{a} \mid a \in D_w\}$. *The language $\mathcal{L}_{\mathsf{Int}}(D_w)$ is defined the same as $\mathcal{L}_{\mathsf{Int}}$, but with atomic formulae of the form $p(t_1, \ldots, t_n)$, where $p \in \mathrm{Pred}$ and $t_1, \ldots, t_n \in \mathrm{Var} \cup \mathrm{Par}_w$.*

*A D_w-*formula *is simply a formula in $\mathcal{L}_{\mathsf{Int}}(D_w)$, and a D_w-*sentence *is a D_w-formula that does not contain any free variables. (NB. Free variables are defined for formulae in $\mathcal{L}_{\mathsf{Int}}(D_w)$ in the same way as for formulae in $\mathcal{L}_{\mathsf{Int}}$, but with the exception that parameters from Par_w are ignored.)*

Last, it should be made explicit that we use $\underline{a}, \underline{b}, \underline{c}, \ldots$ to denote parameters corresponding to objects in $D_w = \{a, b, c, \ldots\}$.

Definition 18 (Satisfaction, Global Truth)**.** *Let $M = (W, \leq, D, V)$ be an* IntQ- *or* IntQC-*model with $w \in W$. The* satisfaction relation *$M, w \Vdash \phi$ between w and a D_w-sentence ϕ is inductively defined as follows:*

- $M, w \Vdash p(\underline{a}_1, \cdots, \underline{a}_n)$ *iff* $(a_1, \cdots, a_n) \in V(p, w)$*;*
- $M, w \nVdash \bot$*;*
- $M, w \Vdash \phi \vee \psi$ *iff* $M, w \Vdash \phi$ *or* $M, w \Vdash \psi$*;*
- $M, w \Vdash \phi \wedge \psi$ *iff* $M, w \Vdash \phi$ *and* $M, w \Vdash \psi$*;*
- $M, w \Vdash \phi \supset \psi$ *iff for all $u \in W$, if $w \leq u$ and $M, u \Vdash \phi$, then $M, u \Vdash \psi$;*
- $M, w \Vdash \forall x \phi$ *iff for all $u \in W$ and all $a \in D_u$, if $w \leq u$, then $M, u \Vdash \phi(\underline{a}/x)$;*
- $M, w \Vdash \exists x \phi$ *iff there exists an $a \in D_w$ such that $M, w \Vdash \phi(\underline{a}/x)$.*

We say that a formula ϕ is globally true *on M, written $M \Vdash \phi$, iff $M, u \Vdash \forall \vec{x} \phi$ for all worlds $u \in W$. A formula ϕ is* IntQ-valid *(*IntQC-valid*), written $\Vdash_{\mathsf{IntQ}} \phi$ ($\Vdash_{\mathsf{IntQC}} \phi$, resp.), iff it is globally true on all* IntQ-*models (*IntQC-*models).*

We note that in the intuitionistic setting the universal quantifier is stronger than the existential quantifier, that is, unlike in the classical setting the two operators fail to be interdefinable (see [Kri65]). The monotonicity condition **(M)**, together with the semantic clauses of Def. 18, necessitates a general form of monotonicity, detailed below:

Lemma 1 (General Monotonicity)**.** *Let M be an* IntQ- *or* IntQC-*model with $w, v \in W$ of M. For any D_w-sentence ϕ, if $M, w \Vdash \phi$ and $w \leq v$, then $M, v \Vdash \phi$.*

Proof. See [GSS09, Lem. 3.2.16] for details. □

Below (Def. 20), we provide axiomatizations for first-order intuitionistic logic with non-constant domains [GSS09, p. 119] and first-order intuitionistic logic with constant domains [GSS09, p. 136]. Both axiomatizations are extensions of the axiomatization for propositional intuitionistic logic which can be found in [GSS09, p. 6]. Axioms A9 and A10 make use of a *substitution* (y/x) of the variable y for the free variable x on a formula ϕ. We define $\phi(y/x)$ in the standard way below as the replacement of all free occurrences of x in ϕ with y, and define additional substitutions that will be used in the sequel. Also, the side condition *y is free for x in ϕ* is imposed on both axioms, and is taken to mean that y does not become bound by a quantifier if substituted for x in ϕ.[4]

Definition 19 (Substitutions)**.** *Let $\phi \in \mathcal{L}_{\mathsf{Int}}(D_w)$ and $t, t', s, t_1, \dots, t_n \in \mathrm{Var} \cup \mathrm{Par}_w$. We define the* substitution (t/s) *inductively as follows:*

- $s(t/t') := t$ *if* $t' = s$ *and* $s(t/t') := s$ *otherwise.*
- $p(t_1, \dots, t_n)(t/s) := p(t_1(t/s), \dots, t_n(t/s))$.
- $\bot(t/s) := \bot$.
- $(\psi \circ \chi)(t/s) := \psi(t/s) \circ \chi(t/s)$ *for* $\circ \in \{\vee, \wedge, \supset\}$.
- $(\forall x \psi)(t/s) := \forall x \psi(t/s)$ *and* $(\exists x \psi)(t/s) := \exists x \psi(t/s)$.

Definition 20 (Axiomatizations HIntQ and HIntQC)**.** *The axiomatization* HIntQ *for the logic* IntQ *is obtained by taking axioms A0–A12 and rules R0 and R1, whereas the axiomatization* HIntQC *for the logic* IntQC *is obtained by taking all axioms and inference rules.*

A0 $\phi \supset (\psi \supset \phi)$

A1 $(\phi \supset (\psi \supset \chi)) \supset ((\phi \supset \psi) \supset (\phi \supset \chi))$

A2 $\phi \supset (\psi \supset (\phi \wedge \psi))$

A3 $(\phi \wedge \psi) \supset \phi$

A4 $(\phi \wedge \psi) \supset \psi$

A5 $\phi \supset (\phi \vee \psi)$

A6 $\psi \supset (\phi \vee \psi)$

A7 $\bot \supset \phi$

A8 $(\phi \supset \chi) \supset ((\psi \supset \chi) \supset ((\phi \vee \psi) \supset \chi))$

A9 $\forall x \phi \supset \phi(y/x)$ *y free for x in* ϕ

A10 $\phi(y/x) \supset \exists x \phi$ *y free for x in* ϕ

A11 $\forall x(\psi \supset \phi) \supset (\psi \supset \forall x \phi)$ $x \notin FV(\psi)$

A12 $\forall x(\phi \supset \psi) \supset (\exists x \phi \supset \psi)$ $x \notin FV(\psi)$

A13 $\forall x(\phi \vee \psi) \supset \forall x \phi \vee \psi$ $x \notin FV(\psi)$

R0 $\dfrac{\phi \qquad \phi \supset \psi}{\psi}\ (mp)$

R1 $\dfrac{\phi}{\forall x \phi}$

[4] See [vD04, pp. 64–66] for a formal definition one variable being *free for* another variable in a formula.

We define the logic IntQ *(and* IntQC*) to be the smallest set of formulae from* $\mathcal{L}_{\mathsf{Int}}$ *that is closed under substitutions of the axioms and applications of the inference rules in* HIntQ *(*HIntQC*, resp.). We say that a formula* $\phi \in \mathcal{L}_{\mathsf{Int}}$ *is an* IntQ-theorem *(*IntQC-theorem*), written* $\vdash_{\mathsf{IntQ}} \phi$ *(*$\vdash_{\mathsf{IntQC}} \phi$*, resp.) iff* $\phi \in \mathsf{IntQ}$ *(*$\phi \in \mathsf{IntQC}$*, resp.).*

The axioms A0–A8, together with modus ponens (rule R0), is an axiomatization for propositional intuitionistic logic Int (cf. [GSS09, p. 6]). (NB. We define *propositional intuitionistic logic* Int to be the smallest set of formulae derivable from the axioms A0–A8 with rule R0.) The sole difference between the axiomatizations for IntQ and IntQC is that the former omits the *constant domain axiom* $\forall x(\phi \lor \psi) \supset \forall x\phi \lor \psi$ with $x \notin FV(\psi)$ (first introduced by A. Grzegorczyk; cf. [GSS09, p. 136]) whereas the latter includes it. The inclusion of this axiom in HIntQC is what causes the logic IntQC to be sound and complete relative to frames with constant domains. Soundness and completeness of each of the above systems can be found, for example, in [GSS09].

Theorem 2 (Soundness and Completeness)**.** *For any* $\phi \in \mathcal{L}_{\mathsf{Int}}$*,* $\Vdash_{\mathsf{IntQ}} \phi$ *(*$\Vdash_{\mathsf{IntQC}} \phi$*) iff* $\vdash_{\mathsf{IntQ}} \phi$ *(*$\vdash_{\mathsf{IntQC}} \phi$*, resp.).*

Proof. The forward direction follows from [GSS09, Cor. 6.2.21], [GSS09, Prop. 6.2.22], [GSS09, Prop. 7.2.9], and [GSS09, Prop. 7.3.6]. The backward direction is obtained from [GSS09, Lem. 3.2.31]. □

2.3 Deontic STIT Logics

Traditional multi-agent STIT logics (with STIT an acronym for ‘Seeing To It That’) have been used to model multi-agent choice making [BP90, BPX01, Hor01]. Such logics employ an atemporal *choice* operator $[i]$ expressing that ‘agent i sees to it that’ (some proposition is realized). For example, the formula $[i]$`door_Closed` might be interpreted as saying that ‘agent i sees to it that the apartment door is closed’, thus expressing that the proposition ‘the door is closed’ is realized via a choice available to (and selected by) the agent i.

Since their inception, STIT logics have been augmented with deontic notions to additionally allow for normative reasoning in multi-agent scenarios [BP90]. Numerous proposals have been put forth regarding the extension of STIT logics with deontic notions. For example, in [BPX01] traditional multi-agent STIT logics were extended with traditional deontic operators (e.g. “It is obligatory that,” “It is permissible that,” and “It is forbidden that” [Åqv84]), in [Hor01, Mur04] utilitarian deontic operators were discussed, and in [vBL19b, vBL21] an assortment of non-utilitarian deontic operators and their interrelations were analyzed. We make use of an agent-specific (non-utilitarian deontic) *obligation* operator $\otimes_i$, interpreted to mean that ‘it ought to be the case for agent i that’ (some proposition is realized). For example, the formula $\otimes_i$`door_Open` expresses that ‘it ought to be the case for agent i that the apartment door is open’, meaning that in an ideal world for agent i the apartment door would be open.

STIT logics and extensions thereof continue to receive considerable attention and have found a range of applications, being applied in epistemic [Bro11a], temporal [vBL19b], and legal [LS15] reasoning. Additionally, such logics have proven fruitful in the clarification of philosophical principles (e.g. [vBL21]) and in the verification of autonomous systems [SBA20].

We note that the deontic STIT logics introduced in this section are closely related to those of Murakami [Mur04]. Yet, rather than supplying a utilitarian, branching-time semantics that orders states of affairs according to an agent's preferences (as is done in [Mur04]), we supply a relational semantics indicating which possible worlds are optimal for an agent (similar to what is done in [vBL19b, vBL21]). This difference necessitates a proof of strong completeness for our axiom systems relative to the new relational semantics, which we prove via a canonical model construction (cf. [BdRV01]) in Sect. 2.3.2. First, however, we introduce our deontic STIT logics and associated preliminary concepts (Sect. 2.3.1).

2.3.1 Logical Preliminaries

The deontic STIT logics we consider employ a variety of modal operators to formalize reasoning about multi-agent choice making and agent dependent obligations. First off, such logics make use of a *settledness* operator $\Box$, which is prefixed to formulae expressing that a proposition is 'settled true' at a specific moment. As mentioned above, the language of each logic also includes a choice operator $[i]$ expressing that 'agent i sees to it that' and an obligation operator $\otimes_i$ expressing that 'it ought to be the case for agent i that'. The operators $\Diamond$, $\langle i \rangle$, and $\ominus_i$ are the duals of $\Box$, $[i]$, and $\otimes_i$, respectively.

Definition 21 (The Language $\mathcal{L}_{\mathsf{DS}^k_n}$). *We let* $\mathrm{Ag} := \{0, 1, \ldots, n\}$ *be our set of agents. The multi-agent language* $\mathcal{L}_{\mathsf{DS}^k_n}$ *(with* $n = |\mathrm{Ag}| \in \mathbb{N}$*) is defined via the following grammar in BNF:*

$$\phi ::= p \mid \neg p \mid (\phi \vee \phi) \mid (\phi \wedge \phi) \mid \Diamond\phi \mid \Box\phi \mid \langle i \rangle\phi \mid [i]\phi \mid \ominus_i \phi \mid \otimes_i \phi$$

where p *is among a denumerable set of propositional variables* $\mathrm{Prop} = \{p, q, r, \ldots\}$ *and* $i \in \mathrm{Ag}$*. We use* ϕ*,* ψ*,* χ*, ... (occasionally with subscripts) to denote formulae in* $\mathcal{L}_{\mathsf{DS}^k_n}$*, and refer to formulae of the form* p *and* $\neg p$ *(with* $p \in \mathrm{Prop}$*) as* literals.

As in the previous two sections, we introduce a measure on our logical formulae called the *complexity*, which counts the number of binary connectives and modalities occurring in a formula. We define this measure on logical formulae from $\mathcal{L}_{\mathsf{DS}^k_n}$ in the usual manner:

Definition 22 (Complexity of a $\mathcal{L}_{\mathsf{DS}^k_n}$ Formula). *We define the* complexity $|\phi|$ *of a formula* $\phi \in \mathcal{L}_{\mathsf{DS}^k_n}$ *inductively as follows:*

- $|p| = |\neg p| = 0$
- $|\psi \wedge \chi| = |\psi \vee \chi| = max\{|\psi|, |\chi|\} + 1$

- $|\Box\psi| = |\Diamond\psi| = |[i]\psi| = |\langle i\rangle\psi| = |\otimes_i \psi| = |\ominus_i \psi| = |\psi| + 1$

Our formulae in $\mathcal{L}_{\mathsf{DS}_n^k}$ are in negation normal form, which—as in the case of grammar logics (Sect. 2.1)—lets us simplify the sequents employed in our calculi and reduce the number of cases we need to consider when proving certain proof-theoretic results (see Sect. 3.4). We stipulate that p and $\neg p$ (for $p \in \text{Prop}$), $\wedge$ and $\vee$, $\Box$ and $\Diamond$, $[i]$ and $\langle i\rangle$ (for $i \in \text{Ag}$), and $\otimes_i$ and $\ominus_i$ (for $i \in \text{Ag}$) are duals of one another. By utilizing the notion of duality, we may define the negation $\neg\phi$ of a formula $\phi \in \mathcal{L}_{\mathsf{DS}_n^k}$ to be the formula obtained by replacing each literal and logical operator with its corresponding dual. For example, given that ϕ is the formula $\Box(\langle i\rangle\neg p \wedge \otimes_i q)$, $\neg\phi$ would be equal to $\Diamond([i]p \vee \ominus_i\neg q)$. The formal definition of negation is given below:

Definition 23 (Negation of a $\mathcal{L}_{\mathsf{DS}_n^k}$ Formula)**.** *We define the* negation $\neg\phi$ *of a formula* $\phi \in \mathcal{L}_{\mathsf{Km}}(\sum)$ *inductively as follows:*

- *If* $\phi = p$*, then* $\neg\phi = \neg p$
- *If* $\phi = \neg p$*, then* $\neg\phi = p$
- *If* $\phi = \psi \wedge \chi$*, then* $\neg\phi = \neg\psi \vee \neg\chi$
- *If* $\phi = \psi \vee \chi$*, then* $\neg\phi = \neg\psi \wedge \neg\chi$
- *If* $\phi = \Box\psi$*, then* $\neg\phi = \Diamond\neg\psi$
- *If* $\phi = \Diamond\psi$*, then* $\neg\phi = \Box\neg\psi$
- *If* $\phi = [i]\psi$*, then* $\neg\phi = \langle i\rangle\neg\psi$
- *If* $\phi = \langle i\rangle\psi$*, then* $\neg\phi = [i]\neg\psi$
- *If* $\phi = \otimes_i\psi$*, then* $\neg\phi = \ominus_i\neg\psi$
- *If* $\phi = \ominus_i\psi$*, then* $\neg\phi = \otimes_i\neg\psi$

Occasionally, we may also use the following abbreviations, where $p \in \text{Prop}$ is a fixed propositional variable:

$$\top := p \vee \neg p, \quad \bot := p \wedge \neg p, \quad \phi \to \psi := \neg\phi \vee \psi, \quad \text{and} \quad \phi \leftrightarrow \psi := (\phi \to \psi) \wedge (\psi \to \phi).$$

Due to the fact that our logic concerns instantaneous decision making, we make use of single moments in time where choices are made, as in [BHT08, vBL21, LvB19]. Therefore, we forgo the use of the (more complex) traditional branching time structures often employed in atemporal STIT logics [BPX01, Hor01]. Making use of this simplified semantics has the benefit that it simplifies our proof calculi (given in Sect. 3.4) and associated automated reasoning procedures (given in Sect. 6.1).

Definition 24 (Frames and Models for Deontic STIT Logics)**.** *For each* $i \in \text{Ag}$*, we let* $R_{[i]}(w) := \{v \in W \mid (w,v) \in R_{[i]}\}$*. A* DS_n^k-frame *is defined to be a tuple* $F = (W, \{R_{[i]} \mid i \in \text{Ag}\}, \{I_{\otimes_i} \mid i \in \text{Ag}\})$*, where* W *is a non-empty set of worlds* w*,* u*,* v*, … and the following hold:*

(S$_1$) *For all* $i \in \text{Ag}$*,* $R_{[i]} \subseteq W \times W$ *is an equivalence relation.*

(S$_2$) *For all* $u_1, ..., u_n \in W$*,* $\bigcap_{i \in \text{Ag}} R_{[i]}(u_i) \neq \emptyset$*.*

(S$_3$) *Let* $k > 0$*. For all* $i \in \text{Ag}$ *and* $w_0, w_1, \cdots, w_k \in W$*,*

$$\bigvee_{0 \leq m \leq k-1,\ m+1 \leq j \leq k} R_{[i]} w_m w_j.$$

($\mathbf{D_1}$) *For all* $i \in \mathrm{Ag}$, $I_{\otimes_i} \subseteq W$.
($\mathbf{D_2}$) *For all* $i \in \mathrm{Ag}$, $I_{\otimes_i} \neq \emptyset$.
($\mathbf{D_3}$) *For all* $i \in \mathrm{Ag}$ *and* $w, v \in W$, *if* $w \in I_{\otimes_i}$ *and* $v \in R_{[i]}(w)$, *then* $v \in I_{\otimes_i}$.

A DS_n^k-model *is a tuple* $M = (F, V)$ *where* F *is a frame and* $V : \mathrm{Prop} \to 2^W$ *is a valuation function mapping propositional variables to subsets of* W.

As in [BHT08, LvB19], W represents a single moment of possible worlds where agents from Ag are making decisions. Also, it should be noted that the parameters n and k (in DS_n^k) represent the number of agents in Ag and the maximum number of choices available to an agent in Ag at any given moment, respectively. Condition **($\mathbf{S_1}$)** ensures that each relation $R_{[i]}$ is a binary relation on W and partitions W into equivalence classes called *choice-cells*, which represents a set of possible worlds that an agent can realize via their choice at a moment. Condition **($\mathbf{S_2}$)**—referred to as the *independence of agents* condition [BPX01]—ensures that all choices made by agents are consistent, i.e., regardless of which choices are made, some state of affairs is realized. Condition **($\mathbf{S_3}$)** limits the number of choices available to an agent at a moment to a maximum of k, though we stipulate that if $k = 0$, then condition **($\mathbf{S_3}$)** is not enforced at all, and agents may have any number of choices available at a moment. The condition **($\mathbf{D_1}$)** ensures that all ideal worlds at the present moment for any agent i (which are those worlds contained in $I_{\otimes_i}$) are in fact possible worlds. Condition **($\mathbf{D_2}$)** ensures that at least one ideal world exists for an agent i at a moment, and **($\mathbf{D_3}$)** states that every ideal world extends to an ideal choice; in other words, **($\mathbf{D_2}$)** and **($\mathbf{D_3}$)** together ensure the existence of an ideal choice for each agent i at a given moment.

Below, we define *satisfaction*, *global truth*, and *validity* relative to these DS_n^k-frames and -models:

Definition 25 (Satisfaction, Global Truth, Validity). *Let* $M = (W, \{R_{[i]} \mid i \in \mathrm{Ag}\}, \{I_{\otimes_i} \mid i \in \mathrm{Ag}\}, V)$ *be a* DS_n^k*-model and let* $w \in W$. *We define the* satisfaction *of a formula* $\phi \in \mathcal{L}_{\mathsf{DS}_n^k}$ *on* M *at* w *(written* $M, w \Vdash \phi$*) inductively as follows:*

- $M, w \Vdash p$ *iff* $w \in V(p)$
- $M, w \Vdash \neg p$ *iff* $w \not\in V(p)$
- $M, w \Vdash \phi \wedge \psi$ *iff* $M, w \Vdash \phi$ *and* $M, w \Vdash \psi$
- $M, w \Vdash \phi \vee \psi$ *iff* $M, w \Vdash \phi$ *or* $M, w \Vdash \psi$
- $M, w \Vdash \Box\phi$ *iff for all* $u \in W$, $M, u \Vdash \phi$
- $M, w \Vdash \Diamond\phi$ *iff for some* $u \in W$, $M, u \Vdash \phi$
- $M, w \Vdash [i]\phi$ *iff for all* $u \in R_{[i]}(w)$, $M, u \Vdash \phi$

- $M, w \Vdash \langle i \rangle \phi$ *iff for some* $u \in R_{[i]}(w)$, $M, u \Vdash \phi$
- $M, w \Vdash \otimes_i \phi$ *iff for all* $u \in I_{\otimes_i}$, $M, u \Vdash \phi$
- $M, w \Vdash \ominus_i \phi$ *iff for some* $u \in I_{\otimes_i}$, $M, u \Vdash \phi$

A formula ϕ *is* globally true *on a model* M *(written* $M \Vdash \phi$*) iff* $M, u \Vdash \phi$ *for all* $u \in W$*. A formula* ϕ *is* DS_n^k-valid *(written* $\Vdash_{\mathsf{DS}_n^k} \phi$*) iff it is globally true on every* DS_n^k*-model. Last, we say that* X *semantically implies* ϕ *(written* X $\Vdash \phi$*) iff for all models* M *and worlds* w *in* M*, if* $M, w \Vdash \psi$ *for all* $\psi \in$ X*, then* $M, w \Vdash \phi$*.*

It is useful to observe that all modalities used in $\mathcal{L}_{\mathsf{DS}_n^k}$ are normal and that the conditions $\mathbf{(S_1)}$–$\mathbf{(S_3)}$ and $\mathbf{(D_1)}$–$\mathbf{(D_3)}$ fall within the class of *Kracht formulae* (cf. [BdRV01, Def. 3.58]). This implies that each (first-order) condition $\mathbf{(S_1)}$–$\mathbf{(S_3)}$ and $\mathbf{(D_1)}$–$\mathbf{(D_3)}$ corresponds to a modal axiom in the Sahlqvist class that is canonical for that property (cf. [BdRV01, Thm. 3.59]). We use this insight to provide sound and complete axiomatizations HDS_n^k for each logic DS_n^k:

Definition 26 (The Axiomatization HDS_n^k)**.** *The axiomatization* HDS_n^k *consists of all axioms and inference rules below, for all* $i \in$ Ag*.*

A0 *All instances of (classical) propositional tautologies*

A1 $\Box(\phi \rightarrow \psi) \rightarrow (\Box\phi \rightarrow \Box\psi)$

A2 $[i](\phi \rightarrow \psi) \rightarrow ([i]\phi \rightarrow [i]\psi)$

A3 $\otimes_i(\phi \rightarrow \psi) \rightarrow (\otimes_i\phi \rightarrow \otimes_i\psi)$

A4 $\Box\phi \rightarrow [i]\phi$

A5 $\Box\phi \rightarrow \otimes_i\phi$

A6 $\Box\phi \rightarrow \phi$

A7 $\Diamond\phi \rightarrow \Box\Diamond\phi$

A8 $[i]\phi \rightarrow \phi$

A9 $\langle i \rangle\phi \rightarrow [i]\langle i \rangle\phi$

A10 $\otimes_i\phi \rightarrow \ominus_i\phi$

A11 $\Diamond \otimes_i \phi \rightarrow \Box \otimes_i \phi$

A12 $\otimes_i\phi \rightarrow \otimes_i[i]\phi$

A13 $\Diamond[0]\phi_0 \wedge \cdots \wedge \Diamond[n]\phi_n \rightarrow \Diamond([0]\phi_0 \wedge \cdots \wedge [n]\phi_n)$

A14 $\Diamond[i]\phi_1 \land \Diamond(\neg\phi_1 \land [i]\phi_2) \land \cdots \land \Diamond(\neg\phi_1 \land \cdots \land \neg\phi_{k-1} \land [i]\phi_k) \rightarrow \phi_1 \lor \cdots \lor \phi_k$

R0 $\dfrac{\phi \qquad \phi \rightarrow \psi}{\psi}$

R1 $\dfrac{\phi}{\Box\phi}$

We define the logic DS_n^k *to be the smallest set of formulae from* $\mathcal{L}_{\mathsf{DS}_n^k}$ *that is closed under substitutions of the axioms and applications of the inference rules in* HDS_n^k*. We say that a formula* $\phi \in \mathcal{L}_{\mathsf{DS}_n^k}$ *is a* DS_n^k-theorem *(written* $\vdash_{\mathsf{DS}_n^k} \phi$*) iff* $\phi \in \mathsf{DS}_n^k$*.*

Axiom A0 ensures that each logic DS_n^k is an extension of classical propositional logic, and R0 is the familiar rule modus ponens. Axioms A1 - A5 and rule R1 ensure that all modalities are normal. Furthermore, the bridge axioms—A4 and A5—correspond to the fact that $R_{[i]} \subseteq W \times W$ in condition $(\mathbf{S_1})$ and that $I_{\otimes_i} \subseteq W$ in condition $(\mathbf{D_1})$, respectively. Axioms A6 - A9 encode the $\mathsf{S5}$ behavior of the $\Box$, $\Diamond$, $[i]$, and $\langle i \rangle$ modalities. Within the deontic logic community, axiom A10 is referred to as the *principle of deontic consistency* [vW51], whereas in the philosophical literature it is commonly called the *ought implies logical possibility* principle [Vra07]. The axiom corresponds to condition $(\mathbf{D_2})$ and establishes that whatever is obligatory is possible (i.e. consistent). Axiom A11 stipulates that whatever is obligatory from the perspective of a world in a model, is obligatory from the perspective of all worlds in the model, thus making every obligation settled-true at any given moment. Axiom A12 corresponds to condition $(\mathbf{D_3})$ and secures that if a state of affairs is obligatory, then there is a choice available to the agent that realizes the obligatory state of affairs. Axiom A13 is the well-known *independence of agents* axiom [BPX01], which coincides with condition $(\mathbf{S_2})$ and encodes the fact that any collection of choices made by the agents are consistent with one another. Last, axiom A14 corresponds to condition $(\mathbf{S_3})$ and limits the number of choices available to each agent at a moment to a maximum of k, though when $k = 0$, we omit this axiom from the axiomatization, thus placing no limit on the number of choices available to each agent. That is to say, when $k > 0$, axiom A14 is included in our axiomatization for each $i \in \text{Ag}$ and semantically corresponds to each agent being limited to a maximum of k choices at a moment, and when $k = 0$, no upper bound on choices is imposed.

Definition 27 (DS_n^k-derivable)**.** *We say that* ϕ is DS_n^k-derivable from a set of premises X*, written* $\text{X} \vdash_{\mathsf{DS}_n^k} \phi$*, iff there exist* $\psi_1, \ldots, \psi_m \in \text{X}$ *such that* $\psi_1 \land \cdots \land \psi_m \rightarrow \phi \in \mathsf{DS}_n^k$*.*

It is quick to verify that the notion of a formula being an element of a deontic STIT logic is equivalent to being derivable from an empty set of premises:

Proposition 1. *For any formula* $\phi \in \mathcal{L}_{\mathsf{DS}_n^k}$*,* $\vdash_{\mathsf{DS}_n^k} \phi$ *iff* $\emptyset \vdash_{\mathsf{DS}_n^k} \phi$*.*

A nice feature of our logics DS_n^k is that the notion of being derivable from a set of premises (Def. 27) coincides with the notion of being semantically implied by a set of premises

(Def. 25), that is, our logics are (strongly) sound and complete (cf. [BdRV01, Def. 4.10]). Soundness is straightforward, and is shown below. Completeness, on the other hand, requires more work, and is proven in the following section (Sect. 2.3.2) using a slight variation canonical model technique for normal modal logics [BdRV01, Ch. 4].

Theorem 3 (Soundness). *Let $\phi \in \mathcal{L}_{\mathsf{DS}^k_n}$ and $\mathrm{X} \subseteq \mathcal{L}_{\mathsf{DS}^k_n}$. If $\mathrm{X} \vdash_{\mathsf{DS}^k_n} \phi$, then $\mathrm{X} \Vdash \phi$.*

Proof. It is straightforward to show that if $\vdash_{\mathsf{DS}^k_n} \phi$, then $\Vdash_{\mathsf{DS}^k_n} \phi$ by showing that each axiom is DS^k_n-valid and that each inference rule of HDS^k_n preserves DS^k_n-validity. We use this fact to prove our claim and suppose that $\mathrm{X} \vdash_{\mathsf{DS}^k_n} \phi$. Then, by Def. 27, we know that there exist $\psi_1, \ldots, \psi_m \in \mathrm{X}$ such that $\psi_1 \wedge \cdots \wedge \psi_m \rightarrow \phi \in \mathsf{DS}^k_n$, i.e. $\vdash_{\mathsf{DS}^k_n} \psi_1 \wedge \cdots \wedge \psi_m \rightarrow \phi$. To prove the desired conclusion, let M be an arbitrary DS^k_n-model with $w \in W$ of M, and assume that for all $\chi \in \mathrm{X}$, $M, w \Vdash \chi$. It follows that $M, w \Vdash \psi_1 \wedge \cdots \wedge \psi_m$, implying that $M, w \Vdash \phi$ (since $\vdash_{\mathsf{DS}^k_n} \psi_1 \wedge \cdots \wedge \psi_m \rightarrow \phi$, which implies that $\Vdash_{\mathsf{DS}^k_n} \psi_1 \wedge \cdots \wedge \psi_m \rightarrow \phi$). Hence, $\mathrm{X} \Vdash \phi$. □

2.3.2 Completeness via Canonical Models

In this section, we prove the completeness of each logic DS^k_n by adapting and adjusting the canonical model construction for normal modal logics from [BdRV01] to our setting. We use the following notation throughout the course of the section: if X is a given set of formulae, then $\bigwedge \mathrm{X}$ represents a conjunction of all formulae (assuming X is finite) and $\heartsuit \mathrm{X}$ is taken to be the set of all formulae in X prefixed with a $\heartsuit$ modality (for $\heartsuit \in \{\Box, \Diamond, [i], \langle i \rangle, \otimes_i, \ominus_i\}$).

As is typical when proving completeness via canonical models for modal logics, we first define a notion of a *maximally consistent set* of formulae, and then show that such sets possess advantageous properties [BdRV01, Ch. 4]. We define such sets below and detail their properties in Lem. 2. Note that such sets will be used as worlds in our canonical models, and, since such sets are inherently founded upon syntactic notions (e.g. being DS^k_n-derivable Def. 27), they will serve as a point of connection between our syntactically defined logics DS^k_n and the semantics. Ultimately, this connection will be leveraged to prove completeness.

Definition 28 (DS^k_n-CS, DS^k_n-MCS). *A set $\mathrm{X} \subset \mathcal{L}_{\mathsf{DS}^k_n}$ is a DS^k_n* consistent set (DS^k_n-CS) *iff $\mathrm{X} \nvdash_{\mathsf{DS}^k_n} \bot$. We call a set $\mathrm{X} \subset \mathcal{L}_{\mathsf{DS}^k_n}$ a DS^k_n*-maximally consistent set (DS^k_n-MCS) *iff X is a DS^k_n-CS and for any set X' such that $\mathrm{X} \subset \mathrm{X}'$, $\mathrm{X}' \vdash_{\mathsf{DS}^k_n} \bot$.*

The following is similar to [BdRV01, Prop. 4.16].

Lemma 2. *Let X be a DS^k_n-MCS. Then, the following hold:*

(i) $\mathrm{X} \vdash_{\mathsf{DS}^k_n} \phi$ iff $\phi \in \mathrm{X}$;

(ii) $\phi \in \mathrm{X}$ *iff* $\neg\phi \notin \mathrm{X}$.

Proof. (i) For the forward direction, assume for a contradiction that $\mathrm{X} \vdash_{\mathsf{DS}^k_n} \phi$ and $\phi \notin \mathrm{X}$. By the latter assumption, $\mathrm{X} \cup \{\phi\} \vdash_{\mathsf{DS}^k_n} \bot$ by Def. 28, from which it follows that $\mathrm{X} \vdash_{\mathsf{DS}^k_n} \neg\phi$, contradicting our other assumption. The backward direction is trivial.

(ii) For the forward direction, assume that $\phi \in \mathrm{X}$ and $\neg\phi \in \mathrm{X}$. Then, $\mathrm{X} \vdash_{\mathsf{DS}^k_n} \bot$, contradicting the fact that X is consistent. For the backward direction, assume that $\neg\phi \notin \mathrm{X}$ and $\phi \notin \mathrm{X}$. By the first assumption, $\mathrm{X} \cup \{\neg\phi\} \vdash_{\mathsf{DS}^k_n} \bot$, implying that $\mathrm{X} \vdash_{\mathsf{DS}^k_n} \phi$, which by part (i), means that $\phi \in \mathrm{X}$, contradicting our second assumption and proving the desired claim. □

Using standard techniques, it is straightforward to show that every DS^k_n-CS can be extended to a DS^k_n-MCS. This fact will be frequently used.

Lemma 3 (Lindenbaum's Lemma)**.** *Every DS^k_n-CS can be extended to a DS^k_n-MCS.*

Proof. Similar to [BdRV01, Lem. 4.17]. □

By using DS^k_n-MCS's as building blocks, we define our canonical models *relative to a DS^k_n-MCS* X. The fact that our canonical models are defined *relatively*, is what contrasts our approach with the standard approach of building canonical models for normal modal logics [BdRV01, Ch. 4]. Typically, the models for normal modal logics integrate an explicit relation for each modality occurring in the language (e.g. $R_{[i]}$ is used to interpret $[i]$ and $I_{\otimes_i}$ is used to interpret $\otimes_i$ in $\mathcal{L}_{\mathsf{DS}^k_n}$). However, as explained in the previous section, we employ a simplified semantics for our DS^k_n logics, which forgoes the introduction of a relation for the $\Box$ and $\Diamond$ modalities. Instead, these modalities are interpreted relative to the set of worlds in a DS^k_n-model, which has two consequences. First, this interpretation of $\Box$ and $\Diamond$ causes the modalities to exhibit $\mathsf{S5}$ behavior, though, such behavior is not only acceptable, but desirable, as these operators are usually interpreted as $\mathsf{S5}$-type modalities in STIT logics [BPX01, BHT08, vBL19b, Mur04]. Second, since the interpretation of $\Box$ and $\Diamond$ depends on the set of worlds, and not on an explicitly associated relation, the standard canonical model definition—which defines a relation over the set of maximally consistent sets, used to interpret the corresponding modalities—is rendered insufficient. Therefore, we slightly adjust the canonical model definition given in [BdRV01, Ch. 4] to account for our simplified semantics.

Definition 29 (Canonical Model)**.** *Let* X *be a* DS^k_n*-MCS. We define the* canonical model (relative to X) *to be the tuple* $M^{\mathrm{X}} = (W^{\mathrm{X}}, \{R^{\mathrm{X}}_{[i]} \mid i \in \mathrm{Ag}\}, \{I^{\mathrm{X}}_{\otimes_i} \mid i \in \mathrm{Ag}\}, V^{\mathrm{X}})$ *such that:*

- $W^{\mathrm{X}} := \{w \mid w$ *is a* DS^k_n*-MCS and for all* $\Box\phi \in \mathcal{L}_{\mathsf{DS}^k_n}$*, if* $\Box\phi \in \mathrm{X}$*, then* $\phi \in w.\}$*;*
- *For all* $w, u \in W^{\mathrm{X}}$*,* $u \in R^{\mathrm{X}}_{[i]}(w)$ *iff for all* $[i]\phi \in w$*,* $\phi \in u$*;*

- *For all $w, u \in W^{\mathrm{X}}$, $w \in I^{\mathrm{X}}_{\otimes_i}$ iff for all $\otimes_i \phi \in u$, $\phi \in w$;*
- $V^{\mathrm{X}}(p) = \{w \in W^{\mathrm{X}} \mid p \in w\}$.

We now show that our canonical models possess favorable properties, which will be used in establishing both the Containment Lemma (Lem. 7) and Truth Lemma (Lem. 8)—both of which are crucial in verifying completeness (Thm. 4).

Lemma 4. *Let w and u be DS^k_n-MCS's. (i) $w \in W^{\mathrm{X}}$ iff for all $\phi \in \mathcal{L}_{\mathsf{DS}^k_n}$, if $\phi \in w$, then $\Diamond\phi \in \mathrm{X}$, (ii) For all $u, w \in W^{\mathrm{X}}$, $u \in R^{\mathrm{X}}_{[i]}(w)$ iff $\phi \in \mathcal{L}_{\mathsf{DS}^k_n}$, if $\phi \in u$, then $\langle i \rangle \phi \in w$, and (iii) For all $w, u \in W^{\mathrm{X}}$, $w \in I^{\mathrm{X}}_{\otimes_i}$ iff $\phi \in \mathcal{L}_{\mathsf{DS}^k_n}$, if $\phi \in w$, then $\ominus_i \phi \in u$.*

Proof. Let w and u be DS^k_n-MCS's. We prove claim (i); claims (ii) and (iii) are similar. (i) For the forward direction, suppose that $w \in W^{\mathrm{X}}$ and $\phi \in w$. Since w is a DS^k_n-MCS by Def. 29, our supposition implies that $\Box\neg\phi \not\in \mathrm{X}$ by Def. 29 as well. It follows that $\neg\Box\neg\phi \in \mathrm{X}$, yielding that $\Diamond\phi \in \mathrm{X}$. For the other direction, we assume that for all $\phi \in \mathcal{L}_{\mathsf{DS}^k_n}$, if $\phi \in w$, then $\Diamond\phi \in \mathrm{X}$, and further suppose that $\Box\psi \in \mathrm{X}$ for an arbitrary ψ. It follows that $\neg\Box\psi \not\in \mathrm{X}$, implying that $\Diamond\neg\psi \not\in \mathrm{X}$. Therefore, $\neg\psi \not\in w$, meaning that $\psi \in w$, showing that $w \in W^{\mathrm{X}}$. □

Lemma 5 (Existence Lemma)**.** *(i) For any world $w \in W^{\mathrm{X}}$, if $\Diamond\phi \in w$, then there exists a world $u \in W^{\mathrm{X}}$ such that and $\phi \in u$. (ii) For any world $w \in W^{\mathrm{X}}$, if $\langle i \rangle \phi \in w$, then there exists a world $u \in W^{\mathrm{X}}$ such that $u \in R^{\mathrm{X}}_{[i]}(w)$ and $\phi \in u$. (iii) For any world $w \in W^{\mathrm{X}}$, if $\ominus_i \phi \in w$, then there exists a world $u \in W^{\mathrm{X}}$ such that $u \in I^{\mathrm{X}}_{\otimes_i}$ and $\phi \in u$.*

Proof. Similar to [BdRV01, Lem. 4.20]. □

Lemma 6. *For all $w, u \in W^{\mathrm{X}}$, $\Diamond\phi \in w$ iff $\Diamond\phi \in u$.*

Proof. For the forward direction, suppose that $\Diamond\phi \in w$. Then, by Lem. 4-(i), we have $\Diamond\Diamond\phi \in \mathrm{X}$, which implies that $\Box\Diamond\phi \in \mathrm{X}$ since X is a DS^k_n-MCS. We therefore have that $\Diamond\phi \in u$. For the backward direction, the proof is almost identical. □

The following lemma ensures that any given canonical model is *contained* in our class of DS^k_n-models.

Lemma 7 (Containment Lemma)**.** *Let* X *be a DS^k_n-MCS. The canonical model M^{X} is a DS^k_n-model.*

Proof. Let X be a DS^k_n-MCS. Observe that $\{\phi \mid \Box\phi \in \mathrm{X}\} \subseteq \mathrm{X}$ due to the fact that X is a DS^k_n-MCS and $\Box\phi \rightarrow \phi$ is an axiom of HDS^k_n. This fact and Def. 29 ensures that $\mathrm{X} \in W^{\mathrm{X}}$, thus proving W^{X} non-empty. We now show that M^{X} satisfies properties $\mathbf{(S_1)}$–$\mathbf{(S_3)}$ and $\mathbf{(D_1)}$–$\mathbf{(D_3)}$. We confirm below that each property holds:

(S₁) The fact that $R^{\mathrm{X}}_{[i]} \subseteq W^{\mathrm{X}} \times W^{\mathrm{X}}$ follows from the definition of $R^{\mathrm{X}}_{[i]}$. To prove that $R^{\mathrm{X}}_{[i]}$ is an equivalence relation, we must show that (i) $R^{\mathrm{X}}_{[i]}$ is reflexive, and (ii) $R^{\mathrm{X}}_{[i]}$ is Euclidean. For (i), let w be an arbitrary world in W^{X}, and assume that $[i]\phi \in w$. Since w is a DS^k_n-MCS and $[i]\phi \to \phi$ is an axiom, we know that $w \vdash_{\mathsf{DS}^k_n} \phi$, which implies that $\phi \in w$ by Lem. 2-(i). By the definition of $R^{\mathrm{X}}_{[i]}$, this implies that $w \in R^{\mathrm{X}}_{[i]}(w)$. For (ii), let w, u, and v be arbitrary worlds in W^{X}, and assume that $u, v \in R^{\mathrm{X}}_{[i]}(w)$. We aim to show that $v \in R^{\mathrm{X}}_{[i]}(u)$. Suppose that $\phi \in v$. It follows that $\langle i \rangle\phi \in w$ by Lem. 4-(ii), and since $\langle i \rangle\phi \to [i]\langle i \rangle\phi$ is an axiom, we know that $w \vdash_{\mathsf{DS}^k_n} [i]\langle i \rangle\phi$, meaning that $[i]\langle i \rangle\phi \in w$ by Lem. 2-(i). Due to the fact that $u \in R^{\mathrm{X}}_{[i]}(w)$, and by the definition of $R^{\mathrm{X}}_{[i]}$ (Def. 29), we have $\langle i \rangle\phi \in u$. Thus, by Lem. 4-(ii), it follows that $v \in R^{\mathrm{X}}_{[i]}(u)$.

(S₂) Let $u_0, \ldots, u_n \in W^{\mathrm{X}}$. We aim to show that there exists a $v \in W^{\mathrm{X}}$ such that $v \in \bigcap_{i \in \mathrm{Ag}} R^{\mathrm{X}}_{[i]}(u_i)$. Let $v' := \bigcup_{i \in \mathrm{Ag}} \{\phi \mid [i]\phi \in u_i\} \cup \{\psi \mid \Box\psi \in \mathrm{X}\}$, and suppose that v' is inconsistent to derive a contradiction. Due to the inconsistency of v', we know that there exist $\phi_1, \ldots, \phi_k \in \bigcup_{i \in \mathrm{Ag}} \{\phi \mid [i]\phi \in u_i\}$ and $\psi_1, \ldots, \psi_m \in \{\psi \mid \Box\psi \in \mathrm{X}\}$ such that $\vdash_{\mathsf{DS}^k_n} \phi_1 \land \cdots \land \phi_k \to \neg\psi_1 \lor \cdots \lor \neg\psi_m$. Define $\Phi_i := \{\phi_l \mid [i]\phi_l \in u_i\} \cap \{\phi_1, \ldots, \phi_k\}$, and observe that for each $i \in \mathrm{Ag}$, $\bigwedge [i]\Phi_i \in u_i$, implying that $[i] \bigwedge \Phi_i \in u_i$, by modal reasoning and Lem. 2. Since $[i] \bigwedge \Phi_i \to \Diamond [i] \bigwedge \Phi_i$ is a theorem of HDS^k_n, it follows that $\Diamond [i] \bigwedge \Phi_i \in u_i$ for all $i \in \mathrm{Ag}$. Consequently, by Lem. 6, for any $i \in \mathrm{Ag}$, we have $\Diamond [0] \bigwedge \Phi_0, \ldots, \Diamond [n] \bigwedge \Phi_n \in u_i$, implying that $\Diamond [0] \bigwedge \Phi_0 \land \cdots \land \Diamond [n] \bigwedge \Phi_n \in u_i$ since u_i is a DS^k_n-MCS. By the independence of agents axiom, $\Diamond([0] \bigwedge \Phi_0 \land \cdots \land [n] \bigwedge \Phi_n) \in u_i$ for any $i \in \mathrm{Ag}$. By Lem. 5-(i), it follows that there exists a world $u' \in W^{\mathrm{X}}$ such that $[0] \bigwedge \Phi_0 \land \cdots \land [n] \bigwedge \Phi_n \in u'$, which further implies that $\bigwedge \Phi_1 \land \cdots \land \bigwedge \Phi_n \in u'$ by modal reasoning and the fact that $[i] \bigwedge \Phi_i \to \bigwedge \Phi_i$ is an axiom instance for each $i \in \mathrm{Ag}$. Hence, $\bigwedge \Phi_1 \land \cdots \land \bigwedge \Phi_n \in u'$, so by the definition of each Φ_i and the fact that $\vdash_{\mathsf{DS}^k_n} \phi_1 \land \cdots \land \phi_k \to \neg\psi_1 \lor \cdots \lor \neg\psi_m$, we have that $u' \vdash_{\mathsf{DS}^k_n} \neg\psi_1 \lor \cdots \lor \neg\psi_m$. Since $u' \in W^{\mathrm{X}}$, we know that $\{\psi \mid \Box\psi \in \mathrm{X}\} \subseteq u'$, which implies that $u' \vdash_{\mathsf{DS}^k_n} \psi_1 \land \cdots \land \psi_m$, giving a contradiction. It follows that v' is a DS^k_n-CS. By Lem. 3, v' can be extended to a DS^k_n-MCS v such that $\{\psi \mid \Box\psi \in \mathrm{X}\} \subseteq v$. Therefore, $v \in W^{\mathrm{X}}$ and for all $[i]\phi \in u_i$, $\phi \in v$, so, by Def. 29, $v \in R^{\mathrm{X}}_{[i]}(u_i)$ for all $i \in \mathrm{Ag}$, which implies the desired result.

(S₃) For a contradiction, suppose that W^{X} contains $k' > k$ choice-cells, this is, there exist worlds $w_0, \ldots w_{k'} \in W^{\mathrm{X}}$ such that $w_j \not\in R^{\mathrm{X}}_{[i]}(w_m)$ for $0 \leq m \leq k' - 1$ and $m + 1 \leq j \leq k'$. Let us first consider the world w_0. By our assumption, we have that $w_j \not\in R^{\mathrm{X}}_{[i]}(w_0)$ for $1 \leq j \leq k'$. This, in conjunction with Def. 29, implies that there exists formulae $[i]\phi_{0,1}, \ldots, [i]\phi_{0,k'} \in w_0$ such that $\phi_{0,1} \not\in w_1$, …, $\phi_{0,k'} \not\in w_{k'}$. Let us define $\bigwedge \Phi_{0,1 \leq j \leq k'} := \phi_{0,1} \land \cdots \land \phi_{0,k'}$. By Def. 2, the fact that w_0 is a DS^k_n-MCS, by modal reasoning, and by the fact that $\phi_{0,j} \not\in w_j$ for $1 \leq j \leq k'$ (entailing $\neg\phi_{0,j} \in w_j$ for $1 \leq j \leq k'$), we know that:

$$[i](\bigwedge \Phi_{0,1 \leq j \leq k'}) \in w_0 \text{ and } \neg \bigwedge \Phi_{0,1 \leq j \leq k'} \in w_j \text{ for } 1 \leq j \leq k'.$$

We continue in the above fashion for the worlds w_m with $1 \leq m \leq k$. By our assumptions, $[i]\phi_{m,j} \in w_m$ and $\phi_{m,j} \not\in w_j$ for $m + 1 \leq j \leq k'$. Furthermore, we let $\bigwedge \Phi_{m,m+1 \leq j \leq k'} :=$

$\phi_{m,m+1} \wedge \cdots \wedge \phi_{m,k'}$. Similar to the w_0 case above, Def. 2, the fact that w_m is a DS_n^k-MCS, by modal reasoning, and by the fact that $\phi_{m,j} \not\in w_j$ for $m+1 \leq j \leq k'$ (meaning that $\neg\phi_{m,j} \in w_j$ for $m+1 \leq j \leq k'$), we have for each m (with $1 \leq m \leq k$) that:

$$[i](\bigwedge \Phi_{m,m+1 \leq j \leq k'}) \in w_m \text{ and } \neg \bigwedge \Phi_{m,m+1 \leq j \leq k'} \in w_j \text{ for } 1 \leq j \leq k'.$$

Without loss of generality, we now consider the world $w_{k'}$. By Lem. 6 and the above facts, the following large conjunction is an element of $w_{k'}$:

$$\begin{aligned}
&\Diamond[i](\bigwedge \Phi_{0,1 \leq j \leq k'}) \wedge \\
&\Diamond\Big(\neg\bigwedge \Phi_{0,1 \leq j \leq k'} \wedge [i]\Big(\bigwedge \Phi_{1,2 \leq j \leq k'})\Big) \wedge \\
&\Diamond\Big(\neg\bigwedge \Phi_{0,1 \leq j \leq k'} \wedge \neg\bigwedge \Phi_{1,2 \leq j \leq k'} \wedge [i]\Big(\bigwedge \Phi_{2,3 \leq j \leq k'})\Big) \wedge \\
&\vdots \\
&\Diamond\Big(\neg\bigwedge \Phi_{0,1 \leq j \leq k'} \wedge \cdots \wedge \neg\bigwedge \Phi_{k-1,k \leq j \leq k'} \wedge [i](\bigwedge \Phi_{k,k \leq j \leq k'})\Big)
\end{aligned}$$

Therefore, by axiom A14, we have the following:

$$\bigwedge \Phi_{0,1 \leq j \leq k'} \vee \cdots \vee \bigwedge \Phi_{k,k \leq j \leq k'} \in w_{k'}$$

Observe that $\phi_{m,k'} \not\in w_{k'}$ for $0 \leq m \leq k$, implying that $\neg\phi_{m,k'} \in w_{k'}$ for $0 \leq m \leq k$. This fact, in conjunction with the definition of $\bigwedge \Phi_{m,m+1 \leq j \leq k'}$ for $0 \leq m \leq k$, by Lem. 2, and by modal reasoning, it follows that $\neg\bigwedge \Phi_{m,m+1 \leq j \leq k'} \in w_{k'}$ for $0 \leq m \leq k$. Hence, by modal reasoning,

$$\neg\bigwedge \Phi_{0,1 \leq j \leq k'} \wedge \cdots \wedge \neg\bigwedge \Phi_{k,k \leq j \leq k'} \in w_{k'},$$

which gives a contradiction. Therefore, it must be the case that the number of choice-cells in M^{X} is a maximum of k.

($\mathbf{D_1}$) Follows from the definition of $I^{\mathrm{X}}_{\otimes_i}$ (Def. 29).

($\mathbf{D_2}$) Fix an $i \in$ Ag. We want to show that there exists a $v \in W^{\mathrm{X}}$ such that $v \in I^{\mathrm{X}}_{\otimes_i}$. Let $v' := \bigcup_{w \in W^{\mathrm{X}}} \{\psi \mid \otimes_i \psi \in w\} \cup \{\phi \mid \Box\phi \in \mathrm{X}\}$ and assume that v' is inconsistent to derive a contradiction. Since v' is inconsistent, we know that $\vdash_{\mathsf{DS}_n^k} \chi_1 \wedge \cdots \wedge \chi_m \rightarrow \bot$. Define $\Phi := \{\chi_1, \ldots, \chi_m\} \cap \{\phi \mid \Box\phi \in \mathrm{X}\}$, and $\Psi_w := \{\chi_1, \ldots, \chi_m\} \cap \{\psi \mid \otimes_i \psi \in w\}$. It follows that $\vdash_{\mathsf{DS}_n^k} \bigwedge \Phi \wedge \bigwedge \Psi_{w_1} \wedge \cdots \wedge \bigwedge \Psi_{w_k} \rightarrow \bot$ for some $w_1, \ldots, w_k \in W^{\mathrm{X}}$. Using modal reasoning, we obtain the following:

$$\vdash_{\mathsf{DS}^k_n} \quad \otimes_i \bigwedge \Phi \land \otimes_i \bigwedge \Psi_{w_1} \land \cdots \land \otimes_i \bigwedge \Psi_{w_k} \to \otimes_i \bot \tag{2.1}$$

$$\vdash_{\mathsf{DS}^k_n} \quad \Box \bigwedge \Phi \land \otimes_i \bigwedge \Psi_{w_1} \land \cdots \land \otimes_i \bigwedge \Psi_{w_k} \to \otimes_i \bot \tag{2.2}$$

$$\vdash_{\mathsf{DS}^k_n} \quad \Box \bigwedge \Phi \land \otimes_i \bigwedge \Psi_{w_1} \land \cdots \land \otimes_i \bigwedge \Psi_{w_k} \to \ominus_i \bot \tag{2.3}$$

$$\vdash_{\mathsf{DS}^k_n} \quad \Box \bigwedge \Phi \land \otimes_i \bigwedge \Psi_{w_1} \land \cdots \land \otimes_i \bigwedge \Psi_{w_k} \to \bot \tag{2.4}$$

$$\vdash_{\mathsf{DS}^k_n} \quad \Box\Box \bigwedge \Phi \land \Box \otimes_i \bigwedge \Psi_{w_1} \land \cdots \land \Box \otimes_i \bigwedge \Psi_{w_k} \to \Box \bot \tag{2.5}$$

$$\vdash_{\mathsf{DS}^k_n} \quad \Box \bigwedge \Phi \land \Box \otimes_i \bigwedge \Psi_{w_1} \land \cdots \land \Box \otimes_i \bigwedge \Psi_{w_k} \to \bot \tag{2.6}$$

$$\vdash_{\mathsf{DS}^k_n} \quad \bigwedge \Box \Phi \land \bigwedge \Box \otimes_i \Psi_{w_1} \land \cdots \land \bigwedge \Box \otimes_i \Psi_{w_k} \to \bot \tag{2.7}$$

$$\vdash_{\mathsf{DS}^k_n} \quad \bigwedge \Box \Phi \land \bigwedge \Diamond \otimes_i \Psi_{w_1} \land \cdots \land \bigwedge \Diamond \otimes_i \Psi_{w_k} \to \bot \tag{2.8}$$

We explain each of the step in the above deduction: 2.2 follows from 2.1 by the bridge axiom A5 (i.e. $\Box\phi \to \otimes_i\phi$), 2.3 follows from 2.2 by the deontic consistency axiom A10 (i.e. $\otimes_i\phi \to \ominus_i\phi$), 2.4 and 2.5 follow from 2.3 by modal reasoning, 2.6 follows from 2.5 by modal reasoning and the fact that $\Box$ is an $\mathsf{S5}$ modality (by axioms A6 and A7), 2.7 follows from 2.6 by modal reasoning, and last, 2.8 follows from 2.7 by axiom A11 (i.e. $\Diamond \otimes_i \phi \to \Box \otimes_i \phi$). Now, observe that $\Box\Phi, \Diamond \otimes_i \Psi_{w_1}, \ldots, \Diamond \otimes_i \Psi_{w_k} \subseteq \mathrm{X}$ (by the definition of v' and Lem. 4-(i)), which implies that $\mathrm{X} \vdash_{\mathsf{DS}^k_n} \bot$. Since X is a DS^k_n-MCS, this gives a contradiction, and shows that v' is consistent. By Lem. 3, v' may be extended to a DS^k_n-MCS v. Since $\{\phi \mid \Box\phi \in \mathrm{X}\} \subseteq v$, we know that $v \in W^{\mathrm{X}}$, and further, because $\{\psi \mid \otimes_i \psi \in w\} \subseteq v$ for each $w \in W^{\mathrm{X}}$, it follows that $v \in I^{\mathrm{X}}_{\otimes_i}$.

(D₃) Let $w, u \in W^{\mathrm{X}}$ and $i \in \mathrm{Ag}$. Assume that $w \in I^{\mathrm{X}}_{\otimes_i}$ and $u \in R^{\mathrm{X}}_{[i]}(w)$. We aim to show that $u \in I^{\mathrm{X}}_{\otimes_i}$. To prove this, we let v be an arbitrary world in W^{X} and suppose that $\otimes_i\phi \in v$, with the aim of showing that $\phi \in u$. Since v is a DS^k_n-MCS and $\otimes_i\phi \to \otimes_i[i]\phi$ is an axiom instance, it follows that $\otimes_i[i]\phi \in v$. By Def. 29 and the assumption that $w \in I^{\mathrm{X}}_{\otimes_i}$, we have that for all $v' \in W^{\mathrm{X}}$ and each $\otimes_i\psi \in v'$, $\psi \in w$. This fact, in conjunction with our supposition above, implies that $[i]\phi \in w$. By Def. 29 and the assumption that $u \in R^{\mathrm{X}}_{[i]}(w)$, we have that $\phi \in u$, which entails the desired result. □

The following is similar to [BdRV01, Lem. 4.21].

Lemma 8 (Truth Lemma)**.** *For any $\phi \in \mathcal{L}_{\mathsf{DS}^k_n}$, $M^{\mathrm{X}}, w \Vdash \phi$ iff $\phi \in w$.*

Proof. We prove the result by induction on the complexity of ϕ. The base case trivially follows from the definition of V^{X} (Def. 29), so we focus on the inductive step. We show the $\lor$, $\Box$, $[i]$, and $\otimes_i$ cases since the dual cases are similar.

Inductive step. We consider each connective in turn.

($\lor$) $M^{\mathrm{X}}, w \Vdash \psi \lor \chi$ iff $M^{\mathrm{X}}, w \Vdash \psi$ or $M^{\mathrm{X}}, w \Vdash \chi$ (by Def. 25) iff $\psi \in w$ or $\chi \in w$ (by IH) iff $\psi \lor \chi \in w$ (by the fact that w is a DS^k_n-MCS).

($\Box$) For the forward direction, suppose that $M^{\mathrm{X}}, w \Vdash \Box\psi$. It follows that for all $u \in W^{\mathrm{X}}$, $M^{\mathrm{X}}, u \Vdash \psi$ (by Def. 25), which implies that for all $u \in W^{\mathrm{X}}$, $\psi \in u$ (by IH). By Lem. 5-(i), $\Box\psi \in w$. For the backward direction, assume that $\Box\psi \in w$ and let $u \in W^{\mathrm{X}}$. By Lem. 4-(i), $\psi \in u$, which implies that $M^{\mathrm{X}}, u \Vdash \psi$ by IH. Since u was arbitrary, it follows that $M^{\mathrm{X}}, w \Vdash \Box\psi$ by Def. 25.

($[i]$) For forward direction, assume that $M^{\mathrm{X}}, w \Vdash [i]\psi$. By Def. 25 and IH, for all $u \in W^{\mathrm{X}}$, if $u \in R_{[i]}(w)$, then $\psi \in u$. By Lem. 5-(ii), $[i]\psi \in w$. For the backward direction, suppose that $[i]\psi \in w$ and let $u \in W^{\mathrm{X}}$ such that $u \in R_{[i]}(w)$. By Def. 29, $\psi \in u$, which implies that $M^{\mathrm{X}}, u \Vdash \psi$ by IH. We may conclude that $M^{\mathrm{X}}, w \Vdash [i]\psi$.

($\otimes_i$) For the forward direction, assume that $M^{\mathrm{X}}, w \Vdash \otimes_i\psi$. By Def. 25 and IH, for all $u \in I^{\mathrm{X}}_{\otimes_i}$, $\psi \in u$. By Lem. 5-(iii), $\otimes_i\psi \in w$. For the backward direction, suppose that $\otimes_i\psi \in w$ and let $u \in I^{\mathrm{X}}_{\otimes_i}$. By Def. 29, $\psi \in u$, implying that $M^{\mathrm{X}}, u \Vdash \psi$ by IH. Therefore, $M^{\mathrm{X}}, w \Vdash \otimes_i\psi$ since u was an arbitrary world in W^{X}. □

Theorem 4 (Completeness)**.** *Let* $\phi \in \mathcal{L}_{\mathsf{DS}^k_n}$ *and* $\mathrm{X} \subseteq \mathcal{L}_{\mathsf{DS}^k_n}$. *If* $\mathrm{X} \Vdash \phi$, *then* $\mathrm{X} \vdash_{\mathsf{DS}^k_n} \phi$.

Proof. We prove the claim by contraposition. Suppose that $\mathrm{X} \nvdash_{\mathsf{DS}^k_n} \phi$. Then, it follows that $\mathrm{X} \cup \{\neg\phi\}$ is consistent. By Lem. 3, we may extend this set to a DS^k_n-MCS X'. By Lem. 8 and the fact that $\neg\phi \in \mathrm{X}'$, we have that $M^{\mathrm{X}'}, \mathrm{X}' \Vdash \psi$ for all $\psi \in \mathrm{X}$, and $M^{\mathrm{X}'}, \mathrm{X}' \Vdash \neg\phi$. Hence, $\mathrm{X} \nVdash \phi$ by Def. 25. □

CHAPTER 3

Labelled Systems for Modal and Constructive Logics

In this chapter, we introduce labelled sequent systems for context-free grammar logics with converse (Sect. 3.2), for first-order intuitionistic logics (Sect. 3.3), and for deontic STIT logics (Sect. 3.4). In each of these sections, we define the sequents and structures used in each class of labelled calculi, prove that such systems possess desirable properties (e.g. hp-invertibility of rules and syntactic cut-elimination), and confirm soundness and completeness relative to the classes of logics we are considering. In the first section (Sect. 3.1), we discuss preliminary concepts of labelled sequent systems, as well as touch on the history of such systems and significant results obtained in the labelled paradigm.

3.1 Preliminaries and Related Work

Labelled sequent systems extend the structure of Gentzen-style systems by explicitly incorporating labels and semantic elements into the syntax of sequents. This idea—of integrating labels directly into the syntax of sequents—stretches back to the work of Kanger [Kan57], who introduced sequent systems for modal logics based on *spotted formulae*. Since then, numerous labelled sequent systems have been provided for large classes of modal and constructive logics [CS02, DN12, Gab96, KO03, Min97, Neg05, Vig00].

The paradigm of labelled sequents has proven itself successful in producing modular calculi that uniformly cover extensive classes of logics, even when other proof-theoretic formalisms fail to do so, that is, through the addition or deletion of structural rules, a labelled sequent system for one logic may be converted into a labelled sequent system for another logic within a given class. In addition to uniform coverage and modularity, general results exist ([DN12, Neg05, Sim94, Vig00]) showing that labelled sequent systems share the same fundamental properties; e.g. height-preserving admissibility of contractions and syntactic cut-elimination. Perhaps one of the most beneficial characteristics however, is the ease with which labelled calculi are constructed. In essence, the semantic clauses and frame properties within a logic's relational semantics are transformed into inference rules, yielding a sound and complete calculus for the associated logic, which precisely demonstrates the semantic reasoning utilized in deriving a theorem. As was explained in the introduction (Ch. 1), this characteristic of the labelled paradigm, when composed with the process of refinement, effectively supplies a method for transforming the semantics of a logic into a calculus of economical structure that is also in possession of desirable properties and is well-suited for applications. After briefly introducing Viganò's labelled sequent formalism below (which is predated by Simpson's labelled sequent formalism for modal intuitionistic logics [Sim94]), we will illustrate how semantic information is transformed into inference rules by considering the semantic clause for the standard $\Box$ modality.

In [Vig00], Viganò demonstrated the utility of the labelled formalism and provided labelled sequent calculi for a substantially broad class of modal, constructive, and relevance logics. The logics Viganò considers are founded upon relational semantics, and allow for the accessibility relation(s) to satisfy any number of Horn formulae of the form:

$$\forall \vec{x}\Big(R_i t_1^1 \ldots t_n^1 \land \cdots \land R_i t_1^m \ldots t_n^m \rightarrow R_i t_1^0 \ldots t_n^0\Big)$$

where R_i is an n-ary relational symbol and each term t_j^k is built from the variables in $\vec{x}$ along with function symbols [Vig00, p. 61]. Viganò's systems employ two types of labelled sequents: (i) labelled sequents of the form $\mathcal{R}, \Gamma \Rightarrow \Delta$, where $\mathcal{R}$ is a multiset of *relational atoms* of the form $R_i t_1 \ldots t_n$, and Γ and Δ are multisets of *labelled formulae* of the form $w : \phi$, and (ii) labelled sequents of the form $\mathcal{R} \Rightarrow R_i t_1 \ldots t_n$, where $\mathcal{R}$ is a multiset of relational atoms. The relational atoms encode information about accessibility relations, and labelled formulae encode the satisfaction relation (i.e. $w : \phi$ can be interpreted as $M, w \Vdash \phi$). The separation between the two types of sequents is based on work by Gabbay [Gab96], whereby labelled proof systems are divided into two parts: a *base calculus* that corresponds to a base logic (with inference rules operating on the sequents of the form $\mathcal{R}, \Gamma \Rightarrow \Delta$), and a *relational algebra* that is used to reason about relational atoms, i.e. the properties of accessibility relations (with inference rules operating on the sequents of the form $\mathcal{R} \Rightarrow R_i t_1 \ldots t_n$) [Vig00, p. 7]. In essence, the base calculus is a proof system consisting of rules for deriving logical formulae, and the relational algebra is a proof system consisting of rules that allow the derivation of relational properties. Viganò argues that the modularity of his labelled sequent calculi is partially due to this division, which allows for a labelled sequent calculus to be constructed for a logic by taking the base calculus and equipping it with the proper relational algebra encoding the properties of that logic's accessibility relation(s). If another logic's accessibility relations possess distinct characteristics, then by modifying the relational algebra accordingly, a new labelled system for the logic can be obtained, showing that the formalism is modular.

To exhibit how a semantic clause can be transformed into a set of inference rules in the labelled formalism, let us consider the standard semantic clause for the $\Box$ modality in the modal logic K (cf. [BdRV01]), which is similar to the semantic clause for the $[x]$ modality used in grammar logics (see Sect. 5).

$$M, w \Vdash \Box\phi \text{ iff for all } u \in W, \text{ if } Rwu, \text{ then } M, u \Vdash \phi.$$

The above semantic clause can be transformed into the following two clauses by means of classical reasoning—one of which expresses when $\Box\phi$ is satisfied, and another which expresses when $\Box\phi$ is unsatisfied (in a relational model $M = (W, R, V)$).

(i) If for all $u \in W$, if $(w, u) \in R$, then $M, u \Vdash \phi$, then $M, w \Vdash \Box\phi$.

(ii) If there exists a $u \in W$ such that $(w, u) \in R$ and $M, u \not\Vdash \phi$, then $M, w \not\Vdash \Box\phi$.

Clause (i) can be written as the inference rule shown top-left below, which states that if for any u, if $(w, u) \in R$, then $M, u \Vdash \phi$ holds, then $M, w \Vdash \Box\phi$ holds. Putting this into sequent notation, we can use "$Rwu \Rightarrow u : \phi$" to represent "if $(w, u) \in R$, then $M, u \Vdash \phi$ holds", and "$\Rightarrow w : \Box\phi$" to represent "$M, w \Vdash \Box\phi$". This notational substitution gives the inference rule shown top-middle below. To obtain the final labelled sequent rule that corresponds to the first clause, we add in contexts $\mathcal{R}$, Γ, and Δ, which gives the final form of the rule shown top-right below. We note that a side condition must be imposed on the rule once contexts are added, i.e. u is not allowed to occur in the conclusion, which ensures that the label is implicitly universally quantified. The inference rule corresponding to clause (ii) is obtained in a similar fashion, and its derivation is shown second below.

$$\frac{\text{if } (w,u) \in R, \text{ then } M, u \Vdash \phi}{M, w \Vdash \Box\phi} \quad \rightsquigarrow \quad \frac{Rwu \Rightarrow u : \phi}{\Rightarrow w : \Box\phi} \quad \rightsquigarrow \quad \frac{\mathcal{R}, Rwu, \Gamma \Rightarrow u : \phi, \Delta}{\mathcal{R}, \Gamma \Rightarrow w : \Box\phi, \Delta}$$

$$\frac{(w,u) \in R \text{ and } M, u \not\Vdash \phi}{M, w \not\Vdash \Box\phi} \rightsquigarrow \frac{\Rightarrow Rwu \qquad u : \phi \Rightarrow}{w : \Box\phi \Rightarrow} \rightsquigarrow \frac{\mathcal{R} \Rightarrow Rwu \qquad \mathcal{R}, u : \phi, \Gamma \Rightarrow \Delta}{\mathcal{R}, w : \Box\phi, \Gamma \Rightarrow \Delta}$$

Building off of Viganò's [Vig00] and Simpson's [Sim94] work, Dyckhoff and Negri [DN12, Neg05], showed that cut-free labelled sequent systems could be provided for an even broader class of modal and constructive logics than those provided by Viganò. In [DN12, Neg05], the authors showed that labelled sequent calculi could be provided for any extension of intuitionistic logic Int, and any extension of the modal logic K, with the accessibility relation satisfying any number of *geometric formulae* [Sim94], that is, formulae of the form shown below left. These classes of logics go beyond those considered by Viganò, as Viganò only considers extensions of Int and K where the accessibility relation satisfies Horn formulae.

$$\forall \vec{x}\Big(\bigwedge_{i=1}^{n} \phi_i \rightarrow \exists \vec{y}(\bigvee_{j=1}^{m} \bigwedge_{k=1}^{l_j} \psi_{j,k})\Big) \qquad \frac{\mathcal{R}, \overline{\phi}, \overline{\psi}_1, \Gamma \Rightarrow \Delta \qquad \cdots \qquad \mathcal{R}, \overline{\phi}, \overline{\psi}_m, \Gamma \Rightarrow \Delta}{\mathcal{R}, \overline{\phi}, \Gamma \Rightarrow \Delta}$$

In the geometric formula (above left), each ϕ_i and $\psi_{j,k}$ is taken to be an atomic formula. Each geometric formula is equivalent to a *geometric structural rule* (introduced in Simpson's PhD thesis [Sim94]) of the form shown above right, where $\overline{\phi} := \phi_1, \ldots, \phi_n$, $\overline{\psi}_j := \psi_{j,1}, \ldots, \psi_{j,l_j}$, and each variable from $\vec{y}$ is an *eigenvariable*, meaning that it does not occur in the conclusion of the rule. Interestingly, the addition of any number of geometric structural rules to a base (labelled) calculus for Int or K yields a sound and complete labelled sequent calculus for that extension of Int or K with the corresponding geometric formulae.

Another distinction between the work of Viganò [Vig00] and the work in [DN12, Neg05], is that the latter abandons the division between a base calculus and the relational algebra, and instead, only employs labelled sequents of the form $\mathcal{R}, \Gamma \Rightarrow \Delta$ (as in Simpson's PhD thesis [Sim94]). Consequently, reasoning about relational atoms (i.e. applying inference rules that manipulate relational atoms) is done directly on labelled sequents of the form $\mathcal{R}, \Gamma \Rightarrow \Delta$, instead of on labelled sequents of the form $\mathcal{R} \Rightarrow R_i t_1 \ldots t_n$. As will be seen in the succeeding chapters (Ch. 4 and 5), this difference is conducive to refining our labelled calculi since it allows for the interaction of rules that would have otherwise been kept separate in Viganò's formalism. Therefore, we will build labelled sequent calculi for grammar logics, first-order intuitionistic logics, and deontic STIT logics within the formalism of [DN12, Neg05, Sim94], thus setting the stage for refinement in Ch. 4 and 5.

Last, we note that although our labelled sequent systems admit syntactic cut-elimination, a *strict* form of analyticity (i.e. the subformula property) fails to hold—as is typical of labelled systems. That is to say, cut-elimination does not immediately imply that every formula occurring within a cut-free derivation is a subformula of the conclusion. The blame for the loss of (a strict notion of) analyticity lies with the incorporation of structural rules that delete relational atoms from premise to conclusion. Nevertheless, since labelled sequent systems regularly include such rules, they are often qualified as analytic if every labelled formula occurring in a derivation is a subformula of some labelled formula in the conclusion [Neg05, Vig00]. It will be seen that this weaker notion of analyticity holds for our labelled sequent systems; yet, interestingly, the systems obtained via refinement (in the following chapters) will either possess a strict form of the subformula property or a higher degree of the property.

Terminology and Notation.

We introduce notation and terminology which is uniformly applicable to the wide variety of labelled sequent systems we consider. First off, *labelled sequents* are syntactic objects of the form $\Lambda := \mathcal{R}, \Gamma \Rightarrow \Delta$. The multiset $\mathcal{R}$ of relational atoms may be empty, or may consist of relational atoms of the form $R_x wu$, $\underline{a} \in D_w$, $w \leq u$, or $R_{[i]}wu$ depending on if the labelled calculus is for a grammar logic, first-order intuitionistic logic, or deontic

STIT logic. The labels w and u are taken to be among a denumerable set of labels Lab, and the multisets Γ and Δ occurring in a labelled sequent may be empty, or may consist of labelled formulae of the form $w : \phi$, where $\phi \in \mathcal{L}_{\mathsf{Km}}(\sum)$, $\phi \in \mathcal{L}_{\mathsf{Int}}$, or $\phi \in \mathcal{L}_{\mathsf{DS}^k_n}$, depending on the logic under consideration. In each of the following sections, we will formally define the set of labelled sequents used. Also, we define the following useful operations on multisets:

Definition 30. *If Γ is a multiset of labelled formulae, then we define $\Gamma \restriction w$ to be the multiset $\{\phi \mid w : \phi \in \Gamma\}$, and if $\Gamma := \phi_1, \ldots, \phi_k$ is a multiset of formulae, then we define $w : \Gamma := w : \phi_1, \ldots, w : \phi_k$.*

A *labelled calculus* is a set of *inference rules*, which are objects of the following form:

$$\frac{\Lambda_1 \qquad \cdots \qquad \Lambda_n}{\Lambda} \; (r)$$

Inference rules are schematic, and when used in practice they will be instantiated with concrete labelled sequents, thus allowing for an instance of Λ (the *conclusion*) to be derived from instances of Λ_1, ..., Λ_n (the *premises*). Also, if $n = 0$ in an inference rule, that is, the inference rule does not contain premises, then we say that the inference rule is *initial* and each labelled sequent given as a conclusion of the rule is an *initial sequent*.

Aside from initial rules, there are two other types of rules that our systems make use of: *logical rules* and *structural rules*. An inference rule is a *logical rule* iff the rule explicitly introduces a complex logical formula, called the *principal formula*, from less complex logical formulae, called the *auxiliary formulae* (all other formulae in the inference are called *side formulae* or *parametric formulae*). Furthermore, we refer to the principal and auxiliary formulae of an inference as *active*, more generally. For example, the following rules are logical rules since they introduce the complex logical formulae $\phi \land \psi$ and $\Box\phi$, respectively.

$$\frac{\mathcal{R}, \Gamma, w : \phi, w : \psi \Rightarrow \Delta}{\mathcal{R}, \Gamma, w : \phi \land \psi \Rightarrow \Delta} \qquad \frac{\mathcal{R}, Rwu, \Gamma \Rightarrow u : \phi, \Delta}{\mathcal{R}, \Gamma \Rightarrow w : \Box\phi, \Delta}$$

In the above rules, the formulae $w : \phi \land \psi$ and $w : \Box\phi$ are principal, $w : \phi, w : \psi$ and $Rwu, u : \phi$ are auxiliary, and all such formulae are active. The formulae in $\mathcal{R}, \Gamma$, and Δ are side (or, parametric) formulae.

In contrast to logical rules, our labelled calculi also contain *structural rules* which are rules that do not make explicit reference to a logical connective being introduced (as in the case of the logical rules), but rather, directly manipulate labelled sequents in a manner consistent with the associated logic's meta-logical properties. For example, the following two rules are labelled structural rules:

$$\frac{\mathcal{R}, w \leq w, \Gamma \Rightarrow \Delta}{\mathcal{R}, \Gamma \Rightarrow \Delta} \qquad \frac{\mathcal{R}, \Gamma \Rightarrow w : \phi, w : \phi, \Delta}{\mathcal{R}, \Gamma \Rightarrow w : \phi, \Delta}$$

The top-left rule deletes a relational atom, and the top-right rule contracts the labelled formula $w : \phi$ from two copies into one copy.

A *proof* is defined to be a tree of labelled sequents such that each labelled sequent is the conclusion of an inference rule with premises also in the proof, and all leaves of the proof are initial sequents. The root of the proof is called the *end sequent*, and the *height* of the proof is the maximal number of labelled sequents that occur from the end sequent to an initial sequent.

We say that a rule (r) is *admissible* (*height-preserving (hp-) admissible*) in a labelled calculus iff derivability of the premise(s) implies derivability of the conclusion with an (r)-free proof (whose height is less than or equal to the height(s) of the derivation(s) of the premise(s)). A rule (r) is *eliminable* in a labelled calculus iff there exists an algorithm that transforms any proof in the labelled calculus into an (r)-free proof (observe that eliminability implies admissibility). Last, we say that a rule is *invertible* (*height-preserving (hp-) invertible*) in a labelled calculus iff if an instance of the conclusion is derivable, then the premise(s) is (are) derivable (with height(s) less than or equal to the height of the derivation of the conclusion).

All of the above proof-theoretic terminology is fairly standard and can be found in most introductory texts on proof theory (e.g. [Bus98, Tak13]). Moreover, we extend such terminology to systems built within other proof theoretic formalisms (e.g. nested sequents).

If a rule is (hp-)admissible or eliminable, then this implies that given a proof Π of the premise of the rule, one can find a proof Π' of the conclusion of the rule. Similarly, if a rule is (hp-)invertible, then this implies that given a proof Π of the conclusion of the rule, one can find a proof Π' of the premise(s) of the rule. When applying an (hp-)admissible, eliminable, or (hp-)invertible rule (r) in a derivation, we will use a dashed line (as shown below left) to represent that the conclusion is not obtained via an inference rule, but rather, is obtained via the (hp-)admissibility, eliminability, or (hp-)invertibility property of the rule. Also, we will use a dashed line to indicate that a labelled sequent is known to be derivable by an assumption or by some result. Last, we use dotted lines (as shown below right) to indicate that two labelled sequents are identical to one another.

$$\begin{array}{c} \mathcal{R}, \Gamma \Rightarrow \Delta \\ \hdashline \mathcal{R}', \Gamma' \Rightarrow \Delta' \end{array} \ (r) \qquad \begin{array}{c} \mathcal{R}, \Gamma \Rightarrow \Delta \\ \hdashline[0.5pt/2pt] \mathcal{R}', \Gamma' \Rightarrow \Delta' \end{array} \ =$$

In order to understand the data structures encoded by labelled sequents, we will transform labelled sequents into graphs. Therefore, it will be useful to define the notion of a (sub)graph, various types of graphs (e.g. trees, DAGs, etc.), and isomorphisms between graphs. In the following chapters, we will see that refining our labelled calculi necessitates a “simplification” in the data structures encoded by labelled sequents in derivations.

Definition 31 (Graph, Induced Subgraph)**.** *A* graph *is an object of the form $G = (V, E, L)$, where V is a set of vertices, the set of edges $E \subseteq V \times V$ is a binary relation on*

V (we also allow the set of edges $E \subseteq V \times V \times S$ to be indexed with elements of a set S), and L is a labelling function mapping elements from V to elements in a set S'.

We define an induced subgraph *$G' = (V', E', L')$ of a graph $G = (V, E, L)$ to be a graph such that $V' \subseteq V$, $E' = E \restriction V'$, and $L' = L \restriction V'$.*

Definition 32 (Tree)**.** *A graph $G = (V, E, L)$ is a* tree *iff there exists a vertex $w \in V$ (called the* root*) such that there is a unique directed path from w to every other vertex $u \in V$. Equivalently, a tree is a graph that is connected, free of directed cycles, and contains no backward branching.*

Definition 33 (Polytree)**.** *We say that a graph $G = (V, E, L)$ is a* polytree *iff it is connected, free of directed cycles, and free of undirected cycles.*

Definition 34 (Forest)**.** *A graph $G = (V, E, L)$ is a* forest *iff it consists of a disjoint union of trees. We refer to each root of a tree in the disjoint union as a* root *of the forest.*

Definition 35 (Directed Acyclic Graph (DAG))**.** *A graph $G = (V, E, L)$ is a* directed acyclic graph (DAG) *iff it is free of directed cycles. We refer to a vertex $v \in V$ as a* root *of a DAG given that no vertices $u \in V$ exist such that there is an edge from u to v.*

Definition 36 (Isomorphism, Isomorphic)**.** *Let $G = (V, E, L)$ and $G' = (V', E', L')$ be two graphs. An* isomorphism *$f : V \mapsto V'$ from G to G' is a function such that:*

- *f is bijective,*
- *$(x, y) \in E$ iff $(fx, fy) \in E'$ (or, $(x, y, s) \in E$ iff $(fx, fy, s) \in E'$, if edges are indexed with elements from a set S), and*
- *$L(w) = L'(fw)$.*

We say G and G' are isomorphic *(and write $G \cong G'$) iff there exists an isomorphism between them.*

3.2 Labelled Calculi for Grammar Logics

Although grammar logics were introduced in 1988 [dCP88], the proof theory for such logics did not come about until a decade later [BGM98]. In [BGM98], Baldoni *et al.* formulated prefixed analytic tableaux for grammar logics and leveraged these systems to show the decidability of right linear grammar logics and the undecidability of context-free grammar logics. In subsequent years, alternative tableaux were proposed for classes of grammar logics, such as (non-prefixed) tableau with automaton-labelled formulae for regular grammar logics [GN05] and tableaux for regular grammar logics with converse [NS09, NS11]. The former were used to provide EXPTIME decision procedures for regular grammar logics as well as to give an effective Craig interpolation lemma, while the latter was used to provide optimal EXPTIME decision procedures for regular grammar logics with converse.

Beyond the tableau formalism, nested sequent calculi were given for context-free grammar logics with converse [TIG12]. In that paper, the authors define proof-search and countermodel extraction algorithms, which are of EXPSPACE complexity when termination holds. Interestingly, the method of refinement (discussed in the next chapter) yields (slight variants of) the nested calculi of [TIG12] when applied to the labelled sequent calculi for context-free grammar logics with converse introduced in this section.

Below, we adapt the formalism and methodology of [DN12, Neg05, Sim94, Vig00] to introduce labelled sequent systems for grammar logics. Our calculi consist of rules that manipulate *labelled sequents*, that is, formulae extending the language $\mathcal{L}_{\mathsf{Km}}(\sum)$ and which are used to derive theorems. We define these as follows:

Definition 37 (Labelled Sequents for Grammar Logics)**.** Labelled sequents *for grammar logics are defined to be syntactic objects of the form* $\Lambda := \mathcal{R} \Rightarrow \Gamma$*, where* $\mathcal{R}$ *(the* antecedent*) and* Γ *(the* consequent*) are defined via the following grammars in BNF:*

$$\mathcal{R} ::= \epsilon \mid R_x wu \mid \mathcal{R}, \mathcal{R} \qquad \Gamma ::= \epsilon \mid w : \phi \mid \Gamma, \Gamma$$

with $x \in \sum$*,* $\phi \in \mathcal{L}_{\mathsf{Km}}(\sum)$*, and where* w *and* u *are among a denumerable set of labels* Lab $:= \{w, u, v, \ldots\}$*. We refer to formulae of the form* $R_x wu$ *as* relational atoms*, to formulae of the form* $w : \phi$ *as* labelled formulae*, and let* $\mathrm{Lab}(\Lambda)$*,* $\mathrm{Lab}(\mathcal{R})$*, and* $\mathrm{Lab}(\Gamma)$ *represent the set of labels in* Λ*,* $\mathcal{R}$*, and* Γ*, respectively.*

We use Λ, Λ', ... (occasionally with subscripts) to denote labelled sequents, $\mathcal{R}$, $\mathcal{R}'$, ... (occasionally with subscripts) to denote multisets of relational atoms, and Γ, Γ', ... (occasionally with subscripts) to denote multisets of labelled formulae. Hence, we take the comma operator to be both associative and commutative; for example, we identify $R_a wu, R_b uv$ with $R_b uv, R_a wu$ and $w : \phi, u : \psi, v : \chi$ with $v : \chi, w : \phi, u : \psi$. This interpretation of comma is what lets us view strings $\mathcal{R}$ and Γ as multisets. The symbol ϵ is taken to represent the *empty string* that acts as an identity element for the comma operator (e.g. $R_a wu, \epsilon, R_b uv$ is identified with $R_a wu, R_b uv$); hence, ϵ will typically be implicit in labelled sequents.

Labelled sequents serve as an abstraction of a $\sum$-model (Def. 4), where the labels from Lab represent the worlds, the relational atoms represent the accessibility relations, and the labelled formulae determine at which world a formula $\phi \in \mathcal{L}_{\mathsf{Km}}(\sum)$ is (un)satisfied. Intuitively, we interpret a labelled sequent $\mathcal{R} \Rightarrow \Gamma$ as saying, 'if all relational atoms in $\mathcal{R}$ hold, then some formula in Γ holds.' See Def. 41 for the formal definition of a labelled sequent interpretation, which also spells out the exact relationship between labelled sequents and $\sum$-models.

Our labelled calculi for grammar logics $\mathsf{Km}(S)$ are displayed in Fig. 3.1, and the derivability relation is defined below. Moreover, examples of concrete derivations in a calculus $\mathsf{G3Km}(S)$ can be found in the proof of Thm. 7 below.

$$\frac{}{\mathcal{R} \Rightarrow w : p, w : \neg p, \Gamma}\ (id) \qquad \frac{\mathcal{R} \Rightarrow w : \phi, w : \psi, \Gamma}{\mathcal{R} \Rightarrow w : \phi \lor \psi, \Gamma}\ (\lor_r)$$

$$\frac{\mathcal{R} \Rightarrow w : \phi, \Gamma \qquad \mathcal{R} \Rightarrow w : \psi, \Gamma}{\mathcal{R} \Rightarrow w : \phi \land \psi, \Gamma}\ (\land_r) \qquad \frac{\mathcal{R}, R_x wu \Rightarrow w : \langle x \rangle \phi, u : \phi, \Gamma}{\mathcal{R}, R_x wu \Rightarrow w : \langle x \rangle \phi, \Gamma}\ (\langle x \rangle)$$

$$\frac{\mathcal{R}, R_x wu \Rightarrow u : \phi, \Gamma}{\mathcal{R} \Rightarrow w : [x]\phi, \Gamma}\ ([x])^{\dagger} \qquad \frac{\mathcal{R}, R_s wu, R_x wu \Rightarrow \Gamma}{\mathcal{R}, R_s wu \Rightarrow \Gamma}\ (p^x_s) \qquad \frac{\mathcal{R}, R_x wu, R_{\overline{x}} uw \Rightarrow \Gamma}{\mathcal{R}, R_x wu \Rightarrow \Gamma}\ (C^x_1)$$

Figure 3.1: The labelled calculus G3Km(S) for the grammar logic Km(S). We have an $([x])$, $(\langle x \rangle)$, and (C^x_1) rule for each $x \in \sum$, and for each production rule $x \longrightarrow s \in S$, we have a corresponding (p^x_s) rule (see Def. 39 below for a description of the correspondence between production rules and structural rules). The side condition † states that the associated rule can be applied only if the label u is an *eigenvariable*, i.e. it does not occur in the conclusion of the rule. Last, we assume that each calculus satisfies the closure condition (given in Def. 40).

Definition 38. *We write* $\vdash_{\mathsf{G3Km}(S)} \Lambda$ *to indicate that a labelled sequent* Λ *is derivable in a calculus* G3Km(S). *(NB. We will use the notation* $\vdash_{\mathsf{X}}$ *throughout the thesis to denote that a sequent is derivable in the calculus* X.*)*

The rules of the calculi are obtained from the notion of a $\sum$-model (Def. 4), and from the semantic clauses for the various connectives (Def. 5). The (id) rule encodes the fact that in a $\sum$-model the valuation function will either make a propositional variable p true at a world, or it will not. The rules $(\land_r)$, $(\lor_r)$, $([x])$, and $(\langle x \rangle)$ are obtained from the semantic clauses of the respective logical connectives, and each rule (p^x_s) lets us derive theorems based on the frame property obtained from the corresponding production rule in S (see Def. 39 for the correspondence between production rules and frame properties). Each (C^x_1) rule encodes the fact that in a $\sum$-model, the relation R_x is the converse of the relation $R_{\overline{x}}$ for any $x \in \sum$, thus corresponding to the converse condition $\mathbf{(C_1)}$ (Def. 4).

As mentioned previously, a nice feature of the labelled paradigm is that it is relatively straightforward to transform the relational semantics of a logic into a labelled calculus.[1] In addition to ease of construction, labelled calculi typically possess desirable proof-theoretic properties, such as (hp-)admissibility of structural rules (e.g. those presented in Fig. 3.2 below) and hp-invertibility of rules [DN12, Neg05, Sim94, Vig00]. In [Neg05], the author shows that such properties are preserved under any extension of the labelled base calculus G3K (for the minimal modal logic K) with geometric structural rules (mentioned into the introduction to this chapter). By analogy, let us think of the set of rules $\{(id), (\land_r), (\lor_r), ([x]), (\langle x \rangle) \mid x \in \sum\}$ as being a base calculus akin to G3K. Despite the

[1] As shown in [CMS13], the process of constructing labelled calculi for intermediate logics can be automated.

fact that [Neg05] does not address the construction and properties of labelled calculi in the multi-modal setting (and considers only the uni-modal setting), if we think of the aforementioned set as our base calculus, and because the rules (p^x_s) and (C^x_1) fit within the geometric rule scheme, it is straightforward to adapt the methods and results from [Neg05] to our setting. Below, we will explain how this is done.

Although we may think of the set of rules $\{(id), (\wedge_r), (\vee_r), ([x]), (\langle x \rangle) \mid x \in \sum\}$ as a base calculus in order to draw an analogy with the method of [Neg05] and transfer results to our setting, the set of rules $\{(id), (\wedge_r), (\vee_r), ([x]), (\langle x \rangle), (C^x_1) \mid x \in \sum\}$ serves as a sound and complete calculus for the base grammar logic $\mathsf{Km} = \mathsf{Km}(\emptyset)$ relative to an alphabet $\sum$ (see Thm. 5 and 7). To obtain a calculus for an extension of a minimal grammar logic Km with a CFCST system S (defined in Def. 8), i.e. for the logic $\mathsf{Km}(S)$, we add a structural rule (p^x_s) for each production rule in S. The correspondence between structural and production rules is detailed in the definition below:

Definition 39 (Production and Structural Rules). *Let S be a CFCST system, and let $x \longrightarrow s \in S$. If $s = x_1 \cdots x_n \in \sum^* - \{\varepsilon\}$, we define $R_s wu := R_{x_1} wv_1, \ldots, R_{x_n} v_{n-1} u$, and if $s = \varepsilon$, then we define $R_s wu := \epsilon$. Depending on if $s = \varepsilon$ or not, the structural rule corresponding to the production rule $x \longrightarrow s$ is defined as follows:*

$s \neq \varepsilon$	$s = \varepsilon$
$\dfrac{\mathcal{R}, R_s wu, R_x wu \Rightarrow \Gamma}{\mathcal{R}, R_s wu \Rightarrow \Gamma}\ (p^x_s)$	$\dfrac{\mathcal{R}, R_x ww \Rightarrow \Gamma}{\mathcal{R} \Rightarrow \Gamma}\ (p^x_s)$

In the rule top right, observe that $R_s wu$ does not occur because $s = \varepsilon$, meaning that $R_s wu$ is the empty string ϵ.

Although labelled calculi generally permit the hp-admissibility of relational atoms contractions $(ctr_{\mathcal{R}})$ (see Fig. 3.2), there is an important caveat. When labelled calculi are extended with geometric structural rules of a specific shape, additional rules must be added to the system to ensure the hp-admissibility of $(ctr_{\mathcal{R}})$ [Neg05]. To motivate and explain the need for such rules, we consider an example: suppose we have a CFCST system $S := \{b \longrightarrow aa, \overline{b} \longrightarrow \overline{aa}\}$. The first production rule gives rise to the (p^b_{aa}) structural rule shown below left. (NB. Since the structural rule (p^b_{aa}) is obtained from the first production rule, we let b instantiate x and aa instantiate s in the schema (p^x_s) as explained in Def. 39 above.) To explain why the hp-admissibility of relational atom contractions $(ctr_{\mathcal{R}})$ is not immediately permitted, let us assume that we are given a derivation ending with the following two inferences shown below right, where an instance of (p^b_{aa}) (with the label w substituted for v and u) is followed by an instance of $(ctr_{\mathcal{R}})$:

$$\dfrac{\mathcal{R}, R_a wv, R_a vu, R_b wu \Rightarrow \Gamma}{\mathcal{R}, R_a wv, R_a vz \Rightarrow \Gamma}\ (p^b_{aa}) \qquad \dfrac{\dfrac{\mathcal{R}, R_a ww, R_a ww, R_b ww \Rightarrow \Gamma}{\mathcal{R}, R_a ww, R_a ww \Rightarrow \Gamma}\ (p^b_{aa})}{\mathcal{R}, R_a ww \Rightarrow \Gamma}\ (ctr_{\mathcal{R}})$$

Typically, when proving a rule (such as $(ctr_{\mathcal{R}})$) hp-admissible, one shows that the rule can be permuted above any rule of the calculus and can be deleted at initial sequents. Notice though, if we attempt to permute $(ctr_{\mathcal{R}})$ above (p^{b}_{aa}) and apply the rule directly to the top sequent, then we obtain the labelled sequent $\mathcal{R}, R_a ww, R_b ww \Rightarrow \Gamma$. This labelled sequent is no longer in a form matching the premise of (p^{b}_{aa}), and so the rule is inapplicable; the other production rule of S does not give rise to a structural rule that could be used to delete $R_b ww$ (in order to derive the desired end sequent) either.

Nevertheless, this obstacle can be overcome by stipulating that our labelled calculi must adhere to the so-called *closure condition* [Neg05]. Essentially, the closure condition states that if a substitution of labels within a structural rule causes a duplication of *principal* relational atoms (i.e. those occurring in the conclusion), then another instance of the rule with those relational atoms contracted must be added to the calculus. Since the closure condition is only enforced when a duplication of principal relational atoms arises in the conclusion of the structural rule, this condition need not be enforced for the (C^{x}_{1}) structural rules, which only contain a *single* principal relational atom in the conclusion. Howbeit, each (p^{x}_{s}) rule may contain more than a single principal relational atom, so we define the closure condition for such rules below:

Definition 40 (Closure Condition)**.** *Let S be a CFCST system, and let (r_1) be either a structural rule (p^{x}_{s}) obtained from a production rule $x \longrightarrow s \in S$, or a structural rule $(p^{x}_{s})_{\ddagger}$ obtained from the closure condition. (NB. We use a ‡ subscript to indicate rules obtained via the closure condition.) For all such rules (r_1), if the following is an instance of (r_1):*

$$\frac{\mathcal{R}, R_t wv, R_y vz, R_y vz, R_r zu, R_x wu \Rightarrow \Gamma}{\mathcal{R}, R_t wv, R_y vz, R_y vz, R_r zu \Rightarrow \Gamma}\ (r_1)$$

then our calculus satisfies the closure condition *iff it also contains the following instance of the rule (with $R_y vz$ contracted in both premise and conclusion):*

$$\frac{\mathcal{R}, R_t wv, R_y vz, R_r zu, R_x wu \Rightarrow \Gamma}{\mathcal{R}, R_t vu, R_y vz, R_r zu \Rightarrow \Gamma}\ (r_2)$$

As stated in Fig. 3.1, we assume that each $\mathsf{G3Km}(S)$ *calculus satisfies the closure condition.*

For any given structural rule (p^{x}_{s}) only a finite number of instances with duplications can arise, implying that the closure condition only adds a finite number of rules for each (p^{x}_{s}) rule. Hence, the above recursively defined condition terminates, meaning that the imposition of the closure condition is unproblematic.

Now that we have defined each calculus $\mathsf{G3Km}(S)$, we show how to interpret labelled sequents on $\sum$-models, giving rise to a notion of validity for sequents. We then utilize this notion to show that our calculi are sound (see Thm. 5 below).

Definition 41 ($\mathsf{G3Km}(S)$ Semantics)**.** *Let $M = (W, \{R_x \mid x \in \sum\}, V)$ be a $\sum$-model satisfying the CFCST system S with I : Lab $\mapsto W$ an* interpretation function *mapping labels to worlds. We define the* satisfaction *of a relational atom $R_x wu$ (written $M, I \models_{\mathsf{Km}(S)} R_x wu$) and labelled formula $w : \phi$ (written $M, I \models_{\mathsf{Km}(S)} w : \phi$) as follows:*

- ▶ *$M, I \models_{\mathsf{Km}(S)} R_x wu$ iff $(I(w), I(u)) \in R_x$*
- ▶ *$M, I \models_{\mathsf{Km}(S)} w : \phi$ iff $M, I(w) \Vdash \phi$*

We say that a multiset of relational atoms $\mathcal{R}$ is satisfied *in M with I (written $M, I \models_{\mathsf{Km}(S)} \mathcal{R}$) iff $M, I \models_{\mathsf{Km}(S)} R_x wu$ for all $R_x wu \in \mathcal{R}$, and we say that a multiset of labelled formulae Γ is* satisfied *in M with I (written $M, I \models_{\mathsf{Km}(S)} \Gamma$) iff $M, I \models_{\mathsf{Km}(S)} w : \phi$ for all $w : \phi \in \Gamma$.*

A labelled sequent $\Lambda = \mathcal{R} \Rightarrow \Gamma$ is satisfied *in M with I (written, $M, I \models_{\mathsf{Km}(S)} \Lambda$) iff if $M, I \models_{\mathsf{Km}(S)} \mathcal{R}$, then $M, I \models_{\mathsf{Km}(S)} \Gamma$. Also, we say that a labelled sequent Λ is* falsified *in M with I iff $M, I \not\models_{\mathsf{Km}(S)} \Lambda$, that is, Λ is not satisfied by M with I.*

Last, a labelled sequent Λ is $\mathsf{Km}(S)$-valid *(written $\models_{\mathsf{Km}(S)} \Lambda$) iff it is satisfiable in every $\sum$-model M satisfying S with every interpretation function I. We say that a labelled sequent Λ is* $\mathsf{Km}(S)$-invalid *iff $\not\models_{\mathsf{Km}(S)} \Lambda$, i.e. Λ is not $\mathsf{Km}(S)$-valid.*

Theorem 5 (Soundness)**.** *If $\vdash_{\mathsf{G3Km}(S)} \Lambda$, then $\models_{\mathsf{Km}(S)} \Lambda$*

Proof. Let S be a CFCST system. We argue that each rule preserves validity by contraposition, that is, we show that if the conclusion of the rule is $\mathsf{Km}(S)$-invalid, then at least one premise of the rule is $\mathsf{Km}(S)$-invalid. We only consider the ($[x]$), ($\langle x \rangle$), (p^x_s), and (C^x_1) cases.

($[x]$) Suppose that the conclusion of ($[x]$) is $\mathsf{Km}(S)$-invalid, that is, there exists a $\sum$-model M satisfying S and an interpretation function I such that $M, I \not\models_{\mathsf{Km}(S)} \mathcal{R} \Rightarrow w : [x]\phi, \Gamma$. It follows that $M, I(w) \not\Vdash [x]\phi$, which further implies that there exists a world v such that $(I(w), v) \in R_x$ and $M, v \not\Vdash \phi$. We define a new interpretation function I' where $I'(z) = I(z)$ if $z \neq u$ and $I'(u) = v$ otherwise. Consequently, $M, I'(u) \not\Vdash \phi$, and since u is an eigenvariable and $(I'(w), I'(u)) \in R_x$, we have that the premise is falsified in M with I'.

($\langle x \rangle$) Suppose that the conclusion of an ($\langle x \rangle$) inference in $\mathsf{Km}(S)$-invalid, i.e. there exists a $\sum$-model M satisfying S and an interpretation function I such that $M, I \not\models_{\mathsf{Km}(S)} \mathcal{R}, R_x wu \Rightarrow w : \langle x \rangle\phi, \Gamma$. It follows that $(I(w), I(u)) \in R_x$ and $M, I(w) \not\Vdash \langle x \rangle\phi$, implying that $M, I(u) \not\Vdash \phi$. This shows that the premise is falsified in M and I.

(p^x_s) Let $x \longrightarrow s$ be the production rule corresponding to (p^x_s). We have two cases to consider: either (i) $s = \varepsilon$ or (ii) $s \neq \varepsilon$. For case (i), suppose that the conclusion of (p^x_s) is $\mathsf{Km}(S)$-invalid. Then, there exists a $\sum$-model M satisfying S and an interpretation function I such that $M, I \not\models_{\mathsf{Km}(S)} \mathcal{R}, R_s wu \Rightarrow \Gamma$. Since our $\sum$-model satisfies all production rules in S, and $x \longrightarrow s \in S$, we know that $R_s \subseteq R_x$ by Def. 10. This implies that $(I(w), I(u)) \in R_x$, showing that the premise is falsified by M with I. For case (ii), suppose that there exists a $\sum$-model M satisfying S and an interpretation function I such that $M, I \not\models_{\mathsf{Km}(S)} \mathcal{R} \Rightarrow \Gamma$. Since our $\sum$-model satisfies S, which includes the production rule $x \longrightarrow \varepsilon$, we know that $R_\varepsilon \subseteq R_x$ by Def. 10. By Def. 7, $(I(w), I(w)) \in R_\varepsilon \subseteq R_x$, implying that the premise is falsified by M with I. Also, note that the soundness of any rule $(p^x_s)_\ddagger$ obtained from (p^x_s) via the closure condition holds as well.

(C^x_1) Suppose the conclusion of (C^x_1) is $\mathsf{Km}(S)$-invlaid, that is, there exists a $\sum$-model M and interpretation function I such that $M, I \not\models_{\mathsf{Km}(S)} \mathcal{R}, R_x wu \Rightarrow \Gamma$. It follows that $(I(w), I(u)) \in R_x$, and since M satisfies the converse condition $\mathbf{(C_1)}$, we have $(I(u), I(w)) \in R_{\overline{x}}$, showing that the premise is falsified in M with I. □

As mentioned previously, we may adapt the work from [Neg05] to our setting and confirm that each calculus $\mathsf{G3Km}(S)$ possesses desirable proof theoretic properties. Not only are these properties useful in refining our labelled calculi, thus leading to applications—e.g. proving that all grammar logics based on CFCST systems have the effective Lyndon interpolation property (Sect. 6.2)—but also help us confirm the completeness of our calculi (Thm. 7). Before listing the properties of each calculus $\mathsf{G3Km}(S)$, we follow [DN12, Neg05] and define the notion of a *label substitution*, which is an operation fundamental in establishing the results that follow.

Definition 42 (Label Substitution)**.** *We define a* label substitution (w/u) *for* $w, u \in$ Lab *on individual relational atoms and labelled formulae as follows:*

- $$(R_x vz)(w/u) = \begin{cases} R_x wz & \text{if } u = v \text{ and } v \neq z \\ R_x ww & \text{if } u = v \text{ and } v = z \\ R_x uw & \text{if } u = z \text{ and } v \neq z \\ R_x vz & \text{otherwise} \end{cases}$$
- $$(v : \phi)(w/u) = \begin{cases} w : \phi & \text{if } u = v \\ v : \phi & \text{otherwise} \end{cases}$$

We define the label substitution (w/u) on a multiset of relational atoms $\mathcal{R}$ and a multiset of labelled formulae Γ to be the multiset obtained by applying (w/u) to each element of the multiset.

We are now in a position to leverage the results of [Neg05] in order to confirm that each calculus $\mathsf{G3Km}(S)$ possesses useful characteristics. As mentioned previously, the work of [Neg05] holds for uni-modal logics, that is, the language makes use of the standard modalities $\Box$ and $\Diamond$ (cf. [BdRV01]) as opposed to a variety of modalities like the ones we are using in the current multi-modal setting. Nevertheless, if we inspect the proofs of certain results in [Neg05], we can see that by identifying $[x]$ with $\Box$ and $\langle x \rangle$ with $\Diamond$,

$$\frac{\mathcal{R} \Rightarrow \Gamma}{\mathcal{R}, \mathcal{R}' \Rightarrow \Gamma', \Gamma}\ (wk) \qquad \frac{\mathcal{R} \Rightarrow \Gamma}{\mathcal{R}(w/u) \Rightarrow \Gamma(w/u)}\ (lsb) \qquad \frac{\mathcal{R}, R_a wu, R_a wu \Rightarrow \Gamma}{\mathcal{R}, R_a wu \Rightarrow \Gamma}\ (ctr_{\mathcal{R}})$$

$$\frac{\mathcal{R} \Rightarrow w : \phi, w : \phi, \Gamma}{\mathcal{R} \Rightarrow w : \phi, \Gamma}\ (ctr_r) \qquad \frac{\mathcal{R} \Rightarrow w : \phi, \Gamma \qquad \mathcal{R} \Rightarrow w : \neg\phi, \Gamma}{\mathcal{R} \Rightarrow \Gamma}\ (cut)$$

Figure 3.2: The set Str($\mathsf{Km}(S)$) of structural rules consists of all rules shown above.

all proofs go through. Moreover, only minor modifications to the proofs are needed to account for the different logical signatures and the fact that our formulae are in negation normal form. (NB. In [Neg05], the logical connectives consist of $\{\bot, \vee, \wedge, \rightarrow, \Box, \Diamond\}$; by contrast, our language uses $\{\neg, \vee, \wedge, [x], \langle x \rangle \mid x \in \sum\}$.) In what follows, we state the various properties possessed by each calculus $\mathsf{G3Km}(S)$, and mention in the corresponding proof what results from [Neg05] are exploited.

Lemma 9. *For all $\phi \in \mathcal{L}_{\mathsf{Km}}(\sum)$, $\vdash_{\mathsf{G3Km}(S)} \mathcal{R} \Rightarrow w : \phi, w : \neg\phi, \Gamma$.*

Proof. The claim is shown by induction on the complexity $|\phi|$ of ϕ. □

Lemma 10. *The (lsb), (wk), $(ctr_{\mathcal{R}})$, and (ctr_r) rules are hp-admissible.*

Proof. All claims are shown by induction on the height of the given derivation and are similar to the proofs of Lem. 4.3, Prop. 4.4, and Thm. 4.12 of [Neg05]. □

Lemma 11. *All rules of $\mathsf{G3Km}(S)$ are hp-invertible.*

Proof. Hp-invertibility of the $(\langle x \rangle)$, (p_s^x), and (C_1^x) rules follows from the hp-admissibility of (wk). The remaining rules are shown hp-invertible by induction on the height of the given derivation and are similar to the proof of Prop. 4.11 of [Neg05]. □

Theorem 6. *The rule (cut) is eliminable in $\mathsf{G3Km}(S)$.*

Proof. The result is shown by induction on the lexicographic ordering of pairs $(|\phi|, h_1+h_2)$, where $|\phi|$ is the complexity of the cut formula ϕ, h_1 is the height of the derivation of the left premise of (cut), and h_2 is the height of the derivation of the right premise of (cut). The proof is similar to the proof of Thm. 4.13 of [Neg05]. □

Let us now harness the above results to prove each calculus $\mathsf{G3Km}(S)$ complete relative to $\mathsf{Km}(S)$. Prior to proving full completeness (i.e. completeness relative to $\mathsf{Km}(S)$), we confirm that the rules (id), $(\vee_r)$, and $(\wedge_r)$ are enough to confirm *classical completeness*, that is, all instances of classical propositional tautologies are derivable in each $\mathsf{G3Km}(S)$ calculus. This is needed as it ensures that the axioms A0 (see Def. 12 for each grammar logic axiomatization $\mathsf{HKm}(S)$) are satisfied. Although this result is needed to confirm full completeness of our calculi, we defer the proof to the appendix (Appendix B on p. 208) to simplify presentation.

Lemma 12 (Classical Completeness)**.** *All instances of classical propositional tautologies in* $\mathcal{L}_{\mathsf{Km}}(\sum)$ *are derivable in* $\mathsf{G3Km}(S)$*.*

Proof. See Appendix B (p. 208) for details. □

Theorem 7 (Completeness)**.** *If* $\vdash_{\mathsf{Km}(S)} \phi$*, then* $\vdash_{\mathsf{G3Km}(S)} \epsilon \Rightarrow w : \phi$*.*

Proof. By Lem. 12, we know that all instances of classical propositional tautologies in $\mathcal{L}_{\mathsf{Km}}(\sum)$ are derivable in $\mathsf{G3Km}(S)$, ensuring that A0 holds. Therefore, we need only show that the axioms A1 – A4 and the rules R0 and R1 (see Def. 12 for the axiomatization of $\mathsf{Km}(S)$) are derivable in $\mathsf{G3Km}(S)$ to show completeness.

Axiom A1.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Pi_1 \qquad \Pi_2}{R_x wu \Rightarrow w : \langle x\rangle(\phi \vee \neg\psi), u : \neg\phi \vee \psi, w : \langle x\rangle\neg\phi, u : \psi}}{R_x wu \Rightarrow w : \langle x\rangle(\phi \vee \neg\psi), w : \langle x\rangle\neg\phi, u : \psi}\,(\langle x\rangle)}{\Rightarrow w : \langle x\rangle(\phi \vee \neg\psi), w : \langle x\rangle\neg\phi, w : [x]\psi}\,([x])}{\Rightarrow w : \langle x\rangle(\phi \vee \neg\psi) \vee (\langle x\rangle\neg\phi \vee [x]\psi)}\,(\vee) \times 2}{\Rightarrow w : [x](\phi \rightarrow \psi) \rightarrow ([x]\phi \rightarrow [x]\psi)}\,=$$

$$\Pi_1 \quad := \left\{ \quad \dfrac{\dfrac{}{R_x wu \Rightarrow w : \langle x\rangle(\neg\phi \vee \psi), u : \phi, w : \langle x\rangle\neg\phi, u : \neg\phi, u : \psi}\,\text{Lem. 9}}{R_x wu \Rightarrow w : \langle x\rangle(\neg\phi \vee \psi), u : \phi, w : \langle x\rangle\neg\phi, u : \psi}\,(\langle x\rangle) \right.$$

$$\Pi_2 \quad := \left\{ \quad \dfrac{}{R_x wu \Rightarrow w : \langle x\rangle(\neg\phi \vee \psi), u : \neg\psi, w : \langle x\rangle\neg\phi, u : \psi}\,\text{Lem. 9} \right.$$

Rule R0. In the derivation of R0, the use of (*cut*) is eliminable due to Thm. 6.

$$\dfrac{\dfrac{\dfrac{\Rightarrow w : \phi \rightarrow \psi}{\Rightarrow w : \neg\phi \vee \psi}\,=}{\Rightarrow w : \neg\phi, w : \psi}\,\text{Lem. 11} \qquad \Rightarrow w : \phi}{\Rightarrow w : \psi}\,(cut)$$

Axiom A2.

$$
\begin{array}{c}
\hdashline
R_x wu, R_{\overline{x}} uw \Rightarrow w : \neg\phi, u : \langle\overline{x}\rangle\phi, w : \phi \quad \text{Lem. 9} \\
\hline
R_x wu, R_{\overline{x}} uw \Rightarrow w : \neg\phi, u : \langle\overline{x}\rangle\phi \quad (\langle x\rangle) \\
\hline
R_x wu \Rightarrow w : \neg\phi, u : \langle\overline{x}\rangle\phi \quad (C^x_1) \\
\hline
\Rightarrow w : \neg\phi, w : [x]\langle\overline{x}\rangle\phi \quad ([x]) \\
\hline
\Rightarrow w : \neg\phi \vee [x]\langle\overline{x}\rangle\phi \quad (\vee) \\
\cdots\cdots\cdots\cdots\cdots\cdots \; = \\
\Rightarrow w : \phi \rightarrow [x]\langle\overline{x}\rangle\phi
\end{array}
$$

Axiom A3 and rule R1. The use of (lsb) in the proof of R1 is admissible due to the use of Lem. 10.

$$
\begin{array}{c}
\hdashline
R_s wu, R_x wu \Rightarrow u : \neg\phi, w : \langle x\rangle\phi, u : \phi \quad \text{Lem. 9} \\
\hline
R_s wu, R_x wu \Rightarrow u : \neg\phi, w : \langle x\rangle\phi \quad (\langle x\rangle) \\
\hline
R_s wu \Rightarrow u : \neg\phi, w : \langle x\rangle\phi \quad (p^x_s) \\
\hline
\Rightarrow w : [s]\neg\phi, w : \langle x\rangle\phi \quad ([x]) \times |s| \\
\hline
\Rightarrow w : [s]\neg\phi \vee \langle x\rangle\phi \quad (\vee) \\
\cdots\cdots\cdots\cdots\cdots\cdots \; = \\
\Rightarrow w : \langle s\rangle\phi \rightarrow \langle x\rangle\phi
\end{array}
\qquad
\begin{array}{c}
\Rightarrow w : \phi \\
\hdashline
R_x uw \Rightarrow w : \psi \quad (wk) \\
\hline
\Rightarrow u : [x]\psi \quad [x] \\
\hdashline
\Rightarrow w : [x]\psi \quad (lsb)
\end{array}
$$

□

An interesting question to ask of labelled calculi is: what structures are necessary to ensure completeness? Once we *refine* our labelled calculi (in the following chapter), we will see that the labelled sequents needed to ensure completeness in the refined versions of our calculi are more minimalistic than those used in the $\mathsf{G3Km}(S)$ calculi. To make this notion of "structure" precise, we define *sequent graphs* (for grammar logics) below:

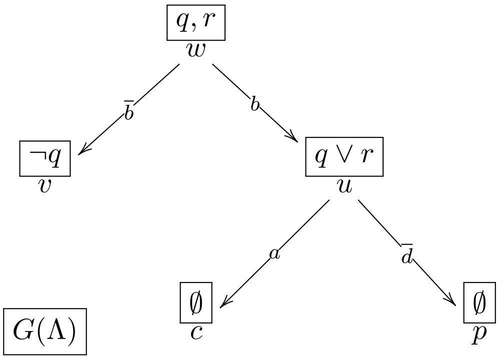
Definition 43 (Sequent Graph for Grammar Logics)**.** *Let* $\Lambda := \mathcal{R} \Rightarrow \Gamma$ *be a labelled sequent. We define the* sequent graph *of* Λ*,* $G(\Lambda) = (V, E, L)$*, as follows:*

- $V = \text{Lab}(\Lambda)$
- $E = \{(w, u, x) \mid R_x wu \in \mathcal{R}\}$
- $L(w) = \Gamma \restriction w$

(NB. Recall that $\Gamma \restriction w$ *was defined in Def. 30.)*

Example 1. *Below, we give an example of a labelled tree sequent* Λ *(see Def. 44) and its corresponding sequent graph* $G(\Lambda)$*.*

$$\Lambda := R_{\overline{b}}wv, R_b wu, R_a uc, R_{\overline{d}}up \Rightarrow$$
$$w : q, w : r, v : \neg q, u : q \vee r$$



In the following chapter, for each calculus $\mathsf{G3Km}(S)$, we will obtain a refined variant which makes use of labelled sequents whose graphs possess less structure, namely, *labelled tree sequents* [GR12]. Such sequents are essentially the same as the labelled nested sequents presented in [Pim18]. We define labelled tree sequents here and provide a theorem below which shows that restricting to such sequents invalidates the completeness of each calculus $\mathsf{G3Km}(S)$, that is, each calculus $\mathsf{G3Km}(S)$ requires labelled sequents with a more liberal syntactic structure to ensure completeness.

Definition 44 (Labelled Tree Sequent for Grammar Logics)**.** *A labelled sequent* Λ *is a* labelled tree sequent *iff* $G(\Lambda) = (V, E, L)$ *is a* tree*.*

Definition 45 (Labelled Tree Proof, Fixed Root Property for Grammar Logics)**.** *We say that a proof is a* labelled tree proof *iff it consists solely of labelled tree sequents.*

Also, we say that a labelled tree proof has the fixed root property *iff the every labelled tree sequent in the proof has the same root.*

As specified in the theorem below, if we restricted $\mathsf{G3Km}(S)$ to only allow labelled tree derivations, then there would be theorems of $\mathsf{Km}(S)$ that would no longer be provable. In such a situation, $\mathsf{G3Km}(S)$ would be *incomplete* relative to $\mathsf{Km}(S)$. We will refer to a calculus as *incomplete* when it does not derive all theorems of its intended, associated logic.

Theorem 8. *Let S be a CFCST system. The calculus* $\mathsf{G3Km}(S)$ *is incomplete relative to labelled tree derivations.*

Proof. Since $\sum$ is non-empty by definition, we know there exists an $x \in \sum$. The claim follows by considering the proof of $p \rightarrow [x]\langle\overline{x}\rangle p$ (an instance of axiom A3), which requires sequents that are not labelled tree sequents. □

3.3 Labelled Calculi for First-Order Intuitionistic Logics

In this section, we provide labelled sequent calculi for first-order intuitionistic logic with non-constant domains (also called, *first-order intuitionistic logic proper*), and first-order intuitionistic logic with constant domains (introduced by Grzegorczyk in [Grz64]). A sequent calculus for the former has been in existence since the initiation of the sequent formalism by Gentzen [Gen35a, Gen35b]. Yet, sequent calculi for the latter were not provided until nearer the close of the century [KS94, FM99]. In [Fit14], Fitting showed that both logics could be uniformly captured within the prefixed tableau and nested sequent formalisms, and also defined translations between the distinct calculi, thus showing their deductive equivalence.[2]

As it so happens, the labelled sequent calculi defined in this section have a natural relationship with Fitting's nested calculi. In [Lyo20b] it was shown that the labelled calculi for the first-order intuitionistic logics IntQ and IntQC can be transformed into Fitting's nested calculi for the logics via the refinement process (with some "additional adjustments"). Also, both [Lyo20a, Pim18] show how labelled calculi for propositional intuitionistic logic can be transformed into nested calculi that are identical to, or equivalent to, Fitting's nested calculus for the logic, respectively. In the next chapter, we will show how refining the labelled calculi for first-order intuitionistic logics begets nested calculi that are related to Fitting's. As will be commented on, these nested variants allow for a higher degree of modularity than Fitting's nested calculi via the modification of side conditions imposed on certain logical rules.

The labelled calculi presented in this section are defined by extending the labelled calculus $\mathsf{G3I}$ for propositional intuitionistic logic of Dyckhoff and Negri [DN12] with quantifier rules and rules concerning domains. Some of these quantifier and domain rules can be found in Viganò [Vig00] and in [NVP11], where the authors provide labelled sequent calculi for first-order modal logics. These works however do not provide rules for the intuitionistic (i.e. strong) universal quantifier or the first-order monotonicity condition **(M)**, and so, we present rules for these in this section. Since our labelled sequent calculi are based on the work and formalisms of [DN12, NVP11, Sim94, Vig00], so are our labelled sequents; these are defined as follows:

Definition 46 (Labelled Sequents for First-Order Intuitionistic Logics)**.** *A* labelled sequent *for first-order intuitionistic logics is a syntactic object of the form $\Lambda := L_1 \Rightarrow L_2$, where L_1 (the* antecedent*) and L_2 (the* consequent*) are defined via the following grammars in BNF:*

$$L_1 ::= \epsilon \mid w : \phi \mid \underline{a} \in D_w \mid w \leq u \mid L_1, L_1 \qquad L_2 ::= \epsilon \mid w : \phi \mid L_2, L_2$$

with $\phi \in \mathcal{L}_{\mathsf{Int}}$, $\underline{a}$ among a denumerable set of parameters $\mathrm{Par} = \{\underline{a}, \underline{b}, \underline{c}, \ldots\}$*, and w, u among a denumerable set of labels* $\mathrm{Lab} = \{w, u, v, \ldots\}$*. We refer to formulae of the forms $w \leq u$ and $\underline{a} \in D_w$ as* relational atoms *(with formulae of the form $\underline{a} \in D_w$ referred to as* domain atoms*, more specifically) and refer to formulae of the form $w : \phi$ as* labelled formulae*.*

[2] In 1983 [Fit83], Fitting sketched a prefixed tableau calculus for *propositional* intuitionistic logic, which stood as a basis for his work in the first-order setting.

We use Λ, Λ', ... (occasionally annotated) to denote labelled sequents as a whole, and due to the two types of formulae occurring in a labelled sequent, we use $\mathcal{R}$, $\mathcal{R}'$, ... (occasionally annotated) to denote multisets of relational atoms, and Γ, Γ', Δ, Δ', ... (occasionally annotated) to denote multisets of labelled formulae, thus distinguishing between the two. We therefore take the comma operator to be commutative and associative; for example, we identify the labelled sequent $w \leq u, w : \phi, \underline{a} \in D_w \Rightarrow v : \psi, u : \chi$ with $\underline{a} \in D_w, w \leq u, w : \phi \Rightarrow u : \chi, v : \psi$. This means that we may write a labelled sequent Λ in a general form as $\mathcal{R}, \Gamma \Rightarrow \Delta$, where we separate the relational atoms from the labelled formulae in the antecedent. Moreover, our interpretation of comma is what lets us view the antecedent $\mathcal{R}, \Gamma$ and the consequent Δ of a labelled sequent $\mathcal{R}, \Gamma \Rightarrow \Delta$ as multisets. We use $\mathrm{Lab}(\Lambda)$, $\mathrm{Lab}(\mathcal{R})$, and $\mathrm{Lab}(\Gamma)$ to denote the sets of labels that occur in a labelled sequent Λ, a multiset $\mathcal{R}$ of relational atoms, and a multiset Γ of labelled formulae, respectively. Also, as in the previous section, we use ϵ to denote the *empty string* which acts as the identity element for the comma operator (e.g. we identify $w \leq v, \epsilon, v : \psi$ with $w \leq v, v : \psi$). Consequently, ϵ will often be implicit in labelled sequents.

We syntactically distinguish between *bound variables* $\{x, y, z, \ldots\}$ and *free variables*, which are replaced with *parameters* $\{\underline{a}, \underline{b}, \underline{c}, \ldots\}$, to avoid clashes between the two categories (cf. [Fit14, Sect. 8]). Instead of making use of formulae from the first-order language $\mathcal{L}_{\mathsf{Int}}$, we use formulae from the first-order language where each freely occurring variable x has been replaced by a distinct parameter $\underline{a}$. For example, we would make use of the labelled formula $w : (\forall x)p(\underline{a}, x) \lor q(\underline{a}, \underline{b})$ instead of $w : (\forall x)p(y, x) \lor q(y, z)$ in a labelled sequent.

We use the notation $\phi(\underline{a}_1, \ldots, \underline{a}_n)$, with $n \in \mathbb{N}$, to denote that the parameters $\underline{a}_1, \ldots, \underline{a}_n$ are all parameters occurring in the formula ϕ. (NB. If $n = 0$, then we assume that the formula ϕ does not contain any parameters). We write $\phi(\vec{\underline{a}})$ as shorthand for $\phi(\underline{a}_1, \ldots, \underline{a}_n)$ and $\vec{\underline{a}} \in D_w$ as shorthand for $\underline{a}_1 \in D_w, \ldots, \underline{a}_n \in D_w$. Furthermore, the notation $\phi(\underline{a}/x)$ represents the formula obtained by substituting the parameter $\underline{a}$ for each free occurrence of x in ϕ. Substitutions of the form $(\underline{a}/x)$ are formally defined as in Def. 19 in Sect. 2.2.

Our labelled calculi are given in Fig. 3.3, and are obtained from the models and semantic clauses of IntQ and IntQC. The derivability relations for our calculi are defined as follows:

Definition 47. *We write* $\vdash_{\mathsf{G3IntQ}} \Lambda$ *and* $\vdash_{\mathsf{G3IntQC}} \Lambda$ *to indicate that the labelled sequent* Λ *is derivable in* $\mathsf{G3IntQ}$ *and* $\mathsf{G3IntQC}$*, respectively.*

The rules for $(\bot_l)$, $(\land_l)$, $(\land_r)$, $(\lor_l)$, $(\lor_r)$, $(\supset_l)$, $(\supset_r)$, $(\exists_l)$, $(\exists_r)$, $(\forall_l)$, and $(\forall_r)$ are rule representations of the semantic clauses given in Def. 18. The (ref) and (tra) rules allow inferences arising from the fact that frames (Def. 16) are reflexive and transitive, whereas (ned), (nd), and (cd) allow inferences based on the fact that the domains of frames are always inhabited (i.e. non-empty), satisfy the nested domain condition **(ND)**, and satisfy the constant domain condition **(CD)**, respectively. The rule (id) encodes the monotonicity condition **(M)** imposed on models (Def. 16). Let us also comment on the new $(\forall_l)$ and $(\forall_r)$ rules for introducing universal quantifiers: since in the first-order intuitionistic setting the satisfaction of a universally quantified formula is determined

$$\frac{}{\mathcal{R}, w \leq u, \vec{\underline{a}} \in D_w, \Gamma, w : p(\vec{\underline{a}}) \Rightarrow u : p(\vec{\underline{a}}), \Delta}\,(id) \qquad \frac{}{\mathcal{R}, \Gamma, w : \bot \Rightarrow \Delta}\,(\bot_l)$$

$$\frac{\mathcal{R}, \Gamma, w : \phi, w : \psi \Rightarrow \Delta}{\mathcal{R}, \Gamma, w : \phi \wedge \psi \Rightarrow \Delta}\,(\wedge_l) \qquad \frac{\mathcal{R}, \Gamma \Rightarrow w : \phi, \Delta \qquad \mathcal{R}, \Gamma \Rightarrow w : \psi, \Delta}{\mathcal{R}, \Gamma \Rightarrow w : \phi \wedge \psi, \Delta}\,(\wedge_r)$$

$$\frac{\mathcal{R}, \Gamma, w : \phi \Rightarrow \Delta \qquad \mathcal{R}, \Gamma, w : \psi \Rightarrow \Delta}{\mathcal{R}, \Gamma, w : \phi \vee \psi \Rightarrow \Delta}\,(\vee_l) \qquad \frac{\mathcal{R}, w \leq w, \Gamma \Rightarrow \Delta}{\mathcal{R}, \Gamma \Rightarrow \Delta}\,(ref)$$

$$\frac{\mathcal{R}, w \leq u, \Gamma, w : \phi \supset \psi \Rightarrow \Delta, u : \phi \qquad \mathcal{R}, w \leq u, \Gamma, w : \phi \supset \psi, u : \psi \Rightarrow \Delta}{\mathcal{R}, w \leq u, \Gamma, w : \phi \supset \psi \Rightarrow \Delta}\,(\supset_l)$$

$$\frac{\mathcal{R}, w \leq u, \Gamma, u : \phi \Rightarrow u : \psi, \Delta}{\mathcal{R}, \Gamma \Rightarrow w : \phi \supset \psi, \Delta}\,(\supset_r)^{\dagger_1} \qquad \frac{\mathcal{R}, \Gamma \Rightarrow w : \phi, w : \psi, \Delta}{\mathcal{R}, \Gamma \Rightarrow w : \phi \vee \psi, \Delta}\,(\vee_r)$$

$$\frac{\mathcal{R}, w \leq u, u \leq v, w \leq v, \Gamma \Rightarrow \Delta}{\mathcal{R}, w \leq u, u \leq v, \Gamma \Rightarrow \Delta}\,(tra) \qquad \frac{\mathcal{R}, \underline{a} \in D_w, \Gamma \Rightarrow \Delta}{\mathcal{R}, \Gamma \Rightarrow \Delta}\,(ned)^{\dagger_2}$$

$$\frac{\mathcal{R}, w \leq u, \underline{a} \in D_u, \Gamma \Rightarrow u : \phi(\underline{a}/x), \Delta}{\mathcal{R}, \Gamma \Rightarrow w : \forall x \phi, \Delta}\,(\forall_r)^{\dagger_3} \qquad \frac{\mathcal{R}, \underline{a} \in D_w, \Gamma \Rightarrow w : \phi(\underline{a}/x), w : \exists x \phi, \Delta}{\mathcal{R}, \underline{a} \in D_w, \Gamma \Rightarrow w : \exists x \phi, \Delta}\,(\exists_r)$$

$$\frac{\mathcal{R}, \underline{a} \in D_w, \Gamma, w : \phi(\underline{a}/x) \Rightarrow \Delta}{\mathcal{R}, \Gamma, w : \exists x \phi \Rightarrow \Delta}\,(\exists_l)^{\dagger_2} \qquad \frac{\mathcal{R}, w \leq u, \underline{a} \in D_u, \Gamma, u : \phi(\underline{a}/x), w : \forall x \phi \Rightarrow \Delta}{\mathcal{R}, w \leq u, \underline{a} \in D_u, \Gamma, w : \forall x \phi \Rightarrow \Delta}\,(\forall_l)$$

$$\frac{\mathcal{R}, w \leq u, \underline{a} \in D_w, \underline{a} \in D_u, \Gamma \Rightarrow \Delta}{\mathcal{R}, w \leq u, \underline{a} \in D_w, \Gamma \Rightarrow \Delta}\,(nd) \qquad \frac{\mathcal{R}, u \leq w, \underline{a} \in D_u, \underline{a} \in D_w, \Gamma \Rightarrow \Delta}{\mathcal{R}, u \leq w, \underline{a} \in D_w, \Gamma \Rightarrow \Delta}\,(cd)$$

Figure 3.3: The labelled calculus G3IntQ for IntQ consists of all rules minus the (cd) rule, and all rules give the calculus G3IntQC for IntQC. The side condition $\dagger_1$ states that the variable u does not occur in the conclusion, $\dagger_2$ states that $\underline{a}$ does not occur in the conclusion, and $\dagger_3$ states that neither $\underline{a}$ nor u occurs in the conclusion. As usual, labels and parameters restricted from occurring in the conclusion of an inference are called *eigenvariables.*

not only by logical information holding at the present world, but at all future worlds, the corresponding logical rules require information at future labels to be considered when applying the rule.

Remark 2. *We note that when the predicate p is of arity* 0 *(i.e. p is a propositional atom), the* (id) *rule is of the following form:*

$$\frac{}{\mathcal{R}, w \leq u, \Gamma, w : p \Rightarrow u : p, \Delta}\,(id)$$

The calculus obtained by taking $(\bot_l)$, $(\wedge_l)$, $(\wedge_r)$, $(\vee_l)$, $(\vee_r)$, $(\supset_l)$, $(\supset_r)$, (ref), (tra), *and* (id) *restricted to the use of propositional atoms (i.e. the form of* (id) *shown above)* *is referred to as **G3I**, and is a sound and complete labelled calculus for propositional intuitionistic logic [DN12]. The above instance of our* (id) *rule generates the initial sequents of **G3I** when we restrict ourselves to the propositional setting [DN12]. Thus, our initial sequents are first-order generalizations of the initial sequents utilized for propositional intuitionistic logic.*

Furthermore, the rules $(\exists_l)$, $(\exists_r)$, (nd), (cd), *and* (ned) *appear in Viganò [Vig00, Ch. 6] and in [NVP11, Ch. 12]; however, the* (id), $(\forall_l)$, *and* $(\forall_r)$ *rules are (to the best of the author's knowledge) new.*

In fact, not only can the above calculi be seen as proof-theoretic encodings of semantic clauses (Def. 18) and frame properties (Def. 16) for IntQ and IntQC, but labelled sequents can be seen as abstractions of IntQ- and IntQC-models. The relationship between the syntactic structures present in labelled sequents (Def. 46) and the semantics for the logics IntQ and IntQC is spelled out formally in the definition below (Def. 48). Moreover, we also leverage the semantics for labelled sequents to define a notion of validity for labelled sequents, which is used in the subsequent theorem (Thm. 9) to prove our calculi sound.

Definition 48 (G3IntQ and G3IntQC Semantics)**.** *Let* $\mathsf{IntX} \in \{\mathsf{IntQ}, \mathsf{IntQC}\}$ *and* $M = (W, \leq, D, V)$ *be an* IntX*-model with* $\mathbf{D} = \bigcup_{w \in W} D_w$. *We let* $I : \mathrm{Lab} \cup \mathrm{Par} \mapsto W \cup \mathbf{D}$ *be an* interpretation function *mapping labels to worlds and parameters to elements of* $\mathbf{D}$.

We define the satisfaction *of a relational atom* $w \leq u$ *or* $\underline{a} \in D_w$ *(written* $M, I \models_{\mathsf{IntX}} w \leq u$ *and* $M, I \models_{\mathsf{IntX}} \underline{a} \in D_w$*, resp.), and labelled formula* $w : \phi$ *(written* $M, I \models_{\mathsf{IntX}} w : \phi$*) as follows:*

- $M, I \models_{\mathsf{IntX}} w \leq u$ *iff* $I(w) \leq I(u)$
- $M, I \models_{\mathsf{IntX}} \underline{a} \in D_w$ *iff* $I(\underline{a}) \in D_{I(w)}$
- $M, I \models_{\mathsf{IntX}} w : \phi$ *iff* $M, I(w) \Vdash \phi$

We say that a multiset of relational atoms $\mathcal{R}$ *is* satisfied *in* M *with* I *(written* $M, I \models_{\mathsf{IntX}} \mathcal{R}$*) iff* $M, I \models_{\mathsf{IntX}} w \leq u$ *and* $M, I \models_{\mathsf{IntX}} \underline{a} \in D_w$ *for all* $w \leq u, \underline{a} \in D_w \in \mathcal{R}$*, and we say that a multiset of labelled formulae* Γ *is* satisfied *in* M *with* I *(written* $M, I \models_{\mathsf{IntX}} \Gamma$*) iff* $M, I \models_{\mathsf{IntX}} w : \phi$ *for all* $w : \phi \in \Gamma$.

A labelled sequent $\Lambda = \mathcal{R}, \Gamma \Rightarrow \Delta$ *is* satisfied *in* M *with* I *(written,* $M, I \models_{\mathsf{IntX}} \Lambda$*) iff if* $M, I \models_{\mathsf{IntX}} \mathcal{R}$ *and* $M, I \models_{\mathsf{IntX}} \Gamma$*, then* $M, I \models_{\mathsf{IntX}} \Delta$*. Also, we say that a labelled sequent* Λ *is* falsified *in* M *with* I *iff* $M, I \not\models_{\mathsf{IntX}} \Lambda$*, that is,* Λ *is not satisfied by* M *with* I.

Last, a labelled sequent Λ *is* IntX-valid *(written* $\models_{\mathsf{IntX}} \Lambda$*) iff it is satisfiable in every* IntX*-model* M *with every interpretation function* I*. We say that a labelled sequent* Λ *is* IntX-invalid *iff* $\not\models_{\mathsf{IntX}} \Lambda$*, i.e.* Λ *is not* IntX*-valid.*

By making use of the above notion of IntQ- and IntQC-validity for labelled sequents, we prove our calculi G3IntQ and G3IntQC sound.

Theorem 9 (Soundness)**.** *Let* $\mathsf{X} \in \{\mathsf{Q}, \mathsf{QC}\}$*. If* $\vdash_{\mathsf{G3IntX}} \Lambda$*, then* $\models_{\mathsf{IntX}} \Lambda$*.*

Proof. Let $\mathsf{IntX} \in \{\mathsf{IntQ}, \mathsf{IntQC}\}$. With the exception of the (id), $(\forall_l)$, and $(\forall_r)$ rules, we can leverage the work of [DN12, NVP11, Vig00] to confirm that all other rules are sound. Therefore, we show that (id) produces only valid sequents, and that $(\forall_l)$ and $(\forall_r)$ preserve validity.

(id) Let $M = (W, \leq, D, V)$ be an arbitrary IntX-model and I be an arbitrary interpretation function. Suppose that the antecedent $\mathcal{R}, w \leq u, \underline{\vec{a}} \in D_w, w : p(\underline{\vec{a}}), \Gamma$ is satisfied in M with I. Then, $I(w) \leq I(u)$, $I(\underline{a}_1) \in D_{I(w)}$, …, $I(\underline{a}_n) \in D_{I(w)}$, and $M, I(w) \Vdash p(\underline{\vec{a}})$. These facts, along with the monotonicity condition **(M)**, implies that $(I(\underline{a}_1), \ldots, I(\underline{a}_n)) \in V(p, I(w)) \subseteq V(p, I(u))$. Hence, $M, I(u) \Vdash p(\underline{\vec{a}})$, showing that any sequent generated by the (id) rule is IntX-valid.

$(\forall_l)$ Let M be an IntX-model and I be an interpretation function such that $M, I \not\models \mathcal{R}, w \leq u, \underline{a} \in D_u, \Gamma, w : \forall x \phi \Rightarrow \Delta$. Thus, $I(w) \leq I(u)$, $I(\underline{a}) \in D_{I(u)}$, and $M, I(w) \Vdash \forall x \phi$. It follows that $M, I(u) \Vdash \phi(\underline{a}/x)$, showing that the premise is falsified by M with I.

$(\forall_r)$ Suppose that M is an IntX-model and I is an interpretation function such that $M, I \not\models \mathcal{R}, \Gamma \Rightarrow w : \forall x \phi, \Delta$. Then, $M, I(w) \nVdash \forall x \phi$, which implies that there exists a world u' and an $a \in D_{u'}$ such that $I(w) \leq u'$ and $M, u' \nVdash \phi(\underline{a}/x)$. Let us define $I'(\underline{b}) = I(\underline{b})$ for $\underline{b} \neq \underline{a}$ and $I'(\underline{a}) = a$, and $I'(v) = I(v)$ for $v \neq u'$ and $I'(u) = u'$. Therefore, $I'(w) \leq I'(u)$, $I'(\underline{a}) \in D_{I'(u)}$, and $M, I'(u) \nVdash \phi(\underline{a}/x)$. Since u and $\underline{a}$ are eigenvariable, it follows that the premise is falsified by M with I'. □

We now go on to show that the labelled calculi possess proof-theoretic properties such as the hp-admissibility of substitutions and structural rules (e.g. (psb) and (wk)), hp-invertibility of all rules, and syntactic cut-elimination. The hp-admissible and eliminable structural rules are presented in Fig. 3.4. Ultimately, these properties will let us prove the completeness of our calculi (Thm. 11). While proving our results, we adapt proofs and results from [DN12, NVP11], though, since these works do not explicitly consider first-order intuitionistic logic with (non-)constant domains, we still consider a few cases in our proofs.

Before moving on to showing that our calculi possess useful properties, we define the notion of a *label substitution* and *parameter substitution*. Intuitively, a label substitution (w/u) on a multiset of relational atoms or labelled formulae replaces all occurrences of the label u with the label w, and a parameter substitution replaces all occurrences of the parameter $\underline{b}$ with the parameter $\underline{a}$. We formally define these operations below:

Definition 49 (Label and Parameter Substitution)**.** *We define a* label substitution (w/u) *for* $w, u \in \mathrm{Lab}$ *on individual relational atoms and labelled formulae as follows:*

$$\frac{\mathcal{R}, \Gamma \Rightarrow \Delta}{\mathcal{R}(w/u), \Gamma(w/u) \Rightarrow \Delta(w/u)} (lsb) \qquad \frac{\mathcal{R}, \Gamma \Rightarrow \Delta}{\mathcal{R}(\underline{a}/\underline{b}), \Gamma(\underline{a}/\underline{b}) \Rightarrow \Delta(\underline{a}/\underline{b})} (psb)$$

$$\frac{\mathcal{R}, \Gamma \Rightarrow \Delta}{\mathcal{R}', \mathcal{R}, \Gamma', \Gamma \Rightarrow \Delta', \Delta} (wk) \quad \frac{\mathcal{R}, \mathcal{R}', \mathcal{R}', \Gamma \Rightarrow \Delta}{\mathcal{R}, \mathcal{R}', \Gamma \Rightarrow \Delta} (ctr_{\mathcal{R}}) \quad \frac{\mathcal{R}, w : \phi, w : \phi, \Gamma \Rightarrow \Delta}{\mathcal{R}, w : \phi, \Gamma \Rightarrow \Delta} (ctr_l)$$

$$\frac{\mathcal{R}, \Gamma \Rightarrow w : \phi, w : \phi, \Delta}{\mathcal{R}, \Gamma \Rightarrow w : \phi, \Delta} (ctr_r) \quad \frac{\mathcal{R}, \Gamma \Rightarrow \Delta, w : A \qquad \mathcal{R}, w : A, \Gamma \Rightarrow \Delta}{\mathcal{R}, \Gamma \Rightarrow \Delta} (cut)$$

Figure 3.4: The set Str(Int) of structural rules consists of all rules shown above.

- $(v \leq z)(w/u) = \begin{cases} w \leq z & \textit{if } u = v \textit{ and } v \neq z \\ w \leq w & \textit{if } u = v \textit{ and } v = z \\ u \leq w & \textit{if } u = z \textit{ and } v \neq z \\ v \leq z & \textit{otherwise} \end{cases}$

- $(\underline{a} \in D_v)(w/u) = \begin{cases} \underline{a} \in D_w & \textit{if } u = v \\ \underline{a} \in D_v & \textit{otherwise} \end{cases}$

- $(v : \phi)(w/u) = \begin{cases} w : \phi & \textit{if } u = v \\ v : \phi & \textit{otherwise} \end{cases}$

We define the label substitution (w/u) *on a multiset of relational atoms* $\mathcal{R}$ *and a multiset of labelled formulae* Γ *to be the multiset obtained by applying* (w/u) *to each element of the multiset.*

We define a parameter substitution $(\underline{a}/\underline{b})$ *for* $\underline{a}, \underline{b} \in \mathrm{Par}$ *on individual relational atoms and labelled formulae as follows:*

- $(w \leq u)(\underline{a}/\underline{b}) := w \leq u$
- $(\underline{c} \in D_v)(\underline{a}/\underline{b}) := \underline{a} \in D_v$ *if* $\underline{b} = \underline{c}$, *and* $(\underline{c} \in D_v)(\underline{a}/\underline{b}) := \underline{c} \in D_v$ *otherwise.*
- $(w : \phi)(\underline{a}/\underline{b}) := w : \phi(\underline{a}/\underline{b})$

Note that for the last displayed parameter substitution (on a labelled formula) we make use of the notion of a parameter substitution on a formula, which is defined as in Def. 19.

We define the parameter substitution $(\underline{a}/\underline{b})$ *on a multiset of relational atoms* $\mathcal{R}$ *and a multiset of labelled formulae* Γ *to be the multiset obtained by applying* $(\underline{a}/\underline{b})$ *to each element of the multiset.*

The first property we prove below is fundamental in securing completeness (Thm. 11), and ensures that all *instances* of the axioms in HIntQ and HIntQC (Def. 20) are derivable in $\mathsf{G3IntQ}$ and $\mathsf{G3IntQC}$, respectively.

Lemma 13. *Let* $\mathsf{G3X} \in \{\mathsf{G3IntQ}, \mathsf{G3IntQC}\}$.

(i) For all $\phi \in \mathcal{L}_{\mathsf{Int}}$, $\vdash_{\mathsf{G3X}} \mathcal{R}, w \leq u, \underline{\vec{a}} \in D_w, w : \phi(\underline{\vec{a}}), \Gamma \Rightarrow u : \phi(\underline{\vec{a}}), \Delta$.

(ii) For all $\phi \in \mathcal{L}_{\mathsf{Int}}$, $\vdash_{\mathsf{G3X}} \mathcal{R}, \underline{\vec{a}} \in D_w, w : \phi(\underline{\vec{a}}), \Gamma \Rightarrow \Delta, w : \phi(\underline{\vec{a}})$.

Proof. We prove claims (i) and (ii) for both calculi $\mathsf{G3IntQ}$ and $\mathsf{G3IntQC}$ simultaneously by mutual induction on the complexity of ϕ.

Base case for (i). The base cases for claim (i) trivially follow from the (id) rule for atomic formulae and propositional variables, and from the $(\bot_l)$ rule for $\bot$.

Inductive step for (i). We provide the cases for when $\phi(\underline{\vec{a}})$ is of the form $\exists x\psi(\underline{\vec{a}})$ and $\forall x\psi(\underline{\vec{a}})$, and omit the remaining cases which follow from [DN12, Lem. 1]. In the $\exists x\psi(\underline{\vec{a}})$ case, we let $\Delta' := u : \exists x\psi(\vec{a}), \Delta$, and in the $\forall x\psi(\underline{\vec{a}})$ case, we let $\mathcal{R}' = \mathcal{R}, w \leq u, u \leq v, \underline{\vec{a}} \in D_w$.

$$\dfrac{\dfrac{\dfrac{}{\mathcal{R}, w \leq u, \underline{b} \in D_w, \underline{\vec{a}} \in D_w, w : \psi(\underline{\vec{a}})(\underline{b}/x), \Gamma \Rightarrow u : \psi(\underline{\vec{a}})(\underline{b}/x), \Delta'}\text{IH}}{\mathcal{R}, w \leq u, \underline{b} \in D_w, \underline{\vec{a}} \in D_w, w : \psi(\vec{a})(\underline{b}/x), \Gamma \Rightarrow u : \exists x\psi(\vec{a}), \Delta}(\exists_r)}{\mathcal{R}, w \leq u, \vec{a} \in D_w, w : \exists x\psi(\vec{a}), \Gamma \Rightarrow u : \exists x\psi(\vec{a}), \Delta}(\exists_l)$$

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\mathcal{R}', w \leq v, \underline{b} \in D_v, v : \psi(\underline{\vec{a}})(\underline{b}/x), w : \forall x\psi(\underline{\vec{a}}), \Gamma \Rightarrow v : \psi(\underline{\vec{a}})(\underline{b}/x), \Delta}\text{IH}}{\mathcal{R}', w \leq v, \underline{b} \in D_v, w : \forall x\psi(\underline{\vec{a}}), \Gamma \Rightarrow v : \psi(\underline{\vec{a}})(\underline{b}/x), \Delta}(\forall_l)}{\mathcal{R}', \underline{b} \in D_v, w : \forall x\psi(\underline{\vec{a}}), \Gamma \Rightarrow v : \psi(\underline{\vec{a}})(\underline{b}/x), \Delta}(tra)}{\mathcal{R}, w \leq u, \underline{\vec{a}} \in D_w, w : \forall x\psi(\vec{a}), \Gamma \Rightarrow u : \forall x\psi(\vec{a}), \Delta}(\forall_r)$$

Base case for (ii). The base case for atomic formulae is shown below (the base case for propositional variables is similar), and the case for when ϕ is of the form $\bot$ is omitted as it is simple to verify using the $(\bot_l)$ rule.

$$\dfrac{\dfrac{}{\mathcal{R}, w \leq w, \underline{\vec{a}} \in D_w, w : p(\underline{\vec{a}}), \Gamma \Rightarrow w : p(\underline{\vec{a}}), \Delta}(id)}{\mathcal{R}, \underline{\vec{a}} \in D_w, w : p(\underline{\vec{a}}), \Gamma \Rightarrow w : p(\underline{\vec{a}}), \Delta}(ref)$$

Inductive step for (ii). We only show the case when $\phi(\underline{\vec{a}})$ is of the form $\forall x\psi(\underline{\vec{a}})$, as all other cases follow from [DN12, Lem. 1] and [NVP11, Lem. 12.1].

$$\dfrac{\dfrac{\dfrac{}{\mathcal{R}, w \leq u, \underline{b} \in D_u, \underline{\vec{a}} \in D_w, u : \psi(\underline{\vec{a}})(\underline{b}/x), w : \forall x\psi(\underline{\vec{a}}), \Gamma \Rightarrow u : \psi(\underline{\vec{a}})(\underline{b}/x), \Delta}\text{IH}}{\mathcal{R}, w \leq u, \underline{b} \in D_u, \underline{\vec{a}} \in D_w, w : \forall x\psi(\vec{a}), \Gamma \Rightarrow u : \psi(\underline{\vec{a}})(\underline{b}/x), \Delta}(\forall_l)}{\mathcal{R}, \underline{\vec{a}} \in D_w, w : \forall x\psi(\underline{\vec{a}}), \Gamma \Rightarrow w : \forall x\psi(\underline{\vec{a}}), \Delta}(\forall_r)$$

□

Lemma 14. *The rule* (lsb) *is hp-admissible in* G3IntQ *and* G3IntQC.

Proof. We prove the result by induction on the height of the given derivation.

Base case. The base case is easily resolved as any application of (lsb) to (id) or ($\bot_l$) yields another instance of the rule.

Inductive step. We need only consider the ($\forall_l$) and ($\forall_r$) cases, as all other cases follow from [DN12, Lem. 3] and [NVP11, Lem. 12.4]. The ($\forall_l$) case is handled by applying IH and then the corresponding rule. The non-trivial ($\forall_r$) case arises when the substitution introduces the eigenvariable of the ($\forall_r$) inference. We show how to resolve this non-trivial case below, and omit the trivial case as it follows by applying IH and then the corresponding rule. In what follows, we let z be a fresh label.

$$
\dfrac{\dfrac{\mathcal{R}, w \leq u, \underline{a} \in D_u, \Gamma \Rightarrow u : \phi(\underline{a}/x), \Delta}{\mathcal{R}, \Gamma \Rightarrow w : \forall x \phi, \Delta}\ (\forall_r)}{\mathcal{R}(u/v), \Gamma(u/v) \Rightarrow (w : \forall x \phi)(u/v), \Delta(u/v)}\ (lsb) \quad \rightsquigarrow
$$

$$
\begin{array}{cl}
\mathcal{R}, w \leq u, \underline{a} \in D_u, \Gamma \Rightarrow u : \phi(\underline{a}/x), \Delta & \\
\hdashline
\mathcal{R}(z/u), (w \leq u)(z/u), (\underline{a} \in D_u)(z/u), \Gamma(z/u) \Rightarrow (u : \phi(\underline{a}/x))(z/u), \Delta(z/u) & \text{IH} \\
\hdashline
\mathcal{R}, w \leq z, \underline{a} \in D_z, \Gamma \Rightarrow z : \phi(\underline{a}/x), \Delta & = \\
\hdashline
\mathcal{R}(u/v), (w \leq z)(u/v), (\underline{a} \in D_z)(u/v), \Gamma(u/v) \Rightarrow (z : \phi(\underline{a}/x))(u/v), \Delta(u/v) & \text{IH} \\
\hdashline
\mathcal{R}(u/v), w \leq z, \underline{a} \in D_z, \Gamma(u/v) \Rightarrow z : \phi(\underline{a}/x), \Delta(u/v) & = \\
\hline
\mathcal{R}(u/v), \Gamma(u/v) \Rightarrow (w : \forall x \phi)(u/v), \Delta(u/v) & (\forall_r)
\end{array}
$$

□

Lemma 15. *The rule* (psb) *is hp-admissible in* G3IntQ *and* G3IntQC.

Proof. We prove the claim by induction on the height of the given derivation.

Base case. The base is trivial as any application of (lsb) to an instance of (id) or ($\bot_l$) yields another instance of the rule.

Inductive step. With the exception of the ($\exists_l$), ($\forall_r$), and (ned) cases, all other cases follow by applying IH and then the corresponding rule. The ($\exists_l$) and (ned) cases are handled as explained in [NVP11, Lem. 12.4]. The non-trivial ($\forall_r$) case arises when the parameter substitution ($\underline{a}/\underline{b}$) introduces the eigenvariable of the ($\forall_r$) inference. We show how to resolve the non-trivial case below, and omit the trivial case as it follows from invoking IH followed by the corresponding rule. In the case below, we let $\underline{c}$ be a fresh parameter.

$$
\dfrac{\dfrac{\mathcal{R}, w \leq u, \underline{a} \in D_u, \Gamma \Rightarrow u : \phi(\underline{a}/x), \Delta}{\mathcal{R}, \Gamma \Rightarrow w : \forall x \phi, \Delta}\ (\forall_r)}{\mathcal{R}(\underline{a}/\underline{b}), \Gamma(\underline{a}/\underline{b}) \Rightarrow (w : \forall x \phi)(\underline{a}/\underline{b}), \Delta(\underline{a}/\underline{b})}\ (psb) \quad \rightsquigarrow
$$

$$\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\mathcal{R}, w \leq u, \underline{a} \in D_u, \Gamma \Rightarrow u : \phi(\underline{a}/x), \Delta}{\mathcal{R}(\underline{c}/\underline{a}), (w \leq u)(\underline{c}/\underline{a}), (\underline{a} \in D_u)(\underline{c}/\underline{a}), \Gamma(\underline{c}/\underline{a}) \Rightarrow (u : \phi(\underline{a}/x))(\underline{c}/\underline{a}), \Delta(\underline{c}/\underline{a})}\,\text{IH}}{\mathcal{R}, w \leq u, \underline{c} \in D_u, \Gamma \Rightarrow u : \phi(\underline{c}/x), \Delta}\,=}{\mathcal{R}(\underline{a}/\underline{b}), (w \leq u)(\underline{a}/\underline{b}), (\underline{c} \in D_u)(\underline{a}/\underline{b}), \Gamma(\underline{a}/\underline{b}) \Rightarrow (u : \phi(\underline{c}/x))(\underline{a}/\underline{b}), \Delta(\underline{a}/\underline{b})}\,\text{IH}}{\mathcal{R}(\underline{a}/\underline{b}), w \leq u, \underline{c} \in D_u, \Gamma(\underline{a}/\underline{b}) \Rightarrow u : \phi(\underline{c}/x)(\underline{a}/\underline{b}), \Delta(\underline{a}/\underline{b})}\,=}{\mathcal{R}(\underline{a}/\underline{b}), \Gamma(\underline{a}/\underline{b}) \Rightarrow (w : \forall x\phi)(\underline{a}/\underline{b}), \Delta(\underline{a}/\underline{b})}\,(\forall_r)$$

□

Lemma 16. *The rule* (wk) *is hp-admissible in* G3IntQ *and* G3IntQC.

Proof. We prove the result by induction on the height of the given derivation.

Base case. The base case is straightforward since any application of (wk) to (id) or $(\bot_l)$ yields another instance of the rule.

Inductive step. The only non-trivial cases concern $(\supset_r)$, $(\exists_l)$, $(\forall_r)$, and (ned), and arise when (wk) introduces a parameter or label identical to an eigenvariable of one of the aforementioned inferences. The $(\supset_r)$ case is resolved as in [DN12, Prop. 1], and the $(\exists_l)$ and (ned) cases are resolved as in [NVP11, Thm. 12.5]. Therefore, we only show how to resolve non-trivial $(\forall_r)$ case below. We let $\underline{c}$ and z be a fresh parameter and label, respectively.

$$\cfrac{\cfrac{\mathcal{R}, w \leq u, \underline{a} \in D_u, \Gamma \Rightarrow u : \phi(\underline{a}/x), \Delta}{\mathcal{R}, \Gamma \Rightarrow w : \forall x\phi, \Delta}\,(\forall_r)}{\mathcal{R}, \mathcal{R}', \Gamma, \Gamma' \Rightarrow w : \forall x\phi, \Delta, \Delta'}\,(wk) \quad \rightsquigarrow$$

$$\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\mathcal{R}, w \leq u, \underline{a} \in D_u, \Gamma \Rightarrow u : \phi(\underline{a}/x), \Delta}{\mathcal{R}(z/u), (w \leq u)(z/u), (\underline{a} \in D_u)(z/u), \Gamma(z/u) \Rightarrow (u : \phi(\underline{a}/x))(z/u), \Delta(z/u)}\,(lsb)}{\mathcal{R}, w \leq z, \underline{a} \in D_z, \Gamma \Rightarrow z : \phi(\underline{a}/x), \Delta}\,=}{\mathcal{R}(\underline{c}/\underline{a}), (w \leq z)(\underline{c}/\underline{a}), (\underline{a} \in D_z)(\underline{c}/\underline{a}), \Gamma(\underline{c}/\underline{a}) \Rightarrow (z : \phi(\underline{a}/x))(\underline{c}/\underline{a}), \Delta(\underline{c}/\underline{a})}\,(psb)}{\mathcal{R}, w \leq z, \underline{c} \in D_z, \Gamma \Rightarrow z : \phi(\underline{c}/x), \Delta}\,=}{\mathcal{R}, \mathcal{R}', w \leq z, \underline{c} \in D_z, \Gamma, \Gamma' \Rightarrow z : \phi(\underline{c}/x), \Delta, \Delta'}\,\text{IH}}{\mathcal{R}, \mathcal{R}', \Gamma, \Gamma' \Rightarrow w : \forall x\phi, \Delta, \Delta'}\,(\forall_r)$$

□

Lemma 17. *All rules are hp-invertible in* G3IntQ *and* G3IntQC.

Proof. The result is shown by induction on the height of the given derivation, and is proven similarly to [DN12, Prop. 2] and [NVP11, Thm. 12.7]. □

Lemma 18. *The rules* $(ctr_{\mathcal{R}})$, (ctr_l), *and* (ctr_r) *are hp-admissible in* G3IntQ *and* G3IntQC.

Proof. The result is shown by induction on the height of the given derivation, and is proven similarly to [DN12, Thm. 3] and [NVP11, Thm. 12.8]. □

Theorem 10. *The rule* (*cut*) *is eliminable in* G3IntQ *and* G3IntQC.

Proof. The result is shown by induction on the lexicographic ordering of pairs $(|\phi|, h_1+h_2)$, where $|\phi|$ is the complexity of the cut formula ϕ, h_1 is the height of the derivation of the left premise of (*cut*), and h_2 is the height of the derivation of the right premise of (*cut*). We may assume w.l.o.g. that (*cut*) is the last rule used in our given derivation, and that no other instances of (*cut*) appear in the given derivation. The general result follows by repeatedly applying the procedure given in the proof to successively eliminate topmost instances of (*cut*) until the derivation is free of such inferences.

Due to the modularity of the labelled formalism, we may leverage the results [DN12, Thm. 4] and [NVP11, Thm. 12.9] to cover all cases with the exception of the case where the cut formula is of the form $\forall x \psi$ and is principal in both premises of (*cut*). This case is resolved as shown below:

$$\Pi_1 \quad = \left\{ \quad \dfrac{\mathcal{R}, w \leq v, w \leq u, \underline{a} \in D_v, \underline{b} \in D_u, \Gamma \Rightarrow \Delta, u : \psi(\underline{b}/x)}{\mathcal{R}, w \leq v, \underline{a} \in D_v, \Gamma \Rightarrow \Delta, w : \forall x \psi} \; (\forall_r) \right.$$

$$\Pi_2 \quad = \left\{ \quad \dfrac{\mathcal{R}, w \leq v, \underline{a} \in D_v, v : \psi(\underline{a}/x), w : \forall x \psi, \Gamma \Rightarrow \Delta}{\mathcal{R}, w \leq v, \underline{a} \in D_v, w : \forall x \psi, \Gamma \Rightarrow \Delta} \; (\forall_l) \right.$$

$$\dfrac{\Pi_1 \qquad \Pi_2}{\mathcal{R}, w \leq v, \underline{a} \in D_v, \Gamma \Rightarrow \Delta} \; (cut) \quad \rightsquigarrow \quad \dfrac{\Pi_1' \qquad \Pi_2'}{\mathcal{R}, w \leq v, \underline{a} \in D_v, \Gamma \Rightarrow \Delta} \; \mathrm{IH}$$

$$\Pi_1' \quad = \left\{ \quad \dfrac{\dfrac{\mathcal{R}, w \leq u, \underline{b} \in D_u, \Gamma \Rightarrow \Delta, u : \psi(\underline{b}/x)}{\mathcal{R}, w \leq v, \underline{b} \in D_v, \Gamma \Rightarrow \Delta, v : \psi(\underline{b}/x)} \; (lsb)}{\mathcal{R}, w \leq v, \underline{a} \in D_v, \Gamma \Rightarrow \Delta, v : \psi(\underline{a}/x)} \; (psb) \right.$$

$$\Pi_2' \quad = \left\{ \quad \dfrac{\dfrac{\dfrac{\mathcal{R}, w \leq v, w \leq u, \underline{a} \in D_v, \underline{b} \in D_u, \Gamma \Rightarrow \Delta, u : \psi(\underline{b}/x)}{\mathcal{R}, w \leq v, w \leq u, \underline{a} \in D_v, \underline{b} \in D_u, v : \psi(\underline{a}/x), \Gamma \Rightarrow \Delta, u : \psi(\underline{b}/x)} \; (wk)}{\mathcal{R}, w \leq v, \underline{a} \in D_v, v : \psi(\underline{a}/x), \Gamma \Rightarrow \Delta, w : \forall x \psi} \; (\forall_r) \qquad \Pi_3'}{\mathcal{R}, w \leq v, \underline{a} \in D_v, v : \psi(\underline{a}/x), \Gamma \Rightarrow \Delta} \; \mathrm{IH} \right.$$

$$\Pi_3' = \left\{ \quad \mathcal{R}, w \leq v, \underline{a} \in D_v, v : \psi(\underline{a}/x), w : \forall x \psi, \Gamma \Rightarrow \Delta \right.$$

Note that we may invoke IH in Π_2' since the sum of the heights $h_1 + h_2$ is one less than the original (*cut*), and we may invoke the second use of IH since the cut formula $v : \psi(\underline{a}/x)$ is of a smaller complexity, that is $|\psi(\underline{a}/x)| < |\forall x \psi| = |\phi|$. □

Theorem 11 (Completeness)**.** *Let $\phi \in \mathcal{L}_{\mathsf{Int}}$, and let $\phi(\underline{\vec{a}}) := \phi(\underline{a}_1/x_1) \cdots (\underline{a}_n/x_n)$, that is, $\phi(\underline{\vec{a}})$ is the formula ϕ where the parameters $\underline{\vec{a}} = a_1, \ldots, a_n$ have been substituted for each respective free variable $x_1, \ldots, x_n$ in ϕ.*

(i) If $\vdash_{\mathsf{IntQ}} \phi$, then $\vdash_{\mathsf{G3IntQ}} \underline{\vec{a}} \in D_w \Rightarrow w : \phi(\underline{\vec{a}})$.

(ii) If $\vdash_{\mathsf{IntQC}} \phi$, then $\vdash_{\mathsf{G3IntQC}} \underline{\vec{a}} \in D_w \Rightarrow w : \phi(\underline{\vec{a}})$.

Proof. By Rmk. 2, we know that our calculi G3IntQ and G3IntQC can derive all axioms A0 – A8, that is, the axioms for propositional intuitionistic logic (cf. Def. 20 and [GSS09, p. 6]). Therefore, we need only show that they can derive the axioms A9 – A12 and A9 – A13, respectively, as well as simulate the inference rules R0 and R1 (see Def. 20 for the axioms of IntQ and IntQC). We show this below, and note that the only proof that makes use of the constant domain rule (*cd*) is the proof of axiom A13 (the constant domain axiom). The conclusion of each derivation displays the axiom derived.

Axiom A9. We present the case below where the universal quantifier is not vacuous, and note that the case where the universal quantifier is vacuous is similar.

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{w \leq u, u \leq u, \underline{\vec{a}} \in D_w, \underline{a} \in D_w, \underline{a} \in D_u, \underline{\vec{a}} \in D_u, u : \forall x \phi(\underline{\vec{a}}, x), u : \phi(\underline{\vec{a}}, \underline{a}) \Rightarrow u : \phi(\underline{\vec{a}}, \underline{a})}\ \text{Lem. 13}}{w \leq u, u \leq u, \underline{\vec{a}} \in D_w, \underline{a} \in D_w, \underline{a} \in D_u, \underline{\vec{a}} \in D_u, u : \forall x \phi(\underline{\vec{a}}, x) \Rightarrow u : \phi(\underline{\vec{a}}, \underline{a})}\ (\forall_l)}{w \leq u, u \leq u, \underline{\vec{a}} \in D_w, \underline{a} \in D_w, u : \forall x \phi(\underline{\vec{a}}, x) \Rightarrow u : \phi(\underline{\vec{a}}, \underline{a})}\ (nd) \times n}{w \leq u, \underline{\vec{a}} \in D_w, \underline{a} \in D_w, u : \forall x \phi(\underline{\vec{a}}, x) \Rightarrow u : \phi(\underline{\vec{a}}, \underline{a})}\ (ref)}{\underline{\vec{a}} \in D_w, \underline{a} \in D_w \Rightarrow w : \forall x \phi(\underline{\vec{a}}, x) \supset \phi(\underline{\vec{a}}, \underline{a})}\ (\supset_r)
$$

Axiom A10. Similar to the previous case, we show the case where the existential quantifier is not vacuous. The case where the quantifier is vacuous is proven similarly.

$$
\dfrac{\dfrac{\dfrac{\dfrac{}{w \leq u, \underline{\vec{a}} \in D_w, \underline{a} \in D_w, \underline{\vec{a}} \in D_u, \underline{a} \in D_u, u : \phi(\underline{\vec{a}}, \underline{a}) \Rightarrow u : \phi(\underline{\vec{a}}, \underline{a}), u : \exists x \phi(\underline{\vec{a}}, x)}\ \text{Lem. 13}}{w \leq u, \underline{\vec{a}} \in D_w, \underline{a} \in D_w, \underline{\vec{a}} \in D_u, \underline{a} \in D_u, u : \phi(\underline{\vec{a}}, \underline{a}) \Rightarrow u : \exists x \phi(\underline{\vec{a}}, x)}\ (\exists_r)}{w \leq u, \underline{\vec{a}} \in D_w, \underline{a} \in D_w, u : \phi(\underline{\vec{a}}, \underline{a}) \Rightarrow u : \exists x \phi(\underline{\vec{a}}, x)}\ (nd) \times n}{\underline{\vec{a}} \in D_w, \underline{a} \in D_w \Rightarrow w : \phi(\underline{\vec{a}}, \underline{a}) \supset \exists x \phi(\underline{\vec{a}}, x)}\ (\supset_r)
$$

Axiom A11. We show how to derive the axiom A11 below when the universal quantifier is not vacuous, and omit the proof of the axiom when the universal quantifier is vacuous as the proof is similar. To save space and improve readability, we let $\mathcal{R} := w \leq v, v \leq u, u \leq z, \underline{\vec{a}} \in D_w, \underline{\vec{b}} \in D_w, \underline{a} \in D_z$, $\mathcal{R}' := \mathcal{R}, v \leq z, z \leq z$, and $\Gamma := u : \psi(\underline{\vec{b}}) \supset \phi(\underline{\vec{a}}, \underline{a}), v : \forall x (\psi(\underline{\vec{b}}) \supset \phi(\underline{\vec{a}}, x))$. Moreover, we suppose that k_1 is equal to the number of domain atoms $\underline{\vec{b}} \in D_w = \underline{b}_1 \in D_w, \ldots, \underline{b}_{k_1} \in D_w$, and k_2 is equal to the number of domain atoms $\underline{\vec{a}} \in D_w = \underline{a}_1 \in D_w, \ldots, \underline{a}_{k_2} \in D_w$.

$$
\Pi_1 \quad = \left\{ \quad \dfrac{\dfrac{}{\mathcal{R}', \underline{\vec{b}} \in D_v, \underline{\vec{b}} \in D_u, \Gamma, u : \psi(\underline{\vec{b}}) \Rightarrow z : \psi(\underline{\vec{b}}), z : \phi(\underline{\vec{a}}, \underline{a})}\ \text{Lem. 13}}{\mathcal{R}', \Gamma, u : \psi(\underline{\vec{b}}) \Rightarrow z : \psi(\underline{\vec{b}}), z : \phi(\underline{\vec{a}}, \underline{a})}\ (nd) \times 2k_1 \right.
$$

$$\Pi_2 \quad = \left\{ \begin{array}{cl} \hdashline \mathcal{R}', \underline{\vec{a}} \in D_v, \underline{\vec{a}} \in D_u, \underline{\vec{a}} \in D_z, \Gamma, u : \psi(\underline{\vec{b}}), z : \phi(\underline{\vec{a}}, \underline{a}) \Rightarrow z : \phi(\underline{\vec{a}}, \underline{a}) & \text{Lem. 13} \\ \hline \mathcal{R}', \Gamma, u : \psi(\underline{\vec{b}}), z : \phi(\underline{\vec{a}}, \underline{a}) \Rightarrow z : \phi(\underline{\vec{a}}, \underline{a}) & (nd) \times 3k_2 \end{array} \right.$$

$$\begin{array}{cl}
\Pi_1 \qquad \Pi_2 & \\
\hline
\mathcal{R}, v \leq z, z \leq z, z : \psi(\underline{\vec{b}}) \supset \phi(\underline{\vec{a}}, \underline{a}), v : \forall x(\psi(\underline{\vec{b}}) \supset \phi(\underline{\vec{a}}, x)), u : \psi(\underline{\vec{b}}) \Rightarrow z : \phi(\underline{\vec{a}}, \underline{a}) & (\supset_l) \\
\hline
\mathcal{R}, v \leq z, z : \psi(\underline{\vec{b}}) \supset \phi(\underline{\vec{a}}, \underline{a}), v : \forall x(\psi(\underline{\vec{b}}) \supset \phi(\underline{\vec{a}}, x)), u : \psi(\underline{\vec{b}}) \Rightarrow z : \phi(\underline{\vec{a}}, \underline{a}) & (ref) \\
\hline
\mathcal{R}, v \leq z, v : \forall x(\psi(\underline{\vec{b}}) \supset \phi(\underline{\vec{a}}, x)), u : \psi(\underline{\vec{b}}) \Rightarrow z : \phi(\underline{\vec{a}}, \underline{a}) & (\forall_l) \\
\hline
\mathcal{R}, v : \forall x(\psi(\underline{\vec{b}}) \supset \phi(\underline{\vec{a}}, x)), u : \psi(\underline{\vec{b}}) \Rightarrow z : \phi(\underline{\vec{a}}, \underline{a}) & (tra) \\
\hline
w \leq v, v \leq u, \underline{\vec{a}} \in D_w, \underline{\vec{b}} \in D_w, v : \forall x(\psi(\underline{\vec{b}}) \supset \phi(\underline{\vec{a}}, x)), u : \psi(\underline{\vec{b}}) \Rightarrow u : \forall x \phi(\underline{\vec{a}}, x) & (\forall_r) \\
\hline
w \leq v, \underline{\vec{a}} \in D_w, \underline{\vec{b}} \in D_w, v : \forall x(\psi(\underline{\vec{b}}) \supset \phi(\underline{\vec{a}}, x)) \Rightarrow v : \psi(\underline{\vec{b}}) \supset \forall x \phi(\underline{\vec{a}}, x) & (\supset_r) \\
\hline
\underline{\vec{a}} \in D_w, \underline{\vec{b}} \in D_w \Rightarrow w : \forall x(\psi(\underline{\vec{b}}) \supset \phi(\underline{\vec{a}}, x)) \supset (\psi(\underline{\vec{b}}) \supset \forall x \phi(\underline{\vec{a}}, x)) & (\supset_r)
\end{array}$$

Axiom A12. We show how to derive axiom A12 below and only consider the case where the quantifiers are non-vacuous, since the case where the quantifiers vacuously quantify ψ is similar. To save space and improve readability, we let $\mathcal{R} := w \leq v, v \leq u, \underline{\vec{a}} \in D_w, \underline{\vec{b}} \in D_w, \underline{a} \in D_u$ and $\Gamma := u : \psi(\underline{\vec{b}}, \underline{a}) \supset \phi(\underline{\vec{a}}), v : \forall x(\psi(\underline{\vec{b}}, x) \supset \phi(\underline{\vec{a}}))$. Also, we let k_1 be equal to the number of domain atoms $\underline{\vec{b}} \in D_w = \underline{b}_1 \in D_w, \ldots, \underline{b}_{k_1} \in D_w$, and k_2 be equal to the number of domain atoms $\underline{\vec{a}} \in D_w = \underline{a}_1 \in D_w, \ldots, \underline{a}_{k_2} \in D_w$.

$$\Pi_1 \quad = \left\{ \begin{array}{cl} \hdashline \mathcal{R}, u \leq u, \underline{\vec{b}} \in D_v, \underline{\vec{b}} \in D_u, \Gamma, u : \psi(\underline{\vec{b}}, \underline{a}) \Rightarrow u : \psi(\underline{\vec{b}}, \underline{a}), u : \phi(\underline{\vec{a}}) & \text{Lem. 13} \\ \hline \mathcal{R}, u \leq u, \Gamma, u : \psi(\underline{\vec{b}}, \underline{a}) \Rightarrow u : \psi(\underline{\vec{b}}, \underline{a}), u : \phi(\underline{\vec{a}}) & (nd) \times 2k_1 \end{array} \right.$$

$$\Pi_2 \quad = \left\{ \begin{array}{cl} \hdashline \mathcal{R}, u \leq u, \underline{\vec{a}} \in D_v, \underline{\vec{a}} \in D_u, \Gamma, u : \psi(\underline{\vec{b}}, \underline{a}), u : \phi(\underline{\vec{a}}) \Rightarrow u : \phi(\underline{\vec{a}}) & \text{Lem. 13} \\ \hline \mathcal{R}, u \leq u, \Gamma, u : \psi(\underline{\vec{b}}, \underline{a}), u : \phi(\underline{\vec{a}}) \Rightarrow u : \phi(\underline{\vec{a}}) & (nd) \times 2k_2 \end{array} \right.$$

$$\begin{array}{cl}
\Pi_1 \qquad \Pi_2 & \\
\hline
\mathcal{R}, u \leq u, u : \psi(\underline{\vec{b}}, \underline{a}) \supset \phi(\underline{\vec{a}}), v : \forall x(\psi(\underline{\vec{b}}, x) \supset \phi(\underline{\vec{a}})), u : \psi(\underline{\vec{b}}, \underline{a}) \Rightarrow u : \phi(\underline{\vec{a}}) & (\supset_l) \\
\hline
\mathcal{R}, u : \psi(\underline{\vec{b}}, \underline{a}) \supset \phi(\underline{\vec{a}}), v : \forall x(\psi(\underline{\vec{b}}, x) \supset \phi(\underline{\vec{a}})), u : \psi(\underline{\vec{b}}, \underline{a}) \Rightarrow u : \phi(\underline{\vec{a}}) & (ref) \\
\hline
\mathcal{R}, v : \forall x(\psi(\underline{\vec{b}}, x) \supset \phi(\underline{\vec{a}})), u : \psi(\underline{\vec{b}}, \underline{a}) \Rightarrow u : \phi(\underline{\vec{a}}) & (\forall_l) \\
\hline
w \leq v, v \leq u, \underline{\vec{a}} \in D_w, \underline{\vec{b}} \in D_w, v : \forall x(\psi(\underline{\vec{b}}, x) \supset \phi(\underline{\vec{a}})), u : \exists x \psi(\underline{\vec{b}}, x) \Rightarrow u : \phi(\underline{\vec{a}}) & (\exists_l) \\
\hline
w \leq v, \underline{\vec{a}} \in D_w, \underline{\vec{b}} \in D_w, v : \forall x(\psi(\underline{\vec{b}}, x) \supset \phi(\underline{\vec{a}})) \Rightarrow v : \exists x \psi(\underline{\vec{b}}, x) \supset \phi(\underline{\vec{a}}) & (\supset_r) \\
\hline
\underline{\vec{a}} \in D_w, \underline{\vec{b}} \in D_w \Rightarrow w : \forall x(\psi(\underline{\vec{b}}, x) \supset \phi(\underline{\vec{a}})) \supset (\exists x \psi(\underline{\vec{b}}, x) \supset \phi(\underline{\vec{a}})) & (\supset_r)
\end{array}$$

Axiom A13. Below, we derive the constant domain axiom A13 and consider the case where the universal quantifier non-vacuously quantifiers over ϕ; the case where the universal quantifier is vacuous is similar. Note that the derivation requires an application of the constant domain rule (cd), showing that the axiom is derivable in G3IntQC, but not G3IntQ (see proof Π_1 below). To save space and improve readability, we let $\mathcal{R} := w \leq v, v \leq u, \underline{\vec{a}} \in D_w, \underline{\vec{b}} \in D_w, \underline{a} \in D_u$.

$$\Pi_1 = \left\{ \dfrac{\dfrac{}{\mathcal{R}, \underline{a} \in D_u, v \leq v, \underline{\vec{a}} \in D_v, v : \phi(\underline{\vec{a}}, \underline{a}), v : \forall x(\phi(\underline{\vec{a}}, x) \lor \psi(\underline{\vec{b}})) \Rightarrow u : \phi(\underline{\vec{a}}, \underline{a}), v : \psi(\underline{\vec{b}})}\text{Lem. 13}}{\mathcal{R}, \underline{a} \in D_u, v \leq v, v : \phi(\underline{\vec{a}}, \underline{a}), v : \forall x(\phi(\underline{\vec{a}}, x) \lor \psi(\underline{\vec{b}})) \Rightarrow u : \phi(\underline{\vec{a}}, \underline{a}), v : \psi(\underline{\vec{b}})} (nd) \times k_1 \right.$$

$$\Pi_2 = \left\{ \dfrac{\dfrac{}{\mathcal{R}, \underline{a} \in D_u, v \leq v, \underline{\vec{b}} \in D_v, v : \psi(\underline{\vec{b}}), v : \forall x(\phi(\underline{\vec{a}}, x) \lor \psi(\underline{\vec{b}})) \Rightarrow u : \phi(\underline{\vec{a}}, \underline{a}), v : \psi(\underline{\vec{b}})}\text{Lem. 13}}{\mathcal{R}, \underline{a} \in D_u, v \leq v, v : \psi(\underline{\vec{b}}), v : \forall x(\phi(\underline{\vec{a}}, x) \lor \psi(\underline{\vec{b}})) \Rightarrow u : \phi(\underline{\vec{a}}, \underline{a}), v : \psi(\underline{\vec{b}})} (nd) \times k_2 \right.$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Pi_1 \qquad \Pi_2}{\mathcal{R}, \underline{a} \in D_u, v \leq v, v : \phi(\underline{\vec{a}}, \underline{a}) \lor \psi(\underline{\vec{b}}), v : \forall x(\phi(\underline{\vec{a}}, x) \lor \psi(\underline{\vec{b}})) \Rightarrow u : \phi(\underline{\vec{a}}, \underline{a}), v : \psi(\underline{\vec{b}})} (\lor_l)}{\mathcal{R}, \underline{a} \in D_u, v \leq v, v : \forall x(\phi(\underline{\vec{a}}, x) \lor \psi(\underline{\vec{b}})) \Rightarrow u : \phi(\underline{\vec{a}}, \underline{a}), v : \psi(\underline{\vec{b}})} (\forall_l)}{\mathcal{R}, v \leq v, v : \forall x(\phi(\underline{\vec{a}}, x) \lor \psi(\underline{\vec{b}})) \Rightarrow u : \phi(\underline{\vec{a}}, \underline{a}), v : \psi(\underline{\vec{b}})} (cd)}{\mathcal{R}, v : \forall x(\phi(\underline{\vec{a}}, x) \lor \psi(\underline{\vec{b}})) \Rightarrow u : \phi(\underline{\vec{a}}, \underline{a}), v : \psi(\underline{\vec{b}})} (ref)}{w \leq v, \underline{\vec{a}} \in D_w, \underline{\vec{b}} \in D_w, v : \forall x(\phi(\underline{\vec{a}}, x) \lor \psi(\underline{\vec{b}})) \Rightarrow v : \forall x \phi(\underline{\vec{a}}, x), v : \psi(\underline{\vec{b}})} (\forall_r)}{w \leq v, \underline{\vec{a}} \in D_w, \underline{\vec{b}} \in D_w, v : \forall x(\phi(\underline{\vec{a}}, x) \lor \psi(\underline{\vec{b}})) \Rightarrow v : \forall x \phi(\underline{\vec{a}}, x) \lor \psi(\underline{\vec{b}})} (\lor_r)}{\underline{\vec{a}} \in D_w, \underline{\vec{b}} \in D_w \Rightarrow w : \forall x(\phi(\underline{\vec{a}}, x) \lor \psi(\underline{\vec{b}})) \supset \forall x \phi(\underline{\vec{a}}, x) \lor \psi(\underline{\vec{b}})} (\supset_r)$$

Rule R0. To show that modus ponens can be simulated, we let $\vec{a}$ be all parameters occurring in ϕ, $\vec{b}$ be all parameters occurring in ψ, and let $\vec{c}$ consist of all parameters occurring in ψ, but not ϕ. The last inference consists of a sequence of k (ned) applications that delete all domains atoms containing parameters from ϕ, but not ψ.

$$\Pi = \left\{ \dfrac{\dfrac{\dfrac{\dfrac{\underline{\vec{a}} \in D_w, \underline{\vec{c}} \in D_w \Rightarrow w : \phi(\underline{\vec{a}}) \supset \psi(\underline{\vec{b}})}{w \leq u, \underline{\vec{a}} \in D_w, \underline{\vec{c}} \in D_w, u : \phi(\underline{\vec{a}}) \Rightarrow u : \psi(\underline{\vec{b}})} \text{Lem. 17}}{(w \leq u)(w/u), (\underline{\vec{a}} \in D_w)(w/u), (\underline{\vec{c}} \in D_w)(w/u), (u : \phi(\underline{\vec{a}}))(w/u) \Rightarrow (u : \psi(\underline{\vec{b}}))(w/u)} (lsb)}{w \leq w, \underline{\vec{a}} \in D_w, \underline{\vec{c}} \in D_w, w : \phi(\underline{\vec{a}}) \Rightarrow w : \psi(\underline{\vec{b}})} =}{\underline{\vec{a}} \in D_w, \underline{\vec{c}} \in D_w, w : \phi(\underline{\vec{a}}) \Rightarrow w : \psi(\underline{\vec{b}})} (ref) \right.$$

$$\dfrac{\dfrac{\dfrac{\underline{\vec{a}} \in D_w \Rightarrow w : \phi(\underline{\vec{a}})}{\underline{\vec{a}} \in D_w, \underline{\vec{c}} \in D_w \Rightarrow w : \phi(\underline{\vec{a}})} (wk) \qquad \Pi}{\underline{\vec{a}} \in D_w, \underline{\vec{c}} \in D_w \Rightarrow w : \psi(\underline{\vec{b}})} (cut)}{\underline{\vec{b}} \in D_w \Rightarrow w : \psi(\underline{\vec{b}})} (ned) \times k$$

Rule R1. Last, we show that the generalization rule R1 can be simulated. We assume there are k domains atoms $\underline{\vec{a}} \in D_w := \underline{\vec{a}}_1 \in D_1, \ldots, \underline{\vec{a}}_k \in D_k$.

$$\begin{array}{cl}
\underline{\vec{a}} \in D_w, \underline{a} \in D_w \Rightarrow w : \phi(\underline{\vec{a}}, \underline{a}) & \\
\hdashline
u \leq w, \underline{\vec{a}} \in D_u, \underline{\vec{a}} \in D_w, \underline{a} \in D_w \Rightarrow w : \phi(\underline{\vec{a}}, \underline{a}) & (wk) \\
\hline
u \leq w, \underline{\vec{a}} \in D_u, \underline{a} \in D_w \Rightarrow w : \phi(\underline{\vec{a}}, \underline{a}) & (nd) \times k \\
\hline
\underline{\vec{a}} \in D_u \Rightarrow u : \forall x \phi(\underline{\vec{a}}, x) & (\forall_r) \\
\hdashline
\underline{\vec{a}} \in D_w \Rightarrow w : \forall x \phi(\underline{\vec{a}}, x) & (lsb)
\end{array}$$

□

As in the previous section, we define sequent graphs for labelled sequents, which uncovers the underlying data structures inherent in our sequents. This theoretical tool will assist us while refining our calculi in the next chapter.

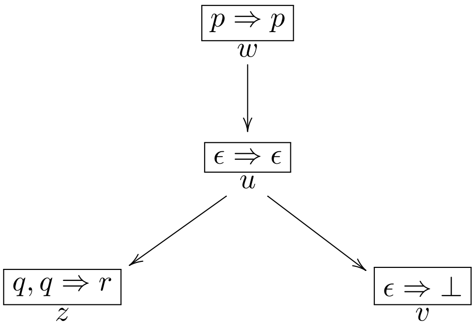
Definition 50 (Sequent Graph for First-Order Intuitionistic Logics)**.** *Let* $\Lambda := \mathcal{R}, \Gamma \Rightarrow \Delta$ *be a labelled sequent for first-order intuitionistic logics. We define the* sequent graph *of* Λ, $G(\Lambda) = (V, E, L)$, *as follows:*

- $V = \mathrm{Lab}(\Lambda)$
- $E = \{(w, u) \mid w \leq u \in \mathcal{R}\}$
- $L(w) = \Gamma \restriction w \Rightarrow \Delta \restriction w$

Last, we define labelled tree sequents*,* labelled tree proofs*, and the* fixed root property *analogous to Def. 44 and 45.*

Example 2. *We provide an example of a labelled tree sequent* Λ_1 *along with its corresponding sequent graph.*

$$\begin{array}{c}
\Lambda_1 = w \leq u, u \leq z, u \leq v, \\
w : p, z : q, z : q \Rightarrow w : p, z : r, v : \bot
\end{array}$$



Theorem 12. *Each calculus* G3IntQ *and* G3IntQC *is incomplete relative to labelled tree derivations.*

Proof. The result follows by considering the proof of $(p \supset q) \supset ((q \supset r) \supset (p \supset r))$, which requires the use of the (tra) rule, thus breaking the labelled tree property. □

3.4 Labelled Calculi for Deontic STIT Logics

Although STIT logics were introduced three decades ago by Belnap and Perloff [BP90], the proof theory for such logics has only recently been developed. Tableau calculi for traditional (non-deontic) multi-agent STIT logics were introduced by Wansing in 2006 [Wan06], with extended versions introduced in 2018 [OW18]. The first cut-free labelled sequent calculi for traditional multi-agent STIT logics were presented in 2019 [vBL19a], along with cut-free labelled sequent calculi for the *temporal* STIT logics: TSTIT [Lor13] and XSTIT [Bro11a, Bro11b]. While the papers [vBL19a, LvB19] provided labelled calculi for traditional, multi-agent STIT logics based on *Kripke semantics* [BHT08], an equivalent set of labelled calculi were given for the same class of logics in [NP20] based on *branching-time semantics* [BPX01]. Furthermore, the proof theory for a class of *deontic* STIT logics was addressed as well, with labelled calculi being introduced in [vBL21] and employed to map out the interrelationships between certain deontic STIT logics and *Ought-implies-Can* principles. The most relevant work for our purposes however is [LvB19], where a simplified semantics for multi-agent STIT logics was exploited to design labelled calculi amenable to the method of refinement. These calculi were then refined and used to provide the first proof-search and counter-model construction algorithms for STIT logics.

In this section, labelled sequent calculi are given for deontic STIT logics based on the author's joint work in [vBL19a, vBL21, LvB19]. It will be shown that these calculi possess useful proof-theoretic properties, and refined versions of the calculi (obtained in Ch. 4) will be applied to provide proof-search and counter-model extraction algorithms for a class of deontic STIT logics (in Ch. 6).

Definition 51 (Labelled Sequents for Deontic STIT Logics)**.** *We define labelled sequents for deontic STIT logics to be syntactic objects of the form $\Lambda := \mathcal{R} \Rightarrow \Gamma$, where $\mathcal{R}$ (the* antecedent*) and Γ (the* consequent*) are defined via the following grammars in BNF:*

$$\mathcal{R} ::= \epsilon \mid R_{[i]}wu \mid I_{\otimes_i}w \mid \mathcal{R},\mathcal{R} \qquad \Gamma ::= \epsilon \mid w : \phi \mid \Gamma,\Gamma$$

with $i \in \mathrm{Ag}$, $\phi \in \mathcal{L}_{\mathsf{DS}_n^k}$, and w, u elements from a denumerable set of labels $\mathrm{Lab} := \{w, u, v, ...\}$*. We refer to formulae of the form $R_{[i]}wu$ and $I_{\otimes_i}w$ as* relational atoms*, and refer to formulae of the form $w : \phi$ as* labelled formulae*.*

We use Λ, Λ', ... (occasionally annotated) to denote labelled sequents, $\mathcal{R}$, $\mathcal{R}'$, ... (occasionally annotated) to denote multisets of relational atoms, and Γ, Γ', ... (occasionally annotated) to denote multisets of labelled formulae. We therefore take the comma operator to commute and associate in $\mathcal{R}$ and Γ; therefore, we would identify $R_{[1]}wu, R_{[2]}wv, I_{\otimes_i}v$ with $R_{[2]}wv, I_{\otimes_i}v, R_{[1]}wu$, and $v : [1]\phi, u : \psi, w : [1]\phi$ with $u : \psi, w : [1]\phi, v : [1]\phi$. This interpretation of comma is what lets us view strings $\mathcal{R}$ and Γ as multisets. As in the previous two sections, we let ϵ represent the *empty string*, which acts as an identity element for comma (e.g. we identify $R_{[1]}wu, \epsilon, R_{[2]}wv$ with $R_{[1]}wu, R_{[2]}wv$). Therefore, the empty string ϵ will usually be implicit in labelled sequents. Last, we use the notation

$$\frac{}{\mathcal{R} \Rightarrow w : p, w : \neg p, \Gamma}\ (id) \qquad \frac{\mathcal{R} \Rightarrow w : \phi, \Gamma \qquad \mathcal{R} \Rightarrow w : \psi, \Gamma}{\mathcal{R} \Rightarrow w : \phi \land \psi, \Gamma}\ (\land_r)$$

$$\frac{\mathcal{R} \Rightarrow w : \phi, w : \psi, \Gamma}{\mathcal{R} \Rightarrow w : \phi \lor \psi, \Gamma}\ (\lor_r) \qquad \frac{\mathcal{R}, R_{[i]}wu, R_{[i]}wv, R_{[i]}uv \Rightarrow \Gamma}{\mathcal{R}, R_{[i]}wu, R_{[i]}wv \Rightarrow \Gamma}\ (euc_i)$$

$$\frac{\mathcal{R} \Rightarrow u : \phi, \Gamma}{\mathcal{R} \Rightarrow w : \Box\phi, \Gamma}\ (\Box)^{\dagger} \qquad \frac{\mathcal{R} \Rightarrow w : \Diamond\phi, u : \phi, \Gamma}{\mathcal{R} \Rightarrow w : \Diamond\phi, \Gamma}\ (\Diamond) \qquad \frac{\mathcal{R}, R_{[i]}ww \Rightarrow \Gamma}{\mathcal{R} \Rightarrow \Gamma}\ (ref_i)$$

$$\frac{\mathcal{R}, I_{\otimes_i}u \Rightarrow w : \ominus_i\phi, u : \phi, \Gamma}{\mathcal{R}, I_{\otimes_i}u \Rightarrow w : \ominus_i\phi, \Gamma}\ (\ominus_i) \qquad \frac{\mathcal{R}, R_{[i]}wu \Rightarrow w : \langle i \rangle\phi, u : \phi, \Gamma}{\mathcal{R}, R_{[i]}wu \Rightarrow w : \langle i \rangle\phi, \Gamma}\ (\langle i \rangle)$$

$$\frac{\mathcal{R}, R_{[i]}wu \Rightarrow u : \phi, \Gamma}{\mathcal{R} \Rightarrow w : [i]\phi, \Gamma}\ ([i])^{\dagger} \qquad \frac{\mathcal{R}, I_{\otimes_i}u \Rightarrow u : \phi, \Gamma}{\mathcal{R} \Rightarrow w : \otimes_i\phi, \Gamma}\ (\otimes_i)^{\dagger} \qquad \frac{\mathcal{R}, I_{\otimes_i}u \Rightarrow \Gamma}{\mathcal{R} \Rightarrow \Gamma}\ (D^i_2)^{\dagger}$$

$$\frac{\mathcal{R}, R_{[0]}w_0u, ..., R_{[n]}w_nu \Rightarrow \Gamma}{\mathcal{R} \Rightarrow \Gamma}\ (IOA)^{\dagger} \qquad \frac{\mathcal{R}, R_{[i]}wu, I_{\otimes_i}w, I_{\otimes_i}u \Rightarrow \Gamma}{\mathcal{R}, R_{[i]}wu, I_{\otimes_i}w \Rightarrow \Gamma}\ (D^i_3)$$

$$\frac{\Big\{\mathcal{R}, R_{[i]}w_mw_j \Rightarrow \Gamma \ \Big|\ 0 \leq m \leq k-1,\ m+1 \leq j \leq k\Big\}}{\mathcal{R} \Rightarrow \Gamma}\ (APC^k_i)$$

Figure 3.5: The calculi $\mathsf{G3DS}^k_n$ (with $|\text{Ag}| = n + 1$ and $n, k \in \mathbb{N}$). The superscript † on $(\Box)$, $([i])$, $(\otimes_i)$, (IOA), and $(\mathsf{D2}_i)$ indicates that u is a eigenvariable, i.e. it does not occur in the conclusion. We have $([i])$, $(\langle i \rangle)$, $(\otimes_i)$, $(\ominus_i)$, (ref_i), (euc_i), (D^i_2), (D^i_3), and (APC^k_i) rules for each $i \in \text{Ag}$. We stipulate that if $k = 0$, then (APC^k_i) is omitted from the calculus.

Lab($\mathcal{R}$), Lab(Γ), and Lab($\mathcal{R} \Rightarrow \Gamma$) to represent the set of labels contained in $\mathcal{R}$, Γ, and $\mathcal{R} \Rightarrow \Gamma$ respectively (e.g., $Lab(R_{[2]}wv \Rightarrow u : p) = \{w, v, u\}$).

The calculus $\mathsf{G3DS}^k_n$ for the logic DS^k_n (with $|\text{Ag}| = n + 1$, k the maximum number of choices available to agents, and $n, k \in \mathbb{N}$) is shown in Fig. 3.5 with the derivability relation defined as follows:

Definition 52. *We write* $\vdash_{\mathsf{G3DS}^k_n} \Lambda$ *to indicate that a labelled sequent* Λ *is derivable in* $\mathsf{G3DS}^k_n$.

The (id) rule encodes that fact that in a DS^k_n-model, either a propositional atom holds at a world, or it does not. The $(\lor_r)$, $(\land_r)$, $(\Diamond)$, $(\Box)$, $(\ominus_i)$, $(\otimes_i)$, $(\langle i \rangle)$, and $([i])$ rules are obtained from the corresponding semantic clauses (Def. 25). The structural rules (ref_i) and (euc_i) encode the fact that choice-cells in a DS^k_n-model are equivalence classes, that is each $R_{[i]}$ relation is both reflexive and Euclidean as dictated by condition $\mathbf{(S_1)}$. The other conditions $\mathbf{(S_2)}$, $\mathbf{(D_2)}$, $\mathbf{(D_3)}$, and $\mathbf{(S_3)}$ imposed on a DS^k_n-model are encoded by the rules, (IOA), (D_2^i), (D_3^i), and (APC_i^k), respectively. Note that when the maximum number of choices $k > 0$, the (APC_i^k) rule contains $k(k+1)/2$ premises, and each sequent $\mathcal{R}, R_{[i]}x_m x_j \Rightarrow \Gamma$ (for $0 \leq m \leq k-1$ and $m+1 \leq j \leq k$) represents a different premise of the rule. For example, if $k = 1$ or $k = 2$, then the form of the (APC_i^k) rule is as shown below top and below bottom, respectively:

$$\frac{\mathcal{R}, R_{[i]}w_0 w_1 \Rightarrow \Gamma}{\mathcal{R} \Rightarrow \Gamma}\ (APC_i^1)$$

$$\frac{\mathcal{R}, R_{[i]}w_0 w_1 \Rightarrow \Gamma \qquad \mathcal{R}, R_{[i]}w_0 w_2 \Rightarrow \Gamma \qquad \mathcal{R}, R_{[i]}w_1 w_2 \Rightarrow \Gamma}{\mathcal{R} \Rightarrow \Gamma}\ (APC_i^2)$$

As is characteristic of labelled systems, our labelled sequents represent an abstraction of a model (*viz.* a DS_n^k-model) with the labels denoting worlds, the relational atoms expressing the relations of the model, and labelled formulae representing that formulae are (un)satisfied at a particular world. To make the correspondence between the syntax of our labelled sequents and the content of a DS_n^k-model, we define how to interpret labelled sequents below. In addition, our sequent semantics will allow us to prove each $\mathsf{G3DS}_n^k$ calculus sound relative to the associated logic DS_n^k, which we show after defining the sequent semantics.

Definition 53 ($\mathsf{G3DS}_n^k$ Semantics)**.** *Let $M = (W, \{\mathcal{R}_{[i]} \mid i \in \mathrm{Ag}\}, \{I_{\otimes_i} \mid i \in \mathrm{Ag}\}, V)$ be a DS_n^k-model with I an* interpretation function *mapping labels to worlds, i.e. $I : \mathrm{Lab} \mapsto W$. We define the* satisfaction *of a relational atom $R_{[i]}wu$ or $I_{\otimes_i}w$ (written $M, I \models_{\mathsf{DS}_n^k} R_{[i]}wu$ and $M, I \models_{\mathsf{DS}_n^k} I_{\otimes_i}w$, resp.) and labelled formula $w : \phi$ (written $M, I \models_{\mathsf{DS}_n^k} w : \phi$) as follows:*

- *$M, I \models_{\mathsf{DS}_n^k} R_{[i]}wu$ iff $(I(w), I(u)) \in R_{[i]}$*
- *$M, I \models_{\mathsf{DS}_n^k} I_{\otimes_i}w$ iff $I(w) \in I_{\otimes_i}$*
- *$M, I \models_{\mathsf{DS}_n^k} w : \phi$ iff $M, I(w) \Vdash \phi$*

We say that a multiset of relational atoms $\mathcal{R}$ is satisfied *in M with I (written $M, I \models_{\mathsf{DS}_n^k} \mathcal{R}$) iff $M, I \models_{\mathsf{DS}_n^k} R_{[i]}wu$ and $M, I \models_{\mathsf{DS}_n^k} I_{\otimes_i}w$ for all $R_{[i]}wu, I_{\otimes_i}w \in \mathcal{R}$, and we say that a multiset of labelled formulae Γ is* satisfied *in M with I (written $M, I \models_{\mathsf{DS}_n^k} \Gamma$) iff $M, I \models_{\mathsf{DS}_n^k} w : \phi$ for all $w : \phi \in \Gamma$.*

A labelled sequent $\Lambda := \mathcal{R} \Rightarrow \Gamma$ is satisfied *in M with I (written, $M, I \models_{\mathsf{DS}_n^k} \Lambda$) iff if $M, I \models_{\mathsf{DS}_n^k} \mathcal{R}$, then $M, I \models_{\mathsf{DS}_n^k} \Gamma$. Also, we say that a labelled sequent Λ is* falsified *in M with I iff $M, I \not\models_{\mathsf{DS}_n^k} \Lambda$, that is, Λ is not satisfied by M with I.*

Last, a labelled sequent Λ is DS_n^k-valid *(written $\models_{\mathsf{DS}_n^k} \Lambda$) iff it is satisfiable in every DS_n^k model M with every interpretation function I. We say that a labelled sequent Λ is* DS_n^k-invalid *iff $\not\models_{\mathsf{DS}_n^k} \Lambda$, i.e. Λ is not DS_n^k-valid.*

Intuitively, the above definition expresses that a sequent $\Lambda := \mathcal{R} \Rightarrow \Gamma$ is satisfied in a DS^k_n-model iff some labelled formula $w : \phi \in \Gamma$ holds, given that all relational atoms in $\mathcal{R}$ hold. Therefore, in traditional fashion, we interpret the comma in the antecedent conjunctively, and the comma in the consequent disjunctively, with the sequent arrow $\Rightarrow$ representing an implication. Using the above definition, we prove our calculi sound:

Theorem 13 (Soundness). *If* $\vdash_{\mathsf{G3DS}^k_n} \Lambda$*, then* $\models_{\mathsf{DS}^k_n} \Lambda$.

Proof. The result is proven by showing that each initial sequent generated by (id) is DS^k_n-valid, and that each additional rule of $\mathsf{G3DS}^k_n$ preserves validity. We prove the latter by contraposition, and show that if the conclusion of the rule is DS^k_n-invalid, then at least one of the premises (or, the premise) of the rule is DS^k_n-invalid. We omit consideration of the (id), $(\vee_r)$, and $(\wedge_r)$ cases, as they are simple.

$(\Box)$ Suppose that $M, I \not\models_{\mathsf{DS}^k_n} \mathcal{R} \Rightarrow w : \Box\phi, \Gamma$. It follows that $M, I(w) \not\Vdash \Box\phi$, implying that there exists some u' such that $M, u' \not\Vdash \phi$. Let us define $I'(v) = I(v)$ if $v \neq u$ and $I'(u) = u'$ otherwise. By definition, we have that $M, I'(u) \not\Vdash \phi$. Since u is an eigenvariable, it follows that the premise is falsified by M with I'.

$([i])$ Assume that $M, I \not\models_{\mathsf{DS}^k_n} \mathcal{R} \Rightarrow w : [i]\phi, \Gamma$. Then, $M, I(w) \not\Vdash [i]\phi$, implying that there exists a world u' such that $(I(w), u') \in R_{[i]}$ and $M, u' \not\Vdash \phi$. We let $I'(v) = I(v)$ if $v \neq u$ and $I'(u) = u'$ otherwise. By definition, $(I'(w), I'(u)) \in R_{[i]}$ and $M, I'(u) \not\Vdash \phi$. Since the label u is an eigenvariable, it follows that the premise is falsified by M with I'.

$(\otimes_i)$ Assume that $M, I \not\models_{\mathsf{DS}^k_n} \mathcal{R} \Rightarrow w : \otimes_i\phi$. It follows that $M, I(w) \not\Vdash w : \otimes_i\phi$, implying that there exists a world u' such that $u' \in I_{\otimes_i}$ and $M, u' \not\Vdash \phi$. We define $I'(v) = I(v)$ if $v \neq u$ and $I'(u) = u'$ otherwise. By definition, $I'(u) \in I_{\otimes_i}$ and $M, I'(u) \not\Vdash \phi$, and since u is an eigenvariable, it follows that the premise is falsified by M with I'.

$(\Diamond)$ Assume that $M, I \not\models_{\mathsf{DS}^k_n} \mathcal{R} \Rightarrow w : \Diamond\phi, \Gamma$. Then, $M, I(w) \not\Vdash \Diamond\phi$, meaning that for all $v \in W$, $M, v \not\Vdash \phi$. Since $I(u) \in W$, this implies that $M, I(u) \not\Vdash \phi$, which shows that the premise is falsified by M with I.

$(\langle i \rangle)$ Assume that $M, I \not\models_{\mathsf{DS}^k_n} \mathcal{R}, R_{[i]}wu \Rightarrow w : \langle i \rangle\phi, \Gamma$. Then, $M, I(w) \not\Vdash [i]\phi$, which implies that for all u', if $(I(w), u') \in R_{[i]}$, then $M, u' \not\Vdash \phi$. Since $(I(w), I(u)) \in R_{[i]}$, it follows that $M, I(u) \not\Vdash \phi$, which shows that the premise is falsified by M with I.

$(\ominus_i)$ Let us suppose that $M, I \not\models_{\mathsf{DS}^k_n} \mathcal{R}, I_{\otimes_i}u \Rightarrow w : \ominus_i\phi, \Gamma$. It follows that $M, I(w) \not\Vdash \ominus_i\phi$, implying that for all u', if $u' \in I_{\otimes_i}$, then $M, u' \not\Vdash \phi$. Since $I(u) \in I_{\otimes_i}$, we have that $M, I(u) \not\Vdash \phi$, which shows that the premise is falsified by M with I.

(ref_i) Let us suppose that $M, I \not\models_{\mathsf{DS}^k_n} \mathcal{R} \Rightarrow \Gamma$. Since the relation $R_{[i]}$ is reflexive, it follows that $(I(w), I(w)) \in R_{[i]}$, showing that the premise is falsified by M with I.

(euc_i) Let us suppose that $M, I \not\models_{\mathsf{DS}^k_n} \mathcal{R}, R_{[i]}wu, R_{[i]}uv \Rightarrow \Gamma$. Therefore, $(I(w), I(u)) \in R_{[i]}$ and $(I(u), I(v)) \in R_{[i]}$. Since the relation $R_{[i]}$ is Euclidean, it follows that $(I(w), I(v)) \in R_{[i]}$, which shows that the premise is falsified by M with I.

(D^i_2) Let us suppose that $M, I \not\models_{\mathsf{DS}^k_n} \mathcal{R} \Rightarrow \Gamma$. Since $I_{\otimes_i}$ satisfies condition **(D₂)**, we know there exists some $u' \in W$ such that $u' \in I_{\otimes_i}$. Let us define $I'(v) = I(v)$ if $v \neq u$, and $I'(u) = u'$ otherwise. By definition, $I'(u) \in I_{\otimes_i}$, and since u is an eigenvariable, it follows that the premise is falsified by M with I'.

(D^i_3) Suppose that $M, I \not\models_{\mathsf{DS}^k_n} \mathcal{R}, R_{[i]}wu, I_{\otimes_i}w \Rightarrow \Gamma$. This entails that $(I(w), I(u)) \in R_{[i]}$ and $I(w) \in I_{\otimes_i}$. Since our model M satisfies the condition **(D₃)**, it follows that $I(u) \in I_{\otimes_i}$, which shows that the premise is falsified by M with I.

(IOA) Suppose that $M, I \not\models_{\mathsf{DS}^k_n} \mathcal{R} \Rightarrow \Gamma$. Since our model M satisfies the **(S₂)** condition, we know that for $I(w_0), \ldots, I(w_n) \in W$, there exists some u' such that $(I(w_i), u') \in R_{[i]}$ for $i \in \{0, \ldots, n\}$. Let us define $I'(v) = I(v)$ if $v \neq u$ and $I'(u) = u'$ otherwise. Then, it follows that $(I(w_i), I(u)) \in R_{[i]}$ for $i \in \{0, \ldots, n\}$. Because u is an eigenvariable, we have that the premise is falsified by M with I'.

(APC^k_i) Suppose that $M, I \not\models_{\mathsf{DS}^k_n} \mathcal{R} \Rightarrow \Gamma$. Then, since our model M satisfies the **(S₃)** condition, we know there exist worlds $I(w_0), \ldots, I(w_k) \in W$ such that

$$\bigvee_{0 \leq m \leq k-1, m+1 \leq j \leq k} (I(w_m), I(w_j)) \in R_{[i]}$$

holds. Hence, there must exist an m and j (with $0 \leq m \leq k-1, m+1 \leq j \leq k$) such that $(I(w_m), I(w_j)) \in R_{[i]}$ holds, showing that one of the premises is falsified by M with I. □

Our labelled calculi for deontic STIT logics possess fundamental proof-theoretic properties such as the hp-admissibility of label substitutions (lsb), weakening (wk), and contractions $(ctr_{\mathcal{R}})$, (ctr_r). Moreover, all rules of the calculi are hp-invertible, and each calculus admits syntactic cut-elimination. All results are shown below, with the methods of proof based upon the work in [Neg05, Sim94, Vig00].

The aforementioned properties are useful not only in establishing the completeness of our calculi (Thm. 15), but are useful in applications such as decidability via proof-search (see Sect. 6.1). Before moving on to establish these results, we define the notion of a *label substitution*, which is an operation that is fundamental for establishing our results.

Definition 54 (Label Substitution)**.** *We define a* label substitution (w/u) *for* $w, u \in \mathrm{Lab}$ *on individual relational atoms and labelled formulae as follows:*

- $$(R_{[i]}vz)(w/u) = \begin{cases} R_{[i]}wz & \text{if } u = v \text{ and } v \neq z \\ R_{[i]}ww & \text{if } u = v \text{ and } v = z \\ R_{[i]}uw & \text{if } u = z \text{ and } v \neq z \\ R_{[i]}vz & \text{otherwise} \end{cases}$$
- $$(I_{\otimes_i}v)(w/u) = \begin{cases} I_{\otimes_i}w & \text{if } u = v \\ I_{\otimes_i}v & \text{otherwise} \end{cases}$$

$$\dfrac{\mathcal{R} \Rightarrow \Gamma}{\mathcal{R}(w/u) \Rightarrow \Gamma(w/u)}\,(lsb) \qquad \dfrac{\mathcal{R} \Rightarrow \Gamma}{\mathcal{R}, \mathcal{R}' \Rightarrow \Gamma, \Gamma'}\,(wk) \qquad \dfrac{\mathcal{R}, \mathcal{R}', \mathcal{R}' \Rightarrow \Gamma}{\mathcal{R}, \mathcal{R}' \Rightarrow \Gamma}\,(ctr_{\mathcal{R}})$$

$$\dfrac{\mathcal{R} \Rightarrow w : \phi, w : \phi, \Gamma}{\mathcal{R} \Rightarrow w : \phi, \Gamma}\,(ctr_r) \qquad \dfrac{\mathcal{R} \Rightarrow w : \phi, \Gamma \qquad \mathcal{R} \Rightarrow w : \neg\phi, \Gamma}{\mathcal{R} \Rightarrow \Gamma}\,(cut)$$

Figure 3.6: The set Str(DS_n^k) of structural rules consists of the rules above.

- ▶ $(v : \phi)(w/u) = \begin{cases} w : \phi & \textit{if } u = v \\ v : \phi & \textit{otherwise} \end{cases}$

We define the label substitution (w/u) on a multiset of relational atoms $\mathcal{R}$ and a multiset of labelled formulae Γ to be the multiset obtained by applying (w/u) to each element of the multiset.

Lemma 19. *For all $\phi \in \mathcal{L}_{\mathsf{Km}}(\sum)$, $\vdash_{\mathsf{G3DS}_n^k} \mathcal{R} \Rightarrow w : \phi, w : \neg\phi, \Gamma$.*

Proof. We prove the result by induction on the complexity of ϕ.

Base case. If $|\phi| = 0$, then ϕ is of the form p or $\neg p$. In either case, the desired result is obtained as an instance of (id). (NB. If ϕ is of the form $\neg p$, then observe that the sequent $\mathcal{R} \Rightarrow w : \neg p, w : \neg\neg p, \Gamma$ is identical to $\mathcal{R} \Rightarrow w : \neg p, w : p, \Gamma$ by the definition of negation (Def. 23), which is an instance of (id).)

Inductive step. We show the cases where ϕ is of the form $\Box\psi$, $\langle i \rangle\psi$, and $\otimes_i\psi$; the other cases when ϕ is of the form $\psi \vee \chi$, $\psi \wedge \chi$, $\Diamond\psi$, $[i]\psi$, or $\ominus_i\psi$ are simple or are shown similarly.

$$\dfrac{\dfrac{\dfrac{}{\mathcal{R} \Rightarrow u : \phi, w : \Diamond\neg\psi, u : \neg\psi, \Gamma}\,\text{IH}}{\mathcal{R} \Rightarrow u : \phi, w : \Diamond\neg\psi, \Gamma}\,(\Diamond)}{\mathcal{R} \Rightarrow w : \Box\psi, w : \Diamond\neg\psi, \Gamma}\,(\Box) \qquad \dfrac{\dfrac{\dfrac{}{\mathcal{R}, R_{[i]}wu \Rightarrow w : \langle i \rangle\psi, u : \psi, u : \neg\psi, \Gamma}\,\text{IH}}{\mathcal{R}, R_{[i]}wu \Rightarrow w : \langle i \rangle\psi, u : \neg\psi, \Gamma}\,(\langle i \rangle)}{\mathcal{R} \Rightarrow w : \langle i \rangle\psi, w : [i]\neg\psi, \Gamma}\,([i])$$

$$\dfrac{\dfrac{\dfrac{}{\mathcal{R}, I_{\otimes_i}u \Rightarrow u : \psi, u : \neg\psi, w : \ominus_i\neg\psi, \Gamma}\,\text{IH}}{\mathcal{R}, I_{\otimes_i}u \Rightarrow u : \psi, w : \ominus_i\neg\psi, \Gamma}\,(\ominus_i)}{\mathcal{R} \Rightarrow w : \otimes_i\psi, w : \ominus_i\neg\psi, \Gamma}\,(\otimes_i)$$

□

Lemma 20. *The rule (lsb) is hp-admissible in $\mathsf{G3DS}_n^k$.*

Proof. We prove the result by induction on the height of the given derivation.

Base case. Any application of (lsb) to an instance of (id) yields another instance of (id), which solves the base case.

Inductive step. With the exception of the $(\Box)$, $([i])$, $(\otimes_i)$, (D_2^i), and (IOA) rules, all cases are resolved by applying IH and then the corresponding rule. Concerning the $(\Box)$, $([i])$, $(\otimes_i)$, (D_2^i), and (IOA) cases, the non-trivial cases occur when the label substituted in is identical to the eigenvariable. We show how to resolve these non-trivial cases below and note that all other cases (i.e. the trivial cases) are resolved by applying IH followed by the corresponding rule. Below, we assume that z is a fresh label.

$$\dfrac{\dfrac{\mathcal{R} \Rightarrow u : \phi, \Gamma}{\mathcal{R} \Rightarrow w : \Box\phi, \Gamma}(\Box)}{\mathcal{R}(u/v) \Rightarrow (w : \Box\phi)(u/v), \Gamma(u/v)}(lsb) \quad \leadsto \quad \begin{array}{cl}
\mathcal{R} \Rightarrow u : \phi, \Gamma & \\ \hdashline
\mathcal{R}(z/u) \Rightarrow (u : \phi)(z/u), \Gamma(z/u) & \text{IH} \\ \cdots\cdots\cdots\cdots & = \\
\mathcal{R} \Rightarrow z : \phi, \Gamma & \\ \hdashline
\mathcal{R}(u/v) \Rightarrow (z : \phi)(u/v), \Gamma(u/v) & \text{IH} \\ \cdots\cdots\cdots\cdots & = \\
\mathcal{R}(u/v) \Rightarrow z : \phi, \Gamma(u/v) & \\ \hline
\mathcal{R}(u/v) \Rightarrow (w : \Box\phi)(u/v), \Gamma(u/v) & (\Box)
\end{array}$$

$$\dfrac{\dfrac{\mathcal{R}, R_{[i]}wu \Rightarrow u : \phi, \Gamma}{\mathcal{R} \Rightarrow w : [i]\phi, \Gamma}([i])}{\mathcal{R}(u/v) \Rightarrow (w : [i]\phi)(u/v), \Gamma(u/v)}(lsb) \quad \leadsto$$

$$\begin{array}{cl}
\mathcal{R}, R_{[i]}wu \Rightarrow u : \phi, \Gamma & \\ \hdashline
\mathcal{R}(z/u), (R_{[i]}wu)(z/u) \Rightarrow (u : \phi)(z/u), \Gamma(z/u) & \text{IH} \\ \cdots\cdots\cdots\cdots & = \\
\mathcal{R}, R_{[i]}wz \Rightarrow z : \phi, \Gamma & \\ \hdashline
\mathcal{R}(u/v), (R_{[i]}wu)(u/v) \Rightarrow (z : \phi)(u/v), \Gamma(u/v) & \text{IH} \\ \cdots\cdots\cdots\cdots & = \\
\mathcal{R}(u/v), (R_{[i]}wu)(u/v) \Rightarrow z : \phi, \Gamma(u/v) & \\ \hline
\mathcal{R}(u/v) \Rightarrow (w : [i]\phi)(u/v), \Gamma(u/v) & ([i])
\end{array}$$

$$\dfrac{\dfrac{\mathcal{R}, I_{\otimes_i}u \Rightarrow u : \phi, \Gamma}{\mathcal{R} \Rightarrow w : \otimes_i\phi, \Gamma}(\otimes_i)}{\mathcal{R}(u/v) \Rightarrow (w : \otimes_i\phi)(u/v), \Gamma(u/v)}(lsb) \quad \leadsto$$

$$\begin{array}{cl}
\mathcal{R}, I_{\otimes_i}u \Rightarrow u : \phi, \Gamma & \\ \hdashline
\mathcal{R}(z/u), (I_{\otimes_i}u)(z/u) \Rightarrow (u : \phi)(z/u), \Gamma(z/u) & \text{IH} \\ \cdots\cdots\cdots\cdots & = \\
\mathcal{R}, I_{\otimes_i}z \Rightarrow z : \phi, \Gamma & \\ \hdashline
\mathcal{R}(u/v), (I_{\otimes_i}z)(u/v) \Rightarrow (z : \phi)(u/v), \Gamma(u/v) & \text{IH} \\ \cdots\cdots\cdots\cdots & = \\
\mathcal{R}(u/v), I_{\otimes_i}z \Rightarrow z : \phi, \Gamma(u/v) & \\ \hline
\mathcal{R}(u/v) \Rightarrow (w : \otimes_i\phi)(u/v), \Gamma(u/v) & (\otimes_i)
\end{array}$$

$$\dfrac{\dfrac{\mathcal{R}, I_{\otimes_i}u \Rightarrow \Gamma}{\mathcal{R} \Rightarrow \Gamma}\,(D^i_2)}{\mathcal{R}(u/v) \Rightarrow \Gamma(u/v)}\,(lsb) \quad\rightsquigarrow\quad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\mathcal{R}, I_{\otimes_i}u \Rightarrow \Gamma}{\mathcal{R}(z/u), (I_{\otimes_i}u)(z/u) \Rightarrow \Gamma(z/u)}\,\text{IH}}{\mathcal{R}, I_{\otimes_i}z \Rightarrow \Gamma}\,=}{\mathcal{R}(u/v), (I_{\otimes_i}z)(u/v) \Rightarrow \Gamma(u/v)}\,\text{IH}}{\mathcal{R}(u/v), I_{\otimes_i}z \Rightarrow \Gamma(u/v)}\,=}{\mathcal{R}(u/v) \Rightarrow \Gamma(u/v)}\,(D^i_2)$$

$$\dfrac{\dfrac{\mathcal{R}, R_{[0]}w_0u, ..., R_{[n]}w_nu \Rightarrow \Gamma}{\mathcal{R} \Rightarrow \Gamma}\,(IOA)}{\mathcal{R}(u/v) \Rightarrow \Gamma(u/v)}\,(lsb) \quad\rightsquigarrow$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\mathcal{R}, R_{[0]}w_0u, ..., R_{[n]}w_nu \Rightarrow \Gamma}{\mathcal{R}(z/u), (R_{[0]}w_0u, ..., R_{[n]}w_nu)(z/u) \Rightarrow \Gamma(z/u)}\,\text{IH}}{\mathcal{R}, R_{[0]}w_0z, ..., R_{[n]}w_nz \Rightarrow \Gamma}\,=}{\mathcal{R}(u/v), (R_{[0]}w_0u, ..., R_{[n]}w_nu)(u/v) \Rightarrow \Gamma(u/v)}\,\text{IH}}{\mathcal{R}(u/v) \Rightarrow \Gamma(u/v)}\,(IOA)$$

□

Lemma 21. *The rule (wk) is hp-admissible in $\mathsf{G3DS}^k_n$.*

Proof. We prove the result by induction on the height of the given derivation.

Base case. The base case is easily resolved as any application of (wk) to an instance of (id) gives another instance of (id).

Inductive step. With the exception of the $(\Box)$, $([i])$, $(\otimes_i)$, (D^i_2), and (IOA) cases, all cases are resolved by invoking IH followed by the corresponding rule. The non-trivial $(\Box)$, $([i])$, $(\otimes_i)$, (D^i_2), and (IOA) cases arise when a relational atom or labelled formula is weakened in containing the eigenvariable of the inference. We show how to resolve these cases below and omit the other (trivial) cases as they follow by invoking IH and then applying the corresponding rule. We let z be a fresh label below.

$$\dfrac{\dfrac{\mathcal{R} \Rightarrow u : \phi, \Gamma}{\mathcal{R} \Rightarrow w : \Box\phi, \Gamma}\,(\Box)}{\mathcal{R}, \mathcal{R}' \Rightarrow w : \Box\phi, \Gamma, \Gamma'}\,(wk) \quad\rightsquigarrow\quad \dfrac{\dfrac{\dfrac{\dfrac{\mathcal{R} \Rightarrow u : \phi, \Gamma}{\mathcal{R}(z/u) \Rightarrow u : \phi(z/u), \Gamma(z/u)}\,(lsb)}{\mathcal{R} \Rightarrow z : \phi, \Gamma}\,=}{\mathcal{R}, \mathcal{R}' \Rightarrow z : \phi, \Gamma, \Gamma'}\,\text{IH}}{\mathcal{R} \Rightarrow w : \Box\phi, \Gamma}\,(\Box)$$

$$\dfrac{\dfrac{\mathcal{R}, R_{[i]}wu \Rightarrow u : \phi, \Gamma}{\mathcal{R} \Rightarrow w : [i]\phi, \Gamma}\,([i])}{\mathcal{R}, \mathcal{R}' \Rightarrow w : [i]\phi, \Gamma, \Gamma'}\,(wk) \quad \rightsquigarrow \quad \begin{array}{cl} \mathcal{R}, R_{[i]}wu \Rightarrow u : \phi, \Gamma & \\ \hdashline \mathcal{R}(z/u), R_{[i]}wu(z/u) \Rightarrow u : \phi(z/u), \Gamma(z/u) & (lsb) \\ \hdashline \mathcal{R}, R_{[i]}wz \Rightarrow z : \phi, \Gamma & = \\ \hdashline \mathcal{R}, \mathcal{R}', R_{[i]}wz \Rightarrow z : \phi, \Gamma, \Gamma' & \text{IH} \\ \hline \mathcal{R}, \mathcal{R}' \Rightarrow w : [i]\phi, \Gamma, \Gamma' & ([i]) \end{array}$$

$$\dfrac{\dfrac{\mathcal{R}, I_{\otimes_i}u \Rightarrow u : \phi, \Gamma}{\mathcal{R} \Rightarrow w : \otimes_i\phi, \Gamma}\,(\otimes_i)}{\mathcal{R}, \mathcal{R}' \Rightarrow w : \otimes_i\phi, \Gamma, \Gamma'}\,(wk) \quad \rightsquigarrow \quad \begin{array}{cl} \mathcal{R}, I_{\otimes_i}u \Rightarrow u : \phi, \Gamma & \\ \hdashline \mathcal{R}(z/u), I_{\otimes_i}u(z/u) \Rightarrow u : \phi(z/u), \Gamma(z/u) & (lsb) \\ \hdashline \mathcal{R}, I_{\otimes_i}z \Rightarrow z : \phi, \Gamma & = \\ \hdashline \mathcal{R}, \mathcal{R}', I_{\otimes_i}z \Rightarrow z : \phi, \Gamma, \Gamma' & \text{IH} \\ \hline \mathcal{R}, \mathcal{R}' \Rightarrow w : \otimes_i\phi, \Gamma, \Gamma' & (\otimes_i) \end{array}$$

$$\dfrac{\dfrac{\mathcal{R}, I_{\otimes_i}u \Rightarrow \Gamma}{\mathcal{R} \Rightarrow \Gamma}\,(D^i_2)}{\mathcal{R}, \mathcal{R}' \Rightarrow \Gamma, \Gamma'}\,(wk) \quad \rightsquigarrow \quad \begin{array}{cl} \mathcal{R}, I_{\otimes_i}u \Rightarrow \Gamma & \\ \hdashline \mathcal{R}(z/u), I_{\otimes_i}u(z/u) \Rightarrow \Gamma(z/u) & (lsb) \\ \hdashline \mathcal{R}, I_{\otimes_i}z \Rightarrow \Gamma & = \\ \hdashline \mathcal{R}, \mathcal{R}', I_{\otimes_i}z \Rightarrow \Gamma, \Gamma' & \text{IH} \\ \hline \mathcal{R}, \mathcal{R}' \Rightarrow \Gamma, \Gamma' & (\otimes_i) \end{array}$$

$$\dfrac{\dfrac{\mathcal{R}, R_{[0]}w_0u, ..., R_{[n]}w_nu \Rightarrow \Gamma}{\mathcal{R} \Rightarrow \Gamma}\,(IOA)}{\mathcal{R}, \mathcal{R}' \Rightarrow \Gamma, \Gamma'}\,(wk) \quad \rightsquigarrow$$

$$\begin{array}{cl} \mathcal{R}, R_{[0]}w_0u, ..., R_{[n]}w_nu \Rightarrow \Gamma & \\ \hdashline \mathcal{R}(z/u), R_{[0]}w_0u(z/u), ..., R_{[n]}w_nu(z/u) \Rightarrow \Gamma(z/u) & (lsb) \\ \hdashline \mathcal{R}, R_{[0]}w_0z, ..., R_{[n]}w_nz \Rightarrow \Gamma & = \\ \hdashline \mathcal{R}, \mathcal{R}', R_{[0]}w_0z, ..., R_{[n]}w_nz \Rightarrow \Gamma, \Gamma' & \text{IH} \\ \hline \mathcal{R}, \mathcal{R}' \Rightarrow \Gamma, \Gamma' & (IOA) \end{array}$$

□

Lemma 22. *All rules in* $\mathsf{G3DS}^k_n$ *are hp-invertible.*

Proof. The hp-invertibility of ($\Diamond$), ($\langle i \rangle$), ($\ominus_i$), (ref_i), (euc_i), (IOA), (D^i_2), and (D^i_3) follow from Lem. 21. The remaining rules are shown to be hp-invertible by induction on the height of the given derivation, and the proof is similar to the proof of [Neg05, Prop. 4.11]. □

Lemma 23. *The rules* ($ctr_{\mathcal{R}}$) *and* (ctr_r) *is hp-admissible in* $\mathsf{G3DS}^k_n$.

Proof. The result is shown by induction on the height of the given derivation. The hp-admissibility of $(ctr_{\mathcal{R}})$ is straightforward, as any application of $(ctr_{\mathcal{R}})$ to an initial sequent yields another initial sequent, and each case of the inductive step follows by applying IH and then the corresponding rule. Therefore, we focus solely on the hp-admissibility proof of (ctr_r).

Base case. Any application of (ctr_r) to an instance of (id) yields another instance of (id), resolving the base case.

Inductive step. For the inductive step, we assume that the given derivation ends with an application of (r) followed by an application of (ctr_r). If the principal formula of (r) is not active in (ctr_r), then the active formulae of (ctr_r) occur in the premise(s) of (r). Hence, the desired conclusion is obtained by applying IH to the premise(s) of (r), followed by an application of (r). We may therefore assume that the principal formula of (r) is active in (ctr_r). If the rule (r) is an instance of $(\Diamond)$, $(\langle i \rangle)$, or $(\ominus_i)$ then since the principal formula occurs in the premise of the rule, the case is resolved by applying IH to the premise, followed by an application of the respective rule. Also, note that (r) cannot be (ref_i), (euc_i), (D_2^i), (D_3^i), (IOA), or (APC_i^k) by our assumption, since it does not have a principal formula on the right. This leaves the $(\vee_r)$, $(\wedge_r)$, $(\Box)$, $([i])$, and $(\otimes_i)$ cases; we show the latter three cases below, as the first two are resolved as in [Neg05, Thm. 4.12].

$$
\dfrac{\dfrac{\mathcal{R} \Rightarrow u : \phi, w : \Box\phi, \Gamma}{\mathcal{R} \Rightarrow w : \Box\phi, w : \Box\phi, \Gamma}\,(\Box)}{\mathcal{R} \Rightarrow w : \Box\phi, \Gamma}\,(ctr_r)
\quad \rightsquigarrow \quad
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\mathcal{R} \Rightarrow u : \phi, w : \Box\phi, \Gamma}{\mathcal{R} \Rightarrow u : \phi, v : \phi, \Gamma}\,\text{Lem. 22}}{\mathcal{R}(u/v) \Rightarrow (u : \phi)(u/v), (v : \phi)(u/v), \Gamma(u/v)}\,(lsb)}{\mathcal{R} \Rightarrow u : \phi, u : \phi, \Gamma}\,=}{\mathcal{R} \Rightarrow u : \phi, \Gamma}\,\text{IH}}{\mathcal{R} \Rightarrow w : \Box\phi, \Gamma}\,(\Box)
$$

$$
\dfrac{\dfrac{\mathcal{R}, R_{[i]}wu \Rightarrow u : \phi, w : [i]\phi, \Gamma}{\mathcal{R} \Rightarrow w : [i]\phi, w : [i]\phi, \Gamma}\,([i])}{\mathcal{R} \Rightarrow w : [i]\phi, \Gamma}\,(ctr_r)
\quad \rightsquigarrow
$$

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\mathcal{R}, R_{[i]}wu \Rightarrow u : \phi, w : [i]\phi, \Gamma}{\mathcal{R}, R_{[i]}wu, R_{[i]}wv \Rightarrow u : \phi, v : \phi, \Gamma}\,\text{Lem. 22}}{\mathcal{R}(u/v), (R_{[i]}wu)(u/v), (R_{[i]}wv)(u/v) \Rightarrow (u : \phi)(u/v), (v : \phi)(u/v), \Gamma(u/v)}\,(lsb)}{\mathcal{R}, R_{[i]}wu, R_{[i]}wu \Rightarrow u : \phi, u : \phi, \Gamma}\,=}{\mathcal{R}, R_{[i]}wu \Rightarrow u : \phi, \Gamma}\,(ctr_{\mathcal{R}}) + \text{IH}}{\mathcal{R} \Rightarrow w : [i]\phi, \Gamma}\,([i])
$$

$$
\dfrac{\dfrac{\mathcal{R}, I_{\otimes_i}u \Rightarrow u : \phi, w : \otimes_i\phi, \Gamma}{\mathcal{R} \Rightarrow w : \otimes_i\phi, w : \otimes_i\phi, \Gamma}\,(\otimes_i)}{\mathcal{R} \Rightarrow w : \otimes_i\phi, \Gamma}\,(ctr_r)
\quad \rightsquigarrow
$$

$$
\begin{array}{cl}
\mathcal{R}, I_{\otimes_i}u \Rightarrow u : \phi, w : \otimes_i \phi, \Gamma & \\
\hdashline
\mathcal{R}, I_{\otimes_i}u, I_{\otimes_i}v \Rightarrow u : \phi, v : \phi, \Gamma & \text{Lem. 22} \\
\hdashline
\mathcal{R}(u/v), (I_{\otimes_i}u)(u/v), (I_{\otimes_i}v)(u/v) \Rightarrow (u : \phi)(u/v), (v : \phi)(u/v), \Gamma(u/v) & (lsb) \\
\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots & = \\
\mathcal{R}, I_{\otimes_i}u, I_{\otimes_i}u \Rightarrow u : \phi, u : \phi, \Gamma & \\
\hdashline
\mathcal{R}, I_{\otimes_i}u \Rightarrow u : \phi, \Gamma & (ctr_{\mathcal{R}}) + \text{IH} \\
\hline
\mathcal{R} \Rightarrow w : \otimes_i \phi, \Gamma & (\otimes_i)
\end{array}
$$

□

Theorem 14. *The rule* (*cut*) *is eliminable in* $\mathsf{G3DS}_n^k$.

Proof. We assume w.l.o.g. that we are given a derivation where (*cut*) is the last rule used, and that no other instances of (*cut*) appear in the given derivation. The general result follows by repeatedly applying the algorithm described below to successively eliminate topmost instances of (*cut*) until the derivation is free of (*cut*) instances. We prove the result by induction on the lexicographic ordering of pairs $(|\phi|, h_1 + h_2)$, where $|\phi|$ is the complexity of the cut formula ϕ, h_1 is the height of the derivation of the left premise of (*cut*), and h_2 is the height of the derivation of the right premise of (*cut*). As is typical when proving cut elimination, there are a large number of cases, and so, we explicitly write the assumptions being made in each case for clarity.

1. One of the premises of (*cut*) *is an instance of* (*id*).

1.1 The left premise of (*cut*) *is an instance of* (*id*). Then, the left premise of (*cut*) is of the form $\mathcal{R} \Rightarrow w : p, w : \neg p, \Delta$, and there are three subcases to consider:

1.1.1 The cut formula is $w : p$. It follows that the right premise of (*cut*) is of the form $\mathcal{R} \Rightarrow w : \neg p, w : \neg p, \Gamma$, and the conclusion of (*cut*) is of the form $\mathcal{R} \Rightarrow w : \neg p, \Gamma$. Observe that applying the hp-admissiblity of (ctr_r) (Lem. 23) to the proof of the right premise yields a proof of the desired conclusion.

1.1.2 The cut formula is $w : \neg p$. This case is similar to the previous case 1.1.1.

1.1.3 The cut formula is in Γ. Then, the conclusion of (*cut*) is of the form $\mathcal{R} \Rightarrow w : p, w : \neg p, \Gamma'$, which is an instance of (*id*).

1.2 The right premise of (*cut*) *is an instance of* (*id*). This case is shown similarly to the previous case 1.1.

2. None of the premises of (*cut*) *are an instance of* (*id*). In this case, we let $\mathcal{R} \Rightarrow w : \phi, \Gamma$ be the left premise of (*cut*) and $\mathcal{R} \Rightarrow w : \neg\phi, \Gamma$ by the right premise of (*cut*).

2.1 The cut formula $w : \phi$ *is not principal in the left premise of* (*cut*). We show the ($\otimes_i$), ($\Diamond$), and (IOA) cases, as the remaining cases are similar. In each case below we may invoke IH as the sum of the heights $h_1 + h_2$ has decreased.

2.1.1 The left premise of (*cut*) *is derived with* ($\otimes_i$).

$$\dfrac{\dfrac{\mathcal{R}, I_{\otimes_i}u \Rightarrow u : \psi, w : \phi, \Gamma}{\mathcal{R} \Rightarrow v : \otimes_i\psi, w : \phi, \Gamma}(\otimes_i) \qquad \mathcal{R} \Rightarrow v : \otimes_i\psi, w : \neg\phi, \Gamma}{\mathcal{R} \Rightarrow v : \otimes_i\psi, \Gamma}(cut) \quad \rightsquigarrow$$

$$\dfrac{\dfrac{\mathcal{R}, I_{\otimes_i}u \Rightarrow u : \psi, w : \phi, \Gamma \qquad \dfrac{\mathcal{R} \Rightarrow v : \otimes_i\psi, w : \neg\phi, \Gamma}{\mathcal{R}, I_{\otimes_i}u \Rightarrow u : \psi, w : \neg\phi, \Gamma}\text{Lem. 22}}{\mathcal{R}, I_{\otimes_i}u \Rightarrow u : \psi, \Gamma}\text{IH}}{\mathcal{R} \Rightarrow v : \otimes_i\psi, \Gamma}(\otimes_i)$$

2.1.2 The left premise of (cut) *is derived with* $(\Diamond)$.

$$\dfrac{\dfrac{\mathcal{R} \Rightarrow v : \Diamond\psi, u : \psi, w : \phi, \Gamma}{\mathcal{R} \Rightarrow v : \Diamond\psi, w : \phi, \Gamma}(\Diamond) \qquad \mathcal{R} \Rightarrow v : \Diamond\psi, w : \neg\phi, \Gamma}{\mathcal{R} \Rightarrow v : \Diamond\psi, \Gamma}(cut) \quad \rightsquigarrow$$

$$\dfrac{\dfrac{\mathcal{R} \Rightarrow v : \Diamond\psi, u : \psi, w : \phi, \Gamma \qquad \dfrac{\mathcal{R} \Rightarrow v : \Diamond\psi, w : \neg\phi, \Gamma}{\mathcal{R} \Rightarrow v : \Diamond\psi, u : \psi, w : \neg\phi, \Gamma}\text{Lem. 22}}{\mathcal{R} \Rightarrow v : \Diamond\psi, u : \psi, \Gamma}\text{IH}}{\mathcal{R} \Rightarrow v : \Diamond\psi, \Gamma}(\Diamond)$$

2.1.3 The left premise of (cut) *is derived with* (IOA).

$$\dfrac{\dfrac{\mathcal{R}, R_{[0]}w_0u, ..., R_{[n]}w_nu \Rightarrow w : \phi, \Gamma}{\mathcal{R} \Rightarrow w : \phi, \Gamma}(IOA) \qquad \mathcal{R} \Rightarrow w : \neg\phi, \Gamma}{\mathcal{R} \Rightarrow \Gamma}(cut) \quad \rightsquigarrow$$

$$\dfrac{\dfrac{\mathcal{R}, R_{[0]}w_0u, ..., R_{[n]}w_nu \Rightarrow w : \phi, \Gamma \qquad \dfrac{\mathcal{R} \Rightarrow w : \neg\phi, \Gamma}{\mathcal{R}, R_{[0]}w_0u, ..., R_{[n]}w_nu \Rightarrow w : \neg\phi, \Gamma}\text{Lem. 22}}{\mathcal{R}, R_{[0]}w_0u, ..., R_{[n]}w_nu \Rightarrow \Gamma}\text{IH}}{\mathcal{R} \Rightarrow \Gamma}(IOA)$$

2.2 The cut formula $w : \phi$ *is principal in the left premise of* (cut) *only.* This case is proven in a similar manner to the previous case 2.1.

2.3 The cut formula $w : \phi$ *is principal both premises of* (cut). We omit the cases where ϕ is of the form $\psi \lor \chi$, $\psi \land \chi$, $[i]\psi$, and $\langle i \rangle\psi$ as these are handled in a similar fashion to the cut-elimination proof [Neg05, Thm. 4.13] for extensions of the modal logic K. Without loss of generality, we show the cases where ϕ is of the form $\Box\psi$ and $\otimes_i\psi$.

2.3.1 The cut formula is of the form $\Box\psi$.

$$\dfrac{\dfrac{\mathcal{R} \Rightarrow v : \psi, \Gamma}{\mathcal{R} \Rightarrow w : \Box\psi, \Gamma}(\Box) \qquad \dfrac{\mathcal{R} \Rightarrow w : \Diamond\neg\psi, u : \neg\psi, \Gamma}{\mathcal{R} \Rightarrow w : \Diamond\neg\psi, \Gamma}(\Diamond)}{\mathcal{R} \Rightarrow \Gamma}(cut) \quad \rightsquigarrow$$

$$\dfrac{\Pi \qquad \dfrac{\dfrac{\mathcal{R} \Rightarrow v : \psi, \Gamma}{\mathcal{R}(u/v) \Rightarrow (v : \psi)(u/v), \Gamma(u/v)}\,(lsb)}{\mathcal{R} \Rightarrow u : \psi, \Gamma}\,=}{\mathcal{R} \Rightarrow \Gamma}\,\text{IH}$$

$$\Pi \quad = \left\{ \quad \dfrac{\dfrac{\dfrac{\mathcal{R} \Rightarrow v : \psi, \Gamma}{\mathcal{R} \Rightarrow v : \psi, u : \neg\psi, \Gamma}\,(wk)}{\mathcal{R} \Rightarrow w : \Box\psi, u : \neg\psi, \Gamma}\,(\Box) \qquad \mathcal{R} \Rightarrow w : \Diamond\neg\psi, u : \neg\psi, \Delta}{\mathcal{R} \Rightarrow u : \neg\psi, \Gamma}\,\text{IH} \right.$$

2.3.2 The cut formula is of the form $\otimes_i\psi$.

$$\dfrac{\dfrac{\mathcal{R}, I_{\otimes_i}v \Rightarrow v : \phi, \Gamma}{\mathcal{R} \Rightarrow w : \otimes_i\phi, \Gamma}\,(\otimes_i) \qquad \dfrac{\mathcal{R}, I_{\otimes_i}u \Rightarrow w : \ominus_i\neg\phi, u : \neg\phi, \Gamma}{\mathcal{R}, I_{\otimes_i}u \Rightarrow w : \ominus_i\neg\phi, \Gamma}\,(\ominus_i)}{\mathcal{R}, I_{\otimes_i}u \Rightarrow \Gamma}\,(cut) \quad \rightsquigarrow$$

$$\dfrac{\Pi \qquad \dfrac{\dfrac{\mathcal{R}, I_{\otimes_i}v \Rightarrow v : \phi, \Gamma}{\mathcal{R}(u/v), (I_{\otimes_i}v)(u/v) \Rightarrow (u : \phi)(u/v), \Gamma(u/v)}\,(lsb)}{\mathcal{R}, I_{\otimes_i}u \Rightarrow u : \phi, \Gamma}\,=}{\mathcal{R}, I_{\otimes_i}u \Rightarrow \Gamma}\,\text{IH}$$

$$\Pi \quad = \left\{ \quad \dfrac{\dfrac{\dfrac{\mathcal{R}, I_{\otimes_i}v \Rightarrow v : \phi, \Gamma}{\mathcal{R}, I_{\otimes_i}u, I_{\otimes_i}v \Rightarrow v : \phi, u : \neg\phi, \Gamma}\,(wk)}{\mathcal{R}, I_{\otimes_i}u \Rightarrow w : \otimes_i\phi, u : \neg\phi, \Gamma}\,(\otimes_i) \qquad \mathcal{R}, I_{\otimes_i}u \Rightarrow w : \ominus_i\neg\phi, u : \neg\phi, \Delta}{\mathcal{R}, I_{\otimes_i}u \Rightarrow u : \neg\phi, \Gamma, \Delta}\,\text{IH} \right.$$

□

As for the calculi Sect. 3.2, we need to show that each calculus $\mathsf{G3DS}^k_n$ is complete relative to classical propositional logic, that is, all classical propositional tautologies are derivable in each calculus $\mathsf{G3DS}^k_n$. This ensures that axioms given via A0 in HDS^k_n (Def. 26) are derivable in $\mathsf{G3DS}^k_n$, which can then be harnessed to show full completeness relative to DS^k_n. We state the classical completeness lemma below, and defer the proof to the appendix (Appendix B on p. 208) to simplify presentation.

Lemma 24 (Classical Completeness)**.** *All instances of classical propositional tautologies in $\mathcal{L}_{\mathsf{DS}^k_n}$ are derivable in $\mathsf{G3DS}^k_n$.*

Proof. See Appendix B (p. 208) for details. □

Theorem 15 (Completeness)**.** *If $\vdash_{\mathsf{DS}^k_n} \phi$, then $\vdash_{\mathsf{G3DS}^k_n} w : \phi$.*

Proof. By Lem. 24, we know that A0 holds. Moreover, the derivations of axioms A1 – A3 are similar to the derivation of axiom A1 in the proof of Thm. 7, R0 is derived similar to the derivation of R0 in the proof of Thm. 7, and R1 is derived similar to the derivation of R1 in the proof of Thm. 7. We therefore show how to derive axioms A4 – A14 below:

Axioms A4 and A5.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{R_{[i]}wu \Rightarrow w : \Diamond\neg\phi, u : \neg\phi, u : \phi}\text{ Lem. 19}}{R_{[i]}wu \Rightarrow w : \Diamond\neg\phi, u : \phi}(\Diamond)}{\Rightarrow w : \Diamond\neg\phi, w : [i]\phi}([i])}{\Rightarrow w : \Diamond\neg\phi \lor [i]\phi}(\lor_r)}{\Rightarrow w : \Box\phi \rightarrow [i]\phi}= \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{I_{\otimes_i}u \Rightarrow w : \Diamond\neg\phi, u : \neg\phi, u : \phi}\text{ Lem. 19}}{I_{\otimes_i}u \Rightarrow w : \Diamond\neg\phi, u : \phi}(\Diamond)}{\Rightarrow w : \Diamond\neg\phi, w : \otimes_i\phi}(\otimes_i)}{\Rightarrow w : \Diamond\neg\phi \lor \otimes_i\phi}(\lor_r)}{\Rightarrow w : \Box\phi \rightarrow \otimes_i\phi}=$$

Axioms A6 and A7.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\Rightarrow w : \Diamond\neg\phi, w : \neg\phi, w : \phi}\text{ Lem. 19}}{\Rightarrow w : \Diamond\neg\phi, w : \phi}(\Diamond)}{\Rightarrow w : \Diamond\neg\phi \lor \phi}(\lor_r)}{\Rightarrow w : \Box\phi \rightarrow \phi}= \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Rightarrow u : \neg\phi, v : \Diamond\phi, u : \phi}\text{ Lem. 19}}{\Rightarrow u : \neg\phi, v : \Diamond\phi}(\Diamond)}{\Rightarrow w : \Box\neg\phi, w : \Box\Diamond\phi}(\Box) \times 2}{\Rightarrow w : \Box\neg\phi \lor \Box\Diamond\phi}(\lor_r)}{\Rightarrow w : \Diamond\phi \rightarrow \Box\Diamond\phi}=$$

Axiom A8

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{R_{[i]}wu \Rightarrow w : \langle i\rangle\neg\phi, u : \neg\phi, u : \phi}\text{ Lem. 19}}{R_{[i]}wu \Rightarrow w : \langle i\rangle\neg\phi, w : \phi}(\langle i\rangle)}{\Rightarrow w : \langle i\rangle\neg\phi, w : \phi}(ref_i)}{\Rightarrow w : \langle i\rangle\neg\phi \lor \phi}(\lor_r)}{\Rightarrow w : [i]\phi \rightarrow \phi}=$$

Axiom A9.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{R_{[i]}wu, R_{[i]}wv, R_{[i]}vu \Rightarrow u : \neg\phi, v : \langle i\rangle\phi, u : \phi}\text{ Lem. 19}}{R_{[i]}wu, R_{[i]}wv, R_{[i]}vu \Rightarrow u : \neg\phi, v : \langle i\rangle\phi}(\langle i\rangle)}{R_{[i]}wu, R_{[i]}wv \Rightarrow u : \neg\phi, v : \langle i\rangle\phi}(euc_i)}{\Rightarrow w : [i]\neg\phi, w : [i]\langle i\rangle\phi}([i]) \times 2}{\Rightarrow w : [i]\neg\phi \lor [i]\langle i\rangle\phi}(\lor_r)}{\Rightarrow w : \langle i\rangle\phi \rightarrow [i]\langle i\rangle\phi}=$$

Axiom A10.

$$
\begin{array}{cl}
\hdashline
I_{\otimes_i}u \Rightarrow u : \phi, u : \neg\phi, w : \ominus_i\neg\phi, w : \ominus_i\phi & \text{Lem. 19}\\
\hline
I_{\otimes_i}u \Rightarrow u : \neg\phi, w : \ominus_i\neg\phi, w : \ominus_i\phi & (\ominus_i)\\
\hline
I_{\otimes_i}u \Rightarrow w : \ominus_i\neg\phi, w : \ominus_i\phi & (\ominus_i)\\
\hline
\Rightarrow w : \ominus_i\neg\phi, w : \ominus_i\phi & (D^i_2)\\
\hline
\Rightarrow w : \ominus_i\neg\phi \lor \ominus_i\phi & (\lor_r)\\
\cdots\cdots\cdots\cdots\cdots\cdots & =\\
\Rightarrow w : \otimes_i\phi \rightarrow \ominus_i\phi &
\end{array}
$$

Axiom A11.

$$
\begin{array}{cl}
\hdashline
I_{\otimes_i}v \Rightarrow v : \neg\phi, v : \phi, u : \ominus_i\neg\phi, v : \otimes_i\phi & \text{Lem. 19}\\
\hline
I_{\otimes_i}v \Rightarrow v : \phi, u : \ominus_i\neg\phi, v : \otimes_i\phi & (\otimes_i)\\
\hline
\Rightarrow u : \ominus_i\neg\phi, v : \otimes_i\phi & (\otimes_i)\\
\hline
\Rightarrow w : \Box \ominus_i \neg\phi, w : \Box \otimes_i \phi & (\Box) \times 2\\
\hline
\Rightarrow w : \Box \ominus_i \neg\phi \lor \Box \otimes_i \phi & (\lor_r)\\
\cdots\cdots\cdots\cdots\cdots\cdots & =\\
\Rightarrow w : \Diamond \otimes_i \phi \rightarrow \Box \otimes_i \phi &
\end{array}
$$

Axiom A12.

$$
\begin{array}{cl}
\hdashline
R_{[i]}uv, I_{\otimes_i}u, I_{\otimes_i}v \Rightarrow w : \ominus_i\neg\phi, v : \neg\phi, v : \phi & \text{Lem. 19}\\
\hline
R_{[i]}uv, I_{\otimes_i}u, I_{\otimes_i}v \Rightarrow w : \ominus_i\neg\phi, v : \phi & (\ominus_i)\\
\hline
R_{[i]}uv, I_{\otimes_i}u \Rightarrow w : \ominus_i\neg\phi, v : \phi & (D^i_3)\\
\hline
I_{\otimes_i}u \Rightarrow w : \ominus_i\neg\phi, u : [i]\phi & ([i])\\
\hline
\Rightarrow w : \ominus_i\neg\phi, w : \otimes_i[i]\phi & (\otimes_i)\\
\hline
\Rightarrow w : \ominus_i\neg\phi \lor \otimes_i[i]\phi & (\lor_r)\\
\cdots\cdots\cdots\cdots\cdots\cdots & =\\
\Rightarrow w : \otimes_i\phi \rightarrow \otimes_i[i]\phi &
\end{array}
$$

Axiom A13. Let $i \in \mathrm{Ag} = \{0, \ldots, n\}$, $\mathcal{R} := R_{[0]}u_0v, \ldots, R_{[n]}u_nv$, and $\Gamma := u_0 : \langle 0\rangle\neg\phi_0, \ldots, u_n : \langle n\rangle\neg\phi_n, w : \Diamond(\bigwedge_{i\in\mathrm{Ag}}[i]\phi_i)$. Also, we define $\mathcal{R}_i$ be equal to $\mathcal{R}$ minus the relational atom $R_{[i]}u_iv$, and Γ_i be equal to Γ minus the labelled formula $u_i : \neg\phi$. The independence of agents axiom A13 is derived as follows:

$$
\begin{array}{cl}
\Pi_1 \qquad \cdots \qquad \Pi_n & \\
\hline
\mathcal{R} \Rightarrow u_0 : \langle 0\rangle\neg\phi_0, \ldots, u_n : \langle n\rangle\neg\phi_n, w : \Diamond(\bigwedge_{i\in\mathrm{Ag}}[i]\phi_i), v : \bigwedge_{i\in\mathrm{Ag}}[i]\phi_i & (\land_r) \times (n-1)\\
\hline
\mathcal{R} \Rightarrow u_0 : \langle 0\rangle\neg\phi_0, \ldots, u_n : \langle n\rangle\neg\phi_n, w : \Diamond(\bigwedge_{i\in\mathrm{Ag}}[i]\phi_i) & (\Diamond)\\
\hline
\Rightarrow u_0 : \langle 0\rangle\neg\phi_0, \ldots, u_n : \langle n\rangle\neg\phi_n, w : \Diamond(\bigwedge_{i\in\mathrm{Ag}}[i]\phi_i) & (IOA)\\
\hline
\Rightarrow w : \Box\langle 0\rangle\neg\phi_0, \ldots, w : \Box\langle n\rangle\neg\phi_n, w : \Diamond(\bigwedge_{i\in\mathrm{Ag}}[i]\phi_i) & (\Box) \times n\\
\hline
\Rightarrow w : \Box\langle 0\rangle\neg\phi_0 \lor \cdots \lor \Box\langle n\rangle\neg\phi_n \lor \Diamond(\bigwedge_{i\in\mathrm{Ag}}[i]\phi_i) & (\lor_r) \times (n-1)\\
\cdots\cdots\cdots\cdots\cdots\cdots & =\\
\Rightarrow w : \Diamond[0]\phi_0 \land \cdots \land \Diamond[n]\phi_n \rightarrow \Diamond(\bigwedge_{i\in\mathrm{Ag}}[i]\phi_i) &
\end{array}
$$

$$\Pi_i \quad = \left\{ \quad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\mathcal{R}_i, R_{[i]}vu, R_{[i]}u_iv, R_{[i]}u_iu \Rightarrow \Gamma_i, u_i : \langle i\rangle\neg\phi_i, u : \neg\phi_i, u : \phi_i}\ \text{Lem. 19}}{\mathcal{R}_i, R_{[i]}vu, R_{[i]}u_iv, R_{[i]}u_iu_i, \mathcal{R}_{[i]}vu_i, \mathcal{R}_{[i]}u_iu \Rightarrow \Gamma_i, u_i : \langle i\rangle\neg\phi_i, u : \phi_i}\ (\langle i\rangle)}{\mathcal{R}_i, R_{[i]}vu, R_{[i]}u_iv, R_{[i]}u_iu_i, \mathcal{R}_{[i]}vu_i \Rightarrow \Gamma_i, u_i : \langle i\rangle\neg\phi_i, u : \phi_i}\ (euc_i)}{\mathcal{R}_i, R_{[i]}vu, R_{[i]}u_iv, R_{[i]}u_iu_i \Rightarrow \Gamma_i, u_i : \langle i\rangle\neg\phi_i, u : \phi_i}\ (euc_i)}{\mathcal{R}_i, R_{[i]}vu, R_{[i]}u_iv \Rightarrow \Gamma_i, u_i : \langle i\rangle\neg\phi_i, u : \phi_i}\ (ref_i)}{\mathcal{R}_i, R_{[i]}u_iv \Rightarrow \Gamma_i, u_i : \langle i\rangle\neg\phi_i, v : [i]\phi_i}\ [i] \right.$$

Axiom A14. For the $\Pi_{0,j}$ derivations below, we let $1 \leq j \leq k$, and for the $\Pi_{m,j}$ derivations, we let $0 < m \leq k-1$ and $m+1 \leq j \leq n$. We let

$$\Gamma := w_0 : \phi_1, \ldots, w_0 : \phi_k$$

and $\Gamma_{0,j}$ be the multiset

$$w_1 : \langle i\rangle\neg\phi_1, w_2 : \phi_1, w_2 : \langle i\rangle\neg\phi_2, \ldots, w_k : \phi_1, \ldots, w_k : \phi_{k-1}, w_k : \langle i\rangle\neg\phi_k, w_0 : \phi_1, \ldots, w_0 : \phi_k$$

minus $w_0 : \phi_j, w_j : \langle i\rangle\neg\phi_j$. Last, we let $\Gamma_{m,j}$ be the multiset

$$w_1 : \langle i\rangle\neg\phi_1, w_2 : \phi_1, w_2 : \langle i\rangle\neg\phi_2, \ldots, w_k : \phi_1, \ldots, w_k : \phi_{k-1}, w_k : \langle i\rangle\neg\phi_k, w_0 : \phi_1, \ldots, w_0 : \phi_k$$

minus $w_k : \langle i\rangle\neg\phi_k, w_j : \phi_k$ and $q = \frac{k^2-k}{2}$. The axiom is derived as follows:

$$\Pi_{0,j} \quad = \left\{ \quad \dfrac{\dfrac{\dfrac{R_{[i]}w_0w_0, R_{[i]}w_0w_j, R_{[i]}w_jw_0 \Rightarrow w_0 : \phi_j, w_j : \langle i\rangle\neg\phi_j, w_0 : \neg\phi_j, \Gamma_{0,j}}{R_{[i]}w_0w_0, R_{[i]}w_0w_j, R_{[i]}w_jw_0, \ldots, w_0 : \phi_j, w_j : \langle i\rangle\neg\phi_j, \Gamma_{0,j}}\ \langle i\rangle}{R_{[i]}w_0w_0, R_{[i]}w_0w_j, \ldots, w_0 : \phi_j, w_j : \langle i\rangle\neg\phi_j, \Gamma_{0,j}}\ (euc_i)}{R_{[i]}w_0w_j \Rightarrow w_0 : \phi_j, w_j : \langle i\rangle\neg\phi_j, \Gamma_{0,j}}\ (ref_i) \right.$$

$$\Pi_{m,j} \quad = \left\{ \quad \dfrac{R_{[i]}w_kw_j \Rightarrow w_k : \langle i\rangle\neg\phi_k, w_j : \neg\phi_k, w_j : \phi_k, \Gamma_{m,j}}{R_{[i]}w_kw_j \Rightarrow w_k : \langle i\rangle\neg\phi_k, w_j : \phi_k, \Gamma_{m,j}}\ \langle i\rangle \right.$$

$$\dfrac{\dfrac{\dfrac{\Big\{\Pi_{m,j} \;\Big|\; 0 \leq m \leq k-1,\, m+1 \leq j \leq k\Big\}}{\Rightarrow w_1 : \langle i\rangle\neg\phi_1, w_2 : \phi_1, w_2 : \langle i\rangle\neg\phi_2, \ldots, w_k : \phi_1, \ldots, w_k : \phi_{k-1}, w_k : \langle i\rangle\neg\phi_k, \Gamma}\ (APC_i^k)}{\Rightarrow w_0 : \Box\langle i\rangle\neg\phi_1, w_0 : \Box(\phi_1 \lor \langle i\rangle\neg\phi_2), \ldots, w_0 : \Box(\phi_1 \lor \cdots \phi_{k-1} \lor \langle i\rangle\neg\phi_k), \Gamma}\ (\Box) \times k + (\lor_r) \times q}{\Rightarrow w_0 : \Box\langle i\rangle\neg\phi_1 \lor \Box(\phi_1 \lor \langle i\rangle\neg\phi_2) \lor \cdots \lor \Box(\phi_1 \lor \cdots \phi_{k-1} \lor \langle i\rangle\neg\phi_k) \lor \phi_1 \lor \cdots \lor \phi_k}\ (\lor_r) \times (2k-1)$$

$$\overset{\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\ =}{\Rightarrow w : \Diamond[i]\phi_1 \land \Diamond(\neg\phi_1 \land [i]\phi_2) \land \cdots \land \Diamond(\neg\phi_1 \land \cdots \land \neg\phi_{k-1} \land [i]\phi_k) \to \phi_1 \lor \cdots \lor \phi_k}$$

□

Similar to the previous two sections, we define sequent graphs for our labelled sequents as well as special types of labelled sequents called *labelled forest sequents* and *labelled DAG sequents*. These reduced structures will naturally emerge in derivations after we refined our deontic STIT calculi (Ch. 4).

Definition 55 (Sequent Graphs for Deontic STIT Logics)**.** *Let* $\Lambda := \mathcal{R} \Rightarrow \Gamma$ *be a labelled sequent for deontic STIT logics. We define the* sequent graph *of* Λ*,* $G(\Lambda) = (V, E, L)$*, as follows:*

- $V = \mathrm{Lab}(\Lambda)$
- $E = \{(w, u, [i]) \mid R_{[i]}wu \in \mathcal{R}\}$
- $L(w) = (\{i \in \mathrm{Ag} \mid I_{\otimes_i}w \in \mathcal{R}\}; \{\phi \mid w : \phi \in \Gamma\})$

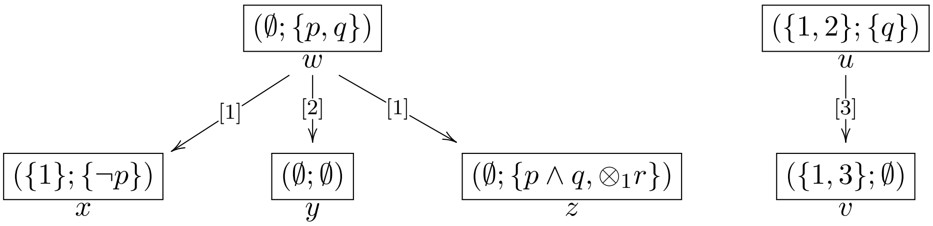
Definition 56 (Labelled Forest/DAG sequent)**.** *A labelled sequent for deontic STIT logics* Λ *is a* labelled forest sequent *(*labelled DAG sequent*) iff* $G(\Lambda) = (V, E, L)$ *is a* forest *(*$G(\Lambda) = (V, E, L)$ *is a* DAG*, resp.).*

To provide additional intuition regarding such labelled sequents, we give an example of a labelled forest sequent and a labelled DAG sequent below:

Example 3. *Below, we give an example of a labelled forest sequent* Λ_1 *along with its corresponding sequent graph* $G(\Lambda_1)$*, and an example of a labelled DAG sequent* Λ_2 *along with its corresponding sequent graph* $G(\Lambda_2)$*. The labels* w *and* u *serve as the roots in* Λ_1*, and the label* w *and* v *serve as the roots in* Λ_2*.*

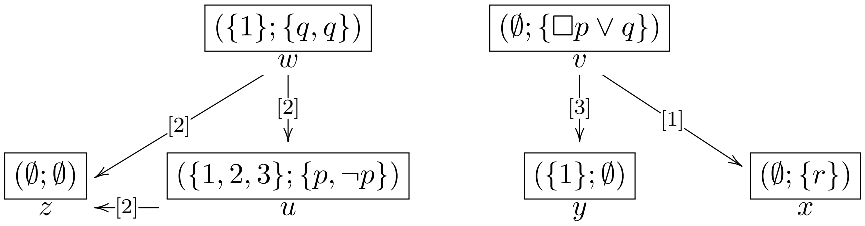
$$\Lambda_1 := R_{[1]}wx, R_{[2]}wy, R_{[1]}wz, R_{[1]}uv, I_{\otimes_1}x, I_{\otimes_1}u, I_{\otimes_2}u, I_{\otimes_1}v, I_{\otimes_3}v \Rightarrow \Gamma_1$$

$$\Gamma_1 := w : p, w : q, x : \neg p, z : p \wedge q, z : \otimes_1 r, u : q$$



$$\Lambda_2 := R_{[2]}wu, R_{[2]}wz, R_{[2]}uz, R_{[3]}vy, R_{[1]}vx, I_{\otimes_1}w, I_{\otimes_1}u, I_{\otimes_2}u, I_{\otimes_3}u, I_{\otimes_1}y \Rightarrow \Gamma_2$$

$$\Gamma_2 := w : q, w : q, u : p, u : \neg p, v : \Box p \vee q, x : r$$



After refining our class of calculi for deontic STIT logics, we will find that certain subclasses of calculi utilize derivations of a certain shape. In the refined setting, when we set $n = 0$, that is, the number of agents $|\mathrm{Ag}| = 1$, we will find that completeness is preserved if we restrict ourselves to the use of labelled forest sequents, and if we set $k = 0$, meaning that no upper bound is imposed on the number of choices available to our agents, then we may restrict ourselves to the use of labelled DAG sequents in derivations. Below, we explicitly define such derivations, and also introduce the *rooted property.* In essence, a labelled forest or DAG derivation possesses the rooted property if, reading the derivation in a bottom-up manner, whenever a label is introduced as a root in the sequent graph of a labelled sequent, it remains a root in all labelled sequents higher up in the derivation. The significance of the rooted property is twofold: first, confirming that the rooted property holds for labelled forest or DAG derivations lets us relate our refined labelled calculi (derived in the next chapter) to the nested and indexed-nested sequent formalisms, where proofs effectively have the rooted property. Second, the rooted property has practical value, as it tells us that information 'flows' from certain points when inference rules are applied to labelled forest or DAG sequents in reverse, i.e. during proof-search, trees and DAGs 'grow' from roots in sequent graphs of labelled sequents. This observation is useful in establishing termination, and will be seen in Sect. 6.1 on proof-search and decidability for deontic STIT logics.

Definition 57 (Labelled Forest/DAG Proof, Rooted Property)**.** *We say that a proof is a* labelled forest proof *(*labelled DAG proof*) iff it consists solely of labelled forest sequents (it consists solely of labelled DAG sequents, resp.).*

Also, we say that a labelled forest (DAG) proof Π *has the* rooted property *iff for all labelled sequents* Λ_1 *and* Λ_2 *occurring in* Π*, if* Λ_1 *occurs below* Λ_2 *and* $w_1, \ldots, w_m$ *are the roots of all connected induced subgraphs in* $G(\Lambda_1)$*, then they are all roots of connected induced subgraphs in* $G(\Lambda_2)$*.*

Last, we prove a theorem below showing that each calculus $\mathsf{G3DS}_n^k$ is incomplete relative to labelled forest and DAG derivations. This theorem is useful in comparing our labelled and refined labelled calculi. In fact, we will confirm that refinement produces systems which permit less underlying structure in their labelled sequents, that is to say, refined labelled systems for certain deontic STIT logics allow for completeness relative to labelled forest and DAG derivations. This topic will be discussed in Sect. 4.2 on refining labelled calculi for deontic STIT logics.

Theorem 16. *Let* $n, k \in \mathbb{N}$*. The calculus* $\mathsf{G3DS}_n^k$ *is incomplete relative to labelled forest and labelled DAG derivations.*

Proof. The theorem follows by considering the proof(s) of $[i]p \rightarrow p = \langle i \rangle \neg p \lor p$ (an instance of axiom A11), which requires the use of (ref_i) as shown below:

$$
\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{}{R_{[i]}ww \Rightarrow w : \langle i \rangle \neg p, w : \neg p, w : p}\;(id)}{R_{[i]}ww \Rightarrow w : \langle i \rangle \neg p, w : p}\;(\langle i \rangle)}{\epsilon \Rightarrow w : \langle i \rangle \neg p, w : p}\;(ref_i)}{\epsilon \Rightarrow w : \langle i \rangle \neg p \vee p}\;(\vee_r)}{\epsilon \Rightarrow w : [i]p \rightarrow p}\;=
$$

When applied bottom-up, (ref_i) adds a loop to the sequent graph of a labelled sequent, which breaks both the labelled tree and labelled DAG property. Moreover, a quick inspection of the rules of $\mathsf{G3DS}_n^k$ will show that (ref_i) is necessary to derive the above theorem, showing the incompleteness of the calculus relative to labelled forest and DAG derivations. □

CHAPTER 4

The Method of Refinement: Modal Propositional Logics

We introduce the method of refinement—a means by which labelled calculi may be simplified through the introduction of propagation rules and the elimination of structural rules—and apply the method in the context of grammar and deontic STIT logics. The next chapter will apply the method of refinement in the context of first-order intuitionistic logics. The central mechanism behind this process is *structural rule elimination*, whereby structural rules encoding frame properties—e.g. (p_s^x) in $\mathsf{G3Km}(S)$ (see Fig. 3.1), (nd) in $\mathsf{G3IntQ}$ (see Fig. 3.3), and (D_2^i) in $\mathsf{G3DS}_n^k$ (see Fig. 3.5)—are permuted upward in a derivation and are either deleted at initial sequents or absorbed into certain logical rules. In order to carry out the elimination of structural rules, we introduce *propagation rules* (see, e.g. [CdCGH97, CLRT20, Fit72, LvB19, Sim94, TIG12]), or a generalization thereof (which we call, *reachability rules*), to our calculi. We note that we will refer to such rules—that allow for the elimination of structural rules to go through—as *conducive*.

The addition of conducive rules (i.e. propagation and reachability rules in our setting) and the elimination of structural rules yields *refined* labelled proof systems that are more economical in the sense that they consist of less rules, produce shorter proofs (compared to the original labelled systems from which they were derived), and need only utilize labelled sequents of a reduced form (e.g. labelled sequents whose sequent graphs are trees; see Def. 44). We will discuss the consequences of these effects along with additional practical consequences of refinement below. However, before discussing the advantages of refinement as well as related work, we give the reader a brief introduction to the methodology. First, we introduce and provide an example of propagation rules, followed by an example of structural rule elimination, and last, give an example of the type of (nested) proof system that is commonly produced via refinement.

Propagation rules have a unique operation with bottom-up applications corresponding to the propagation of a formula along a path within the data structure (or, graph) encoded by the sequent. For example, we might have a propagation rule such as $(Pr^{1}_{\langle a \rangle})$ shown below left, which makes use of the labelled formula $w : \langle a \rangle \phi$ to bottom-up introduce the labelled formula $w : \phi$ (observe that there is a path of length 0 between the label of the principal and auxiliary formula, i.e. the labels are identical). Another example of a propagation rule is $(Pr^{2}_{\langle a \rangle})$, which is shown below right, and makes use of the labelled formula $w : \langle a \rangle \phi$ and the path of relational atoms $R_a wv, R_a vu$ to bottom-up introduce the labelled formula $u : \phi$ associated with the label u at the end of the relational path of length 2.

$$\frac{\mathcal{R} \Rightarrow w : \langle a \rangle \phi, w : \phi, \Gamma}{\mathcal{R} \Rightarrow w : \langle a \rangle \phi, \Gamma} \ (Pr^{1}_{\langle a \rangle}) \qquad \frac{\mathcal{R}, R_a wv, R_a vu \Rightarrow w : \langle a \rangle \phi, u : \phi, \Gamma}{\mathcal{R}, R_a wv, R_a vu \Rightarrow w : \langle a \rangle \phi, \Gamma} \ (Pr^{2}_{\langle a \rangle})$$

The use of propagation rules goes at least as far back as 1972, where Fitting employed rules of a similar functionality within the context of prefixed tableaux for normal modal logics [Fit72].[1] Such rules have been integrated into numerous proof-theoretic frameworks for diverse classes of logics, such as in the labelled sequent formalism for intuitionistic modal logics [Sim94], tense logics [CLRT20], and STIT logics [LvB19]; in the nested sequent formalism for grammar logics [TIG12]; and in the context of prefixed tableaux for normal modal logics [CdCGH97, Fit72]. While propagation rules function by propagating formulae along paths within a sequent (when applied bottom-up), applications of reachability rules (which will be discussed in Ch. 5) additionally depend on the (non-)existence of data occurring along paths within a sequent.

To provide the reader with intuition concerning the process of structural rule elimination and the utility of propagation rules, we consider an example derivation in the labelled calculus $\mathsf{G3Km}(S)$ (see Fig. 3.1) for the grammar logic $\mathsf{Km}(S)$ (see Def. 12) where $S := \{a \longrightarrow \varepsilon, \overline{a} \longrightarrow \varepsilon\}$ and $\sum := \{a, \overline{a}\}$ (i.e. the alphabet is the set of characters $\{a, \overline{a}\}$). The calculus consists of the rules shown below, where we have a (p^{x}_{ε}), (C^{x}_{1}), $(\langle x \rangle)$, and $([x])$ rule for each $x \in \sum$. Also, as usual, the label u is an eigenvariable in the $([x])$ rule.

$$\frac{}{\mathcal{R} \Rightarrow w : p, w : \neg p, \Gamma} \ (id) \qquad \frac{\mathcal{R} \Rightarrow w : \phi, w : \psi, \Gamma}{\mathcal{R} \Rightarrow w : \phi \lor \psi, \Gamma} \ (\lor_r) \qquad \frac{\mathcal{R}, R_x ww \Rightarrow \Gamma}{\mathcal{R} \Rightarrow \Gamma} \ (p^{x}_{\varepsilon})$$

$$\frac{\mathcal{R} \Rightarrow w : \phi, \Gamma \qquad \mathcal{R} \Rightarrow w : \psi, \Gamma}{\mathcal{R} \Rightarrow w : \phi \land \psi, \Gamma} \ (\land_r) \qquad \frac{\mathcal{R}, R_x wu \Rightarrow w : \langle x \rangle \phi, u : \phi, \Gamma}{\mathcal{R}, R_x wu \Rightarrow w : \langle x \rangle \phi, \Gamma} \ (\langle x \rangle)$$

$$\frac{\mathcal{R}, R_x wu \Rightarrow u : \phi, \Gamma}{\mathcal{R} \Rightarrow w : [x] \phi, \Gamma} \ ([x]) \qquad \frac{\mathcal{R}, R_x wu, R_{\overline{x}} uw \Rightarrow \Gamma}{\mathcal{R}, R_x wu \Rightarrow \Gamma} \ (C^{x}_{1})$$

[1] The term *propagation rule* was not used in Fitting's paper [Fit72] and seems to have been coined in [CdCGH97].

Let us now consider the elimination of the structural rule (p^a_ε) from a given derivation. We assume that we are given the derivation shown top-left below, and make use of the propagation rule $(Pr^1_{\langle a \rangle})$, which was introduced above, to permute the (p^a_ε) rule upward at the first step. Thus, the $(\langle a \rangle)$ and (p^a_ε) inferences are transformed into a (p^a_ε) inference followed by a $(Pr^1_{\langle a \rangle})$ inference. Since the conclusion of the (p^a_ε) inference in the top-right derivation below is an instance of (id), we may delete the (p^a_ε) instance altogether, giving the output derivation shown below bottom.

$$\dfrac{\dfrac{\dfrac{}{R_a ww \Rightarrow w : \langle a \rangle p, w : p, w : \neg p}\,(id)}{R_a ww \Rightarrow w : \langle a \rangle p, w : \neg p}\,(\langle a \rangle)}{\epsilon \Rightarrow w : \langle a \rangle p, w : \neg p}\,(p^a_\varepsilon) \quad \rightsquigarrow \quad \dfrac{\dfrac{\dfrac{}{R_a ww \Rightarrow w : \langle a \rangle p, w : p, w : \neg p}\,(id)}{\epsilon \Rightarrow w : \langle a \rangle p, w : p, w : \neg p}\,(p^a_\varepsilon)}{\epsilon \Rightarrow w : \langle a \rangle p, w : \neg p}\,(Pr^1_{\langle a \rangle})$$

$$\rightsquigarrow \quad \dfrac{\dfrac{}{\epsilon \Rightarrow w : \langle a \rangle p, w : p, w : \neg p}\,(id)}{\epsilon \Rightarrow w : \langle a \rangle p, w : \neg p}\,(Pr^1_{\langle a \rangle})$$

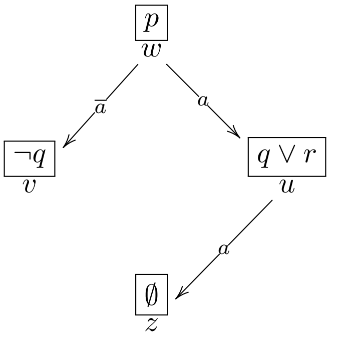
Typically, structural rule elimination has the effect of simplifying the data structure underlying labelled sequents in a proof. This fact can be seen in the example above: observe that the sequent graphs (Def. 55) of the top two sequents in the input (i.e. top-left) derivation contain a loop due to the presence of the relational atom $R_a ww$, whereas the sequent graphs of all sequents in the output derivation consist of a single point. As explained above, introducing propagation or reachability rules to a calculus, which permits the elimination of structural rules, produces labelled calculi that employ simpler syntactic structures in their proofs and—interestingly—tend to be notational variants of *nested sequent systems* [Bul92, Kas94, Str13, TIG12]. It should be noted that refinement does not always produce refined labelled calculi that are notational variants of nested systems, as will be discussed and demonstrated in Sect. 4.2; similar to the systems in [MS14] for (intuitionistic) modal logics, our calculi for deontic STIT logics may contain a mixture of propagation rules and structural rules since it is not known if propagation rules exist which allow for certain structural rules to be eliminated. Still, as will be discussed in that section as well, the data structures underlying labelled sequents in a proof are nevertheless reduced after refinement has taken place.

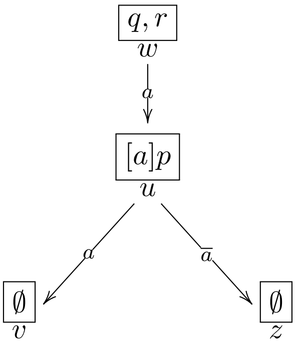
The creation of the nested sequent formalism is often attributed to Bull [Bul92] and Kashima [Kas94] (and is equivalent to the prefixed tableaux formalism introduced much earlier by Fitting [Fit72]). The formalism is characterized by the use of generalized versions of Gentzen-style sequents, which utilize nesting constructors to organize Gentzen-style sequents (or, multisets of formulae) into trees. To give the reader an idea of nested sequent systems and an idea of the output of the refinement process, we introduce nested sequents for the grammar logic $\mathsf{Km}(S)$ with $S := \{a \longrightarrow \varepsilon, \overline{a} \longrightarrow \varepsilon\}$ and $\sum := \{a, \overline{a}\}$ (just introduced above), and show the nested calculus that results from refining the labelled calculus $\mathsf{G3Km}(S)$ (which was just introduced above as well).

The following grammar in BNF defines nested sequents for the grammar logic $\mathsf{Km}(S)$ considered above:

$$X ::= \epsilon \mid \phi \mid X, X \mid (a)\{X\} \mid (\overline{a})\{X\}$$

where $\phi \in \mathcal{L}_{\mathsf{Km}}(\sum)$ and ϵ is the empty sequent. One can readily check that such syntactic structures encode trees; for example, the sequent graph encoded by the nested sequent $p, (\overline{a})\{\neg q\}, (a)\{q \lor r, (a)\{\epsilon\}\}$ is shown below left and the sequent graph encoded by the nested sequent $q, r, (a)\{[a]p, (a)\{\epsilon\}, (\overline{a})\{\epsilon\}\}$ is shown below right. Both graphs are trees in the sense of Def. 32.





If we were to eliminate the structural rules (p^x_ε) and (C^x_1) for each $x \in \sum = \{a, \overline{a}\}$ from the labelled calculus $\mathsf{G3Km}(S)$ introduced above (by expanding the calculus with sufficient propagation rules), then after switching from labelled to nested notation, we would obtain the nested calculus shown below, where we have a $([x])$ and $(Pr_{\langle x \rangle})$ rule for each $x \in \sum$. (NB. As will be discussed in Sect. 4.1 below, it is sometimes advantageous to work with labelled notation as opposed to nested notation. Therefore, translating refined labelled calculi into nested calculi is an *optional* last step in the refinement process.) The notation $X[Y]$ and $X[Y][Z]$ is used to represent that Y and/or Z occur at some level of the nestings in the nested sequent X. For example, if our nested sequent X is $p, (a)\{q, (\overline{a})\{r \land \neg r\}\}$, then $X[p]$, $X[q, (\overline{a})\{r \land \neg r\}]$, and $X[r \land \neg r]$ would all be correct representations of X in this notation. Last, we note that the propagation rule $(Pr_{\langle x \rangle})$ possesses a side condition † stipulating how to use $\langle x \rangle \phi$ to bottom-up propagate the formula ϕ to the point witnessing Z in the nested sequent X. We omit the exact description of this side condition here as it is rather complex, and note that such rules will be defined in the succeeding section (Sect. 4.1).

$$\frac{}{X[p, \neg p, Y]}\ (id) \qquad \frac{X[\phi, \psi, Y]}{X[\phi \lor \psi, Y]}\ (\lor_r) \qquad \frac{X[\phi, Y] \qquad X[\psi, Y]}{X[\phi \land \psi, Y]}\ (\land_r)$$

$$\frac{X[Y, (x)\{\phi\}]}{X[Y, [x]\phi]}\ ([x]) \qquad \frac{X[\langle x \rangle \phi, Y][\phi, Z]}{X[\langle x \rangle \phi, Y][Z]}\ (Pr_{\langle x \rangle})^{\dagger}$$

Refining labelled calculi yields a variety of advantages: first, the replacement of structural rules with propagation or reachability rules leads to a compression in the size of proofs, a decrease in the total number of inference rules, and a reduction in the complexity of the data structure underlying labelled sequents in a proof. This can lead to a savings in space and an increase in efficiency for implementations or automated reasoning algorithms (e.g. proof-search, counter-model extraction, effective interpolation). Second, the removal of certain semantic elements and structures—e.g. domain atoms $\underline{a} \in D_w$ (see Def. 46) used in labelled sequents—from the syntax of sequents, makes the calculi easier to understand and handle. Third, proving decidability for logics via (unrefined) labelled calculi is often complicated and/or uses *ad hoc* methods that are logic dependent [Neg05, Vig00]. By contrast, if we employ more refined sequents (e.g. trees, forests, DAGs) in our calculi, then analyzing and demonstrating the termination of proof-search procedures becomes easier and has proven to be *uniform*; for example, terminating proof-search with nested sequent calculi is shown uniformly for a class of grammar logics in [TIG12]. Fourth, the refinement method has been applied to translate between the labelled paradigm and non-labelled proof-theoretic formalisms (e.g. [CLRT20]), thus allowing for results to be transferred between the distinct settings, for formalisms to be changed when one is better suited for the task at hand, and elucidating the semantic information inherent in syntactic structures of the non-labelled formalism (which has explanatory value). Last, the calculi we obtain are *modular* with respect to their associated classes of logics, that is, by the alteration of side conditions or the deletion/addition of rules, any refined calculus can be transformed into another refined calculus with the same properties (e.g. invertibility of rules, cut-admissibility, etc.), but for another logic in the considered class.

Another benefit of refinement concerns how the method may be composed with methods and results from the labelled sequent paradigm. Although refinement is intimately tied to simplification, the method can be seen as part of a more extensive method whereby the semantics of a logic may be transformed into a proof calculus possessing desirable properties and which is suited for certain applications. As explained in Ch. 1 and 3, an advantage of the labelled formalism is that it allows for the straightforward extraction of (labelled) proof systems from semantics. However, such calculi involve complicated structures, violate the subformula property to a high degree, and tend to be unwieldy. By composing the construction process for labelled calculi with the simplification process obtained from structural rule elimination, we obtain a strategy for transforming the semantics of a logic into a refined calculus in possession of 'nice' properties (e.g. such systems satisfy the desiderata of [Wan94] to a large extent).[2]

The idea of refinement is foreshadowed by works studying the relationship between labelled and alternative proof systems. Such works include [GR12], where translations between labelled and tree-hypersequent/nested sequent calculi were introduced for Gödel-Löb provability logic; the papers [CLR18, CLRT20], which provided translations between labelled, display, and nested calculi for tense logics; and the paper [Pim18], which

[2] See the introduction (Ch. 1) for a discussion of Wansing's desiderata, used to characterize 'nice' proof systems.

provided translations between labelled and nested calculi for intuitionistic, normal, and non-normal modal propositional logics. In each case, the labelled calculi were shown equivalent to calculi within a reduced proof-theoretic formalism, and made use of structural rule elimination arguments to obtain the more economical calculi. The refinement methodology abstracts from such works and gives a more general strategy for simplifying labelled calculi within a wide range of settings. A significant departure from the aforementioned works concerns the use of grammar theoretic machinery (*viz.* CFCST systems; see Def. 8) in defining propagation and reachability rules. As will be seen in Sect. 4.1 below and the following chapter (Ch. 5), the use of such machinery allows for relatively broad classes of propagation and reachability rules to be defined, thus permitting the elimination of structural rules in a large number of scenarios. This allows us to provide the first results showing that labelled calculi for grammar, deontic STIT, and first-order intuitionistic logics can be 'simplified' and transformed into nested calculi in many cases.[3]

This chapter is based on work from [vBL21, CLR18, CLRT20, LvB19] and is organized as follows: In the first section (Sect. 4.1), we show that each labelled calculus for a grammar logic can be algorithmically transformed into a refined labelled calculus with structural rules replaced by propagation rules (Thm. 17) and where each derivation of a theorem only makes use of labelled tree sequents (Thm. 18). We additionally prove that all refined labelled calculi are notational variants of (slight reformulations of) the nested calculi from [TIG12] (Thm. 20 and 21). The section ends with a discussion on the relationship between refined labelled calculi and the shallow nested (i.e. display) calculi from [TIG12]. In the second section (Sect. 4.2), we refine the labelled calculi for our deontic STIT logics, demonstrating that structural rules may be eliminated in the presence of certain propagation rules (Thm. 22), begetting systems that are complete relative to labelled DAG derivations (Thm. 24) or labelled forest derivations (Cor. 8) depending on the deontic STIT logic considered. In the following chapter (Ch. 5), we discuss and apply refinement in the first-order setting, which, due to its more complex nature, will require us to improve upon and expand the refinement methodology. Details of how the methodology is augmented in a first-order context will be discussed there.

4.1 Refining Labelled Calculi for Grammar Logics

We apply the refinement method to each labelled calculus $\mathsf{G3Km}(S)$ for the grammar logic $\mathsf{Km}(S)$ to obtain a new calculus $\mathsf{Km}(S)\mathsf{L}$. Before giving formal proofs confirming the extraction of the latter from the former, we motivate and explain how the conductive (i.e. propagation) rules (shown in Fig. 4.2) of $\mathsf{Km}(S)\mathsf{L}$ arise naturally when attempting to eliminate structural rules—*viz.* (p_s^x) and (C_1^x) (see Fig. 3.1)—from $\mathsf{G3Km}(S)$. The goal, therefore, is not simply to claim that refinement can be done, but to demonstrate the

[3] It should be noted that labelled calculi for traditional STIT logics and first-order intuitionistic logics were refined in [LvB19] and [Lyo20a, Lyo20b], essentially yielding nested systems, so similar results have been shown in more restricted settings.

$$\begin{tikzcd}
\mathsf{G3Km}(S) \arrow[r, bend left, "\text{Thm. 17}"] & \mathsf{Km}(S)\mathsf{L} \arrow[l, bend left, "\text{Thm. 19}"] \arrow[r, dotted, bend left, "\text{Thm. 20}", "\mathfrak{N}" description] & \mathsf{DKm}(S) \arrow[l, dotted, bend left, "\text{Thm. 21}", "\mathfrak{L}" description]
\end{tikzcd}$$

Figure 4.1: Transformations and translations between grammar logic calculi.

type of analysis sufficient to allow for refinement to be performed. After deriving each calculus $\mathsf{Km}(S)\mathsf{L}$ from its parent calculus $\mathsf{G3Km}(S)$, we show that each refined calculus is a notational variant of a nested calculus $\mathsf{DKm}(S)$ from [TIG12]. The insight that refinement generates (slight variants of) known nested calculi (discovered independently of the method), suggests a naturalness to the method of refinement and of the associated nested calculi.

Fig. 4.1 summarizes the main transformations and translations between the calculi studied in this section. We use solid arrows to represent *transformations*, which preserve the language of the calculus (e.g. labelled sequents) while algorithmically mapping proofs between the two systems (as in Thm. 17 and 19), and we use dotted arrows to represent *translations*, which not only algorithmically map proofs between the two systems, but also change the language in the process (as in Thm. 20 and 21). The dotted arrows are annotated with the symbols $\mathfrak{N}$ and $\mathfrak{L}$, which are the translation functions introduced in Def. 81 and 82, respectively.

4.1.1 Extracting $\mathsf{Km}(S)\mathsf{L}$ from $\mathsf{G3Km}(S)$

We begin by analyzing the elimination of a (p^{x}_{s}) rule in a labelled calculus $\mathsf{G3Km}(S)$, and observe under what conditions (p^{x}_{s}) cannot be permuted upward in a derivation. Such situations will motivate and suggest the addition of propagation rules to $\mathsf{G3Km}(S)$ that allow for the permutation to go through. In the labelled setting, such rules propagate formulae along paths of relational atoms occurring in a labelled sequent (when read bottom-up). The rules introduced here are based on the propagation rules for nested and labelled sequents introduced in [CLRT20, LvB19, GPT11, TIG12]. Before defining our propagation rules (which are provided in Fig. 4.2), we will explain how such rules arise naturally when considering the elimination of a (p^{x}_{s}) rule.

For the sake of illustration, let us suppose that we have the following CFCST system $S := \{a \longrightarrow ab, \overline{a} \longrightarrow \overline{a}\overline{b}\}$. We will attempt to prove the admissibility of the structural rule (p^{a}_{ab}) by showing that the rule can always be permuted upward in a derivation. The reader can verify that the (p^{a}_{ab}) rule is permutable with every rule in $\mathsf{G3Km}(S)$ with the exception of the $(\langle x \rangle)$ and (C^{x}_{1}) rules. We fix the characters $a, b \in \sum$ and focus on permuting the (p^{a}_{ab}) rule above the $(\langle a \rangle)$ rule, as analyzing this case will be sufficient to motivate and explain the definition of propagation rules and their addition to $\mathsf{G3Km}(S)$. Furthermore, we will not consider the permutation of (p^{a}_{ab}) above any (C^{x}_{1}) structural rule for the following reason: since our aim is to prove the (p^x_s) and (C^x_1) rules admissible via an elimination algorithm, we can consider topmost occurrences of rules in a given derivation and successively eliminate them via their respective elimination procedures; therefore, we never need to consider the permutation of one rule above the other. As stated previously, our analysis will motivate the definition of a propagation rule, and thus provide the reader with insight into how such rules are discovered and defined via the process of structural rule elimination.

Let us assume that we have a derivation ending with an $(\langle a \rangle)$ inference followed by a (p^a_{ab}) inference. We want to show that the rule can always be moved upward in a derivation and still allow for the same conclusion to be derived.

$$\dfrac{\dfrac{\mathcal{R}, R_a wv, R_b vu, R_a wu \Rightarrow w : \langle a \rangle \phi, u : \phi, \Gamma}{\mathcal{R}, R_a wv, R_b vu, R_a wu \Rightarrow w : \langle a \rangle \phi, \Gamma} (\langle a \rangle)}{\mathcal{R}, R_a wv, R_b vu \Rightarrow w : \langle a \rangle \phi, \Gamma} (p^a_{ab})$$

Since our aim is to eliminate the (p^a_{ab}) rule, we will apply the rule directly to the top sequent shown above with the goal of using other rules in $\mathsf{G3Km}(S)$ to derive the same conclusion. Applying (p^a_{ab}) directly to the top sequent shown above gives us the following:

$$\dfrac{\mathcal{R}, R_a wv, R_b vu, R_a wu \Rightarrow w : \langle a \rangle \phi, u : \phi, \Gamma}{\mathcal{R}, R_a wv, R_b vu \Rightarrow w : \langle a \rangle \phi, u : \phi, \Gamma} (p^a_{ab})$$

In order to derive the desired conclusion, we need to find a set of rules in $\mathsf{G3Km}(S)$ that lets us delete $u : \phi$; however, a quick glance at the rules of $\mathsf{G3Km}(S)$ will demonstrate that no rules are applicable allowing for $u : \phi$ to be deleted. Nevertheless, notice that if we extend our calculus with the (r_1) rule below, then the desired conclusion can be derived by applying the rule directly to the conclusion of the inference above.

$$\dfrac{\mathcal{R}, R_a wv, R_b vu \Rightarrow w : \langle a \rangle \phi, u : \phi, \Gamma}{\mathcal{R}, R_a wv, R_b vu \Rightarrow w : \langle a \rangle \phi, \Gamma} (r_1)$$

The rule (r_1) essentially states that if there is a sequence of relational atoms of the form $R_a wv, R_b vu$ between a label w and a label u in a sequent, with the labelled formulae $w : \langle a \rangle \phi$ and $u : \phi$ also occurring in the sequent, then $u : \phi$ may be deleted from premise to conclusion. If we consider the sequent semantics (Def. 41), we can see that this rule is sound since if a $\mathsf{Km}(S)$-model satisfies $R_a wv, R_b vu$, then because $a \longrightarrow ab \in S$, the model will also satisfy $R_a wu$ as $R_{ab} \subseteq R_a$. This last fact implies that we can think of the relational atom $R_a wu$ as being *implicit* in both premise and conclusion of (r_1), showing that the rule is essentially a special instance of the (sound) $(\langle a \rangle)$ rule.

In accordance with what has been said, we can see that if we extend our calculus $\mathsf{G3Km}(S)$ with the rule (r_1), then the (p^a_{ab}) rule can be permuted above $(\langle a \rangle)$. Still, this does not immediately imply that (p^a_{ab}) can be eliminated from any given derivation, as we now have to check if the (p^{a}_{ab}) rule can be permuted above (r_1). We will consider this below, and observe the problematic case:

$$\dfrac{\dfrac{\mathcal{R}, R_a wz, R_b zv, R_a wv, R_b vu \Rightarrow w : \langle a\rangle\phi, u : \phi, \Gamma}{\mathcal{R}, R_a wz, R_b zv, R_a wv, R_b vu \Rightarrow w : \langle a\rangle\phi, \Gamma}\ (r_1)}{\mathcal{R}, R_a wz, R_b zv, R_b vu \Rightarrow w : \langle a\rangle\phi, \Gamma}\ (p^{a}_{ab})$$

Again, since our goal is to eliminate the (p^{a}_{ab}) rule, we apply it directly to the top sequent above (giving the inference below), and then aim to derive the same conclusion using other rules of $\mathsf{G3Km}(S) + (r_1)$.

$$\dfrac{\mathcal{R}, R_a wz, R_b zv, R_a wv, R_b vu \Rightarrow w : \langle a\rangle\phi, u : \phi, \Gamma}{\mathcal{R}, R_a wz, R_b zv, R_b vu \Rightarrow w : \langle a\rangle\phi, u : \phi, \Gamma}\ (p^{a}_{ab})$$

As in the case of permuting (p^{a}_{ab}) above $(\langle a\rangle)$, no rules are present in $\mathsf{G3Km}(S) + (r_1)$ that will allow us to derive the desired conclusion. We could, as before, add a new rule to our calculus that will bring about the desired permutation:

$$\dfrac{\mathcal{R}, R_a wz, R_b zv, R_b vu \Rightarrow w : \langle a\rangle\phi, u : \phi, \Gamma}{\mathcal{R}, R_a wz, R_b zv, R_b vu \Rightarrow w : \langle a\rangle\phi, \Gamma}\ (r_2)$$

It is easy to see that applying this rule to the conclusion of the former (p^{a}_{ab}) inference allows for us to derive the desired conclusion, thus showing that (p^{a}_{ab}) can be permuted above (r_1) in the calculus $\mathsf{G3Km}(S) + \{(r_1), (r_2)\}$. As the reader might have noticed, although the addition of (r_2) to our calculus allows for (p^{a}_{ab}) to be permuted above (r_1), we now have to check if (p^{a}_{ab}) can be permuted above (r_2). If such a situation is analyzed, then one would observe that in order for the (p^{a}_{ab}) rule to be permuted above (r_2), a new rule (r_3) would have to be added to our calculus, which would necessitate the addition of another rule (r_4) to allow for (p^{a}_{ab}) to be permuted above (r_3)—this phenomenon would continue ad infinitum. Although we could add the whole infinite lot of such rules to our calculus in order to secure the elimination of (p^{a}_{ab}), there is a more elegant solution to our problem.

Observe that the active relational atoms in (r_1) are of the form $R_a wv, R_b vu = R_{ab} wu$ and the active relational atom in (r_2) are of the form $R_a wz, R_b zv, R_b vu = R_{abb} wu$. Recall that the production rule $a \longrightarrow ab$ is element of our CFCST system S, implying that $a \twoheadrightarrow^{*}_{S} ab$ and $a \twoheadrightarrow^{*}_{S} abb$. A quick comparison of the strings derived from a and the active relational atoms of (r_1) and (r_2) shows that the strings derived from a serve as the indices in the active relational atoms. In fact, if one observes the infinite set of rules generated by trying to prove the eliminability of (p^{a}_{ab}), they would find that the active relational atoms of all such rules correspond to strings derivable from a in S. In other words, the indices correspond to strings in the language $L_S(a)$. Therefore, instead of adding an infinite number of rules to $\mathsf{G3Km}(S)$ to allow for the elimination of (p^{a}_{ab}), we could instead add a single rule of the form:

$$\frac{\mathcal{R} \Rightarrow w : \langle a \rangle \phi, u : \phi, \Gamma}{\mathcal{R} \Rightarrow w : \langle a \rangle \phi, \Gamma} \; (Pr_{\langle a \rangle})$$

where we impose a side condition stating that a sequence of relational atoms must exist in $\mathcal{R}$ 'corresponding to' strings in the language $L_S(a)$. The exact meaning of 'corresponding to' relies on the formulation of a *propagation graph* and *propagation path*, which are formally defined in Def. 58 and Def. 59 below, respectively, and are based on the work in [CLRT20, GPT11]. The first notion transforms a labelled sequent into a graph, and the second notion defines a correspondence between paths in the graph and strings in a language generated by a CFCST system. After giving these definitions, examples of a propagation graph, path, and rule are provided for clarity, followed by Fig. 4.2 which defines the propagation rules $(Pr_{\langle x \rangle})$.

Definition 58 (Propagation Graphs for $\mathsf{Km}(S)\mathsf{L}$)**.** *Let $\Lambda = \mathcal{R} \Rightarrow \Gamma$ be a labelled sequent for grammar logics. We define the* propagation graph $PG(\Lambda) = (\mathrm{V}, \mathrm{E})$ *to be the directed graph such that*

- $\mathrm{V} := \mathrm{Lab}(\Lambda)$*;*
- $\mathrm{E} := \{(w, u, a), (u, w, \overline{a}) \mid R_a wu \in \mathcal{R} \text{ or } R_{\overline{a}} uw \in \mathcal{R}\}$.

We will often write $w \in PG(\Lambda)$ to mean $w \in \mathrm{V}$, and $(w, u, x) \in PG(\Lambda)$ to mean $(w, u, x) \in \mathrm{E}$, for $x \in \sum$.

Definition 59 (Propagation Path for $\mathsf{Km}(S)\mathsf{L}$)**.** *Let Λ be a labelled sequent with $PG(\Lambda) = (\mathrm{V}, \mathrm{E})$. We define a* propagation path from w_1 to w_n *in $PG(\Lambda)$ to be an alternating sequence of vertices $w_1, \ldots, w_n \in \mathrm{V}$ and characters $x_1, \ldots, x_{n-1} \in \sum$ of the form:*

$$\pi(w_1, w_n) := w_1, x_1, w_2, x_2, \ldots, x_{n-1}, w_n$$

such that $(w_1, w_2, x_1), (w_2, w_3, x_2), \ldots, (w_{n-1}, w_n, x_{n-1}) \in \mathrm{E}$. Given a propagation path $\pi(w_1, w_n) = w_1, x_1, w_2, x_2, \ldots, x_{n-1}, w_n$, we let $s_\pi(w_1, w_n) = x_1 x_2 \cdots x_{n-1}$ denote the string of the propagation path from w_1 to w_n*.*

We define the converse of a propagation path *as follows:*

$$\overline{\pi}(w_n, w_1) := w_n, \overline{x}_n, w_{n-1}, \ldots, w_2, \overline{x}_1, w_1 \textit{ iff } \pi(w_1, w_n) := w_1, x_1, w_2, \ldots, w_{n-1}, x_{n-1}, w_n.$$

Also, the converse string of a propagation path *as defined follows:*

$$\overline{s}_\pi(w_n, w_1) := \overline{x}_n \overline{x}_{n-1} \cdots \overline{x}_1 \textit{ iff } s_\pi(w_1, w_n) = x_1 x_2 \cdots x_{n-1}.$$

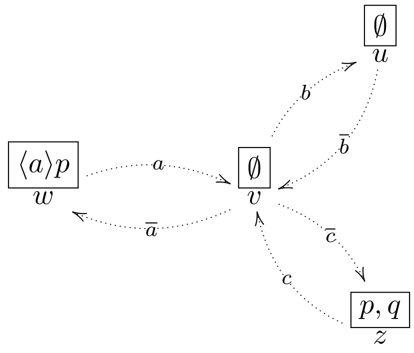
Last, we let $\lambda(w, w) := w$ represent the empty path *that holds between any vertex $w \in V$ and itself, with the string of the empty path defined as follows: $s_\lambda(w, w) := \varepsilon$.*

We are now in a position to properly define our propagation rules. Fig. 4.2 defines a propagation rule $(Pr_{\langle x\rangle})$ for each $x \in \sum$ and gives the side condition dictating its (in)application. For the moment, let us refer to the premise of the propagation rule $(Pr_{\langle x\rangle})$ as Λ and its conclusion as Λ'. The side condition '$\exists\pi(s_{\pi}(w,u) \in L_{S}(x))$' should be read as stating that 'there exists a propagation path $\pi(w,u)$ in the propagation graph $PG(\Lambda) = PG(\Lambda')$ such that $s_{\pi}(w,u) \in L_{S}(x)$.' Observe that since the propagation graph of the premise is identical to the propagation graph of the conclusion, that is $PG(\Lambda) = PG(\Lambda')$, propagation rules can be just as easily applied bottom-up as they can be applied top-down. We will use a similar notation to denote the side conditions of propagation (and reachability) rules in Ch. 5 as well. Last, each refined labelled calculus $\mathsf{Km}(S)\mathsf{L}$ for each grammar logic $\mathsf{Km}(S)$ is displayed in Fig. 4.2 with the derivability relation for each calculus defined as follows:

Definition 60. *We write $\vdash_{\mathsf{Km}(S)\mathsf{L}} \Lambda$ to indicate that a labelled sequent Λ is derivable in a calculus $\mathsf{Km}(S)\mathsf{L}$.*

To supply the reader with additional intuition regarding propagation graphs, paths, and rules, an example unifying all such concepts is provided below:

Example 4. *We give a pictorial representation of the propagation graph $PG(\Lambda)$, where Λ is the labelled sequent defined below. Although it is not technically an aspect of the definition of a propagation graph, we also decorate the vertices of the propagation graph to show which formulae are associated with what vertices.*



$$\Lambda := R_{a}wv, R_{\overline{b}}uv, R_{c}zv \Rightarrow w : \langle a\rangle p, z : p, z : q$$

Let us suppose that our CFCST system is $S := \{a \longrightarrow ab\overline{b}\overline{c}, \overline{a} \longrightarrow \overline{a}\overline{b}bc\}$. Observe that the path $\pi(w,z) := w, a, v, b, u, \overline{b}, v, \overline{c}, z$ exists between w and z with $\langle a\rangle p$ occurring at the starting vertex w, and p occurring at the terminal vertex z. Since $s_{\pi}(w,z) = ab\overline{b}\overline{c} \in L_{S}(a)$ (due to the first production rule of S), we can apply the propagation rule $(Pr_{\langle a\rangle})$ to Λ to delete the labelled formula $z : p$, giving $R_{a}wv, R_{\overline{b}}uv, R_{c}zv \Rightarrow w : \langle a\rangle p, z : q$ as the conclusion.

Lemma 25. *For each $x \in \sum$, the rule $(\langle x\rangle)$ is an instance of $(Pr_{\langle x\rangle})$.*

$$\frac{}{\mathcal{R} \Rightarrow w : p, w : \neg p, \Gamma}\ (id)$$

$$\frac{\mathcal{R} \Rightarrow w : \phi, w : \psi, \Gamma}{\mathcal{R} \Rightarrow w : \phi \vee \psi, \Gamma}\ (\vee_r) \qquad \frac{\mathcal{R} \Rightarrow w : \phi, \Gamma \qquad \mathcal{R} \Rightarrow w : \psi, \Gamma}{\mathcal{R} \Rightarrow w : \phi \wedge \psi, \Gamma}\ (\wedge_r)$$

$$\frac{\mathcal{R}, R_x wu \Rightarrow u : \phi, \Gamma}{\mathcal{R} \Rightarrow w : [x]\phi, \Gamma}\ [x]^{\dagger_1} \qquad \frac{\mathcal{R} \Rightarrow w : \langle x \rangle \phi, u : \phi, \Gamma}{\mathcal{R} \Rightarrow w : \langle x \rangle \phi, \Gamma}\ (Pr_{\langle x \rangle})^{\dagger_2}$$

Figure 4.2: The refined calculus $\mathsf{Km}(S)\mathsf{L}$ for the grammar logic $\mathsf{Km}(S)$. We have a $[x]$ and $(Pr_{\langle x \rangle})$ rule for each $x \in \sum$. The side condition $\dagger_1$ states that the associated rule can be applied only if the label u is an eigenvariable. The side condition $\dagger_2$ states that $\exists \pi (s_\pi(w,u) \in L_S(x))$.

Proof. Let $x \in \sum$. The premise of $(\langle x \rangle)$ is of the form $\mathcal{R}, R_x wu \Rightarrow w : \langle x \rangle \phi, u : \phi, \Gamma$ and its propagation graph contains the propagation path w, x, u (due to the occurrence of $R_x wu$). Since $x \in L_S(x)$, the side condition of the propagation rule $(Pr_{\langle x \rangle})$ is satisfied, allowing for the conclusion $\mathcal{R}, R_x wu \Rightarrow w : \langle x \rangle \phi, \Gamma$ to be derived. □

The above lemma tells us that each propagation rule $(Pr_{\langle x \rangle})$ subsumes $(\langle x \rangle)$, meaning that we need only consider the permutation of a (p^x_s) rule above $(Pr_{\langle x \rangle})$ and not $(\langle x \rangle)$. Therefore, since the only non-trivial case of proving the eliminability of (p^x_s) concerns its permutation with $(\langle x \rangle)$, if we are able to prove that the structural rule (p^x_s) can always be permuted above the propagation rule $(Pr_{\langle x \rangle})$, then we will have successfully solved the problem discussed at the onset of this section, namely, the problem of eliminating each (p^x_s) rule from a given derivation. In the following lemma, we show that the propagation rules are in fact sufficient to allow for the upward mobility of each (p^x_s) rule in a derivation; after showing this result, we give a concrete example showing how to permute a (p^x_s) rule above a $(Pr_{\langle x \rangle})$ rule for clarity.

Lemma 26. *Let S be a CFCST system with $x, x_0, \ldots, x_n, y \in \sum$, and define the following:*

- $R_s wu := R_{x_0} wu_1, \ldots R_{x_n} u_n u$
- $\Lambda := \mathcal{R}, R_s wu, R_x wu \Rightarrow v : \langle y \rangle \phi, z : \phi, \Gamma$
- $\Lambda' := \mathcal{R}, R_s wu \Rightarrow v : \langle y \rangle \phi, z : \phi, \Gamma$

Moreover, let $\pi(v,z)$ be a propagation path between v and z occurring in $PG(\Lambda)$.

Suppose we are given a derivation in $\mathsf{G3Km}(S) + \{(Pr_{\langle x \rangle}) \mid x \in \sum\}$ ending with the following inferences:

$$\dfrac{\dfrac{\mathcal{R}, R_s wu, R_x wu \Rightarrow v : \langle y \rangle \phi, z : \phi, \Gamma}{\mathcal{R}, R_s wu, R_x wu \Rightarrow v : \langle y \rangle \phi, \Gamma}\ (Pr_{\langle y \rangle})}{\mathcal{R}, R_s wu \Rightarrow v : \langle y \rangle \phi, \Gamma}\ (p^x_s)$$

where $s_{\pi}(v, z) \in L_S(y)$. Then, there exists a path $\pi'(v, z)$ in $PG(\Lambda')$ such that $s_{\pi'}(v, z) \in L_S(y)$, that is to say, the (p^x_s) rule may be permuted above the $(Pr_{\langle y \rangle})$ rule to derive the same end sequent as shown below:

$$\dfrac{\dfrac{\mathcal{R}, R_s wu, R_x wu \Rightarrow v : \langle y \rangle \phi, z : \phi, \Gamma}{\mathcal{R}, R_s wu \Rightarrow v : \langle y \rangle \phi, z : \phi, \Gamma}\ (p^x_s)}{\mathcal{R}, R_s wu \Rightarrow v : \langle y \rangle \phi, \Gamma}\ (Pr_{\langle y \rangle})$$

Note that (p^x_s) may represent a structural rule obtained via the closure condition.

Proof. Let our assumptions be those expressed in the statement of the lemma above. We consider two cases: either (i) the relational atom $R_x wu$ is not active in the $(Pr_{\langle y \rangle})$ inference, or (ii) the relational atom $R_x wu$ is active in the $(Pr_{\langle y \rangle})$ inference.

(i) Our assumption implies that the propagation path $\pi(v, z)$ does not depend on the relational atom $R_x wu$, but rather, only depends on the relational atoms $\mathcal{R}, R_s wu$. Therefore, we may take the desired propagation path $\pi'(v, z)$ in $PG(\Lambda')$ such that $s_{\pi'}(v, z) \in L_S(y)$ to be the propagation path $\pi(v, z)$, thus showing that the two rules are permutable.

(ii) For the second case, suppose that the relational atom $R_x wu$ is active in the $(Pr_{\langle y \rangle})$ inference. To prove the claim we need to show the existence of a propagation path $\pi'(v, z)$ in $PG(\Lambda')$ such that $s_{\pi'}(v, z) \in L_S(y)$. We construct such a propagation path by simultaneously performing the following replacements on the propagation path $\pi(v, z)$:

- Replace each occurrence of w, x, u in $\pi(u, z)$ with $w, x_0, u_1, \ldots, u_n, x_n, u$, and
- replace each occurrence of $u, \overline{x}, w$ in $\pi(u, z)$ with $u, \overline{x}_n, u_n, \ldots, u_1, \overline{x}_0, w$.

We call the resulting path, after the above replacements on $\pi(v, z)$ have been performed, $\pi'(v, z)$. First, observe that the former propagation paths correspond to the edges $(w, u_1, x_0), \ldots, (u_n, u, x_n) \in PG(\Lambda')$ obtained from the relational atoms $R_s wu$ that occur in Λ' (by Def. 58). Second, observe that the latter propagation paths correspond to the edges $(u, u_n, \overline{x}_n), \ldots, (u_1, w, \overline{x}_0) \in PG(\Lambda')$ which are also obtained from the relational atoms $R_s wu$ that occur in Λ' (by Def. 58). Therefore, since the only difference between $PG(\Lambda)$ and $PG(\Lambda')$ is that the former contains the edges (w, u, x) and $(u, w, \overline{x})$ corresponding the relational atom $R_x wu$, whereas the latter does not, and because $\pi'(v, z)$ omits use of paths w, x, u and $u, \overline{x}, w$ (corresponding to edges (w, u, x) and $(u, w, \overline{x})$, resp.), it follows that $\pi'(v, z)$ occurs in $PG(\Lambda')$.

To finish the proof, we need to show that $s_{\pi'}(v, z) \in L_S(y)$. By assumption, we know that $s_{\pi}(v, z) \in L_S(y)$, which by Def. 9, implies that $y \twoheadrightarrow^*_S s_{\pi}(v, z)$. Moreover, since (p^x_s)

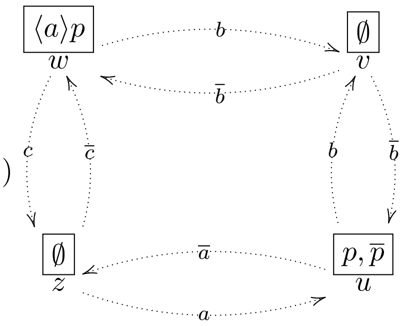
is in our calculus, as explained in Fig. 3.1, there must exist a corresponding production rule $x \longrightarrow s \in S$. Also, since S is a CFCST system, we know that $\overline{x} \longrightarrow \overline{s} \in S$ by the closure condition $\mathbf{(C_2)}$. Now, by the assumption of case (ii), we know that either x or $\overline{x}$ occurs in $s_{\pi}(v,z)$. Observe that applying $x \longrightarrow s$ to each occurrence of x in $s_{\pi}(v,z)$ (obtained from the relational atom $R_x wu$), and $\overline{x} \longrightarrow \overline{s}$ to each occurrence of $\overline{x}$ in $s_{\pi}(v,z)$ (obtained from the relational atom $R_x wu$), yields the string $s_{\pi'}(v,z)$. Hence, $y \twoheadrightarrow^{*}_{S} s_{\pi'}(v,z)$, which implies that $s_{\pi'}(v,z) \in L_S(y)$ by Def. 9. □

Example 5. *Let* $\sum := \{a, \overline{a}, b, \overline{b}, c, \overline{c}\}$ *and* $s := \overline{b}ca \in \sum^{*}$. *We give an example of permuting the structural rule* $(p^{\overline{b}}_{s})$ *above a propagation rule* $(Pr_{\langle a \rangle})$. *Let our CFCST system be:*

$$S := \{a \longrightarrow b\overline{b}, \overline{b} \longrightarrow \overline{b}ca, \overline{a} \longrightarrow a\overline{b}, b \longrightarrow \overline{ac}b\}.$$

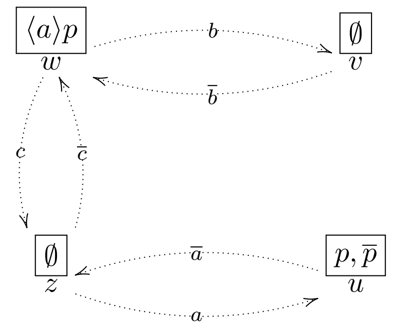
In the derivation below, we assume that $(Pr_{\langle a \rangle})$ *is applied due to the occurrence of the propagation path* $\pi(w,u) = w, b, v, \overline{b}, u$ *based on the relational atoms* $R_{\overline{b}}vw, R_{\overline{b}}vu$, *which further implies that* $s_{\pi}(w,u) = b\overline{b} \in L_S(a)$. *This holds since the production rule* $a \longrightarrow b\overline{b}$ *occurs in* S. *Concerning the* $(p^{\overline{b}}_{s})$ *rule, we assume that it is obtained from the production rule* $\overline{b} \longrightarrow \overline{b}ca \in S$, *and that the relational atom* $R_{\overline{b}}vu$ *is deleted from premise to conclusion due to the existence of the relational atoms* $R_{\overline{b}}vw, R_cwz, R_azu$. *The propagation graph of the top sequent of the derivation is shown below right:*

$$\dfrac{\dfrac{\dfrac{}{R_{\overline{b}}vw, R_cwz, R_azu, R_{\overline{b}}vu \Rightarrow w : \langle a \rangle p, u : p, u : \neg p}(id)}{R_{\overline{b}}vw, R_cwz, R_azu, R_{\overline{b}}vu \Rightarrow w : \langle a \rangle p, u : \neg p}(Pr_{\langle a \rangle})}{R_{\overline{b}}vw, R_cwz, R_azu \Rightarrow w : \langle a \rangle p, u : \neg p}(p^{\overline{b}}_{s})$$



Let us now apply the $(p^{\overline{b}}_{s})$ *to the top sequent of the derivation above in attempt to permute the two rules. If we apply* $(p^{\overline{b}}_{s})$ *to the top sequent, then we obtain the derivation shown below left. Below right, we provide the propagation graph of the conclusion of the derivation:*

$$\dfrac{\dfrac{}{R_{\overline{b}}vw, R_cwz, R_azu, R_buv \Rightarrow w : \langle a \rangle p, u : p, u : \neg p}(id)}{R_{\overline{b}}vw, R_cwz, R_azu \Rightarrow w : \langle a \rangle p, u : p, u : \neg p}(p^{\overline{b}}_{s})$$



In order to apply the propagation rule $(Pr_{\langle a \rangle})$*, we need to show that there exists a propagation path* $\pi'(w,u)$ *in the propagation graph of the conclusion such that* $s_{\pi'}(w,u) \in L_{S}(a)$*. To show such a propagation path exists, we apply the procedure explained in the proof of Lem. 26. In the initial derivation, we had the propagation path* $\pi(w,u) = w, b, v, \overline{b}, u$*. Applying our procedure, we replace each occurrence of* u, b, v *with* $u, \overline{a}, z, \overline{c}, w, b, v$ *and each occurrence of* $v, \overline{b}, u$ *with* $v, \overline{b}, w, c, z, a, u$*. Since no path* u, b, v *exists in* $\pi(w,u)$*, only the latter replacement of* $v, \overline{b}, u$ *is applicable, and yields the propagation path* $\pi'(w,u) := w, b, v, \overline{b}, w, c, z, a, u$*, which indeed exists in the propagation graph of the conclusion. Moreover, the derivation*

$$a \longrightarrow b\overline{b} \longrightarrow b\overline{b}ca$$

in S *confirms that* $s_{\pi'}(w,u) = b\overline{b}ca \in L_{S}(a)$*. Therefore, we may apply* $(Pr_{\langle a \rangle})$ *to the conclusion above to derive the desired sequent and successfully permute the two rules.*

We are now in a position to prove two critical lemmata, which respectively claim that each structural rule (p^{x}_{s}) and (C^{x}_{1}) is eliminable in $\mathsf{G3Km}(S) + \{(Pr_{\langle x \rangle}) \mid x \in \sum\}$. Note that when proving (p^{x}_{s}) elimination, we need not consider permuting the rule above (C^{x}_{1}), or vice-versa. As explained previously, given a derivation in $\mathsf{G3Km}(S) + \{(Pr_{\langle x \rangle}) \mid x \in \sum\}$, we may consider topmost occurrences of either a (p^{x}_{s}) or (C^{x}_{1}) rule, and successively delete such topmost occurrences via the elimination algorithms provided in the following two lemmata. Processing the input derivation in this way has the consequence that a (p^{x}_{s}) rule will never be permuted above a (C^{x}_{1}) rule, or vice-versa, since topmost occurrences are eliminated first.

Up until this point, we have primarily focused on securing the elimination of each (p^{x}_{s}) rule. Coincidentally, although our analysis of (p^{x}_{s}) elimination gave rise to the definition of a propagation rule, it just so happens that propagation rules are sufficient for proving the elimination of each (C^{x}_{1}) rule as well. This is primarily based on the fact that propagation rules rely on the notion of a propagation graph, which ‘builds in’ the information of each (C^{x}_{1}) rule; the specifics of how each (C^{x}_{1}) rule is ‘built into’ the definition of a propagation graph will be explained after the following lemma:

Lemma 27. *Let* S *be a CFCST system. For each* $x \longrightarrow s \in S$*, the rule* (p^{x}_{s}) *is eliminable in the calculus* $\mathsf{G3Km}(S) + \{(Pr_{\langle x \rangle}) \mid x \in \sum\} - \{(C^{x}_{1}) \mid x \in \sum\}$*.*

Proof. We prove the result by induction on the height of the given derivation, and assume that our derivation contains one application of the (p^{x}_{s}) rule, which is the last inference of the derivation. The general result follows by successively applying the procedure given below to delete topmost occurrences of (p^{x}_{s}) rules until the derivation is free of such inferences.

Base case. The base case follows from the fact that any application of a (p^{x}_{s}) rule to an instance of (id) yields another instance of (id).

Inductive step. By Lem. 25, we need not consider the $(\langle x \rangle)$ case. Also, with the exception of the $(Pr_{\langle x \rangle})$ case, all cases are resolved by invoking IH followed by an application of the corresponding rule. The $(Pr_{\langle x \rangle})$ case follows from Lem. 26 above. □

Having confirmed that each (p^{x}_{s}) rule is eliminable, we now focus on showing (C^{x}_{1}) elimination. As mentioned above, the information inherent in the (C^{x}_{1}) rule, namely, that any relations R_{x} and $R_{\overline{x}}$ in a $\sum$-model are converses of one another, is incorporated into the definition of a propagation graph. If one observes the definition of a propagation graph $PG(\Lambda) = (V, E)$ of a labelled sequent Λ (Def. 58), they will notice that the edges (w, u, x) and $(u, w, \overline{x})$ are added to the set of edges E regardless of if the relational atom $R_{x}wu$ or the relational atom $R_{\overline{x}}uw$ occurs in Λ. Therefore, since the (C^{x}_{1}) rule transforms a labelled sequent of the form $\mathcal{R}, R_{x}wu, R_{\overline{x}}uw \Rightarrow \Gamma$ into a labelled sequent of the form $\mathcal{R}, R_{x}wu \Rightarrow \Gamma$, the edges (w, u, x) and $(u, w, \overline{x})$ in the propagation graph of the former will still be present in the propagation graph of the latter because of the continued presence of $R_{x}wu$. To put it another way, the propagation graph of the premise of a (C^{x}_{1}) rule is identical to the propagation graph of the conclusion of the rule. As shown below, the fact that propagation graphs are invariant under applications of (C^{x}_{1}) allows for the permutation of (C^{x}_{1}) above an propagation rule. We will make use of this insight to prove (C^{x}_{1}) elimination below:

Lemma 28. *Let S be a CFCST system with alphabet $\sum$. For each $x \in \sum$, the (C^{x}_{1}) rule is eliminable in $\mathsf{G3Km}(S) + \{(Pr_{\langle x \rangle}) \mid x \in \sum\} - \{(p^{x}_{s}) \mid x \longrightarrow s \in S\}$.*

Proof. We prove the result by induction on the height of the given derivation, and assume that only one instance of (C^{x}_{1}) occurs as the last inference of the derivation. The general result follows by successively deleting topmost occurrences of (C^{x}_{1}) in a derivation, until it is free of such inferences.

Base case. The base case follows from the fact that any application of (C^{x}_{1}) to an instance of (id) yields another instance of (id).

Inductive step. By Lem. 25, we need not consider the permutation of (C^{x}_{1}) above $(\langle x \rangle)$. With the exception of the $(Pr_{\langle y \rangle})$ rule, all cases trivially follow by invoking IH followed by the corresponding rule. Let us now consider the case of permuting (C^{x}_{1}) above a $(Pr_{\langle y \rangle})$ inference.

Suppose that our derivation ends with a propagation inference $(Pr_{\langle y \rangle})$ followed by a (C^{x}_{1}) inference as shown below left. Also, let $\Lambda := \mathcal{R}, R_{x}wu, R_{\overline{x}}uw \Rightarrow v : \langle y \rangle \phi, z : \phi, \Gamma$ with $PG(\Lambda) = (V, E)$, and take note that $(w, u, x), (u, w, \overline{x}) \in E$. Due to the side condition of the propagation rule, we know there exists a propagation path $\pi(v, z)$ in $PG(\Lambda)$ such that $s_{\pi}(v, z) \in L_{S}(y)$. If we invoke IH and delete $R_{\overline{x}}uw$ from Λ, then since the relational atom $R_{x}wu$ is still present, the edges $(w, u, x), (u, w, \overline{x})$ are still in E. In fact, the propagation graph of the conclusion $\mathcal{R}, R_{x}wu \Rightarrow v : \langle y \rangle \phi, z : \phi, \Gamma$ is identical to $PG(\Lambda)$. Therefore, the side condition for $(Pr_{\langle y \rangle})$ is still satisfied, meaning the rule may be applied.

$$\dfrac{\dfrac{\mathcal{R}, R_x wu, R_{\overline{x}} uw \Rightarrow v : \langle y \rangle \phi, z : \phi, \Gamma}{\mathcal{R}, R_x wu, R_{\overline{x}} uw \Rightarrow v : \langle y \rangle \phi, \Gamma}\,(Pr_{\langle y \rangle})}{\mathcal{R}, R_x wu \Rightarrow v : \langle x \rangle \phi, \Gamma}\,(C^x_1) \quad \rightsquigarrow$$

$$\dfrac{\begin{array}{c} \mathcal{R}, R_x wu, R_{\overline{x}} uw \Rightarrow v : \langle y \rangle \phi, z : \phi, \Gamma \\ \hdashline \mathcal{R}, R_x wu \Rightarrow v : \langle y \rangle \phi, z : \phi, \Gamma \end{array}\;\text{IH}}{\mathcal{R}, R_x wu \Rightarrow v : \langle y \rangle \phi, \Gamma}\,(Pr_{\langle y \rangle})$$

□

The two previous lemmata establish the following theorem:

Theorem 17. *Every derivation in* G3Km(S) *can be algorithmically transformed into a derivation in* Km(S)L*.*

Proof. The result follows from Lem. 27 and 28 by successively eliminating topmost occurrences of each (p^x_s) and (C^x_1) rule. □

By considering the elimination of structural rules, we have found that each calculus G3Km(S) can be transformed into a calculus Km(S)L that contains fewer rules overall and which are purely *formula driven rules*, i.e. each rule's (bottom-up) applicability is solely determined by the occurrence of a complex logical formula (from $\mathcal{L}_{\mathsf{Km}}(\sum)$) in the conclusion. Each calculus G3Km(S) is different in this regard, since structural rules such as (p^x_s) and (C^x_1) are not formula driven in the above sense, as bottom-up applications of the rules introduce relational atoms independent of which labelled formulae occur. In addition, an analysis of the proofs of Lem. 27 and 28 shows that the proofs of Km(S)L are linearly compressed versions of the proofs from G3Km(S). Moreover, as the following theorem establishes, our refined calculi only require the use of labelled tree sequents in derivations (Def. 44), whereas each G3Km(S) calculus requires the use of more complex structures (Thm. 8); this is significant as it shows that through structural rule elimination we have essentially obtained nested calculi for grammar logics (which we discuss in more detail in Sect. 4.1.2 below). In these respects, each Km(S)L calculus is simpler than the G3Km(S) calculus it was derived from; of course, the trade-off is that each $(\langle x \rangle)$ rule has absorbed each (p^x_s) and (C^x_1) rule, thus generating the $(Pr_{\langle x \rangle})$ rule that possesses a more complex functionality.

Theorem 18. *Let S be a CFCST system. Every derivation in* Km(S)L *of a labelled formula $w : \phi$ is a labelled tree derivation with the fixed root property.*[4]

Proof. To see that the derivation of $w : \phi$ is a labelled tree derivation with the fixed root property (see Def. 45) observe that applying inference rules of Km(S)L bottom-up either preserve relational structure or add forward relational structure (with the $([x])$ rule), thus constructing a tree emanating from w. □

[4] The *fixed root property* is defined in Def. 45 on p. 53.

Although we have derived $\mathsf{Km}(S)\mathsf{L}$ from $\mathsf{G3Km}(S)$, we have not confirmed whether such calculi possess similar proof-theoretic properties. One course of action could be to prove the (hp-)admissibility of the rules $\mathrm{Str}(\mathsf{Km}(S))$ as well as the (hp-)invertibility of all rules in $\mathsf{Km}(S)\mathsf{L}$ (as we did for each $\mathsf{G3Km}(S)$ calculus). However, we opt for another course of action, and show that every derivation in a $\mathsf{Km}(S)\mathsf{L}$ calculus can be transformed into a derivation in the corresponding $\mathsf{G3Km}(S)$ calculus. This backward transformation kills two birds with one stone: First, it demonstrates the deductive correspondence between $\mathsf{G3Km}(S)$ and $\mathsf{Km}(S)\mathsf{L}$, and second, it establishes that $\mathsf{Km}(S)\mathsf{L}$ inherits proof-theoretic properties from $\mathsf{G3Km}(S)$. Regarding the second point, leveraging both transformations (Thm. 17 and 19), lets us establish the soundness and completeness of $\mathsf{Km}(S)\mathsf{L}$ (see Cor. 3 below), and shows that the refined calculi possess similar admissibility and invertiblity properties (detailed in Cor. 1 and 2 below) that are useful in applications; e.g. the extraction of interpolants (see Sect. 6.2).

Before showing that any derivation in a calculus $\mathsf{Km}(S)\mathsf{L}$ can be transformed into a derivation in $\mathsf{G3Km}(S)$, we first need to show a lemma relating derivations in a CFCST system to what is provable in $\mathsf{G3Km}(S)$.

Lemma 29. *Let $\Lambda := \mathcal{R} \Rightarrow \Gamma$. Suppose $\mathcal{R}, R_x wu \Rightarrow \Gamma$ is derivable in* $\mathsf{G3Km}(S)$*, and that $\pi(w,u)$ is a propagation path (potentially empty) in $PG(\Lambda)$ such that $x \twoheadrightarrow^*_S s_\pi(w,u)$. Then, $\Lambda = \mathcal{R} \Rightarrow \Gamma$ is derivable in* $\mathsf{G3Km}(S)$.

Proof. We prove the result by induction on the length (Def. 9) of the derivation $x \twoheadrightarrow^*_S s_\pi(w,u)$.

Base case. For the base case, we consider (i) a derivation of length 0, meaning that $x \twoheadrightarrow^*_S x$, and (ii) a derivation of length 1, meaning that $x \longrightarrow s_\pi(w,u) \in S$. Both cases are respectively shown below, where $\mathcal{R} := \mathcal{R}', R_x wu$ in the first case, and $\mathcal{R} := \mathcal{R}', R_s wu$ with $s := s_\pi(w,u)$ in the second case.

$$\dfrac{\Pi_1}{\mathcal{R}', R_x wu, R_x wu \Rightarrow \Gamma} \quad \rightsquigarrow \quad \dfrac{\dfrac{\Pi_1}{\mathcal{R}', R_x wu, R_x wu \Rightarrow \Gamma}}{\mathcal{R}', R_x wu \Rightarrow \Gamma}\ (ctr_{\mathcal{R}})$$

$$\dfrac{\Pi_2}{\mathcal{R}', R_s wu, R_x wu \Rightarrow \Gamma} \quad \rightsquigarrow \quad \dfrac{\dfrac{\Pi_2}{\mathcal{R}', R_s wu, R_x wu \Rightarrow \Gamma}}{\mathcal{R}', R_s wu \Rightarrow \Gamma}\ (p^s_x)$$

Inductive step. Let $\Lambda := \mathcal{R} \Rightarrow \Gamma$ and assume that we have a proof Π of $\mathcal{R}, R_x wu \Rightarrow \Gamma$. Suppose our derivation $x \twoheadrightarrow^*_S s_\pi(w,u)$ is of length $n+1$, that is, it consists of a derivation $x \twoheadrightarrow^*_S t$ of length n followed by a one-step derivation $t \twoheadrightarrow_S s_\pi(w,u)$. Hence, there exist strings $r_0, r_1 \in \sum^*$ and a production rule $y \longrightarrow r \in S$ such that $t = r_0 y r_1$ and $r_0 r r_1 = s_\pi(w,u)$. This implies the existence of a propagation path $\pi_{r_0}(w,v), \pi_r(v,z), \pi_{r_1}(z,u)$ in $PG(\Lambda)$, where $v, z \in \mathrm{Lab}(\Lambda)$. We give a proof below showing how to derive the desired result.

First, we apply hp-admissibility of (wk) (Lem. 10) to the proof Π of $\mathcal{R}, R_x wu \Rightarrow \Gamma$, introducing the relational atom $R_y vz$ and giving a proof of the labelled sequent $\Lambda' := \mathcal{R}, R_y vz, R_x wu \Rightarrow \Gamma$. Observe that the propagation path $\pi_{r_0}(w,v), \pi_y(v,z), \pi_{r_1}(z,u)$ exists in $PG(\Lambda')$. Also, by assumption, we know that $x \twoheadrightarrow^*_S t = r_0 y r_1$ with a derivation of length n. Thus, we may invoke IH to derive the sequent $\Lambda'' := \mathcal{R}, R_y vz \Rightarrow \Gamma$. Last, by our assumption that $y \longrightarrow r \in S$, and the fact that $\pi_r(v,z)$ occurs in $PG(\Lambda'')$, we know that there exists a structural rule (p^y_s) in $\mathsf{G3Km}(S)$ that is applicable to Λ''. Applying this rule lets us derive the desired conclusion.

$$
\dfrac{\dfrac{\dfrac{\dfrac{\Pi}{\mathcal{R}, R_x wu \Rightarrow \Gamma}}{\mathcal{R}, R_y vz, R_x wu \Rightarrow \Gamma}\;(wk)}{\mathcal{R}, R_y vz \Rightarrow \Gamma}\;\text{IH}}{\mathcal{R} \Rightarrow \Gamma}\;(p^y_s)
$$

□

Theorem 19. *Every derivation in* $\mathsf{Km}(S)\mathsf{L} + \mathrm{Str}(\mathsf{Km}(S))$ *can be algorithmically transformed into a derivation in* $\mathsf{G3Km}(S)$.

Proof. We prove the result by induction on the height of the given derivation.

Base case. Any instance of (id) in $\mathsf{Km}(S)\mathsf{L} + \mathrm{Str}(\mathsf{Km}(S))$ is an instance of (id) in $\mathsf{G3Km}(S)$.

Inductive step. As usual, we prove the inductive step by a case-distinction on the last rule applied in the given derivation. If the last rule applied is a rule in the set $\mathrm{Str}(\mathsf{Km}(S))$, then we invoke the (hp-)admissibility result of the corresponding rule (Lem. 10 and Thm. 6). With the exception of $(Pr_{\langle x\rangle})$, all other cases are handled by invoking IH and then applying the same rule. We show how to resolve the case when the last inference is $(Pr_{\langle x\rangle})$.

$$
\dfrac{\mathcal{R} \Rightarrow w : \langle x\rangle\phi, u : \phi, \Gamma}{\mathcal{R} \Rightarrow w : \langle x\rangle\phi, \Gamma}\;(Pr_{\langle x\rangle}) \quad \rightsquigarrow \quad \dfrac{\dfrac{\dfrac{\dfrac{}{\mathcal{R} \Rightarrow w : \langle x\rangle\phi, u : \phi, \Gamma}\;\text{IH}}{\mathcal{R}, R_x wu \Rightarrow w : \langle x\rangle\phi, u : \phi, \Gamma}\;(wk)}{\mathcal{R}, R_x wu \Rightarrow w : \langle x\rangle\phi, \Gamma}\;(\langle x\rangle)}{\mathcal{R} \Rightarrow w : \langle x\rangle\phi, \Gamma}\;\text{Lem. 29}
$$

By the side condition imposed on the $(Pr_{\langle x\rangle})$ rule, we know that there exists a propagation path $\pi(w,u)$ in $PG(\mathcal{R} \Rightarrow w : \langle x\rangle\phi, u : \phi, \Gamma)$ such that $s_\pi(w,u) \in L_S(x)$. The applicability of Lem. 29 follows from the fact that $PG(\mathcal{R} \Rightarrow w : \langle x\rangle\phi, u : \phi, \Gamma) = PG(\mathcal{R} \Rightarrow w : \langle x\rangle\phi, \Gamma)$, implying that $\pi(w,u)$ exists in $PG(\mathcal{R} \Rightarrow w : \langle x\rangle\phi, \Gamma)$. □

We may now utilize our translations between $\mathsf{G3Km}(S)$ and $\mathsf{Km}(S)\mathsf{L}$ (Thm. 17 and 19) to show that our refined calculi inherit proof-theoretic properties from their parent labelled calculi.

Corollary 1. *The rules in* $\mathrm{Str}(\mathsf{Km}(S))$ *are admissible in* $\mathsf{Km}(S)\mathsf{L}$.

Proof. Follows from the (hp-)admissibility of the rules $\mathrm{Str}(\mathsf{Km}(S))$ in $\mathsf{G3Km}(S)$ (Lem. 10 and Thm. 6), as well as Thm. 17 and 19. □

Corollary 2. *All rules of* $\mathsf{Km}(S)\mathsf{L}$ *are invertible.*

Proof. The invertibility of $(Pr_{\langle x \rangle})$ follows from the admissibility of (wk) (Cor. 1 above). The invertibility of $(\vee_r)$, $(\wedge_r)$, and $([x])$ is argued as follows: Let Λ be an instance of a conclusion of $(\vee_r)$, $(\wedge_r)$, or $([x])$, and assume Λ is derivable in $\mathsf{Km}(S)\mathsf{L}$. Then, by Thm. 19 we know that Λ is derivable in $\mathsf{G3Km}(S)$, and so, by the (hp-)invertibility of the rules in $\mathsf{G3Km}(S)$ (Lem. 11), we know that the corresponding premise(s) Λ' (and Λ'') of the rule is (are) derivable. Hence, by Thm. 17 Λ' (and Λ'') is (are) derivable in $\mathsf{Km}(S)\mathsf{L}$. □

Corollary 3 (Soundness and Completeness of $\mathsf{Km}(S)\mathsf{L}$)**.** *Let S be a CFCST system.*

(i) If $\vdash_{\mathsf{Km}(S)\mathsf{L}} \Lambda$*, then* $\models_{\mathsf{Km}(S)} \Lambda$.

(ii) If $\vdash_{\mathsf{Km}(S)} \phi$*, then* $\vdash_{\mathsf{Km}(S)\mathsf{L}} \epsilon \Rightarrow w : \phi$.

Proof. Follows from the soundness of $\mathsf{G3Km}(S)$ (Thm. 5), the completeness of $\mathsf{G3Km}(S)$ (Thm. 7), as well as Thm. 17 and 19. □

4.1.2 Relationship to Nested Sequent Formalism

In this section, we prove that each refined labelled calculus $\mathsf{Km}(S)\mathsf{L}$ is a notational variant of a deep nested calculus $\mathsf{DKm}(S)$ (displayed in Fig. 4.3 below). It should be noted that the deep nested calculi presented here are slight reformulations of the nested calculi for grammar logics introduced in [TIG12]. Therefore, due to the minute differences between the nested calculi introduced here and those given in [TIG12], we assign our calculi the same names and refer to each calculus (for a CFCST system S) as $\mathsf{DKm}(S)$.

The observation that refinement yields variants of known nested systems (as will be shown for first-order intuitionistic logics as well), suggests that the discovery of such systems was—in some sense—not accidental, and that refinement is a natural procedure connecting the relational semantics of a logic to a nested calculus for the logic. In order to establish the equivalence between refined labelled and nested systems, we will first introduce the $\mathsf{DKm}(S)$ nested calculi below, and then define translations permitting us to switch between labelled and nested notation. Ultimately, these translations will be used to construct (relatively simple) algorithms that allow for proofs to be translated between the refined labelled and nested calculi.

Let us now define the building blocks for our $\mathsf{DKm}(S)$ calculi: *nested sequents* for grammar logics [TIG12].[5]

Definition 61 (Nested Sequents for Grammar Logics [TIG12])**.** Nested sequents for grammar logics *are syntactic objects X defined via the following grammar in BNF:*

$$X ::= \epsilon \mid \phi \mid X, X \mid (x)\{X\}$$

where $x \in \sum$ and $\phi \in \mathcal{L}_{\mathsf{Km}}(\sum)$.

We use X, Y, Z, ... (possibly annotated) to denote nested sequents, and we let ϵ represent the *empty string* (i.e. empty nested sequent). As with labelled sequents, ϵ is an identity element for comma, and comma associates and commutes. We use the notation $X[Y]$ (and $X[Y][Z]$) to mean that Y (Y and Z, resp.) occurs (occur, resp.) at some depth in the nestings of X. For example, if the nested sequent X is $p, (a)\{q, (b)\{\neg p, r\}\}$, then $X[p]$, $X[q, (b)\{\neg p, r\}]$, $X[q][\neg p, r]$, etc. are all valid representations of X.

The Value of Switching Notation.

Before we proceed, it is interesting to wonder if switching from labelled to nested notation possesses any utility in its own right, that is, does the option of converting notation offer any advantage? Two reasons come to mind which justify a positive answer. First, switching from labelled to nested notation ensures that the language of our calculi enjoys a degree of parsimony, that is, the language does not allow for syntactic structures that go too far beyond what is needed for completeness. Second, switching to a more restrictive notation has practical value as it provides a priori knowledge about the structure of proofs; e.g. one can be certain that all sequents within a given proof encode trees prior to observing any proof within the associated system. To ground this second point further, we consider a concrete case: in [LvB19] the first proof-search procedures for STIT logics were introduced using refined labelled calculi, and to ensure the correctness of each proof-search algorithm, it was necessary to prove that all labelled sequents generated throughout the course of proof-search were labelled forest sequents (see Def. 56). Had the notation of the refined labelled calculi been translated to a *rigid* notation only allowing for sequents of a forest shape (e.g. like how nested sequents only allow for sequents of a tree shape) prior to the writing of the proof-search procedure, then confirming that all sequents generated throughout the course of proof-search were of a forest shape would have been rendered unnecessary. Due to the rigidity of the notation used, which—by its very definition—only allows for certain structures, it would have been known in advance that all sequents possessed the desired shape. Therefore, *rigidifying* notation (i.e. switching from a more liberal notation like that of labelled sequents to a more restrictive notation like that of nested sequents) appears to be of value.

[5] Note that the notation used for nested sequents in [TIG12] was obtained from the notation put forth in Kashima's paper [Kas94] and uses brackets of the form { and } as opposed to brackets of the form [and] for nesting data. This is the reverse of the notation used by many writers who opt to nest data with the latter brackets; e.g. [Brü09, Bul92, Fit14, Str13].

Furthermore, translating in the opposite direction—from nested to labelled notation—appears to be worthwhile as well. For instance, in [LTGC20] a purely syntactic method of proving interpolation for logics via their nested sequent calculi was introduced. In that paper, the nested sequent calculi were translated into labelled notation as the notation was easier to work with and allowed for simpler definitions. Hence, translations provide one with the freedom to switch proof-theoretic formalisms when one is better suited for a particular task.

In order to translate between labelled and nested sequents, we introduce *sequent graphs* for nested sequents, which are objects intermediary in the translation.

Definition 62 (Sequent Graph of a Nested Sequent for $\mathsf{Km}(S)$)**.** *We define the* sequent graph of a nested sequent X *inductively on the depth of the nestings of* X *as shown below. Notice that we make use of sequences of natural numbers*

$$\sigma \in \bigcup_{n \in \mathbb{N}} \mathbb{N}^n$$

to represent our vertices similar to the prefixes used in prefixed tableaux (cf. [Fit72]). We represent sequences as $n_1.n_2.\ldots.n_{k-1}.n_k$ *and use* σ *(possible annotated) to denote them. Our inductive definition of* $G(X) := G_0(X)$ *is as follows:*

- *If* $X = \phi_1, \ldots, \phi_k$*, then* $G_\sigma(X) := (V_\sigma, E_\sigma, L_\sigma)$*, where*

$$(i) \quad V_\sigma := \{\sigma\} \qquad (ii) \quad E_\sigma := \emptyset \qquad (iii) \quad L_\sigma := \{(\sigma, \{\phi_1, \ldots, \phi_k\})\}$$

- *Let* $X := \phi_1, \ldots, \phi_k, (x_1)\{Z_1\}, \ldots, (x_n)\{Z_n\}$ *and suppose that each* $G_{\sigma.i}(Z_i) = (V_{\sigma.i}, E_{\sigma.i}, L_{\sigma.i})$ *(with* $i \in \{1, \ldots n\}$ *and* $n \in \mathbb{N}$*) is already defined. We define* $G_\sigma(X) := (V_\sigma, E_\sigma, L_\sigma)$ *as shown below:*

  - $V_\sigma := \{\sigma\} \cup \bigcup_{1 \leq i \leq n} V_{\sigma.i}$
  - $E_\sigma := \{(\sigma, \sigma.i, x_i) \mid 1 \leq i \leq n\} \cup \bigcup_{1 \leq i \leq n} E_{\sigma.i}$
  - $L_\sigma := \{(\sigma, \{\phi_1, \ldots, \phi_k\})\} \cup \bigcup_{1 \leq i \leq n} L_{\sigma.i}$

Note that when $k = 0$*, the multiset* $\phi_1, \ldots, \phi_k$ *is taken to be the empty string* ϵ*. Also, we will often use* w*,* u*,* v*,* $\ldots$ *to represent vertices as opposed to sequences of natural numbers.*

Even though we may use the notation $X[Y]$ or $X[Y][Z]$ to denote a nested sequent X, we also allow for the notation $X[Y]_w$ and $X[Y]_w[Z]_u$ to denote that Y is associated with the vertex w (meaning that $L(w) = Y$ in $G(X) = (V, E, L)$), and to denote that Y and Z are associated with the vertices w and u (meaning that $L(w) = Y$ and $L(u) = Z$ in $G(X) = (V, E, L)$), respectively.

Since each calculus $\mathsf{DKm}(S)$ is a notational variant of a refined labelled calculus $\mathsf{Km}(S)\mathsf{L}$, each nested calculus likewise employs a set of propagation rules. Therefore, it is necessary to define propagation graphs for nested sequents. Still, the definition of a propagation path and the string of a propagation path, along with their converses, remains the same in the nested setting and are as given in Def. 59. This follows from the fact that propagation graphs for nested sequents are identical to propagation graphs for labelled (tree) sequents.

Definition 63 (Propagation Graphs for $\mathsf{DKm}(S)$)**.** *Let X be a nested sequent for grammar logics with sequent graph $G(X) = (V, E, L)$. We define the* propagation graph $PG(X) = (\mathrm{V}', \mathrm{E}')$ *to be the directed graph such that*

- $\mathrm{V}' := V$*;*
- $\mathrm{E}' := \{(w, u, x), (u, w, \overline{x}) \mid (w, u, x) \in E \textit{ or } (u, w, \overline{x}) \in E\}$*.*

We will often write $w \in PG(X)$ to mean $w \in \mathrm{V}'$, and $(w, u, x) \in PG(X)$ to mean $(w, u, x) \in \mathrm{E}'$, for $x \in \sum$.

Each nested calculus $\mathsf{DKm}(S)$ for a given CFCST system S is displayed in Fig. 4.3. The derivability relation for each calculus is defined as follows:

Definition 64. *We write $\vdash_{\mathsf{DKm}(S)} X$ to indicate that the nested sequent X is derivable in $\mathsf{DKm}(S)$.*

With the exception of each propagation rule $(Pr_{\langle x \rangle})$ (for $x \in \sum$), all rules are identical to those given in [TIG12]. The distinguishing feature between the propagation rules presented here and the propagation rules presented in [TIG12] concerns the side condition imposed. The side condition of a $(Pr_{\langle x \rangle})$ rule states that $\exists \pi (s_{\pi}(w, u) \in L_S(x))$ must hold if the rule is applied, i.e. there must exist a propagation path $\pi(w, u)$ in the propagation graph of the premise (or, equivalently, the conclusion) such that $s_{\pi}(w, u) \in L_S(x)$. By contrast, the side condition of a propagation rule from [TIG12] checks if a certain string is both accepted by a certain automaton and within a certain language [TIG12, p. 523]. Nevertheless, due to the equivalence between formal grammars and automata, it turns out that both side conditions are equivalent.

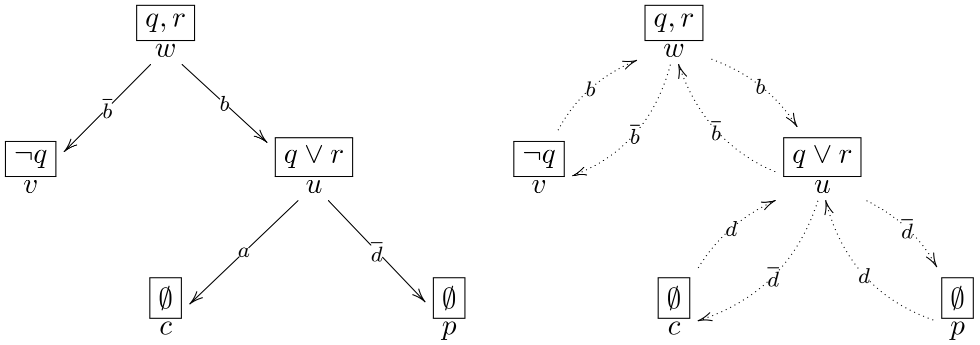
Nested sequents encode a tree structure by definition, with the nestings representing edges occurring within the tree. In this regard, nested sequents differ from labelled sequents, which encode edges via their relational atoms. In addition, the nested notation is more rigid than the labelled notation, since it only allows for trees to be represented, whereas labelled sequents can represent arbitrary graphs. To provide more intuition regarding the tree that corresponds to a nested sequent, we give an example of a nested sequent and the tree it encodes below; an example of the propagation graph of the nested sequent is also provided.

$$\frac{}{X[p, \neg p, Y]}\ (id) \qquad \frac{X[\phi, \psi, Y]}{X[\phi \lor \psi, Y]}\ (\lor_r) \qquad \frac{X[\phi, Y] \qquad X[\psi, Y]}{X[\phi \land \psi, Y]}\ (\land_r)$$

$$\frac{X[Y, (x)\{\phi\}]}{X[Y, [x]\phi]}\ ([x]) \qquad \frac{X[\langle x \rangle \phi, Y]_w[\phi, Z]_u}{X[\langle x \rangle \phi, Y]_w[Z]_u}\ (Pr_{\langle x \rangle})^{\dagger}$$

Figure 4.3: The nested calculus $\mathsf{DKm}(S)$, which is a variant of a calculus from [TIG12] of the same name. The calculus has a $([x])$ and $(Pr_{\langle x \rangle})$ rule for each $x \in \sum$. The side condition † states that the rule is applicable only if $\exists \pi (s_\pi(w, u) \in L_S(x))$.

Example 6. *The nested sequent* $X := q, r, (\overline{b})\{\neg q\}, (b)\{q \lor r, (a)\{\epsilon\}, (\overline{d})\{\epsilon\}\}$ *encodes the tree* $G(X)$ *shown below left and has the propagation graph* $PG(X)$ *shown below right:*



Perhaps the reader will find the above sequent graph $G(X)$ familiar, and in fact it is. The graph is identical to the sequent graph $G(\Lambda)$ of the labelled tree sequent Λ given in Ex. 1 of Sect. 3.2. This demonstrates a natural correspondence between labelled tree sequents and nested sequents. We now define the translation $\mathfrak{N}$ mapping labelled tree sequents to nested sequents, and the translation $\mathfrak{L}$ mapping nested sequents to labelled tree sequents—both of which rely on the notion of a sequent graph. After defining these two translations, we introduce an algorithm that converts derivations in $\mathsf{Km}(S)\mathsf{L}$ to derivations in $\mathsf{DKm}(S)$ (Thm. 20), and an algorithm that translates derivations in $\mathsf{DKm}(S)$ to derivations in $\mathsf{Km}(S)\mathsf{L}$ (Thm. 21).

Definition 65 (Downward Closure)**.** *Let X be a nested sequent and Λ be a labelled tree sequent. Also, let $G(X) := (V, E, L)$ and $G(\Lambda) := (V, E, L)$ with $w \in V$. (NB. The ambiguity in notation is of no consequence and simplifies presentation.) We define the* w-downward closure *$G_w(X) = (V', E', L')$ and $G_w(\Lambda) = (V', E', L')$ to be the smallest induced subgraph (Def. 31) of $G(X)$ and $G(\Lambda)$, respectively, such that $w \in V'$ and*

- *if $v \in V'$ and $(v, u) \in E$, then $u \in V'$;*

- $E' = E \restriction V'$
- $L' = L \restriction V'$

Remark 3. *If X is a nested sequent and Λ is a labelled tree sequent with w the root of the sequent graph $G(X)$ and $G(\Lambda)$, then $G_w(X) = G(X)$ and $G_w(\Lambda) = G(\Lambda)$, respectively.*

Definition 66 (The Translation $\mathfrak{N}$)**.** *Let $\Lambda := \mathcal{R} \Rightarrow \Gamma$ be the a labelled tree sequent with $G(\Lambda) = (V, E, L)$ and $w \in V$ the root. We define the translation $\mathfrak{N}(\Lambda) := \mathfrak{N}(G_w(\Lambda))$ inductively as follows:*

- *If $G(\Lambda) = (V, E, L)$ with $V = \{w\}$, $E = \emptyset$, and $L = \{(w, \{\phi_1, \ldots, \phi_k\})\}$, then*
$$\mathfrak{N}(G_w(\Lambda)) := \phi_1, \ldots, \phi_k$$
- *If $G(\Lambda) = (V, E, L)$ with $w, u_1, \ldots, u_n \in V$, $(w, u_i, x_i) \in E$ (for $i \in \{1, \ldots, n\}$), then*
$$\mathfrak{N}(G_w(\Lambda)) := L(w), (x_1)\{\mathfrak{N}(G_{u_1}(\Lambda))\}, \ldots, (x_n)\{\mathfrak{N}(G_{u_n}(\Lambda))\}$$

We demonstrate the operation of the translation function $\mathfrak{N}$ with an example:

Example 7. *We show how to translate the labelled tree sequent*

$$\Lambda := R_a wu, R_a uv, R_{\overline{b}} wz \Rightarrow w : q, w : \langle b \rangle q, u : p \wedge \neg p, v : q, v : r, z : c, z : p$$

into a nested sequent via the computation below:

$$\begin{aligned}
\mathfrak{N}(\Lambda) \; &:= \mathfrak{N}(G_w(\Lambda)) \\
&= q, \langle b \rangle q, (a)\{\mathfrak{N}(G_u(\Lambda))\}, (\overline{b})\{\mathfrak{N}(G_z(\Lambda))\} \\
&= q, \langle b \rangle q, (a)\{p \wedge \neg p, (a)\{\mathfrak{N}(G_v(\Lambda))\}\}, (\overline{b})\{c, p\} \\
&= q, \langle b \rangle q, (a)\{p \wedge \neg p, (a)\{q, r\}\}, (\overline{b})\{c, p\}
\end{aligned}$$

In order to define the reverse translation from nested to labelled notation, it is helpful to introduce the notion of a *sequent composition*. Since this concept will be used in the sequel, we define the notion for labelled sequents in general.

Definition 67 (Labelled Sequent Composition)**.** *Given two labelled sequents $\Lambda_1 := \mathcal{R}_1, \Gamma_1 \Rightarrow \Delta_1$ and $\Lambda_2 := \mathcal{R}_2, \Gamma_2 \Rightarrow \Delta_2$, we define the* sequent composition *as follows:*

$$\Lambda_1 \circ \Lambda_2 := \mathcal{R}_1, \mathcal{R}_2, \Gamma_1, \Gamma_2 \Rightarrow \Delta_1, \Delta_2$$

Definition 68 (The Translation $\mathfrak{L}$)**.** *Let X be the a nested sequent. We define the translation $\mathfrak{L}(X) := \mathfrak{L}(G_0(X))$ inductively as follows:*

- *If* $G_{\sigma}(X) = (V, E, L)$ *with* $V = \{\sigma\}$, $E = \emptyset$, *and* $L = \{(\sigma, Y)\}$, *then*

$$\mathfrak{L}(G_{\sigma}(X)) := \epsilon \Rightarrow w_{\sigma} : Y$$

- *If* $G_{\sigma}(X) = (V, E, L)$ *with* $\sigma, \sigma.1, \ldots, \sigma.n \in V$, $(\sigma, \sigma.1, x_1), \ldots, (\sigma, \sigma.n, x_n) \in E$, *and* $(\sigma, Y) \in L$, *then*

$$\mathfrak{L}(G_{\sigma}(X)) := (R_{x_1} w_{\sigma} w_{\sigma.1}, \ldots, R_{x_m} w_{\sigma} w_{\sigma.n} \Rightarrow w_{\sigma} : Y) \circ \mathfrak{L}(G_{\sigma.1}(X)) \circ \cdots \circ \mathfrak{L}(G_{\sigma.n}(X))$$

In practice, we will often use labels such as w, u, v, $\ldots$ *as opposed to labels indexed with sequences of natural numbers for simplicity.*

Similar to before, we demonstrate how the translation function works with an example:

Example 8. *Below, we show how to translate the nested sequent*

$$X := q, \langle b \rangle q, (a)\{p \wedge \neg p, (a)\{q, r\}\}, (\bar{b})\{c, p\}$$

into a labelled (tree) sequent:

$$\begin{aligned}
\mathfrak{L}(X) \quad & := \mathfrak{L}(G_0(X)) \\
& = (R_a wu, R_{\bar{b}} wz \Rightarrow w : q, w : \langle b \rangle q) \circ \mathfrak{L}(G_u(X)) \circ \mathfrak{L}(G_z(X)) \\
& = (R_a wu, R_{\bar{b}} wz \Rightarrow w : q, w : \langle b \rangle q) \circ (R_a uv \Rightarrow u : p \wedge \neg p) \circ \\
& \quad \mathfrak{L}(G_v(X)) \circ (\epsilon \Rightarrow z : c, z : p) \\
& = (R_a wu, R_{\bar{b}} wz \Rightarrow w : q, w : \langle b \rangle q) \circ (R_a uv \Rightarrow u : p \wedge \neg p) \circ \\
& \quad (\epsilon \Rightarrow v : q, v : r) \circ (\epsilon \Rightarrow z : c, z : p) \\
& = R_a wu, R_a wz, R_{\bar{b}} uv \Rightarrow w : q, w : \langle b \rangle q, u : p \wedge \neg p, v : q, v : r, z : c, z : p
\end{aligned}$$

Based on the above definitions, it is not hard to see that the sequent graph of a labelled tree sequent Λ is isomorphic to the sequent graph of $\mathfrak{N}(\Lambda)$ or that the sequent graph of a nested sequent X is isomorphic to the sequent graph of $\mathfrak{L}(X)$. (NB. The relation of being *isomorphic* is defined in Def. 36 in Ch. 3.) Thus, we have the following lemma:

Lemma 30. *Let* Λ *be a labelled tree sequent and* X *be a nested sequent (for grammar logics). Then,*

(i) $G(\Lambda) \cong G(\mathfrak{N}(\Lambda))$

(ii) $G(X) \cong G(\mathfrak{L}(X))$

Let us now leverage our translation functions to show that for any CFCST system S, the calculi $\mathsf{Km}(S)\mathsf{L}$ and $\mathsf{DKm}(S)$ are notational variants of one another. Translating proofs between refined labelled and nested systems for grammar logics is straightforward and merely consists of changing notation.

Theorem 20. *Let S be a CFCST system. Every derivation in $\mathsf{Km}(S)\mathsf{L}$ is algorithmically translatable to a derivation in $\mathsf{DKm}(S)$.*

Proof. We prove the result by induction on the height of the given derivation.

Base case. The base case is straightforward as the principal formulae $w : p, w : \neg p$ will be included together in the output nested sequent, thus ensuring that it is initial. We let $Y := \Gamma \restriction w$ in the nested sequent X below.

$$\dfrac{}{\mathcal{R} \Rightarrow w : p, w : \neg p, \Gamma}\,(id) \quad \rightsquigarrow \quad \begin{array}{cl} \dfrac{}{\mathfrak{N}(\mathcal{R} \Rightarrow w : p, w : \neg p, \Gamma)}\,(id) & \\ \hdashline X[p, \neg p, Y]_w & = \end{array}$$

Inductive step. Each case of the inductive step is straightforward and is resolved as shown below. In all cases, we let $Y := \Gamma \restriction w$, and in the $(Pr_{\langle x \rangle})$ case we let $Z := \Gamma \restriction u$. Also, note that the side condition in the $(Pr_{\langle x \rangle})$ case holds due to Lem. 30 above, which ensures that $PG(\mathcal{R} \Rightarrow w : \langle x \rangle \phi, u : \phi, \Gamma) \cong PG(X[\langle x \rangle \phi, Y]_w[\phi, Z]_u)$.

$$\dfrac{\mathcal{R} \Rightarrow w : \phi, w : \psi, \Gamma}{\mathcal{R} \Rightarrow w : \phi \vee \psi, \Gamma}\,(\vee_r) \quad \rightsquigarrow \quad \begin{array}{cl} \mathfrak{N}(\mathcal{R} \Rightarrow w : \phi, w : \psi, \Gamma) & \\ \hdashline \dfrac{X[\phi, \psi, Y]_w}{X[\phi \vee \psi, Y]_w}\,(\vee_r) & = \\ \hdashline \mathfrak{N}(\mathcal{R} \Rightarrow w : \phi \vee \psi, \Gamma) & = \end{array}$$

$$\dfrac{\mathcal{R} \Rightarrow w : \phi, \Gamma \qquad \mathcal{R} \Rightarrow w : \psi, \Gamma}{\mathcal{R} \Rightarrow w : \phi \wedge \psi, \Gamma}\,(\wedge_r) \quad \rightsquigarrow \quad \begin{array}{cl} \dfrac{\begin{array}{cl} \mathfrak{N}(\mathcal{R} \Rightarrow w : \phi, \Gamma) & \\ \hdashline X[\phi, Y]_w & = \end{array} \qquad \begin{array}{cl} \mathfrak{N}(\mathcal{R} \Rightarrow w : \psi, \Gamma) & \\ \hdashline X[\psi, Y]_w & = \end{array}}{X[\phi \wedge \psi, Y]_w}\,(\wedge_r) & \\ \hdashline \mathfrak{N}(\mathcal{R} \Rightarrow w : \phi \wedge \psi, \Gamma) & = \end{array}$$

$$\dfrac{\mathcal{R}, R_x uw \Rightarrow u : \phi, \Gamma}{\mathcal{R} \Rightarrow w : [x]\phi, \Gamma}\,([x]) \quad \rightsquigarrow \quad \begin{array}{cl} \mathfrak{N}(\mathcal{R}, R_x uw \Rightarrow u : \phi, \Gamma) & \\ \hdashline \dfrac{X[Y, (x)\{\phi\}]_w}{X[Y, [x]\phi]_w}\,([x]) & = \\ \hdashline \mathfrak{N}(\mathcal{R} \Rightarrow w : [x]\phi, \Gamma) & = \end{array}$$

$$\dfrac{\mathcal{R} \Rightarrow w : \langle x \rangle \phi, u : \phi, \Gamma}{\mathcal{R} \Rightarrow w : \langle x \rangle \phi, \Gamma}\,(Pr_{\langle x \rangle}) \quad \rightsquigarrow \quad \begin{array}{cl} \mathfrak{N}(\mathcal{R} \Rightarrow w : \langle x \rangle \phi, u : \phi, \Gamma) & \\ \hdashline \dfrac{X[\langle x \rangle \phi, Y]_w[\phi, Z]_u}{X[\langle x \rangle \phi, Y]_w[Z]_u}\,(Pr_{\langle x \rangle}) & = \\ \hdashline \mathfrak{N}(\mathcal{R} \Rightarrow w : \langle x \rangle \phi, u : \phi, \Gamma) & = \end{array}$$

□

Theorem 21. *Let S be a CFCST system. Every derivation in $\mathsf{DKm}(S)$ is algorithmically translatable to a derivation in $\mathsf{Km}(S)\mathsf{L}$.*

Proof. The theorem is proven by induction on the height of the given derivation and is similar to the proof of the previous theorem (Thm. 20). In essence, the proof puts forth a simple algorithm showing that one can switch the notation of a given derivation from nested to labelled through applications of the $\mathfrak{L}$ function. □

An example is provided below that demonstrates the correspondence between proofs in a refined labelled calculus $\mathsf{Km}(S)\mathsf{L}$ and nested calculus $\mathsf{DKm}(S)$.

Example 9. *The derivation on the left translates via $\mathfrak{N}$ to the derivation on the right, and the derivation on the right translates via $\mathfrak{L}$ to the derivation on the left (up to a change of labels).*

$$\dfrac{\dfrac{\dfrac{\dfrac{}{R_x wu \Rightarrow w : \neg p, w : p, u : \langle \overline{x} \rangle p}(id)}{R_x wu \Rightarrow w : \neg p, u : \langle \overline{x} \rangle p}(Pr_{\langle x \rangle})}{\Rightarrow w : \neg p, w : [x]\langle \overline{x} \rangle p}([x])}{\Rightarrow w : \neg p \vee [x]\langle \overline{x} \rangle p}(\vee_r) \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{\neg p, p, (x)\{\langle \overline{x} \rangle p\}}(id)}{\neg p, (x)\{\langle \overline{x} \rangle p\}}(Pr_{\langle x \rangle})}{\neg p, [x]\langle \overline{x} \rangle p}([x])}{\neg p \vee [x]\langle \overline{x} \rangle p}(\vee_r)$$

4.1.3 A Note on the Relationship between Refined Labelled and Display Calculi

Before concluding this section, we mention an interesting property that holds for each $\mathsf{Km}(S)\mathsf{L}$ calculus, and is largely reminiscent of the *display/residuation* rules employed in *display/shallow nested* calculi [Bel82, GPT11, Kas94, LIEG17, TIG12, Wan02].[6] Display calculi were introduced by Belnap in [Bel82], and their one-sided variants were introduced by Kashima in [Kas94]. Such calculi generalize Gentzen's sequent calculus formalism by extending the formalism with additional structural connectives that go beyond the comma utilized in Gentzen-style sequents. Moreover, a significant feature of the display calculus/shallow-nested sequent calculus formalism is that it admits a general cut-elimination theorem stating that any calculus built within the formalism, which also satisfies eight easily verifiable syntactic conditions, allows for cut-elimination [Bel82]. This property has proven the formalism useful in uniformly providing cut-free calculi for large classes of logics, independent of the logical connectives or semantics employed.

The paper [TIG12] not only introduced (slight reformulations of) the $\mathsf{DKm}(S)$ calculi presented above, but also introduced shallow nested calculi SKm(S) for context-free grammar logics with converse. Such calculi make use of the same nested sequents, but restrict applications of logical rules (*viz.* $(\vee_r)$, $(\wedge_r)$, $([x])$) to the root, or top-level, of a nested sequent. Furthermore, propagation rules are replaced by two sets of rules: rules that introduce $\langle x \rangle$ formulae, much like the $(\langle x \rangle)$ rules, and structural rules, much like the (p_s^x) rules. We do not introduce such rules here since they are unneeded in our analysis of $\mathsf{Km}(S)\mathsf{L}$. The interested reader is referred to [TIG12, p. 521] for a formal introduction

[6] The shallow nested calculi of [GPT11, Kas94, TIG12] can be seen as one-sided versions of display calculi (cf. [Bel82, Wan02]).

of shallow nested calculi SKm(S) for context-free grammar logics with converse. The essential characteristic of shallow nested calculi however, is the incorporation of the following residuation rule in each calculus:

$$\frac{X, (x)\{Y\}}{(\overline{x})\{X\}, Y}\ (r)$$

The residuation rule (r) allows for a nesting to be 'flipped' (much like turning the page of a book) from one structure Y within the nested sequent to another structure X, so long as the index is changed from x to $\overline{x}$ in the process. Keeping this in mind, we will prove the following proposition concerning $\mathsf{Km}(S)\mathsf{L}$, which verifies the validity of 'flipping' relational atoms, and afterwards, will briefly discuss the relevance of this property with respect to shallow nested calculi and 'display-style reasoning'.

Proposition 2. *Let S be a CFCST system. Every proof of a sequent $\mathcal{R}, R_x wu \Rightarrow \Gamma$ in $\mathsf{Km}(S)\mathsf{L}$ is algorithmically transformable to a proof of $\mathcal{R}, R_{\overline{x}} uw \Rightarrow \Gamma$ with the same height or less, and vice-versa.*

Proof. We prove the result by induction on the height of the given derivation.

Base case. The base case is resolved as shown below:

$$\frac{}{\mathcal{R}, R_x wu \Rightarrow v : p, v : \neg p, \Gamma}\ (id) \quad \rightsquigarrow \quad \frac{}{\mathcal{R}, R_{\overline{x}} uw \Rightarrow v : p, v : \neg p, \Gamma}\ (id)$$

Inductive step. We show how to resolve the case when the last rule is a $(Pr_{\langle y \rangle})$ inference, as all other cases are resolved by applying IH and then the corresponding rule. We solve and explain the $(Pr_{\langle y \rangle})$ case below:

$$\frac{\mathcal{R}, R_x wu \Rightarrow v : \langle y \rangle \phi, z : \phi, \Gamma}{\mathcal{R}, R_x wu \Rightarrow v : \langle y \rangle \phi, \Gamma}\ (Pr_{\langle y \rangle}) \quad \rightsquigarrow \quad \frac{\mathcal{R}, R_{\overline{x}} wu \Rightarrow v : \langle y \rangle \phi, z : \phi, \Gamma}{\mathcal{R}, R_{\overline{x}} wu \Rightarrow v : \langle y \rangle \phi, \Gamma}\ (Pr_{\langle y \rangle})$$

Let $\Lambda := \mathcal{R}, R_x wu \Rightarrow v : \langle y \rangle \phi, z : \phi, \Gamma$ and $\Lambda' := \mathcal{R}, R_{\overline{x}} wu \Rightarrow v : \langle y \rangle \phi, z : \phi, \Gamma$. The side condition of the $(Pr_{\langle y \rangle})$ inference on the left implies that there exists a path $\pi(v, z)$ in $PG(\Lambda)$ such that $s_\pi(v, z) \in L_S(x)$. By Def. 58, $PG(\Lambda) = PG(\Lambda')$, implying that the path $\pi(v, z)$ exists in $PG(\Lambda')$. Therefore, the derivation on the left may be transformed into the derivation on the right. □

The above proposition is interesting in that it implies the hp-admissibility of a labelled version of the residuation rule (r), namely:

$$\frac{\mathcal{R}, R_x wu \Rightarrow \Gamma}{\mathcal{R}, R_{\overline{x}} uw \Rightarrow \Gamma}\ (l)$$

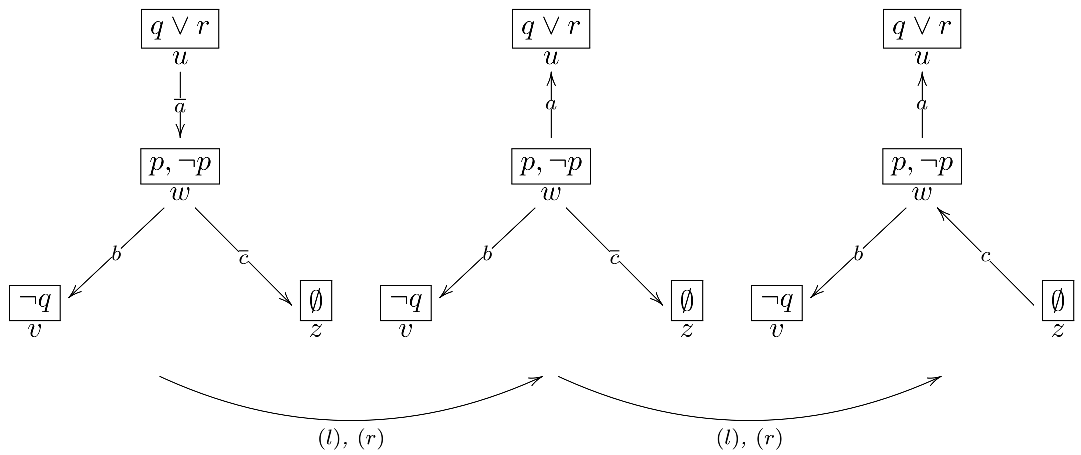
The hp-admissibility of the above rule suggests that our refined labelled calculi are capable of simulating the display-style reasoning inherent in the shallow nested calculi of [TIG12]. In fact, by the work done in this section, and the work in [TIG12], we know that for any CFCST system S, every derivation in $\mathsf{Km}(S)\mathsf{L}$ can be algorithmically transformed into a derivation in the shallow nested calculus SKm(S), and vice-versa. This conclusion is justified by the fact that each calculus $\mathsf{Km}(S)\mathsf{L}$ is a notational variant of a nested calculus $\mathsf{DKm}(S)$, and in [TIG12] it is shown that (a slight variant of) each nested calculus $\mathsf{DKm}(S)$ can be algorithmically transformed into a shallow nested calculus SKm(S).

What is more noteworthy perhaps, is the interplay between the more liberal notation of labelled sequents and the display-style reasoning engendered by the hp-admissible (l) rule. The residuation rule (r) transforms one nested sequent into another nested sequent, effectively transforming an object encoding a tree into a new object encoding a tree. By contrast, the flexibility of labelled notation allows for (l) to not only transform a labelled tree sequent into another labelled tree sequent (thus simulating the behavior of (r) in the nested setting), but applications of (l) allow for the production of intermediate labelled sequents outside the class of labelled tree sequents as well (which do not correspond to nested sequents via the translation $\mathfrak{N}$). That is to say, the labelled notation coupled with the labelled residuation rule (l) allows for a more unrestrained version of display-style reasoning. To make this point concrete, an example comparing applications of (l) to (r) is provided below.

Example 10. *In the first part of the example, we consider a derivation of a labelled tree sequent Λ_2 (defined below) that consists of two applications of the (l) rule to a labelled tree sequent Λ_1 (also defined below). This derivation translates (in its entirety) to a derivation of $\mathfrak{N}(\Lambda_2)$ from $\mathfrak{N}(\Lambda_1)$ consisting of two applications of the (r) rule. In addition, we display the sequent graphs of the labelled tree and nested sequents to illustrate how applications of (l) and (r) manipulate the underlying data structure of the associated sequents. In the second part of the example, we give a derivation of Λ_2 from Λ_1 that uses alternative instances of the (l) rule and does not translate via $\mathfrak{N}$ (in its entirety) to a derivation in the nested setting since it includes a labelled sequent Λ which is not a labelled tree sequent. This observation demonstrates the additional methods of proof offered in the labelled setting.*

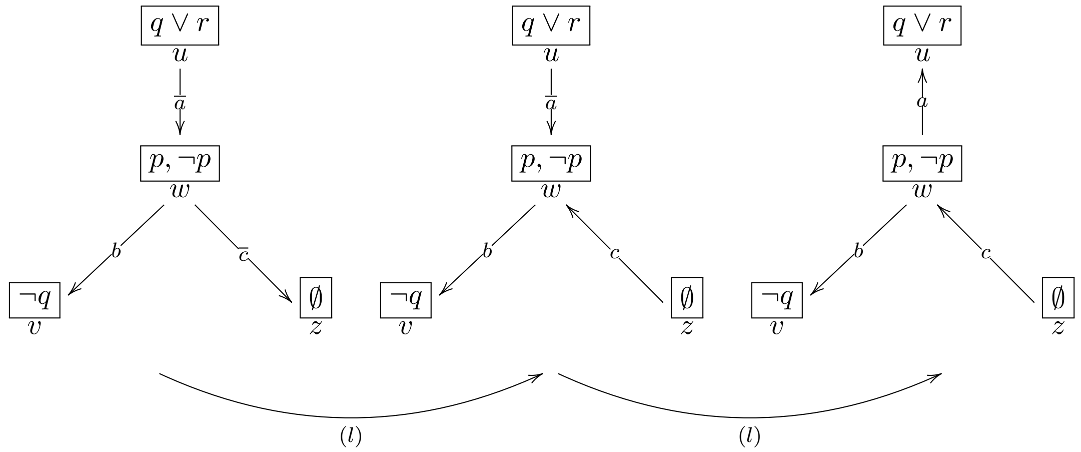
$$\begin{array}{rcl}
\Lambda_1 := & R_{\overline{a}}uw, R_b wv, R_{\overline{c}}wz \Rightarrow u : q \lor r, w : p, w : \neg p, z : \neg q & \\
\cline{2-2}
& R_{\overline{a}}uw, R_b wv, R_c zw \Rightarrow u : q \lor r, w : p, w : \neg p, z : \neg q & (l) \\
\cline{2-2}
\Lambda_2 := & R_{\overline{a}}wu, R_b wv, R_c zw \Rightarrow u : q \lor r, w : p, w : \neg p, z : \neg q & (l)
\end{array}$$

$$\begin{array}{cl}
\mathfrak{N}(R_{\overline{a}}uw, R_b wv, R_{\overline{c}}wz \Rightarrow u : q \lor r, w : p, w : \neg p, z : \neg q) & \\
\hdashline
q \lor r, (\overline{a})\{p, \neg p, (b)\{\epsilon\}, (\overline{c})\{\neg q\}\} & = \\
\cline{1-1}
(a)\{q \lor r\}, p, \neg p, (b)\{\epsilon\}, (\overline{c})\{\neg q\} & (r) \\
\cline{1-1}
(c)\{(a)\{q \lor r\}, p, \neg p, (b)\{\epsilon\}\}, \neg q & (r) \\
\hdashline
\mathfrak{N}(R_{\overline{a}}wu, R_b wv, R_c zw \Rightarrow u : q \lor r, w : p, w : \neg p, z : \neg q) & =
\end{array}$$



In the above derivation the root of Λ_1 (and $\mathfrak{N}(\Lambda_1)$) switches from u to w, and then from w to z. This behavior is not exhibited in the alternative proof of Λ_2 from Λ_1, which includes an intermediary labelled sequent Λ whose sequent graph is not a tree, but rather, is a polytree (Def. 33). This derivation and the corresponding sequent graphs are shown below:

$$
\begin{array}{ll}
\Lambda_1 := & R_{\bar{a}}uw, R_b wv, R_{\bar{c}}wz \Rightarrow u : q \vee r, w : p, w : \neg p, z : \neg q \\
\cline{2-2}
\Lambda \ := & R_a wu, R_b wv, R_{\bar{c}}wz \Rightarrow u : q \vee r, w : p, w : \neg p, z : \neg q \quad (l) \\
\cline{2-2}
\Lambda_2 := & R_a wu, R_b wv, R_c zw \Rightarrow u : q \vee r, w : p, w : \neg p, z : \neg q \quad (l)
\end{array}
$$



The above example demonstrates the added flexibility embedded within labelled notation, which not only explains why labelled systems can simulate other systems (e.g. nested and display) with relative ease [CLR18, CLRT20, GR12, Pim18, Res06], but perhaps explains why labelled systems possess stronger proof-theoretic properties in relation to other systems (which use a more rigid or restrictive notation). For example, certain linear nested sequent systems (e.g. [KL18, Lyo20c]) only allow for the *admissibility* of contractions as opposed to the *height-preserving admissibility* of contractions, which seems to be due to the fact that the manipulation of data must occur within a linear structure.

Last, the paper [CLRT20] discusses the relationship between labelled, refined labelled, nested, and shallow nested (i.e. display) calculi for a class of tense logics. The logics considered in that paper can be seen as a restricted class of grammar logics that use an alphabet of the form $\sum = \{a, \overline{a}\}$. The translations in [CLRT20] may be generalized to define mutual translations between (refined) labelled and shallow nested calculi for context-free grammar logics with converse.

4.2 Refining Labelled Calculi for Deontic STIT Logics

In this section, we will refine each deontic STIT calculus $\mathsf{G3DS}_n^k$ (see Fig. 3.5 in Sect. 3.4). Unlike in the previous section, we forgo a detailed analysis of structural rule elimination as such an analysis would proceed along the same lines—observe which permutations of structural rules in $\mathsf{G3DS}_n^k$ cannot be performed, and extract propagation rules that allow for the permutations to go through. Instead, we simply state which propagation rules were obtained from such an analysis. After defining such rules, we show that each calculus $\mathsf{G3DS}_n^k$ can be refined, yielding a deductively equivalent calculus $\mathsf{DS}_n^k\mathsf{L}$ that inherits the properties of $\mathsf{G3DS}_n^k$. Last, we show that derivations in $\mathsf{DS}_n^k\mathsf{L}$ need only use labelled forest or DAG sequents (see Def. 56), depending on the values of $n, k \in \mathbb{N}$, thus demonstrating that the refined calculi employ simpler structures in deriving theorems relative to the labelled calculi $\mathsf{G3DS}_n^k$.

One of the more interesting observations made in this section concerns the conditions under which refinement can be performed. It will be shown that the structural rules (ref_i), (euc_i), (D_2^i), and (D_3^i) (see Fig. 3.5 in Sect. 3.4) are eliminable from any derivation in a calculus $\mathsf{G3DS}_n^k$, given that the calculus is extended with certain conducive rules. However, we will not achieve complete structural rule elimination as we did with the grammar logic calculi since the rules (IOA) and (APC_i^k) will still be present in our refined labelled calculi, showing that refinement may still be performed in the presence of certain structural rules. In our situation, this holds true because both the (IOA) and (APC_i^k) rules lack active relational atoms in their conclusion, and so, any structural rule may be permuted above them. This observation suggests that other rules of such a shape would allow for refinement as well.

Due to the presence of (IOA) and (APC_i^k) in our refined labelled systems $\mathsf{DS}_n^k\mathsf{L}$ (shown in Fig. 4.4), such systems will not be notational variants of nested systems in general. Even so, if we unbound the number of choices available to an agent (i.e. $k = 0$), or limit ourselves to the use of a single agent (i.e. $n = 0$), then the refined labelled calculi $\mathsf{DS}_n^0\mathsf{L}$ and $\mathsf{DS}_0^k\mathsf{L}$ only require labelled DAG proofs and labelled forest proofs (Def. 56), respectively, showing that behind the scenes a reduction in structure is still present. The former class of deontic STIT calculi (*viz.* $\mathsf{DS}^{0}_{n}\mathsf{L}$) can be seen as close relatives of indexed nested sequent calculi [Fit15, MS17], which generalize the nested sequent notation to encode directed acyclic graphs, while the latter class of calculi (*viz.* $\mathsf{DS}^{k}_{0}\mathsf{L}$) can be viewed as proper nested calculi.

Another distinguishing feature of our refined calculi for deontic STIT logics is that we replace the use of CFCST systems in our propagation rules with the use of *undirected paths* (cf. [Lyo20b]). Intuitively, an undirected path within a labelled sequent is a sequence of relational atoms of the form $R_{[i]}vz$ connecting two labels w and u, but where we ignore the 'orientation' of the relational atoms. For example, the path $R_{[i]}wv, R_{[i]}vu$ moves forward from w to v and then forward from v to u, yielding in path from w to u, but $R_{[i]}vw, R_{[i]}vu$ gives an (undirected) path from w to u as well, since we first move backward from w to v, and then forward from v to u. We make use of such structures and forgo the use of our grammar theoretic machinery as it simplifies the presentation and application of our propagation rules. Still, such simplicity comes at a cost since we can no longer 'plug in' a new CFCST system to obtain a refined labelled calculus for a new logic (as was done in the previous section). Although this is a shortcoming in a sense, it is actually not a drawback in the current setting since deontic STIT logics were specifically designed to possess certain properties with the goal of modeling multi-agent normative reasoning. Hence, we do not desire to change the properties of our deontic STIT logics, as this would damage their correspondence with the underlying philosophical reasons for having imposed such properties in the first place. Therefore, we are justified in trading modularity for simplicity in the current setting.

Let us now define undirected paths, and afterward, we will discuss and define our propagation rules that make use of such objects.

Definition 69 (Undirected i-Path). *Let $w \sim_i u \in \{R_{[i]}wu, R_{[i]}uw\}$ and $\Lambda = \mathcal{R} \Rightarrow \Gamma$. An* undirected i-path *of relational atoms from a label w to u occurs in $\mathcal{R}$ (and therefore, in Λ), written $w \sim^{\mathcal{R}}_{i} u$, iff $w = u$, $w \sim_i u$, or there exist labels $v_j \in$* Lab(Λ) *(with $j \in \{1, \ldots, m\}$) such that $w \sim_i v_1, \ldots, v_m \sim_i u$ occurs in $\mathcal{R}$.*

Lemma 31. *Let $\Lambda := \mathcal{R} \Rightarrow \Gamma$ be a labelled sequent. The undirected i-path relation $\sim^{\mathcal{R}}_{i}$ is an equivalence relation over the set of labels* Lab(Λ)*.*

Proof. Follows from Def. 69 above. □

Our refined labelled calculi are displayed in Fig. 4.4 and employ three new rules: $(Pr_{\langle i \rangle})$, $(Pr^{1}_{\ominus_i})$, and $(Pr^{2}_{\ominus_i})$. The propagation rule $(Pr_{\langle i \rangle})$ is obtained from the $(\langle i \rangle)$, (ref_i), and (euc_i) rules, the $(Pr^{1}_{\ominus_i})$ rule is obtained from the $(\ominus_i)$, (ref_i), (euc_i), and (D^{i}_{3}) rules, and last, the $(Pr^{2}_{\ominus_i})$ rule is obtained from $(Pr^{1}_{\ominus_i})$ and (D^{i}_{2}). The $(Pr_{\langle i \rangle})$ and $(Pr^{1}_{\ominus_i})$ rules make use of undirected i-paths in their side conditions, which arise naturally when analyzing the permutation of (ref_i) and (euc_i) upwards in a $\mathsf{G3DS}^{k}_{n}$ derivation. Moreover, undirected i-paths can be seen as syntactic encodings of the $\mathbf{(S_1)}$ condition imposed on DS^{k}_{n}-frames (Def. 24), which forces the relations $\{R_{[i]} \mid i \in \text{Ag}\}$ to be equivalence classes,

$$\frac{}{\mathcal{R} \Rightarrow w : p, w : \neg p, \Gamma}\,(id) \qquad \frac{\mathcal{R} \Rightarrow w : \phi, \Gamma \qquad \mathcal{R} \Rightarrow w : \psi, \Gamma}{\mathcal{R} \Rightarrow w : \phi \land \psi, \Gamma}\,(\land_r)$$

$$\frac{\mathcal{R}, I_{\otimes_i} u \Rightarrow u : \phi, \Gamma}{\mathcal{R} \Rightarrow w : \otimes_i \phi, \Gamma}\,(\otimes_i)^{\dagger_1} \qquad \frac{\mathcal{R}, R_{[0]} w_0 u, ..., R_{[n]} w_n u \Rightarrow \Gamma}{\mathcal{R} \Rightarrow \Gamma}\,(IOA)^{\dagger_1}$$

$$\frac{\mathcal{R} \Rightarrow w : \phi, w : \psi, \Gamma}{\mathcal{R} \Rightarrow w : \phi \lor \psi, \Gamma}\,(\lor_r) \qquad \frac{\mathcal{R} \Rightarrow w : \Diamond\phi, u : \phi, \Gamma}{\mathcal{R} \Rightarrow w : \Diamond\phi, \Gamma}\,(\Diamond) \qquad \frac{\mathcal{R}, R_{[i]} wu \Rightarrow u : \phi, \Gamma}{\mathcal{R} \Rightarrow w : [i]\phi, \Gamma}\,([i])^{\dagger_1}$$

$$\frac{\mathcal{R} \Rightarrow u : \langle i \rangle \phi, v : \phi, \Gamma}{\mathcal{R} \Rightarrow u : \langle i \rangle \phi, \Gamma}\,(Pr_{\langle i \rangle})^{\dagger_2} \qquad \frac{\mathcal{R}, I_{\otimes_i} u \Rightarrow w : \ominus_i \phi, v : \phi, \Gamma}{\mathcal{R}, I_{\otimes_i} u \Rightarrow w : \ominus_i \phi, \Gamma}\,(Pr^1_{\ominus_i})^{\dagger_2}$$

$$\frac{\mathcal{R} \Rightarrow u : \phi, \Gamma}{\mathcal{R} \Rightarrow w : \Box\phi, \Gamma}\,(\Box)^{\dagger_1} \qquad \frac{\mathcal{R}, I_{\otimes_i} u \Rightarrow w : \ominus_i \phi, u : \phi, \Gamma}{\mathcal{R} \Rightarrow w : \ominus_i \phi, \Gamma}\,(Pr^2_{\ominus_i})^{\dagger_1}$$

$$\frac{\Big\{ \mathcal{R}, R_{[i]} w_m w_j \Rightarrow \Gamma \;\Big|\; 0 \leq m \leq k-1,\, m+1 \leq j \leq k \Big\}}{\mathcal{R} \Rightarrow \Gamma}\,(APC^k_i)$$

Figure 4.4: The calculi $\mathsf{DS}^k_n\mathsf{L}$ (with $|\mathrm{Ag}| = n + 1$ and $n, k \in \mathbb{N}$). The side condition $\dagger_1$ on $(\Box)$, $([i])$, $(\otimes_i)$, (IOA), and $(Pr^2_{\ominus_i})$ indicates that u is a eigenvariable, and the side condition $\dagger_2$ states that $u \sim^{\mathcal{R}}_i v$. Last, we have $([i])$, $(Pr_{\langle i \rangle})$, $(\otimes_i)$, $(Pr^1_{\ominus_i})$, $(Pr^2_{\ominus_i})$, and (APC^k_i) rules for each $i \in \mathrm{Ag}$. We stipulate that if $k = 0$, then (APC^k_i) is omitted from the calculus.

or (equivalently) to satisfy the properties of reflexivity and Euclideanity. Since we will repeatedly make reference to the $(Pr_{\langle i \rangle})$, $(Pr^1_{\ominus_i})$, and $(Pr^2_{\ominus_i})$ rules while refining our calculi, we collect them into a set defined as follows:

Definition 70 (The Set $\mathrm{Pr}(\mathsf{DS}^k_n)$)**.** *Let* $\mathrm{Ag} = \{0, \dots, n\} \subseteq \mathbb{N}$ *be our set of agents. We define our set of conducive rules as follows:*

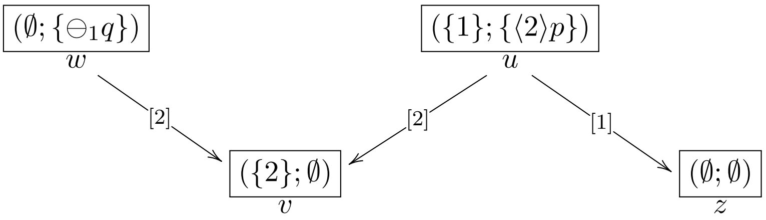
$$\mathrm{Pr}(\mathsf{DS}^k_n) := \{(Pr_{\langle i \rangle}), (Pr^1_{\ominus_i}), (Pr^2_{\ominus_i}) \mid i \in \mathrm{Ag}\}$$

As with the propagation rules we saw in the previous section, the side condition imposed on $(Pr_{\langle i \rangle})$ and $(Pr^1_{\ominus_i})$ can be determined whether we are applying the rules top-down or bottom-up. Regarding the $(Pr_{\langle i \rangle})$ rule, this follows from the fact that the side condition only requires data from $\mathcal{R}$ and the principal formula $u : \langle i \rangle \phi$ (see Fig. 4.4) to determine the rules (in)applicability, and all such data occurs in both the premise and conclusion—the same holds for the $(Pr^1_{\ominus_i})$ rule. This feature renders our propagation rules suitable for proof-search methods which apply inference rules in reverse. To provide more intuition regarding the functionality of $(Pr_{\langle i \rangle})$ and $(Pr^1_{\ominus_i})$ we provide an example below, showing how the rules may be applied bottom-up as would be done during proof-search.

Example 11. *We examine applications of propagation rules to the labelled sequent:*

$$\Lambda := I_{\otimes_1}u, I_{\otimes_2}v, R_{[2]}wv, R_{[2]}uv, R_{[1]}uz \Rightarrow w : \ominus_1 q, u : \langle 2\rangle p$$

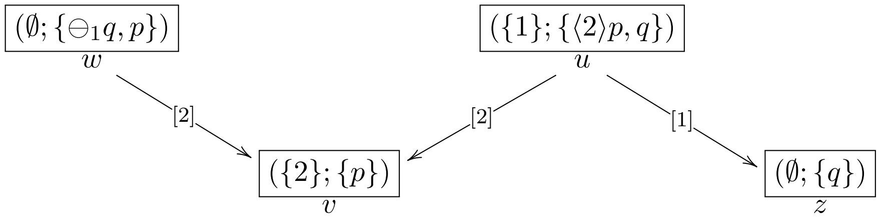
whose sequent graph $G(\Lambda)$ (Def. 55) is shown below:



Due to the undirected 2-path $R_{[2]}uv$, the $(Pr_{\langle 2\rangle})$ rule may be applied (bottom-up) to propagate the propositional variable p to v, effectively adding $v : p$ to our labelled sequent Λ. In addition, we may make use of the undirected 2 path $R_{[2]}uv, R_{[2]}wv$ from u to w to (bottom-up) apply $(Pr_{\langle 2\rangle})$ another time and propagate the propositional atom p to w, i.e. adding $w : p$ to our labelled sequent Λ. Due to the occurrence of $I_{\otimes_1}u$ and $w : \ominus_1 q$ in Λ, and because there is an undirected 1-path from u to itself, and an undirected 1-path $R_{[1]}uz$ from u to z, we may apply $(Pr^1_{\ominus_1})$ (bottom-up) two times to add a q at u and a q at z, effectively adding $u : q, z : q$ to Λ. After performing these four inferences, we obtain the following labelled sequent:

$$\Lambda' := I_{\otimes_1}u, I_{\otimes_2}v, R_{[2]}wv, R_{[2]}uv, R_{[1]}uz \Rightarrow w : \ominus_1 q, u : \langle 2\rangle p, w : p, v : p, u : q, z : q$$

whose sequent graph $G(\Lambda')$ is as follows:


$(\emptyset; \{\ominus_1 q, p\})$
w
$(\{1\}; \{\langle 2\rangle p, q\})$
u
[2]
[2]
[1]
$(\{2\}; \{p\})$
v
$(\emptyset; \{q\})$
z

We now move on to showing that (ref_i), (euc_i), (D^i_2), and (D^i_3) are eliminable in $\mathsf{G3DS}^k_n + \mathrm{Pr}(\mathsf{DS}^k_n)$. As in the last section, when proving a rule eliminable, we need not consider its permutation above the other structural rules we aim to eliminate; e.g. when considering (ref_i) elimination, we do not consider permuting the rule above (euc_i), (D^i_2), or (D^i_3). The reason being, if we consider any derivation in $\mathsf{G3DS}^k_n + \mathrm{Pr}(\mathsf{DS}^k_n)$, we can always successively eliminate topmost occurrences of (ref_i), (tra_i), (D^i_2), and (D^i_3), yielding a proof free of such inferences and without any of the rules ever being permuted above each other in the process. We first show that our new rules $(Pr_{\langle i\rangle})$ and $(Pr^1_{\ominus_i})$ subsume $(\langle i\rangle)$ and $(\ominus_i)$, respectively, as this will let us simplify proofs of subsequent results:

Lemma 32. *The ($\langle i \rangle$) and ($\ominus_i$) rules are instances of the ($Pr_{\langle i \rangle}$) and ($Pr^{1}_{\ominus_i}$) rules, respectively.*

Proof. Observe that the premise of the ($Pr_{\langle i \rangle}$) inference below contains the relational atom $R_{[i]}wu$. Hence, there exists an undirected i-path from w to u, meaning that the side condition of ($Pr_{\langle i \rangle}$) holds. Regarding the ($Pr^{1}_{\ominus_i}$) inference, let $\mathcal{R}' := \mathcal{R}, I_{\otimes_i}u$. The application of the rule is valid as well since $u \sim^{\mathcal{R}'}_{i} u$ by Def. 69, that is, the side condition holds.

$$\frac{\mathcal{R}, R_{[i]}wu \Rightarrow w : \langle i \rangle \phi, u : \phi, \Gamma}{\mathcal{R}, R_{[i]}wu \Rightarrow w : \langle i \rangle \phi, \Gamma}\ (Pr_{\langle i \rangle}) \qquad \frac{\mathcal{R}, I_{\otimes_i}u \Rightarrow w : \ominus_i \phi, u : \phi, \Gamma}{\mathcal{R}, I_{\otimes_i}u \Rightarrow w : \ominus_i \phi, \Gamma}\ (Pr^{1}_{\ominus_i})$$

□

Lemma 33. *For all $i \in \mathrm{Ag}$, the (ref_i) rule is eliminable in the calculus* $\mathsf{G3DS}^{k}_{n} + \mathrm{Pr}(\mathsf{DS}^{k}_{n}) - \{(euc_i), (D^{i}_{2}), (D^{i}_{3}) \mid i \in \mathrm{Ag}\}$.

Proof. We prove the claim by induction on the height of the given derivation.

Base case. Suppose that we have an instance of (id) followed by an instance of (ref_i) as shown below left. Then, the conclusion is an instance of (id) as shown below right. Also, note that w may or may not be identical to the label u.

$$\frac{\dfrac{}{\mathcal{R}, R_{[i]}uu \Rightarrow w : p, w : \neg p, \Gamma}\ (id)}{\mathcal{R}, \Rightarrow w : p, w : \neg p, \Gamma}\ (ref_i) \quad \rightsquigarrow \quad \frac{}{\mathcal{R} \Rightarrow w : p, w : \neg p, \Gamma}\ (id)$$

Inductive step. By Lem. 32, we need not consider permutations above the ($\langle i \rangle$) or ($\ominus_i$) rules. With the exception of ($Pr_{\langle i \rangle}$) and ($Pr^{1}_{\ominus_i}$), the rule (ref_i) freely permutes above all other rules in our calculus. Below, we consider the ($Pr_{\langle i \rangle}$) case as the ($Pr^{1}_{\ominus_i}$) case is similar, and suppose our derivation ends with the following:

$$\frac{\dfrac{\mathcal{R}, R_{[i]}vv \Rightarrow w : \langle i \rangle \phi, u : \phi, \Gamma}{\mathcal{R}, R_{[i]}vv \Rightarrow w : \langle i \rangle \phi, \Gamma}\ (Pr_{\langle i \rangle})}{\mathcal{R} \Rightarrow w : \langle i \rangle \phi, \Gamma}\ (ref_i)$$

Let $\mathcal{R}' := \mathcal{R}, R_{[i]}vv$. Since the ($Pr_{\langle i \rangle}$) rule was applied, we know that there exists an undirected i-path $w \sim^{\mathcal{R}'}_{i} u$. We have two cases to consider: either $R_{[i]}vv$ occurs in the undirected i-path, or it does not. In the latter case, we may freely permute the two rules, so we focus on the former case.

Suppose that $R_{[i]}vv$ occurs in $w \sim^{\mathcal{R}'}_{i} u$. Observe that by deleting each occurrence of $R_{[i]}vv$ in $w \sim^{\mathcal{R}'}_{i} u$, we obtain an alternative undirected i-path $w \sim^{\mathcal{R}}_{i} u$ that does not rely on $R_{[i]}vv$. Therefore, we may apply (ref_i) to the top sequent above, and use this path to satisfy the side condition, thus allowing for the two rules to be permuted. □

Lemma 34. *For all $i \in \mathrm{Ag}$, the (euc_i) rule is eliminable in the calculus* $\mathsf{G3DS}_n^k + \mathrm{Pr}(\mathsf{DS}_n^k) - \{(ref_i), (D_2^i), (D_3^i) \mid i \in \mathrm{Ag}\}$.

Proof. We prove the claim by induction on the height of the given derivation.

Base case. Any application of (euc_i) to an instance of (id) yields another instance of (id), which resolves the base case.

Inductive step. By Lem. 32, we need not consider permutations above the $(\langle i \rangle)$ or $(\ominus_i)$ rules. With the exception of $(Pr_{\langle i \rangle})$ and $(Pr^1_{\ominus_i})$, the (euc_i) rule freely permutes above all other rules in our calculus. We consider only the $(Pr^1_{\ominus_i})$ case below as the $(Pr_{\langle i \rangle})$ case is similar, and assume our derivation ends with the following:

$$
\cfrac{\cfrac{\mathcal{R}, R_{[i]}wu, R_{[i]}wv, R_{[i]}uv, I_{\otimes_i}y \Rightarrow z : \ominus_i \phi, x : \phi, \Gamma}{\mathcal{R}, R_{[i]}wu, R_{[i]}wv, R_{[i]}uv, I_{\otimes_i}y \Rightarrow z : \ominus_i \phi, \Gamma}\ (Pr^1_{\ominus_i})}{\mathcal{R}, R_{[i]}wu, R_{[i]}wv, I_{\otimes_i}y \Rightarrow z : \ominus_i \phi, \Gamma}\ (euc_i)
$$

Let $\mathcal{R}' := \mathcal{R}, R_{[i]}wu, R_{[i]}wv, R_{[i]}uv, I_{\otimes_i}y$. Since $(Pr^1_{\ominus_i})$ was applied, we know that the side condition holds, implying that there exists an undirected i-path $y \sim_i^{\mathcal{R}'} x$ in the premise of the inference. There are two possible cases, either $R_{[i]}uv$ occurs in the aforementioned i-path, or it does not. In the latter case, the two rules may be freely permuted, so we focus on the former case. We replace each occurrence of $R_{[i]}uv$ in $y \sim_i^{\mathcal{R}'} x$ with $R_{[i]}wu, R_{[i]}wv$. Observe that this yields an undirected i-path from y to x that does not rely on $R_{[i]}uv$. Using this path, we may apply (euc_i) first, and then $(Pr^1_{\ominus_i})$, since the side condition will be satisfied. □

Lemma 35. *For all $i \in \mathrm{Ag}$, the (D_2^i) rule is eliminable in the calculus* $\mathsf{G3DS}_n^k + \mathrm{Pr}(\mathsf{DS}_n^k) - \{(ref_i), (euc_i), (D_3^i) \mid i \in \mathrm{Ag}\}$.

Proof. We prove the result by induction on the height of the given derivation.

Base case. The base case follows from the fact that any application of (D_2^i) to an instance of (id) begets another instance of (id).

Inductive step. By Lem. 32, we need not consider permutations above the $(\langle i \rangle)$ or $(\ominus_i)$ rules. With the exception of the $(Pr^1_{\ominus_i})$ rule, (D_2^i) permutes above all other rules in our calculus. We show how to resolve the non-trivial case of permuting (D_2^i) above $(Pr^1_{\ominus_i})$ below:

$$
\cfrac{\cfrac{\mathcal{R}, I_{\otimes_i}u \Rightarrow w : \ominus_i \phi, u : \phi, \Gamma}{\mathcal{R}, I_{\otimes_i}u \Rightarrow w : \ominus_i \phi, \Gamma}\ (Pr^1_{\ominus_i})}{\mathcal{R} \Rightarrow w : \ominus_i \phi, \Gamma}\ (D_2^i)
\quad \rightsquigarrow \quad
\cfrac{\mathcal{R}, I_{\otimes_i}u \Rightarrow w : \ominus_i \phi, u : \phi, \Gamma}{\mathcal{R} \Rightarrow w : \ominus_i \phi, \Gamma}\ (Pr^2_{\ominus_i})
$$

□

Lemma 36. *For all $i \in \mathrm{Ag}$, the (D_3^i) rule is eliminable in the calculus* $\mathsf{G3DS}_n^k + \mathrm{Pr}(\mathsf{DS}_n^k) - \{(ref_i), (euc_i), (D_2^i) \mid i \in \mathrm{Ag}\}$.

Proof. We prove the result by induction on the height of the given derivation.

Base case. The base case is resolved by observing that any application of (D^i_3) to (id) gives another instance of (id).

Inductive step. By Lem. 32, we need not consider permutations above the $(\langle i \rangle)$ or $(\ominus_i)$ rules. With the exception of $(Pr^1_{\ominus_i})$, the (D^i_3) rule permutes above all other rules in our calculus. We show how to resolve the case of permuting (D^i_3) above $(Pr^1_{\ominus_i})$, and assume our derivation ends with the derivation shown top left.

$$\dfrac{\dfrac{\mathcal{R}, R_{[i]}wu, I_{\otimes_i}w, I_{\otimes_i}u \Rightarrow v : \ominus_i\phi, z : \phi, \Gamma}{\mathcal{R}, R_{[i]}wu, I_{\otimes_i}w, I_{\otimes_i}u \Rightarrow v : \ominus_i\phi, \Gamma}\,(Pr^1_{\ominus_i})}{\mathcal{R}, R_{[i]}wu, I_{\otimes_i}w \Rightarrow v : \ominus_i\phi, \Gamma}\,(D^i_3) \quad \rightsquigarrow$$

$$\dfrac{\dfrac{\mathcal{R}, R_{[i]}wu, I_{\otimes_i}w, I_{\otimes_i}u \Rightarrow v : \ominus_i\phi, z : \phi, \Gamma}{\mathcal{R}, R_{[i]}wu, I_{\otimes_i}w \Rightarrow v : \ominus_i\phi, z : \phi, \Gamma}\,(D^i_3)}{\mathcal{R}, R_{[i]}wu, I_{\otimes_i}w \Rightarrow v : \ominus_i\phi, \Gamma}\,(Pr^1_{\ominus_i})$$

Let $\mathcal{R}' := \mathcal{R}, R_{[i]}wu$. We have two cases to consider: either $I_{\otimes_i}u$ is active in the $(Pr^1_{\ominus_i})$ inference, or it is not. In the latter case, the two rules may be freely permuted. In the former case, if $I_{\otimes_i}u$ is active in the $(Pr^1_{\ominus_i})$ inference (in the above left derivation), then we know that there is an undirected i-path $u \sim^{\mathcal{R}'}_i z$ due to the side condition of $(Pr^1_{\ominus_i})$. Observe that the side condition of $(Pr^1_{\ominus_i})$ continues to hold after (D^i_3) is applied since $I_{\otimes_i}w$ occurs in the multiset of relational atoms (as can be seen in the lower right derivation above) and by combining $R_{[i]}wu$ with the undirected i-path $u \sim^{\mathcal{R}'}_i z$, we obtain an undirected i-path $w \sim^{\mathcal{R}'}_i z$. □

Theorem 22. *Every derivation in* $\mathsf{G3DS}^k_n$ *can be algorithmically transformed into a derivation in* $\mathsf{DS}^k_n\mathsf{L}$.

Proof. The result follows from Lem. 32 – 36 above. □

At this stage, we could opt to show that our refined labelled calculi possess proof-theoretic properties (e.g. hp-admissibility of (ctr_r), hp-invertibility of rules, or syntactic (cut) elimination) by carrying out proofs as we did for all $\mathsf{G3DS}^k_n$ calculi. We opt for a different method of proof however, and instead, establish a correspondence between each $\mathsf{DS}^k_n\mathsf{L}$ and $\mathsf{G3DS}^k_n$ calculus as this uncovers the relationship between derivations in the different settings. We have already shown that every derivation in a calculus $\mathsf{G3DS}^k_n$ can be transformed into a derivation in $\mathsf{DS}^k_n\mathsf{L}$ (Thm. 22 above), so to establish our correspondence, we show that proof transformations can be performed in the opposite direction as well. We will then leverage these proof transformations to show that each $\mathsf{DS}^k_n\mathsf{L}$ calculus inherits favorable proof-theoretic properties from its associated $\mathsf{G3DS}^k_n$ calculus (Cor. 4–6). We first prove a valuable lemma (Lem. 37 below), which states that if an undirected i-path $w \sim^{\mathcal{R}}_i u$ occurs in the relational atoms $\mathcal{R}$ of a derivable labelled sequent $\mathcal{R}, R_{[i]}wu \Rightarrow \Gamma$, then the relational atom $R_{[i]}wu$ is superfluous, and so, the labelled sequent $\mathcal{R} \Rightarrow \Gamma$ may be derived on its own. After proving this lemma, we provide an algorithm showing that proofs in $\mathsf{DS}^{k}_{n}\mathsf{L}$ can be transformed into proofs in $\mathsf{G3DS}^{k}_{n}$.

Lemma 37. *If $\mathcal{R}, R_{[i]}wu \Rightarrow \Gamma$ is derivable in $\mathsf{G3DS}^{k}_{n}$ and $w \sim^{\mathcal{R}}_{i} u$, then $\mathcal{R} \Rightarrow \Gamma$ is derivable in $\mathsf{G3DS}^{k}_{n}$.*

Proof. We prove the claim by induction on the length of the undirected i-path between w and u in $\mathcal{R}$, and let Π be the derivation of $\mathcal{R}, R_{[i]}wu \Rightarrow \Gamma$ in $\mathsf{G3DS}^{k}_{n}$.

Base case. If the undirected i-path between w and u is of length 0, then $w = u$, and the case is resolved as shown below left. If the undirected i-path between w and u is of length 1, then the case is resolved as shown below right and makes use of the hp-admissibility of $(ctr_{\mathcal{R}})$ (Lem. 23). Note that in the second case, $\mathcal{R} = \mathcal{R}', R_{[i]}wu$.

$$
\begin{array}{c}
\Pi \\ \hline
\mathcal{R}, R_{[i]}ww \Rightarrow \Gamma \\ \hline
\mathcal{R} \Rightarrow \Gamma
\end{array}
\ (ref_i)
\qquad
\begin{array}{c}
\Pi \\ \hline
\mathcal{R}, R_{[i]}wu \Rightarrow \Gamma \\ \hline
\mathcal{R}', R_{[i]}wu, R_{[i]}wu \Rightarrow \Gamma \\ \hdashline
\mathcal{R}', R_{[i]}wu \Rightarrow \Gamma \\ \cdots\cdots\cdots\cdots \\
\mathcal{R} \Rightarrow \Gamma
\end{array}
\begin{array}{l}
= \\ (ctr_{\mathcal{R}}) \\ =
\end{array}
$$

Inductive step. We assume that the undirected i-path $w \sim^{\mathcal{R}}_{i} u$ between w and u is of length $m + 1$. Therefore, there exists a v such that there is an undirected i-path $w \sim^{\mathcal{R}}_{i} v$ between w and v of length m and an undirected i-path between v and u of length 1. Since the length of the undirected i-path $w \sim^{\mathcal{R}}_{i} v$ between w and v is m, we may invoke IH (in both cases) below to delete the relational atom $R_{[i]}wv$. Additionally, since there is an undirected i-path between v and u of length 1, either $\mathcal{R} = \mathcal{R}', R_{[i]}uv$ or $\mathcal{R} = \mathcal{R}', R_{[i]}vu$ by Def. 69. The first case is resolved as shown below top, and the second case is resolved as shown below bottom. Also, we indicate active relational atoms in the premise of a (euc_i) inference with an asterisk $*$ to improve comprehensibility.

$$
\begin{array}{cl}
\Pi & \\ \hline
\mathcal{R}', R_{[i]}uv, R_{[i]}wu, \Rightarrow \Gamma & \\ \hdashline
\mathcal{R}', R_{[i]}uv, R_{[i]}wv, R_{[i]}ww, R^{*}_{[i]}uu, R_{[i]}vv, R_{[i]}vw, R_{[i]}vu, R^{*}_{[i]}uw, R^{*}_{[i]}wu \Rightarrow \Gamma & (wk) \\ \hline
\mathcal{R}', R_{[i]}uv, R_{[i]}wv, R_{[i]}ww, R_{[i]}uu, R_{[i]}vv, R^{*}_{[i]}vw, R^{*}_{[i]}vu, R^{*}_{[i]}uw \Rightarrow \Gamma & (euc_i) \\ \hline
\mathcal{R}', R^{*}_{[i]}uv, R_{[i]}wv, R_{[i]}ww, R^{*}_{[i]}uu, R_{[i]}vv, R_{[i]}vw, R^{*}_{[i]}vu \Rightarrow \Gamma & (euc_i) \\ \hline
\mathcal{R}', R_{[i]}uv, R^{*}_{[i]}wv, R^{*}_{[i]}ww, R_{[i]}uu, R_{[i]}vv, R^{*}_{[i]}vw \Rightarrow \Gamma & (euc_i) \\ \hline
\mathcal{R}', R_{[i]}uv, R_{[i]}wv, R_{[i]}ww, R_{[i]}uu, R_{[i]}vv \Rightarrow \Gamma & (euc_i) \\ \hline
\mathcal{R}', R_{[i]}uv, R_{[i]}wv \Rightarrow \Gamma & (ref_i) \times 3 \\ \hdashline
\mathcal{R}', R_{[i]}uv \Rightarrow \Gamma & \text{IH} \\ \cdots\cdots\cdots\cdots & = \\
\mathcal{R} \Rightarrow \Gamma &
\end{array}
$$

$$
\begin{array}{cl}
\Pi & \\ \hline
\mathcal{R}', R_{[i]}vu, R_{[i]}wu \Rightarrow \Gamma & \\ \hdashline
\mathcal{R}', R^{*}_{[i]}vu, R^{*}_{[i]}vw, R^{*}_{[i]}wu, R_{[i]}wv, R_{[i]}ww \Rightarrow \Gamma & (wk) \\ \hline
\mathcal{R}', R_{[i]}vu, R^{*}_{[i]}vw, R^{*}_{[i]}wv, R^{*}_{[i]}ww \Rightarrow \Gamma & (euc_i) \\ \hline
\mathcal{R}', R_{[i]}vu, R_{[i]}wv, R_{[i]}ww \Rightarrow \Gamma & (euc_i) \\ \hline
\mathcal{R}', R_{[i]}vu, R_{[i]}wv \Rightarrow \Gamma & (ref_i) \\ \hdashline
\mathcal{R}', R_{[i]}vu \Rightarrow \Gamma & \text{IH} \\ \hdashline
\mathcal{R} \Rightarrow \Gamma & =
\end{array}
$$

□

Theorem 23. *Every derivation in* $\mathsf{DS}^k_n\mathsf{L} + \mathrm{Str}(\mathsf{DS}^k_n)$ *can be algorithmically transformed into a derivation in* $\mathsf{G3DS}^k_n$.

Proof. We prove the result by induction on the height of the given derivation in $\mathsf{DS}^k_n\mathsf{L} + \mathrm{Str}(\mathsf{DS}^k_n)$.

Base case. Any instance of (id) in $\mathsf{DS}^k_n\mathsf{L} + \mathrm{Str}(\mathsf{DS}^k_n)$ is an instance of (id) in $\mathsf{G3DS}^k_n$, which resolves the base case.

Inductive step. We prove the inductive step by considering the last rule applied in $\mathsf{DS}^k_n\mathsf{L} + \mathrm{Str}(\mathsf{DS}^k_n)$. All rules in $\mathrm{Str}(\mathsf{DS}^k_n)$ are handled by applying IH followed by the hp-admissibility or eliminability of the rule (Lem. 20, 21, 23, and Thm. 14). With the exception of the $(Pr_{\langle i\rangle})$, $(Pr^1_{\ominus_i})$, and $(Pr^2_{\ominus_i})$ rules, all other cases are easily resolved since each is a rule in $\mathsf{G3DS}^k_n$. We show the $(Pr_{\langle i\rangle})$, $(Pr^1_{\ominus_i})$, and $(Pr^2_{\ominus_i})$ cases below, and note that in the $(Pr_{\langle i\rangle})$ case the side condition $w \sim^{\mathcal{R}}_i u$ holds, and in the $(Pr^1_{\ominus_i})$ case the side condition $u \sim^{\mathcal{R}}_i v$ holds, which allows for the invocation of Lem. 37.

$$
\begin{array}{cl}
\mathcal{R} \Rightarrow w : \langle i\rangle\phi, u : \phi, \Gamma & \\ \hline
\mathcal{R} \Rightarrow w : \langle i\rangle\phi, \Gamma & (Pr_{\langle i\rangle})
\end{array}
\quad \rightsquigarrow \quad
\begin{array}{cl}
 & \\ \hdashline
\mathcal{R} \Rightarrow w : \langle i\rangle\phi, u : \phi, \Gamma & \text{IH} \\ \hdashline
\mathcal{R}, R_{[i]}wu \Rightarrow w : \langle i\rangle\phi, u : \phi, \Gamma & (wk) \\ \hline
\mathcal{R}, R_{[i]}wu \Rightarrow w : \langle i\rangle\phi, \Gamma & (\langle i\rangle) \\ \hdashline
\mathcal{R} \Rightarrow w : \langle i\rangle\phi, \Gamma & \text{Lem. 37}
\end{array}
$$

$$
\begin{array}{cl}
\mathcal{R}, I_{\otimes_i}u \Rightarrow w : \ominus_i\phi, v : \phi, \Gamma & \\ \hline
\mathcal{R}, I_{\otimes_i}u \Rightarrow w : \ominus_i\phi, \Gamma & (Pr^1_{\ominus_i})
\end{array}
\quad \rightsquigarrow
$$

$$
\begin{array}{cl}
 & \\ \hdashline
\mathcal{R}, I_{\otimes_i}u \Rightarrow w : \ominus_i\phi, v : \phi, \Gamma & \text{IH} \\ \hdashline
\mathcal{R}, R_{[i]}uv, I_{\otimes_i}u, I_{\otimes_i}v \Rightarrow w : \ominus_i\phi, v : \phi, \Gamma & (wk) \\ \hline
\mathcal{R}, R_{[i]}uv, I_{\otimes_i}u, I_{\otimes_i}v \Rightarrow w : \ominus_i\phi, \Gamma & (\ominus_i) \\ \hline
\mathcal{R}, R_{[i]}uv, I_{\otimes_i}u \Rightarrow w : \ominus_i\phi, \Gamma & (D^i_3) \\ \hdashline
\mathcal{R}, I_{\otimes_i}u \Rightarrow w : \ominus_i\phi, \Gamma & \text{Lem. 37}
\end{array}
$$

$$\dfrac{\mathcal{R}, I_{\otimes_i} u \Rightarrow w : \ominus_i \phi, u : \phi, \Gamma}{\mathcal{R} \Rightarrow w : \ominus_i \phi, \Gamma}\,(Pr^2_{\ominus_i}) \quad \rightsquigarrow \quad \begin{array}{cl}
\hdashline
\mathcal{R}, I_{\otimes_i} u \Rightarrow w : \ominus_i \phi, u : \phi, \Gamma & \text{IH} \\
\hdashline
\mathcal{R}, I_{\otimes_i} u \Rightarrow w : \ominus_i \phi, \Gamma & (\ominus_i) \\
\hline
\mathcal{R} \Rightarrow w : \ominus_i \phi, \Gamma & (D^i_2)
\end{array}$$

□

Example 12. *The top derivation below is in* $\mathsf{DS}^k_n\mathsf{L}$ *and the bottom derivation below is in* $\mathsf{G3DS}^k_n$. *The first* $(Pr_{\langle i \rangle})$ *instance in the top derivation translates to the first* $(\langle i \rangle)$ *instance and* (euc_i) *instance in the bottom derivation, the second* $(Pr_{\langle i \rangle})$ *inference in the top derivation translates to the second* $(\langle i \rangle)$ *inference along with the* (ref_i) *instance in the bottom derivation, and the* $(Pr^2_{\ominus_i})$ *instance translates to the* $(\ominus_i)$ *and* (D^i_2) *instances in the bottom derivation. To transform the bottom derivation into the top derivation, the* (euc_i), (ref_i), *and* (D^i_2) *inferences are eliminated as explained in Lem. 34, 33, and 35, respectively. Last, we let* $\phi := [i]\neg p \lor [i]\langle i \rangle \langle i \rangle p$ *to save space.*

$$\begin{array}{cl}
\hline
I_{\otimes_i} w, R_{[i]} wu, R_{[i]} wv \Rightarrow z : \ominus_i \phi, v : \langle i \rangle \langle i \rangle p, v : \langle i \rangle p, u : p, u : \neg p & (id) \\
\hline
I_{\otimes_i} w, R_{[i]} wu, R_{[i]} wv \Rightarrow z : \ominus_i \phi, v : \langle i \rangle \langle i \rangle p, v : \langle i \rangle p, u : \neg p & (Pr_{\langle i \rangle}) \\
\hline
I_{\otimes_i} w, R_{[i]} wu, R_{[i]} wv \Rightarrow z : \ominus_i \phi, v : \langle i \rangle \langle i \rangle p, u : \neg p & (Pr_{\langle i \rangle}) \\
\hline
I_{\otimes_i} w \Rightarrow z : \ominus_i \phi, w : [i]\neg p, w : [i]\langle i \rangle \langle i \rangle p & ([i]) \times 2 \\
\hline
I_{\otimes_i} w \Rightarrow z : \ominus_i \phi, w : [i]\neg p \lor [i]\langle i \rangle \langle i \rangle p & (\lor_r) \\
\hdashline
I_{\otimes_i} w \Rightarrow z : \ominus_i([i]\neg p \lor [i]\langle i \rangle \langle i \rangle p), w : [i]\neg p \lor [i]\langle i \rangle \langle i \rangle p & = \\
\hline
\epsilon \Rightarrow z : \ominus_i([i]\neg p \lor [i]\langle i \rangle \langle i \rangle p) & (Pr^2_{\ominus_i})
\end{array}$$

$$\begin{array}{cl}
\hline
I_{\otimes_i} w, R_{[i]} wu, R_{[i]} wv, R_{[i]} vu, R_{[i]} vv \Rightarrow z : \ominus_i \phi, v : \langle i \rangle \langle i \rangle p, v : \langle i \rangle p, u : p, u : \neg p & (id) \\
\hline
I_{\otimes_i} w, R_{[i]} wu, R_{[i]} wv, R_{[i]} vu, R_{[i]} vv \Rightarrow z : \ominus_i \phi, v : \langle i \rangle \langle i \rangle p, v : \langle i \rangle p, u : \neg p & (\langle i \rangle) \\
\hline
I_{\otimes_i} w, R_{[i]} wu, R_{[i]} wv, R_{[i]} vv \Rightarrow z : \ominus_i \phi, v : \langle i \rangle \langle i \rangle p, v : \langle i \rangle p, u : \neg p & (euc_i) \\
\hline
I_{\otimes_i} w, R_{[i]} wu, R_{[i]} wv, R_{[i]} vv \Rightarrow z : \ominus_i \phi, v : \langle i \rangle \langle i \rangle p, u : \neg p & (\langle i \rangle) \\
\hline
I_{\otimes_i} w, R_{[i]} wu, R_{[i]} wv \Rightarrow z : \ominus_i \phi, v : \langle i \rangle \langle i \rangle p, u : \neg p & (ref_i) \\
\hline
I_{\otimes_i} w \Rightarrow z : \ominus_i \phi, w : [i]\neg p, w : [i]\langle i \rangle \langle i \rangle p & ([i]) \times 2 \\
\hline
I_{\otimes_i} w \Rightarrow z : \ominus_i \phi, w : [i]\neg p \lor [i]\langle i \rangle \langle i \rangle p & (\lor_r) \\
\hdashline
I_{\otimes_i} w \Rightarrow z : \ominus_i([i]\neg p \lor [i]\langle i \rangle \langle i \rangle p), w : [i]\neg p \lor [i]\langle i \rangle \langle i \rangle p & = \\
\hline
I_{\otimes_i} w \Rightarrow z : \ominus_i([i]\neg p \lor [i]\langle i \rangle \langle i \rangle p) & (\ominus_i) \\
\hline
\epsilon \Rightarrow z : \ominus_i([i]\neg p \lor [i]\langle i \rangle \langle i \rangle p) & (D^i_2)
\end{array}$$

We may now leverage our proof-theoretic transformations to show that our refined labelled calculi possess admissibility and invertibility properties, and in addition, are sound and complete.

Corollary 4. *The rules in* $\mathrm{Str}(\mathsf{DS}^k_n)$ *are admissible in* $\mathsf{DS}^k_n\mathsf{L}$.

Proof. Follows from Thm. 23 and Thm. 22. □

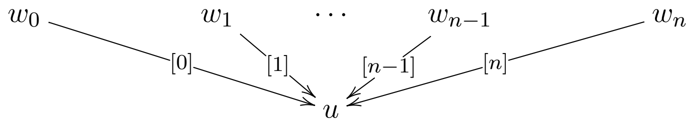
Corollary 5. *All rules of* $\mathsf{DS}^{k}_{n}\mathsf{L}$ *are invertible.*

Proof. Follows from Thm. 23 and Thm. 22, as well as Lem. 22. □

Corollary 6 (Soundness and Completeness of $\mathsf{DS}^{k}_{n}\mathsf{L}$)**.** *Let* $n, k \in \mathbb{N}$. *Then,*
(i) If $\vdash_{\mathsf{DS}^{k}_{n}\mathsf{L}} \Lambda$, *then* $\models_{\mathsf{DS}^{k}_{n}} \Lambda$.
(ii) If $\vdash_{\mathsf{DS}^{k}_{n}} \phi$, *then* $\vdash_{\mathsf{DS}^{k}_{n}\mathsf{L}} \epsilon \Rightarrow w : \phi$.

Proof. Follows from Thm. 13, 15, 22, and Thm. 23. □

Before concluding the section, we discuss the reduction in sequential structure that arises through refinement, and recall that each $\mathsf{G3DS}^{k}_{n}$ calculus is incomplete relative to labelled DAG and forest derivations (Thm. 16). If we consider the calculi in the class $\{\mathsf{DS}^{0}_{n}\mathsf{L} \mid n \in \mathbb{N}\}$, where the maximum number of choices is unrestricted (i.e. $k = 0$), then it is relatively straightforward to show that any derivation of a labelled theorem $w : \phi$ in such a calculus is a labelled DAG derivation. The reason why derivations for such calculi require labelled DAG sequents, as opposed to more minimalistic structures (e.g. labelled forest sequents), is due to the presence of the (IOA) rule, which introduces relational atoms that 'converge' to the same point when applied bottom-up. This is illustrated in the graphic below, which shows the kind of structure that would appear in the sequent graph of a labelled sequent if (IOA) was applied in reverse, introducing the relational atoms $R_{[0]}w_0u, R_{[1]}w_1u, \ldots, R_{[n-1]}w_{n-1}u, R_{[n]}w_nu$.



Therefore, as it currently stands, each refined labelled calculus $\mathsf{DS}^{0}_{n}\mathsf{L}$ (with $n > 0$) appears to require labelled DAG derivations as the (IOA) rule is vital for completeness.

Theorem 24. *Let* $n \in \mathbb{N}$ *and* $k = 0$. *Every derivation in* $\mathsf{DS}^{0}_{n}\mathsf{L}$ *of a labelled formula* $w : \phi$ *is a labelled DAG derivation with the rooted property.*

Proof. First, we recall that if $k = 0$, then (APC^{k}_{i}) is omitted from the calculus for each $i \in \mathrm{Ag}$ (see Fig. 4.4). To prove the result, let us consider a derivation in $\mathsf{DS}^{0}_{n}\mathsf{L}$ of the labelled sequent $\epsilon \Rightarrow w : \phi$ in a bottom-up manner. The only rules that add new labels (i.e. eigenvariables)—and therefore, change the set of vertices or edges in the sequent graph of a labelled sequent—are the $([i])$, $(\otimes_i)$, $(Pr^{2}_{\ominus_i})$, and (IOA) rules. If the $([i])$ rule is applied bottom-up to a labelled sequent Λ, then it will introduce a forward edge to the sequent graph of Λ. If the $(\otimes_i)$ or $(Pr^{2}_{\ominus_i})$ rule is applied bottom-up to a labelled sequent Λ, then it will introduce a new, disconnected vertex. If the (IOA) rule is applied bottom-up to a labelled sequent Λ, then it introduces $|\mathrm{Ag}| = n + 1$ new edges all converging to a fresh vertex in the sequent graph of Λ. Therefore, one can see that applying rules in reverse to the labelled sequent $\epsilon \Rightarrow w : \phi$ cannot introduce a cycle since new edges always connect to fresh vertices in sequent graphs. Furthermore, when new labels are introduced via bottom-up applications of $([i])$, $(\otimes_i)$, $(Pr^2_{\ominus_i})$, or (IOA), either forward edges are added to sequent graphs, or new disconnected vertices are added; this shows that if a label serves as a root, then no edges can be introduced pointing to that label, and hence, the labelled derivation possesses the rooted property. □

Last, we note that in the single-agent setting, i.e. for the class of calculi $\{\mathsf{DS}^k_0\mathsf{L} \mid k \in \mathbb{N}\}$, completeness only requires labelled forest derivations. This minimalism is partly due to the fact that the (IOA) rule is redundant when $|\mathrm{Ag}| = 1$, implying that convergent structures—such as the example presented above—need not be introduced. Still, the (APC^k_i) rule could break the property of being a labelled forest sequent when applied bottom-up, but it turns out that applications of (APC^k_i) can be restricted to ensure that the forest structure of a labelled sequent is preserved. We do not prove that each $\mathsf{DS}^k_0\mathsf{L}$ calculus only requires labelled forest derivations (with the rooted property) here, as the proof requires more sophisticated methods, which will be introduced in Sect. 6.1, and follows as a corollary from our work on proof-search for deontic STIT logics.

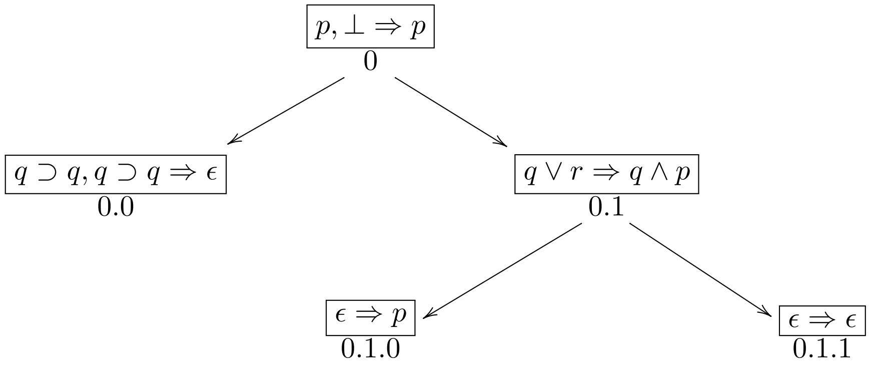
CHAPTER 5

The Method of Refinement: First-Order Intuitionistic Logics

We demonstrate how refining the labelled calculi G3IntQ and G3IntQC (see Fig. 3.3) for the first-order intuitionistic logics IntQ and IntQC (see Def. 20) yields new nested calculi for the logics (which we refer to as DIntQ and DIntQC, respectively; see Fig. 5.8 on p. 167). The characteristic feature of these nested calculi is the employment of *nested sequents* in deriving theorems, which—in the current context—are formulae encoding trees of two-sided Gentzen-style sequents. For example, the nested sequent:

$$p, \bot \Rightarrow p, \{q \supset q, q \supset q \Rightarrow \epsilon\}, \{q \lor r \Rightarrow q \land p, \{\epsilon \Rightarrow p\}, \{\epsilon \Rightarrow \epsilon\}\}$$

encodes the following tree of sequents:

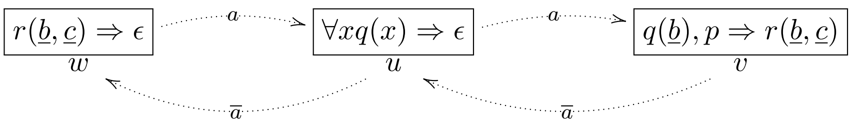

$p, \bot \Rightarrow p$
0
$q \supset q, q \supset q \Rightarrow \epsilon$
0.0
$q \lor r \Rightarrow q \land p$
0.1
$\epsilon \Rightarrow p$
0.1.0
$\epsilon \Rightarrow \epsilon$
0.1.1

The nested calculi we obtain via the method of refinement are distinct from the existing nested calculi for IntQ and IntQC provided by Fitting in [Fit14], which have been included in Appendix A (see p. 207). As will be argued below, the new nested systems DIntQ and

DIntQC obtained via refinement possess certain qualities that improve upon Fitting's nested calculi in a number of ways. Perhaps the most significant and advantageous distinction concerns the theoretical apparatus underpinning inference rules in DIntQ and DIntQC; we make use of CFCST systems (see Def. 8) to define inference rules for $\supset$, $\exists$, and $\forall$ and in the definition of initial sequents. Therefore, such rules are parameterized by formal grammars dictating (in)applications of each rule—just as was done with propagation rules in Sect. 4.1. To provide the reader with intuition concerning how formal grammars play a role in our nested systems, we give a concrete example of how an initial sequent is determined in DIntQ and DIntQC:

$$X := r(\underline{b}, \underline{c}) \Rightarrow \epsilon, \{\forall x q(x) \Rightarrow \epsilon, \{q(\underline{b}), p \Rightarrow r(\underline{b}, \underline{c})\}\}$$

Determining that X is an initial sequent consists of two steps: *first*, we transform the nested sequent into a *propagation graph* (much like the propagation graphs introduced in the previous chapter; see Def. 63), which is analogous to a finite deterministic automaton.[1] A pictorial representation of the propagation graph of X is shown below:


$r(\underline{b}, \underline{c}) \Rightarrow \epsilon$
w
a
$\forall x q(x) \Rightarrow \epsilon$
u
a
$q(\underline{b}), p \Rightarrow r(\underline{b}, \underline{c})$
v
$\overline{a}$
$\overline{a}$

For the propagation graph of a nested sequent, we make use of the character a to encode a *forward* move, that is, a transition to a (two-sided) Gentzen-style sequent that is *deeper* in the nestings, and make use of its converse $\overline{a}$ to encode a *backward* move to a *shallower* Gentzen-style sequent. Labeling the edges of the propagation graph with characters is useful because sequences of edges can be thought of as strings generated by a CFCST system encoding how formulae should be correctly positioned (or, correctly propagated—in the case of inference rules for $\supset$, $\exists$, and $\forall$) throughout the graph of the given nested sequent; this brings us to our *second* step in determining that X is initial, which relies on matching strings generated by a CFCST system to paths in the propagation graph of X.

For reasons that will be made clear in the subsequent section (Sect. 5.1), a nested sequent is considered initial in DIntQ and DIntQC so long as there is a sequence of edges in its propagation graph from a Gentzen-style sequent of the form $Y_1, p(\underline{\vec{a}}) \Rightarrow Z_1$ to another of the form $Y_2 \Rightarrow p(\underline{\vec{a}}), Z_2$ spelling a string from the set $\{\varepsilon, a, aa, aaa, \ldots\}$, where Y_1, Y_2, Z_1, and Z_2 are multisets of formulae from $\mathcal{L}_{\mathsf{Int}}$ (see Def. 13) and $p(\underline{\vec{a}})$ is an atomic formula. We can re-phrase this condition in grammar-theoretic terms by stating that a nested sequent is initial so long as there is a sequence of edges in its propagation graph from a Gentzen-style sequent $Y_1, p(\underline{\vec{a}}) \Rightarrow Z_1$ to another of the form $Y_2 \Rightarrow p(\underline{\vec{a}}), Z_2$ that spells a string generated by a with the following CFCST system:

$$S := \{a \longrightarrow \varepsilon, a \longrightarrow aa, \overline{a} \longrightarrow \varepsilon, \overline{a} \longrightarrow \overline{a}\overline{a}\},$$

[1] See [Sip12, Ch. 1] for an introduction to finite deterministic automata.

that is, the string is in the language $L_S(a)$ (see Def. 9). Note that $L_S(a) = \{\varepsilon, a, aa, aaa, \ldots\}$ and that strings within the language can be seen as encoding a notion of *reachability* in the propagation graph of a nested sequent with ε meaning that there is a forward path of length 0 between $Y_1, p(\vec{\underline{a}}) \Rightarrow Z_1$ and $Y_2 \Rightarrow p(\vec{\underline{a}}), Z_2$, with a meaning that there is a forward path of length 1 between $Y_1, p(\vec{\underline{a}}) \Rightarrow Z_1$ and $Y_2 \Rightarrow p(\vec{\underline{a}}), Z_2$, with aa meaning that there is a forward path of length 2 between $Y_1, p(\vec{\underline{a}}) \Rightarrow Z_1$ and $Y_2 \Rightarrow p(\vec{\underline{a}}), Z_2$, and so on. In terms of the semantics of IntQ and IntQC, this condition on initial sequents is equivalent to the monotonicity condition $(\mathbf{M})$ (see Def. 16) imposed on IntQ- and IntQC-models, which states that if an atomic formula $p(\vec{\underline{a}})$ is true at a world, then it is true at all future worlds. One can easily verify that the nested sequent X is initial via this condition since there is a sequence of (two) forward edges from w to v spelling the string aa—which can also be generated by a with the second production rule in S—where the initial node w is associated with $r(\underline{b}, \underline{c}) \Rightarrow \epsilon$ and the terminal node v is associated with $q(\underline{b}), p \Rightarrow r(\underline{b}, \underline{c})$.

The method of defining rules that make reference to formal grammars produces interesting and beneficial consequences. First, it allows us to recognize a class of rules that generalize the behavior of propagation rules, which we dub *reachability rules*. While propagation rules function by propagating a formula to the terminal node of a path, reachability rules function by (additionally) checking if data exists at the end of a (potentially separate) path. It is easy to imagine this behavior being generalized even further with reachability rules propagating formulae along an arbitrary number of paths conditional on the existence of data found along some number of paths, albeit we will only consider rather simple versions of such rules here. A second outcome of our grammar/language theoretical approach to defining rules is a substantial increase in the modularity of our nested systems compared to the systems put forth by Fitting. Fitting employs a rule—referred to as *lift* (see Appendix A on p. 207)—which encodes the transitivity and monotonicity properties of intuitionistic logics. We note that the *lift* rule is a reformulation of the propagation rules introduced by Postniece in [Pos09, Pos10], which were used to provide a deep-inference nested sequent system for bi-intuitionistic logic amenable to proof-search. In the approach presented here, monotonicity is encoded into an initial, reachability rule, and transitivity is encoded in a propagation rule for the $\supset$ connective and a reachability rule for the $\forall$ connective. By comparison then, our approach is more fine-grained than Fitting's as it distinguishes the monotonicity and transitivity properties as opposed to unifying them in a single rule. Consequently, this appears to bring about an increase in modularity as transforming our systems into systems for sub-intuitionistic logics [Cor87, Res94] should follow by the modification of our reachability and/or propagation rules. In fact, by 'plugging-in' alternative formal grammars and/or adding certain rules for new logical connectives, it appears that we can uniformly obtain proof calculi for a sizable class of propositional and first-order (sub-)(bi-)intuitionistic logics (such as those discussed in [Cor87, IK07, PU18, Rau80, Res94]), though such investigations are left to future work as they go beyond the logics considered in this thesis. Last, connecting our proof systems (and therefore, our logics) to formal grammars opens up the possibility of transferring results (e.g. (un)decidability results) between formal language theory and logic.

As mentioned at the onset, the primary goal of this chapter is to show how (new) nested calculi can be obtained for IntQ and IntQC via the method of refinement. In contrast to the propositional setting where refinement primarily proceeds via structural rule elimination, in the first-order setting we must also consider the removal of domain atoms (e.g. $\underline{a} \in D_w$) from labelled syntax. Thus, structural rule elimination will beget the *quasi-refined* labelled calculi IntQL^* and IntQCL^* containing rules whose applicability still depends on the occurrence of domain atoms. After obtaining these calculi, we then redefine our rules so that their applicability is independent of the occurrence of domain atoms, producing our *(fully) refined* labelled calculi IntQL and IntQCL, from which the nested calculi DIntQ and DIntQC may be obtained. To make our strategy for deriving DIntQ and DIntQC via refinement explicit, we clearly list the steps involved in the process, which also explains the main content of each of the following four sections:

1. Sect. 5.1: Discover through analyzing the elimination of structural rules, what conducive rules (i.e propagation and reachability rules) permit the elimination of all structural rules from the *labelled* calculi G3IntQ and G3IntQC (defined in Fig. 3.3).[2]
2. Sect. 5.2: Derive the *quasi-refined* labelled calculi IntQL^* and IntQCL^* (defined in Def. 74) from the labelled calculi G3IntQ and G3IntQC via structural rule elimination.
3. Sect. 5.3 (Part 1): Remove the domain atoms from the syntax of IntQL^* and IntQCL^* and re-define relevant inference rules so that applications of the rules are independent of the existence of domain atoms to obtain the *(fully) refined* labelled calculi IntQL and IntQCL (defined in Fig. 5.6).
4. Sect. 5.3 (Part 2): Show that the refined labelled calculi IntQL and IntQCL possess favorable proof-theoretic properties.
5. Sect. 5.4: Show that the *nested* calculi DIntQ and DIntQC (defined in Fig. 5.8) can be obtained from, and inherit the properties of, the refined labelled calculi IntQL and IntQCL.

The relationships established between the various proof calculi considered are displayed in Fig. 5.1. Solid arrows indicate that proofs in one calculus may be transformed into proofs in another calculus (which preserves the language of the calculus) and dotted arrows signify that proofs in one calculus may be translated into proofs in another calculus (which changes the language of the calculus). The symbol $\subset$ is used to indicate that one calculus is a restricted version of another; e.g. G3IntQ is a restricted version of G3IntQC as the former omits the (cd) rule and the latter includes it. Furthermore, arrows are annotated with the name of the result that establishes the transformation or translation, and the dotted arrows are additionally annotated with the $\mathfrak{N}$ and $\mathfrak{L}$ symbols, indicating the functions that translate from labelled to nested notation and the opposite, respectively, which are introduced in Sect. 5.4.

[2] Recall that *conducive rules* are rules that permit the elimination of structural rules when added to our calculi.

$$\begin{tikzcd}
\mathsf{G3IntQ} \arrow[r, bend left, "\text{Thm. 25}"] \arrow[d, phantom, "\cap"] & \mathsf{IntQL}^{*} \arrow[l, bend left, "\text{Thm. 26}"] \arrow[d, phantom, "\cap"] & \mathsf{IntQL} \arrow[l, "\text{Lem. 45}"'] \arrow[r, dotted, bend left, "\overset{\text{Thm. 33}}{\mathfrak{N}}"] \arrow[d, phantom, "\cap"] & \mathsf{DIntQ} \arrow[l, dotted, bend left, "\underset{\text{Thm. 34}}{\mathfrak{L}}"] \arrow[d, phantom, "\cap"] \\
\mathsf{G3IntQC} \arrow[r, bend left, "\text{Thm. 25}"] & \mathsf{IntQCL}^{*} \arrow[l, bend left, "\text{Thm. 26}"] & \mathsf{IntQCL} \arrow[l, "\text{Lem. 45}"'] \arrow[r, dotted, bend left, "\overset{\text{Thm. 33}}{\mathfrak{N}}"] & \mathsf{DIntQC} \arrow[l, dotted, bend left, "\underset{\text{Thm. 34}}{\mathfrak{L}}"]
\end{tikzcd}$$

Figure 5.1: Transformations and translations between the intuitionistic calculi considered.

5.1 Refinement Part I: Analysis

We analyze the issues that arise when attempting to perform structural rule elimination in $\mathsf{G3IntQ}$ and $\mathsf{G3IntQC}$ (see Fig. 3.3). This analysis will lead to the discovery of (and motivate the definition of) conducive rules that permit the elimination of (ref), (tra), (nd), (cd), and (ned) (see Fig. 3.3) from our labelled calculi. Our analysis of structural rule elimination will consist of three parts: (i) Considering the permutation of (ref) and (tra) above the (id) and $(\supset_l)$ rules, (ii) Considering the permutation of (nd) above the (id) and $(\exists_r)$ rules, and (iii) Considering the permutation of (ned) above the $(\forall_l)$ rule. Despite the fact that the case of permuting (ref) and (tra) above $(\forall_l)$, permuting (cd) (with (nd)) above (id), $(\exists_r)$, and $(\forall_l)$, and permuting (ned) above $(\exists_r)$ are all interesting cases, the understanding forged from considering the cases in (i) – (iii) above will be sufficient to motivate the definition of our reachability and propagation rules. Before going through our analysis, we introduce some helpful terminology:

Definition 71 (Principal, Auxiliary Label)**.** *We refer to the label of the principal formula in a labelled inference rule as the* principal label*. Also, the label(s) of the auxiliary formula(e) in a labelled inference rule is (are) referred to as the* auxiliary label(s)*. Principal and auxiliary labels are referred to as* active labels*, more generally.*

Let us begin part (i) of our analysis. We first consider the problematic cases that arise when attempting to permute (ref) above (id) and $(\supset_l)$, and after, look at the problematic cases that arise when attempting to permute (tra) above the two rules. The (ref) cases we consider are as follows:

$$\cfrac{\cfrac{}{\mathcal{R}, w \leq w, \underline{\vec{a}} \in D_w, \Gamma, w : p(\underline{\vec{a}}) \Rightarrow w : p(\underline{\vec{a}}), \Delta}\;(id)}{\mathcal{R}, \underline{\vec{a}} \in D_w, \Gamma, w : p(\underline{\vec{a}}) \Rightarrow w : p(\underline{\vec{a}}), \Delta}\;(ref)$$

$$\cfrac{\cfrac{\mathcal{R}, w \leq w, \Gamma, w : \phi \supset \psi \Rightarrow \Delta, w : \phi \qquad \mathcal{R}, w \leq w, \Gamma, w : \phi \supset \psi, w : \psi \Rightarrow \Delta}{\mathcal{R}, w \leq w, \Gamma, w : \phi \supset \psi \Rightarrow \Delta}\,(\supset_l)}{\mathcal{R}, \Gamma, w : \phi \supset \psi \Rightarrow \Delta}\,(ref)$$

In order to successfully eliminate (ref) in the above (id) case, the end sequent must be derivable without the use of (ref). However, by inspection of G3IntQ and G3IntQC, it is apparent that none of the rules can be used to derive the end sequent without the use of (ref). A similar issue arises when attempting to permute the (ref) rule upwards in the $(\supset_l)$ derivation, as shown below:

$$\cfrac{\mathcal{R}, w \leq w, \Gamma, w : \phi \supset \psi \Rightarrow \Delta, w : \phi}{\mathcal{R}, \Gamma, w : \phi \supset \psi \Rightarrow \Delta, w : \phi}\,(ref) \qquad \cfrac{\mathcal{R}, w \leq w, \Gamma, w : \phi \supset \psi, w : \psi \Rightarrow \Delta}{\mathcal{R}, \Gamma, w : \phi \supset \psi, w : \psi \Rightarrow \Delta}\,(ref)$$

In the above cases, no rule exists within G3IntQ or G3IntQC permitting us to derive the desired conclusion of the original proof. Nevertheless, if we add the following (sound) rules to our calculi, then the permutations with (ref) may be accomplished:

$$\cfrac{}{\mathcal{R}, \underline{\vec{a}} \in D_w, \Gamma, w : p(\underline{\vec{a}}) \Rightarrow w : p(\underline{\vec{a}}), \Delta}\,(id_0)$$

$$\cfrac{\mathcal{R}, \Gamma, w : \phi \supset \psi \Rightarrow \Delta, w : \phi \qquad \mathcal{R}, \Gamma, w : \phi \supset \psi, w : \psi \Rightarrow \Delta}{\mathcal{R}, \Gamma, w : \phi \supset \psi \Rightarrow \Delta}\,(\supset_l^0)$$

Notice that both of the above rules share a common characteristic. In the (id_0) rule, what was the (principal) labelled formula $u : p(\underline{\vec{a}})$ on the right of the sequent arrow $\Rightarrow$ has now become the labelled formula $w : p(\underline{\vec{a}})$, thus matching the principal formula on the left. By comparison, in the $(\supset_l^0)$ rule, the auxiliary formulae $u : \phi$ (in the left premise) and $u : \psi$ (in the right premise) have also *propagated* backward and are associated with the label w with which the principal formula $w : \phi \supset \psi$ is associated. This observation suggests that for the elimination of (ref), we require new versions of the (id), $(\supset_l)$, and $(\forall_l)$ rules (notice that all such rules include a relational atom of the form $w \leq u$ in their conclusion, which causes them to 'interact' with (ref)) where the auxiliary labels are reachable with a path of length zero from the principal label.[3] We will keep this fact in mind as we continue our analysis and observe the problematic case of permuting (tra) above (id) and $(\supset_l)$. These cases are as follows:

$$\cfrac{\cfrac{}{\mathcal{R}, w \leq v, v \leq u, w \leq u, \underline{\vec{a}} \in D_w, \Gamma, w : p(\underline{\vec{a}}) \Rightarrow u : p(\underline{\vec{a}}), \Delta}\,(id)}{\mathcal{R}, w \leq v, v \leq u, \underline{\vec{a}} \in D_w, \Gamma, w : p(\underline{\vec{a}}) \Rightarrow u : p(\underline{\vec{a}}), \Delta}\,(tra)$$

[3] Recall that we are omitting consideration of certain cases (e.g. $(\forall_l)$) in our analysis to keep it concise. In spite of this fact, for the curious reader, the new version of the $(\forall_l)$ rule would be as follows:

$$\cfrac{\mathcal{R}, \underline{a} \in D_w, \Gamma, w : \phi(\underline{a}/x), w : \forall x \phi \Rightarrow \Delta}{\mathcal{R}, \underline{a} \in D_w, \Gamma, w : \forall x \phi \Rightarrow \Delta}\,(\forall_l^0)$$

$$\Lambda_1 := \mathcal{R}, w \leq v, v \leq u, w \leq u, \Gamma, w : \phi \supset \psi \Rightarrow \Delta, u : \phi$$

$$\Lambda_2 := \mathcal{R}, w \leq v, v \leq u, w \leq u, \Gamma, w : \phi \supset \psi, u : \psi \Rightarrow \Delta$$

$$\dfrac{\dfrac{\Lambda_1 \qquad \Lambda_2}{\mathcal{R}, w \leq v, v \leq u, w \leq u, \Gamma, w : \phi \supset \psi \Rightarrow \Delta}(\supset_l)}{\mathcal{R}, w \leq v, v \leq u, \Gamma, w : \phi \supset \psi \Rightarrow \Delta}(tra)$$

As before, the end sequent of the first proof is not derivable without the use of (tra) in G3IntQ and G3IntQC, suggesting that the end sequent should be taken as an initial sequent in its own right. Also, permuting the (tra) rule upwards in the $(\supset_l)$ proof yields proofs ending with the following two inferences:

$$\dfrac{\mathcal{R}, w \leq v, v \leq u, w \leq u, \Gamma, w : \phi \supset \psi \Rightarrow \Delta, u : \phi}{\mathcal{R}, w \leq v, v \leq u, \Gamma, w : \phi \supset \psi \Rightarrow \Delta, u : \phi}(tra)$$

$$\dfrac{\mathcal{R}, w \leq v, v \leq u, w \leq u, \Gamma, w : \phi \supset \psi, u : \psi \Rightarrow \Delta}{\mathcal{R}, w \leq v, v \leq u, \Gamma, w : \phi \supset \psi, u : \psi \Rightarrow \Delta}(tra)$$

Since no rule is applicable to the conclusions above that would allow us to derive the desired conclusion, the $(\supset_l)$ case also suggests the need of a new $(\supset_l)$ rule. The new versions of (id) and $(\supset_l)$ suggested by our analysis are as shown below, with $\mathcal{R}_2 := \mathcal{R}, w \leq v, v \leq u$. It is crucial to observe that a directed path of length two (due to the $w \leq v, v \leq u$ relational atoms) occurs between the principal and auxiliary labels in the rules below:

$$\dfrac{}{\mathcal{R}_2, \underline{\vec{a}} \in D_w, \Gamma, w : p(\underline{\vec{a}}) \Rightarrow u : p(\underline{\vec{a}}), \Delta}(id_2)$$

$$\dfrac{\mathcal{R}_2, \Gamma, w : \phi \supset \psi \Rightarrow \Delta, u : \phi \qquad \mathcal{R}_2, \Gamma, w : \phi \supset \psi, u : \psi \Rightarrow \Delta}{\mathcal{R}_2, \Gamma, w : \phi \supset \psi \Rightarrow \Delta}(\supset_l^2)$$

Adding the above rules to our calculi would permit the permutation of (tra) above (id) and $(\supset_l)$, yet, if such rules are added to our calculi, then we must consider permuting (tra) above them as well. Let us examine the problematic cases that arise when attempting to permute (tra) above (id_2) and $(\supset_l^2)$, and after, we will summarize and discuss the conducive rules suggested by part (i) of our analysis.

$$\dfrac{\dfrac{}{\mathcal{R}, w \leq v, v \leq z, z \leq u, w \leq u, \underline{\vec{a}} \in D_w, \Gamma, w : p(\underline{\vec{a}}) \Rightarrow u : p(\underline{\vec{a}}), \Delta}(id_2)}{\mathcal{R}, w \leq v, v \leq z, z \leq u, \underline{\vec{a}} \in D_w, \Gamma, w : p(\underline{\vec{a}}) \Rightarrow u : p(\underline{\vec{a}}), \Delta}(tra)$$

$$\Lambda_1 := \mathcal{R}, w \leq v, v \leq z, z \leq u, w \leq u, \Gamma, w : \phi \supset \psi \Rightarrow \Delta, u : \phi$$

$$\Lambda_2 := \mathcal{R}, w \leq v, v \leq z, z \leq u, w \leq u, \Gamma, w : \phi \supset \psi, u : \psi \Rightarrow \Delta$$

$$\dfrac{\dfrac{\Lambda_1 \qquad \Lambda_2}{\mathcal{R}, w \leq v, v \leq z, z \leq u, w \leq u, \Gamma, w : \phi \supset \psi \Rightarrow \Delta}\,(\supset_l)}{\mathcal{R}, w \leq v, v \leq z, z \leq u, \Gamma, w : \phi \supset \psi \Rightarrow \Delta}\,(tra)$$

At this point, the reader might have guessed that the above cases cannot be resolved within our calculi, unless new versions of the (id_2) and $(\supset_l^2)$ rules are added. Such rules would have the following form, with $\mathcal{R}_3 := \mathcal{R}, w \leq v, v \leq z, z \leq u$, thus requiring the existence of a (directed) relational path of length three between principal and auxiliary labels.

$$\dfrac{}{\mathcal{R}_3, \underline{\vec{a}} \in D_w, \Gamma, w : p(\underline{\vec{a}}) \Rightarrow u : p(\underline{\vec{a}}), \Delta}\,(id_3)$$

$$\dfrac{\mathcal{R}_3, \Gamma, w : \phi \supset \psi \Rightarrow \Delta, u : \phi \qquad \mathcal{R}_3, \Gamma, w : \phi \supset \psi, u : \psi \Rightarrow \Delta}{\mathcal{R}_3, \Gamma, w : \phi \supset \psi \Rightarrow \Delta}\,(\supset_l^3)$$

Of course, extending our calculi with the above rules would require considering their permutability with (tra), thus necessitating the addition of new rules (id_4) and $(\supset_l^4)$ (having relational paths of length four between their auxiliary labels and principal labels), which would further require the addition of new rules to permit their permutation with (tra), ad infinitum.

A pattern has emerged from our analysis: The rules (id_0) and $(\supset_l^0)$ require relational paths of length zero between auxiliary labels and principal labels, the original rules (id) and $(\supset_l)$ require relational paths of length one (i.e. a single relational atom) between auxiliary labels and principal labels, the (id_2) and $(\supset_l^2)$ rules require relational paths of length two (i.e. there are two relational atoms $w \leq v, v \leq u$) between auxiliary labels and principal labels, etc. Similar to Sect. 4.1, it appears that what is needed to allow for the elimination of (ref) and (tra) is the addition of propagation rules to our calculi, where, the side condition permits an application of the rule given that the auxiliary labels are *reachable* (with a directed path) from the principal label. There are a variety of ways in which we could define such rules, but since we have CFCST systems at our disposal, we will make use of such objects to define our propagation rules, analogous to what was done in Sect. 4.1 for grammar logics. (NB. This approach differs from the author's approach in [Lyo20a, Lyo20b] where applications of rules rely on the notion of a *path*.)

Recall that propagation rules view labelled sequents as propagation graphs, and allow for formulae to be introduced at terminal nodes of propagation paths corresponding to strings generated by a CFCST system. By part (i) of our analysis, we found that we should employ rules that propagate formulae to *reachable* labels. Hence, the question arises; how do we encode a notion of reachability in a CFCST system? Answering this question will tell us what side conditions to impose on our propagation rules and will dictate the definition of a propagation graph (given below).

As previously implied, a label u in a labelled sequent is reachable from another label w iff there exists a sequence of relational atoms $w \leq v_1, \ldots, v_n \leq u$ of length zero or greater from w to u in the labelled sequent. Let us fix the minimal alphabet $\sum := \{a, \overline{a}\}$ for the remainder of the next four sections (Sect. 5.1 – 5.4), i.e. any reference to $\sum$ will be a reference to the alphabet $\{a, \overline{a}\}$ in Sect. 5.1 – 5.4. If we think of the character a as encoding a single relational atom of the form $w \leq u$, then all of the strings $\varepsilon, a, aa, aaa, \ldots \in (\sum^{+})^{*}$ can be thought of as encoding sequences of relational atoms. Note that all such sequences are generated by the following CFCST system:[4]

$$S4 := \{a \longrightarrow \varepsilon, a \longrightarrow aa, \overline{a} \longrightarrow \varepsilon, \overline{a} \longrightarrow \overline{a}\overline{a}\}$$

Observe that the above CFCST system is sufficient to encode a notion of reachability as $L_{S4}(a) = (\sum^{+})^{*}$. Furthermore, since we are thinking of the character a as encoding a relational atom of the form $w \leq u$, we will use a to index forward edges in our propagation graphs, and $\overline{a}$ to index backward edges, as described in the definition below:

Definition 72 (Propagation Graphs for IntQL and IntQCL). *Let $\Lambda = \mathcal{R}, \Gamma \Rightarrow \Delta$ be a labelled sequent for first-order intuitionistic logics. We define the* propagation graph *$PG(\Lambda) = (\mathrm{V}, \mathrm{E})$ to be the directed graph such that*

- $\mathrm{V} := \mathrm{Lab}(\Lambda)$*;*
- $\mathrm{E} := \{(w, u, a), (u, w, \overline{a}) \mid w \leq u \in \mathcal{R}\}$.

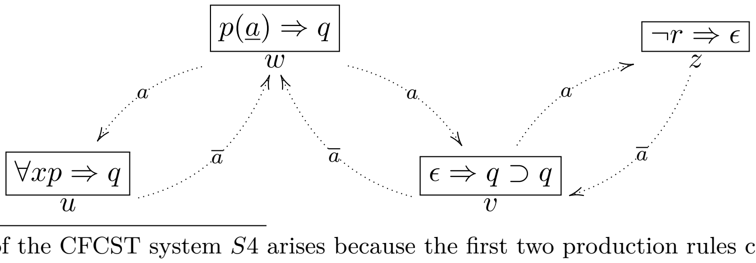
We will often write $w \in PG(\Lambda)$ to mean $w \in \mathrm{V}$, and $(w, u, x) \in PG(\Lambda)$ to mean $(w, u, x) \in \mathrm{E}$, for $x \in \sum$.

Propagation paths, strings of propagation paths, and their converses are defined as in Def. 59, so we need not repeat the definitions here. However, we provide an example of all such concepts below for review, as well an an example of a propagation graph of a labelled sequent for first-order intuitionistic logics.

Example 13. *Let our labelled sequent be the following:*

$$\Lambda := w \leq u, w \leq v, v \leq z, \underline{a} \in D_{w}, w : p(\underline{a}), u : \forall x p(x), z : \neg r \Rightarrow w : q, u : q \supset q, v : q$$

The propagation graph $PG(\Lambda)$ is shown below:


$p(\underline{a}) \Rightarrow q$
w
$\neg r \Rightarrow \epsilon$
z
a
a
a
$\overline{a}$
$\overline{a}$
$\overline{a}$
$\forall xp \Rightarrow q$
u
$\epsilon \Rightarrow q \supset q$
v

[4] The name of the CFCST system $S4$ arises because the first two production rules correspond to the properties of reflexivity and transitivity, respectively, as shown in Fig. 2.1. Since the modal logic S4 is the minimal normal modal logic that is sound and complete relative to reflexive and transitive Kripke frames, the name $S4$ for the above CFCST systems is apt.

$$\frac{}{\mathcal{R}, \underline{a}_1 \in D_{u_1}, \ldots, \underline{a}_n \in D_{u_n}, \Gamma, w : p(\vec{\underline{a}}) \Rightarrow u : p(\vec{\underline{a}}), \Delta} \ (id^{\mathsf{X}})^{\dagger_1(\mathsf{X})}$$

$$\frac{\mathcal{R}, \underline{a} \in D_u, \Gamma \Rightarrow \Delta, w : \phi(\underline{a}/x), w : \exists x \phi}{\mathcal{R}, \underline{a} \in D_u, \Gamma \Rightarrow \Delta, w : \exists x \phi} \ (\exists^{\mathsf{X}}_{r1})^{\dagger_2(\mathsf{X})}$$

$$\frac{\mathcal{R}, \underline{a} \in D_u, \Gamma \Rightarrow w : \phi(\underline{a}/x), w : \exists x \phi, \Delta}{\mathcal{R}, \Gamma \Rightarrow w : \exists x \phi, \Delta} \ (\exists^{\mathsf{X}}_{r2})^{\dagger_3(\mathsf{X})}$$

$$\frac{\mathcal{R}, \underline{a} \in D_u, w : \forall x \phi, v : \phi(\underline{a}/x), \Gamma \Rightarrow \Delta}{\mathcal{R}, \underline{a} \in D_u, w : \forall x \phi, \Gamma \Rightarrow \Delta} \ (\forall^{\mathsf{X}}_{l1})^{\dagger_4(\mathsf{X})}$$

$$\frac{\mathcal{R}, \underline{a} \in D_u, \Gamma, w : \forall x \phi, v : \phi(\underline{a}/x) \Rightarrow \Delta}{\mathcal{R}, \Gamma, w : \forall x \phi \Rightarrow \Delta} \ (\forall^{\mathsf{X}}_{l2})^{\dagger_5(\mathsf{X})}$$

$$\frac{\mathcal{R}, w : \phi \supset \psi, \Gamma \Rightarrow \Delta, u : \phi \qquad \mathcal{R}, w : \phi \supset \psi, u : \psi, \Gamma \Rightarrow \Delta}{\mathcal{R}, w : \phi \supset \psi, \Gamma \Rightarrow \Delta} \ (Pr_{\supset})^{\dagger_6(\mathsf{X})}$$

Figure 5.2: Reachability and propagation rules. The rules when $\mathsf{X} = \mathsf{Q}$ will be added to $\mathsf{G3IntQ}$ to refine the calculus, and the rules when $\mathsf{X} = \mathsf{QC}$ will be added to $\mathsf{G3IntQC}$ to refine the calculus. The side conditions $\dagger_1(\mathsf{Q}) - \dagger_6(\mathsf{Q})$ are defined in Fig. 5.3 and the side conditions $\dagger_1(\mathsf{QC}) - \dagger_6(\mathsf{QC})$ are defined in Fig. 5.4.

Examples of propagation paths include the propagation path $\pi(w, z) = w, a, v, a, z$ and the propagation path $\pi'(v, v) := v, \overline{a}, w, a, v$, with converses $\overline{\pi}(z, w) = z, \overline{a}, v, \overline{a}, w$ and $\overline{\pi}'(v, v) := v, \overline{a}, w, a, v$, respectively. The string of each propagation path is $s_\pi(w, z) = aa$ and $s_{\pi'}(v, w) = \overline{a}a$ with the converse strings $\overline{s}_\pi(z, w) = \overline{a}\overline{a}$ and $\overline{s}_{\pi'}(w, v) = \overline{a}a$, respectively.

We are now in a position to put forth the propagation rules resulting from part (i) of our analysis. The propagation rule $(Pr_{\supset})$ which replaces the $(\supset_l)$ rule in our (quasi-)refined calculi is shown in Fig. 5.2. The propagation rule (Pr_{id}) for (id) resulting from part (i) of our analysis is as follows:

$$\frac{}{\mathcal{R}, \vec{\underline{a}} \in D_w, \Gamma, w : p(\vec{\underline{a}}) \Rightarrow u : p(\vec{\underline{a}}), \Delta} \ (Pr_{id})^{\dagger}$$

where the side condition † states that $\exists \pi (s_\pi(w, u) \in L_{S4}(a))$, that is, there must exist a propagation path $\pi(w, u)$ in the propagation graph of the labelled sequent such that $s_\pi(w, u) \in L_{S4}(a)$. It should be noted however, that the above rule will not occur in our quasi-refined labelled calculi for IntQ and IntQC. The reason being, the occurrence of the domain atoms $\vec{\underline{a}} \in D_w$ cause the rule to interact with the (nd) and (cd) rules, and so, via step (ii) of our analysis the (Pr_{id}) rule will be generalized to a proper reachability rule,

Name	Side Condition
$\dagger_1(\mathsf{Q})$	$\exists \pi_i (s_{\pi_i}(u_i, w) \in L_{S4}(a))$ for each $i \in \{1, \ldots, n\}$, and $\exists \pi (s_\pi(w, u) \in L_{S4}(a))$
$\dagger_2(\mathsf{Q})$	$\exists \pi (s_\pi(u, w) \in L_{S4}(a))$
$\dagger_3(\mathsf{Q})$	$\underline{a}$ is an eigenvariable and $\exists \pi (s_\pi(u, w) \in L_{S4}(a))$
$\dagger_4(\mathsf{Q})$	$\exists \pi (s_\pi(u, v) \in L_{S4}(a))$ and $\exists \pi (s_\pi(w, v) \in L_{S4}(a))$
$\dagger_5(\mathsf{Q})$	$\underline{a}$ is an eigenvariable, $\exists \pi (s_\pi(u, v) \in L_{S4}(a))$ and $\exists \pi (s_\pi(w, v) \in L_{S4}(a))$
$\dagger_6(\mathsf{Q})$	$\exists \pi (s_\pi(w, u) \in L_{S4}(a))$

Figure 5.3: Side conditions for the reachability and propagation rules used to refine G3IntQ.

yielding the final version of the rule that will occur in our quasi-refined labelled calculi. With that being said, let us begin part (ii) of our analysis.

In part (ii) of our analysis, we consider the permutation of (nd) above the (id) and $(\exists_r)$ rules. Notwithstanding, since we discovered in part (i) that our quasi-refined systems require the propagation rule (Pr_{id}) instead of (id) (which is subsumed by (Pr_{id}) as shown in Lem. 38 below), we will analyze permutations of (nd) above (Pr_{id}) rather than above (id). To ease our analysis, we assume the existence of a single principal domain atom in (Pr_{id}). Below, we show the non-trivial cases of applying (nd) to (Pr_{id}) and $(\exists_r)$, along with the result of attempting to permute (nd) above $(\exists_r)$ by applying (nd) to the premise of $(\exists_r)$.

$$\dfrac{\dfrac{}{\mathcal{R}, v \leq w, \underline{a} \in D_v, \underline{a} \in D_w, \Gamma, w : p(\underline{a}) \Rightarrow u : p(\underline{a}), \Delta}(Pr_{id})}{\mathcal{R}, v \leq w, \underline{a} \in D_v, \Gamma, w : p(\underline{a}) \Rightarrow u : p(\underline{a}), \Delta}(nd)$$

$$\dfrac{\dfrac{\mathcal{R}, v \leq w, \underline{a} \in D_v, \underline{a} \in D_w, \Gamma \Rightarrow w : \phi(\underline{a}/x), w : \exists x \phi, \Delta}{\mathcal{R}, v \leq w, \underline{a} \in D_v, \underline{a} \in D_w, \Gamma \Rightarrow w : \exists x \phi, \Delta}(\exists_r)}{\mathcal{R}, v \leq w, \underline{a} \in D_v, \Gamma \Rightarrow w : \exists x \phi, \Delta}(nd) \quad \rightsquigarrow$$

$$\dfrac{\mathcal{R}, v \leq w, \underline{a} \in D_v, \underline{a} \in D_w, \Gamma \Rightarrow w : \phi(\underline{a}/x), w : \exists x \phi, \Delta}{\mathcal{R}, v \leq w, \underline{a} \in D_v, \Gamma \Rightarrow w : \phi(\underline{a}/x), w : \exists x \phi, \Delta}(nd)$$

By inspecting the rules of our calculi, one can confirm that the end sequent of the (Pr_{id}) proof cannot be derived by other means. The issue is that the domain atom $\underline{a} \in D_v$ is associated with a label v that is one step in the past of the label w (due to the $v \leq w$ relational atom) from where it needs to occur. Similarly, no rule is applicable in the $(\exists_r)$ case to allow for the permutation to go through, and the domain atom has also shifted one step into the past (after applying (nd) to the premise of $(\exists_r)$). As we have typically done, we can add new rules to our calculi to allow for the permutation of (nd) above (Pr_{id}) and $(\exists_r)$ to go through. The (sound) rules suggested by the above analysis are the following:

$$\frac{}{\mathcal{R}, v \leq w, \underline{a} \in D_v, \Gamma, w : p(\underline{a}) \Rightarrow u : p(\underline{a}), \Delta} \ (Pr_{id}^{-1})$$

$$\frac{\mathcal{R}, v \leq w, \underline{a} \in D_v, \Gamma \Rightarrow w : \phi(\underline{a}/x), w : \exists x \phi, \Delta}{\mathcal{R}, v \leq w, \underline{a} \in D_v, \Gamma \Rightarrow w : \exists x \phi, \Delta} \ (\exists_r^{-1})$$

Adding the above rules to our labelled calculi will allow for the permutation of (nd) above (Pr_{id}) and $(\exists_r)$ to go through, however, their addition implies that we must consider their permutability with (nd). The non-trivial cases of applying (nd) to (Pr_{id}^{-1}) and $(\exists_r^{-1})$ are shown below, along with the application of (nd) to the premise of $(\exists_r^{-1})$ in attempt to permute the two rules.

$$\cfrac{\cfrac{}{\mathcal{R}, z \leq v, v \leq w, \underline{a} \in D_z, \underline{a} \in D_v, \Gamma, w : p(\underline{a}) \Rightarrow u : p(\underline{a}), \Delta} \ (Pr_{id}^{-1})}{\mathcal{R}, z \leq v, v \leq w, \underline{a} \in D_z, \Gamma, w : p(\underline{a}) \Rightarrow u : p(\underline{a}), \Delta} \ (nd)$$

$$\cfrac{\cfrac{\mathcal{R}, z \leq v, v \leq w, \underline{a} \in D_z, \underline{a} \in D_v, \Gamma \Rightarrow w : \phi(\underline{a}/x), w : \exists x \phi, \Delta}{\mathcal{R}, z \leq v, v \leq w, \underline{a} \in D_z, \underline{a} \in D_v, \Gamma \Rightarrow w : \exists x \phi, \Delta} \ (\exists_r^{-1})}{\mathcal{R}, z \leq v, v \leq w, \underline{a} \in D_z, \Gamma \Rightarrow w : \exists x \phi, \Delta} \ (nd) \quad \rightsquigarrow$$

$$\frac{\mathcal{R}, z \leq v, v \leq w, \underline{a} \in D_z, \underline{a} \in D_v, \Gamma \Rightarrow w : \phi(\underline{a}/x), w : \exists x \phi, \Delta}{\mathcal{R}, z \leq v, v \leq w, \underline{a} \in D_z, \Gamma \Rightarrow w : \phi(\underline{a}/x), w : \exists x \phi, \Delta} \ (nd)$$

Yet again, inspecting the rules of our labelled calculi will prove that the end sequent of the (Pr_{id}^{-1}) inference cannot be derived by other means, and that no rules permit the permutation of (nd) above $(\exists_r^{-1})$. Nonetheless, what is interesting is that both the end sequent of the (Pr_{id}^{-1}) derivation and the concluding sequent obtained from applying (nd) to the premise of $(\exists_r^{-1})$ contain domain atoms $\underline{a} \in D_z$ associated with a label z that is two steps in the past of the principal label w (due to the relational atoms $z \leq v, v \leq w$). Therefore, similar to how permutations of the (tra) rule caused formulae to be propagated forward, we have found that permutations of the (nd) rule propagate domain atoms *backward.* Likewise, an analysis of permuting the (cd) rule above (Pr_{id}) and $(\exists_r)$ would have us conclude the *forward* propagation of active domain atoms. In the constant

Name	Side Condition
$\dagger_1(\mathsf{QC})$	$\exists\pi_i(s_{\pi_i}(u_i, w) \in L_{S5}(a))$ for each $i \in \{1, \ldots, n\}$, and $\exists\pi(s_\pi(w, u) \in L_{S4}(a))$
$\dagger_2(\mathsf{QC})$	$\exists\pi(s_\pi(u, w) \in L_{S5}(a))$
$\dagger_3(\mathsf{QC})$	$\underline{a}$ is an eigenvariable and $\exists\pi(s_\pi(u, w) \in L_{S5}(a))$
$\dagger_4(\mathsf{QC})$	$\exists\pi(s_\pi(u, v) \in L_{S5}(a))$ and $\exists\pi(s_\pi(w, v) \in L_{S4}(a))$
$\dagger_5(\mathsf{QC})$	$\underline{a}$ is an eigenvariable, $\exists\pi(s_\pi(u, v) \in L_{S5}(a))$ and $\exists\pi(s_\pi(w, v) \in L_{S4}(a))$
$\dagger_6(\mathsf{QC})$	$\exists\pi(s_\pi(w, u) \in L_{S4}(a))$

Figure 5.4: Side conditions for the reachability and propagation rules used to refine G3IntQC.

domain setting then (where both (nd) and (cd) are present), successive permutations of (nd) and (cd) could cause the active domain atom(s) to propagate to any label within the labelled sequent (via a sequence of forward and backward 'shifts' along relational atoms of the form $w \leq u$).

Such findings suggest that in the non-constant domain setting, we ought to impose a side condition on (Pr_{id}) and $(\exists_r)$ stating that the principal label w is reachable from the label(s) associated with the active domain atom(s) (via a directed path of relational atoms). Alternatively, in the constant domain setting, we ought to impose the side condition that the principal label w is reachable from the label(s) associated with the active domain atom(s), *but not necessarily with a directed path.* Imposing these side conditions yields the reachability rules (id^{X}) and $(\exists^{\mathsf{X}}_{r1})$ shown in Fig. 5.2. The reachability rule $(\forall^{\mathsf{X}}_{l1})$ is also displayed in that figure; the form of the rule and side conditions imposed were discovered through an analysis of permuting (ref), (tra), (nd) and (cd) above the $(\forall_l)$ rule, similar to what was done for (id), $(\supset_l)$, and $(\exists_r)$ above.

By what was said in part (i) of our analysis, we found that the CFCST system $S4$ encodes a notion of *directed* reachability. To encode a notion of *undirected* reachability—used to formulate side conditions of rules in the constant domain setting—we make use of the following CFCST system:[5]

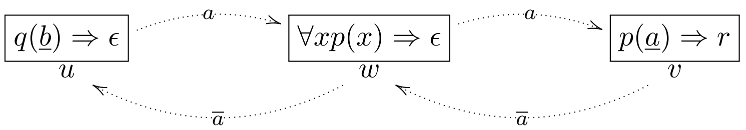
$$S5 := \{a \longrightarrow \varepsilon, a \longrightarrow \overline{a}a, \overline{a} \longrightarrow \varepsilon, \overline{a} \longrightarrow a\overline{a}\}$$

[5] The name of $S5$ was chosen because the first two production rules correspond to the properties of reflexivity and Euclideanity, respectively, as shown in Fig. 2.1. The name alludes to the logic S5, which is the minimal normal modal logic sound and complete on all reflexive and Euclidean frames.

Recall that in the propagation graph of a labelled sequent (Def. 72) the character a is used to encode a forward movement along a relational atom of the form $w \leq u$, and the converse character $\overline{a}$ is used to encode a reverse movement. Hence, two labels are reachable (*in an undirected sense*) iff there is a propagation path whose string lies within the set $\sum^{*} = \{a, \overline{a}\}^{*}$. Since $\sum^{*} = L_{S5}(a)$, we know that the above CFCST system properly encodes the notion of undirected reachability. Before moving onto step (iii) of our analysis, we consider an example of a concrete application of the $(\forall^{\mathsf{Q}}_{l1})$ rule to demonstrate the functionality of a reachability rule in the current setting.

Example 14. *Let us consider the labelled sequent* Λ *below, which has the propagation graph* $PG(\Lambda)$ *shown below bottom. Note that the vertices of the propagation graph have been decorated to make the corresponding formulae explicit.*

$$\Lambda := u \leq w, w \leq v, \underline{a} \in D_u, u : q(\underline{b}), w : \forall v p(x), v : p(\underline{a}) \Rightarrow v : r$$


$q(\underline{b}) \Rightarrow \epsilon$
u
a
$\forall x p(x) \Rightarrow \epsilon$
w
a
$p(\underline{a}) \Rightarrow r$
v
$\overline{a}$
$\overline{a}$

Since there exists a propagation path $\pi(u, v) := u, a, w, a, v$ *such that* $s_{\pi}(u, v) = aa \in L_{S4}(a)$ *and a propagation path* $\pi'(w, v) := w, a, v$ *such that* $s_{\pi}(w, v) = a \in L_{S4}(a)$*, we can apply the* $(\forall^{\mathsf{Q}}_{l1})$ *rule to delete the labelled formula* $v : p(\underline{a})$ *and derive the following labelled sequent:*

$$\Lambda' := u \leq w, w \leq v, \underline{a} \in D_u, u : q(\underline{b}), w : \forall v p(x) \Rightarrow v : r$$

A useful feature of our propagation and reachability rules (shown in Fig. 5.2) is that the propagation graph of the premise of the rule is identical to the propagation graph of the conclusion, meaning that the side condition can be checked regardless of if we are applying the rule top-down or bottom-up (the latter fact being useful for proof-search). Also, it should be noted that the side conditions are read in a similar manner to the side conditions of propagation rules for grammar logics introduced in Sect. 4.1. For instance, the side condition '$\exists \pi (s_{\pi}(u, v) \in L_{S5}(a))$' states there exists a propagation path $\pi(u, v)$ in the propagation graph of the premise or conclusion (depending on if the rule is being applied top-down or bottom-up) such that $s_{\pi}(u, v) \in L_{S5}(a)$. Often times when discussing the side conditions of propagation or reachability rules we will omit mention of the propagation graph and take it for the granted that the propagation path exists in the propagation graph of the premise and/or conclusion of the rule being discussed.

Let us now move onto step (iii) of our analysis. We have determined that instead of employing the $(\exists_r)$ and $(\forall_l)$ rules, we ought to make use of the reachability rules $(\exists^{\mathsf{X}}_{r1})$ and $(\forall^{\mathsf{X}}_{l1})$, which subsume $(\exists_r)$ and $(\forall_l)$, respectively (see Lem. 38 below). It so happens that permuting (*ned*) above $(\exists^{\mathsf{X}}_{r1})$ and $(\forall^{\mathsf{X}}_{l1})$ are the only non-trivial cases that occur when analyzing (*ned*) elimination. We analyze the non-trivial case that occurs when

Name		Set of Rules
$\mathrm{R}(\mathsf{Q})$	$:=$	$\{(id^{\mathsf{Q}}), (\exists^{\mathsf{Q}}_{r1}), (\exists^{\mathsf{Q}}_{r2}), (\forall^{\mathsf{Q}}_{l1}), (\forall^{\mathsf{Q}}_{l2}), (Pr_{\supset})\}$
$\mathrm{R}(\mathsf{QC})$	$:=$	$\{(id^{\mathsf{QC}}), (\exists^{\mathsf{QC}}_{r1}), (\exists^{\mathsf{QC}}_{r2}), (\forall^{\mathsf{QC}}_{l1}), (\forall^{\mathsf{QC}}_{l2}), (Pr_{\supset})\}$

Figure 5.5: Sets of reachability and propagation rules.

attempting to permute (ned) above $(\forall^{\mathsf{X}}_{l1})$, which will motivate how both $(\exists^{\mathsf{X}}_{r1})$ and $(\forall^{\mathsf{X}}_{l1})$ ought to be reformulated to allow for (ned) elimination.

$$\dfrac{\dfrac{\mathcal{R}, \underline{a} \in D_u, w : \forall x \phi, v : \phi(\underline{a}/x), \Gamma \Rightarrow \Delta}{\mathcal{R}, \underline{a} \in D_u, w : \forall x \phi, \Gamma \Rightarrow \Delta}\,(\forall^{\mathsf{X}}_{l1})}{\mathcal{R}, w : \forall x \phi, \Gamma \Rightarrow \Delta}\,(ned)$$

The issue in the above derivation is that due to the existence of $v : \phi(\underline{a}/x)$, which contains the parameter $\underline{a}$, (ned) is inapplicable to the top sequent. The solution to our problem then, is to absorb the (ned) inference, yielding a new rule of the form:

$$\dfrac{\mathcal{R}, \underline{a} \in D_u, w : \forall x \phi, v : \phi(\underline{a}/x), \Gamma \Rightarrow \Delta}{\mathcal{R}, w : \forall x \phi, \Gamma \Rightarrow \Delta}\,(\forall^{\mathsf{X}}_{l2})$$

We obtain the $(\exists^{\mathsf{X}}_{r2})$ rule in a similar fashion. Both of the reachability rules $(\exists^{\mathsf{X}}_{r2})$ and $(\forall^{\mathsf{X}}_{l2})$ are shown in Fig. 5.2.

At this stage, we have completed our analysis, and move on to proving that the conducive rules shown in Fig. 5.2 allow for the elimination of (ref), (tra), (nd), (cd), and (ned) in G3IntQ and G3IntQC. To simplify notation, the conducive rules are organized into two sets $\mathrm{R}(\mathsf{Q})$ and $\mathrm{R}(\mathsf{QC})$, shown in Fig. 5.5, with the former collecting the conducive rules for G3IntQ and the latter collecting the conducive rules for G3IntQC. Moreover, we use the symbol Q to index rules and side conditions that are relevant to refining the calculus G3IntQ and use the symbol QC to index rules and side conditions that are relevant to refining the calculus G3IntQC. When reference is made to a rule or side condition indexed with X, it is taken to represent both the rule or side condition indexed with Q and the rule or side condition indexed with QC (e.g. $(id^{\mathsf{X}}) \in \{(id^{\mathsf{Q}}), (id^{\mathsf{QC}})\}$. Last, we present a formal definition for the CFCST systems $S4$ and $S5$ for reference.

Definition 73 (The CFCST Systems $S4$ and $S5$)**.** *We define the CFCST systems $S4$ and $S5$ as follows:*

$$S4 := \{a \longrightarrow \varepsilon, a \longrightarrow aa, \overline{a} \longrightarrow \varepsilon, \overline{a} \longrightarrow \overline{a}\overline{a}\}$$

$$S5 := \{a \longrightarrow \varepsilon, a \longrightarrow \overline{a}a, \overline{a} \longrightarrow \varepsilon, \overline{a} \longrightarrow a\overline{a}\}$$

5.2 Refinement Part II: Structural Rule Elimination

In this section, we formally prove that the conducive rules shown in Fig. 5.2 allow for the refinement of $\mathsf{G3IntQ}$ and $\mathsf{G3IntQC}$, and ultimately, allow for the extraction of the nested calculi DIntQ and DIntQC for IntQ and IntQC, respectively (discussed in Sect. 5.4). Our first step is to show that the reachability and propagation rules (id^{X}), $(Pr_{\supset})$, $(\exists^{\mathsf{X}}_{r1})$, and $(\forall^{\mathsf{X}}_{l1})$ subsume the rules (id), $(\supset_{l})$, $(\exists_{r})$, and $(\forall_{l})$. Afterward, we show that the structural rules (*viz.* (ref), (tra), (nd), (cd), and (ned); see Fig. 3.3) are eliminable in the labelled calculi $\mathsf{G3IntQ}$ and $\mathsf{G3IntQC}$ when extended with the proper reachability and propagation rules (see Thm. 25 below), yielding the quasi-refined labelled calculi IntQL^{*} and IntQCL^{*} (defined in Def. 74 below). Last, we show that the quasi-refined labelled calculi IntQL^{*} and IntQCL^{*} are deductively equivalent to the labelled calculi $\mathsf{G3IntQ}$ and $\mathsf{G3IntQC}$ (see Thm. 26 below), and therefore inherit proof-theoretic properties from their parental labelled calculi (see Thm. 27 below). In the following section, we show how to rewrite the rules of the quasi-refined calculi IntQL^{*} and IntQCL^{*}, rendering domain atoms expendable, and producing the fully refined labelled calculi IntQL and IntQCL from which the nested calculi DIntQ and DIntQC may be easily extracted.

Lemma 38.

(i) The rule (id) *is an instance of* (id^{Q}) *and* (id^{QC})*.*

(ii) The $(\supset_{l})$ *rule is an instance of the* $(Pr_{\supset})$ *rule.*

(iii) The $(\exists_{r})$ *rule is an instance of the rules* $(\exists^{\mathsf{Q}}_{r1})$ *and* $(\exists^{\mathsf{QC}}_{r1})$*.*

(iv) The $(\forall_{l})$ *rule is an instance of the rules* $(\forall^{\mathsf{Q}}_{l1})$ *and* $(\forall^{\mathsf{QC}}_{l1})$*.*

Proof. We prove claim (i) below and defer the proofs of (ii)–(iv) to Appendix C (p. 211).

(i) We show that (id) is an instance of (id^{QC}) since showing that (id) is an instance of (id^{Q}) is similar. By Def. 73, we know that $\varepsilon \in L_{S5}(a)$ (due to the first production rule in $S5$). Moreover, by Def. 59, we know that the empty path $\lambda(w, w) = w$ always holds between a label and itself, and has the string $s_{\lambda}(w, w) = \varepsilon$. Therefore, $s_{\lambda}(w, w) \in L_{S5}(a)$.

In the inference below, observe that the relational atom $w \leq u$ implies the existence of a propagation path $\pi(w, u) := w, a, u$, where $s_{\pi}(w, u) = a$. Since $a \in L_{S4}(a)$, it follows that $s_{\pi}(w, u) \in L_{S4}(a)$. Therefore, by all that has been argued, we know that the side condition $\dagger_{1}(\mathsf{QC})$ of (id^{QC}) from Fig. 5.4 is indeed satisfied, and the following inference is an instance of (id^{QC}):

$$\dfrac{}{\mathcal{R}, w \leq u, \underline{a}_1 \in D_w, \ldots, \underline{a}_n \in D_w, \Gamma, w : p(\underline{\vec{a}}) \Rightarrow u : p(\underline{\vec{a}}), \Delta}\;(id^{\mathsf{QC}})$$
$$\overset{\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\;=}{\mathcal{R}, w \leq u, \underline{\vec{a}} \in D_w, \Gamma, w : p(\underline{\vec{a}}) \Rightarrow u : p(\underline{\vec{a}}), \Delta}$$

□

The above lemma confirms that our reachability and propagation rules are strengthened versions of the rules (id), $(\supset_l)$, $(\exists_r)$, and $(\forall_l)$ from which they were extracted. Additionally, the above lemma will assist us in proving the elimination of structural rules from $\mathsf{G3IntQ}$ and $\mathsf{G3IntQC}$ as less cases will need to be considered. We now show that all structural rules in our labelled calculi for first-order intuitionistic logics can be eliminated.

Lemma 39.

(i) The rule (ref) is eliminable in $\mathsf{G3IntQ} + \mathrm{R}(\mathsf{Q}) - \{(tra), (nd), (ned)\}$.

(ii) The rule (ref) is eliminable in $\mathsf{G3IntQC} + \mathrm{R}(\mathsf{QC}) - \{(tra), (nd), (cd), (ned)\}$.

Proof. We prove the result by induction on the height of the given derivation and show claim (ii) as it subsumes claim (i).

Base case. The $(\bot_l)$ case is simple to verify, and we need not consider the (id) case due to Lem. 38. Hence, we consider the (id^{QC}) case, which is resolved as shown below:

$$\dfrac{\dfrac{}{\mathcal{R}, v \leq v, \underline{a}_1 \in D_{u_1}, \ldots, \underline{a}_n \in D_{u_n}, \Gamma, w : p(\underline{\vec{a}}) \Rightarrow u : p(\underline{\vec{a}}), \Delta}\,(id^{\mathsf{QC}})}{\mathcal{R}, \underline{a}_1 \in D_{u_1}, \ldots, \underline{a}_n \in D_{u_n}, \Gamma, w : p(\underline{\vec{a}}) \Rightarrow u : p(\underline{\vec{a}}), \Delta}\,(ref) \quad \rightsquigarrow$$

$$\dfrac{}{\mathcal{R}, \underline{a}_1 \in D_{u_1}, \ldots, \underline{a}_n \in D_{u_n}, \Gamma, w : p(\underline{\vec{a}}) \Rightarrow u : p(\underline{\vec{a}}), \Delta}\,(id^{\mathsf{QC}})$$

If the relational atom $v \leq v$ that is active in (ref) occurs along one or more propagation paths $\pi_i(u_i, w)$ or along the propagation path $\pi(w, u)$, then by replacing each occurrence of the path v, a, v and $v, \overline{a}, v$ with the empty path $\lambda(v, v) = v$ in each such $\pi_i(u_i, w)$ and in $\pi(w, u)$, we obtain propagation paths $\pi'_i(u_i, w)$ and a propagation path $\pi'(w, u)$ that do not rely on the relational atom $v \leq v$. By assumption, we know that $s_{\pi_i}(u_i, w) \in L_{S5}(a)$ and $s_\pi(w, u) \in L_{S4}(a)$. By applying the production rules $a \longrightarrow \varepsilon, \overline{a} \longrightarrow \varepsilon \in S5$ and $a \longrightarrow \varepsilon, \overline{a} \longrightarrow \varepsilon \in S4$ to each a and $\overline{a}$ obtained from the path v, a, v and $v, \overline{a}, v$ (respectively) in $s_{\pi_i}(u_i, w)$ and $s_\pi(w, u)$, we obtain the strings $s_{\pi'_i}(u_i, w)$ and $s_{\pi'}(w, u)$, which, by the fact that they are derived from $s_{\pi_i}(u_i, w)$ and $s_\pi(w, u)$ using the production rules $a \longrightarrow \varepsilon$ and $\overline{a} \longrightarrow \varepsilon$ ensures that they are elements of $L_{S5}(a)$ and $L_{S4}(a)$, respectively.

Inductive step. With the exception of the $(Pr_\supset)$, $(\exists^{\mathsf{QC}}_{r1})$, and $(\forall^{\mathsf{QC}}_{l1})$ rules, all cases are handled by invoking IH and then applying the corresponding rule. (NB. By Lem. 38, we may omit consideration of the $(\supset_l)$, $(\exists_r)$, and $(\forall_l)$ rules.) We consider the $(Pr_\supset)$, $(\exists^{\mathsf{QC}}_{r1})$, and $(\forall^{\mathsf{QC}}_{l1})$ cases.

$(Pr_\supset)$. Let $\pi(w, u)$ be a propagation path in the premises of the top derivation below such that $s_\pi(w, u) \in L_{S4}(a)$. If the relational atom $v \leq v$ that is active in (ref) occurs along the propagation path $\pi(w, u)$, then by replacing each occurrence of v, a, v and $v, \overline{a}, v$ (obtained from the relational atom $v \leq v$) in $\pi(w, u)$ with the empty path $\lambda(v, v) = v$, we obtain a new propagation path $\pi'(w, u)$ that does not rely on the relational atom $v \leq v$. Since $a \longrightarrow \varepsilon, \overline{a} \longrightarrow \varepsilon \in S4$, we may apply the production rules to each a and $\overline{a}$ in $s_\pi(w, u)$ that was obtained from the path v, a, v and the path $v, \overline{a}, v$ (respectively), yielding the string $s_{\pi'}(w,u)$. Due to the fact that $s_{\pi}(w,u) \in L_{S4}(a)$ and $s_{\pi}(w,u) \twoheadrightarrow^{*}_{S4} s_{\pi'}(w,u)$, we know that $s_{\pi'}(w,u) \in L_{S4}(a)$, thus showing that the side condition of the $(Pr_{\supset})$ inference in the second derivation holds, allowing for the permutation to go through.

$$\dfrac{\dfrac{\mathcal{R}, v \leq v, w : \phi \supset \psi, \Gamma \Rightarrow \Delta, u : \phi \qquad \mathcal{R}, v \leq v, w : \phi \supset \psi, u : \psi, \Gamma \Rightarrow \Delta}{\mathcal{R}, v \leq v, w : \phi \supset \psi, \Gamma \Rightarrow \Delta}\,(Pr_{\supset})}{\mathcal{R}, \Gamma, w : \phi \supset \psi \Rightarrow \Delta}\,(ref) \qquad \rightsquigarrow$$

$$\dfrac{\dfrac{\mathcal{R}, v \leq v, , \Gamma, w : \phi \supset \psi \Rightarrow \Delta, u : \phi}{\mathcal{R}, \Gamma, w : \phi \supset \psi \Rightarrow \Delta, u : \phi}\,(ref) \qquad \dfrac{\mathcal{R}, v \leq v, , \Gamma, w : \phi \supset \psi, u : \psi \Rightarrow \Delta}{\mathcal{R}, \Gamma, w : \phi \supset \psi, u : \psi \Rightarrow \Delta}\,(ref)}{\mathcal{R}, \Gamma, w : \phi \supset \psi \Rightarrow \Delta}\,(Pr_{\supset})$$

$(\exists^{\mathsf{QC}}_{r1})$. Let $\pi(u,w)$ be a propagation path in the premise of the top left derivation below such that $s_{\pi}(u,w) \in L_{S5}(a)$. If the relational atom $v \leq v$ that is active in the (ref) inference occurs along the propagation path $\pi(u,w)$, then by replacing each occurrence of the path v, a, v and $v, \overline{a}, v$ (obtained from the relational atom $v \leq v$ and where $x \in \sum$) with the empty path $\lambda(v,v) = v$, we obtain a new propagation path $\pi'(u,w)$ that does not rely on the relational atom $v \leq v$. Because $a \longrightarrow \varepsilon, \overline{a} \longrightarrow \varepsilon \in L_{S5}(a)$, we can derive $s_{\pi'}(u,w)$ from $s_{\pi}(u,w)$ by applying the production rules to each a and $\overline{a}$ in $s_{\pi}(u,w)$ obtained from the paths v, a, v and $v, \overline{a}, v$, respectively. It follows that $s_{\pi'}(u,w) \in L_{S5}(a)$, implying that we may apply $(\exists^{\mathsf{QC}}_{r1})$ after applying (ref).

$$\dfrac{\dfrac{\mathcal{R}, v \leq v, \underline{a} \in D_u, \Gamma \Rightarrow \Delta, w : \phi(\underline{a}/x), w : \exists x \phi}{\mathcal{R}, v \leq v, \underline{a} \in D_u, \Gamma \Rightarrow \Delta, w : \exists x \phi}\,(\exists^{\mathsf{QC}}_{r1})}{\mathcal{R}, \underline{a} \in D_u, \Gamma \Rightarrow \Delta, w : \exists x \phi}\,(ref) \qquad \rightsquigarrow$$

$$\dfrac{\dfrac{\mathcal{R}, v \leq v, \underline{a} \in D_u, \Gamma \Rightarrow \Delta, w : \phi(\underline{a}/x), w : \exists x \phi}{\mathcal{R}, \underline{a} \in D_u, \Gamma \Rightarrow \Delta, w : \phi(\underline{a}/x), w : \exists x \phi}\,(ref)}{\mathcal{R}, \underline{a} \in D_u, \Gamma \Rightarrow \Delta, w : \exists x \phi}\,(\exists^{\mathsf{QC}}_{r1})$$

$(\forall^{\mathsf{QC}}_{l1})$. Let $\pi_1(u,v)$ be the propagation path in the premise of the top left derivation below such that $s_{\pi_1}(u,v) \in L_{S5}(a)$ and $\pi_2(w,v)$ be the propagation path in the premise of the top left derivation below such that $s_{\pi_2}(w,v) \in L_{S4}(a)$. If the relational atom $z \leq z$ that is active in the (ref) inference occurs along the propagation path $\pi_1(u,v)$ or $\pi_2(w,v)$, then by replacing each path z, a, z and $z, \overline{a}, z$ with the empty path $\lambda(z,z) = z$, we obtain new propagation paths $\pi'_1(u,v)$ and $\pi'_2(w,v)$ that do not rely on the relational atom $z \leq z$. Since $a \longrightarrow \varepsilon, \overline{a} \longrightarrow \varepsilon \in S5$ and $a \longrightarrow \varepsilon, \overline{a} \longrightarrow \varepsilon \in S4$ (Def. 73), we may apply the production rules to each a and $\overline{a}$ in $s_{\pi_1}(u,v)$ and $s_{\pi_2}(w,v)$ obtained from the paths z, a, z and $z, \overline{a}, z$ (respectively), letting us derive the strings $s_{\pi'_1}(u,v)$ and $s_{\pi'_2}(w,v)$. It follows that $s_{\pi'_1}(u,v) \in L_{S5}(a)$ and $s_{\pi'_2}(w,v) \in L_{S4}(a)$, implying that we may apply (ref) prior to $(\forall^{\mathsf{QC}}_{l1})$.

$$\cfrac{\cfrac{\mathcal{R}, z \leq z, \underline{a} \in D_u, w : \forall x\phi, v : \phi(\underline{a}/x), \Gamma \Rightarrow \Delta}{\mathcal{R}, z \leq z, \underline{a} \in D_u, w : \forall x\phi, \Gamma \Rightarrow \Delta}\ (\forall^{\mathsf{QC}}_{l1})}{\mathcal{R}, \underline{a} \in D_u, w : \forall x\phi, \Gamma \Rightarrow \Delta}\ (ref) \quad \rightsquigarrow$$

$$\cfrac{\cfrac{\mathcal{R}, z \leq z, \underline{a} \in D_u, w : \forall x\phi, v : \phi(\underline{a}/x), \Gamma \Rightarrow \Delta}{\mathcal{R}, \underline{a} \in D_u, w : \forall x\phi, v : \phi(\underline{a}/x), \Gamma \Rightarrow \Delta}\ (ref)}{\mathcal{R}, \underline{a} \in D_u, w : \forall x\phi, \Gamma \Rightarrow \Delta}\ (\forall^{\mathsf{QC}}_{l1})$$

□

Lemma 40.

(i) The rule (tra) *is eliminable in* $\mathsf{G3IntQ} + \mathrm{R}(\mathsf{Q}) - \{(ref), (nd), (ned)\}$.

(ii) The rule (tra) *is eliminable in* $\mathsf{G3IntQC} + \mathrm{R}(\mathsf{QC}) - \{(ref), (nd), (cd), (ned)\}$.

Proof. We defer the proof to Appendix C (p. 211). □

Lemma 41.

(i) The rule (nd) *is eliminable in* $\mathsf{G3IntQ} + \mathrm{R}(\mathsf{Q}) - \{(ref), (tra), (ned)\}$.

(ii) The rule (nd) *is eliminable in* $\mathsf{G3IntQC} + \mathrm{R}(\mathsf{QC}) - \{(ref), (tra), (cd), (ned)\}$.

Proof. We prove the result by induction on the height of the given derivation and only show claim (ii) as it subsumes claim (i).

Base case. The $(\bot_l)$ case is straightforward and we need not consider the (id) case by Lem. 38. If we apply the (nd) rule to (id^{QC}), and none of the principal domain atoms of (id^{QC}) are active in the application of (nd), then the conclusion is an instance of (id^{QC}). Let us suppose then that one of the principal domain atoms of (id^{QC}) is active in the application of (nd). We assume w.l.o.g. that the domain atom is $\underline{a}_1 \in D_{u_1}$, and argue how the case is resolved below:

$$\cfrac{\cfrac{}{\mathcal{R}, v \leq u_i, \underline{a}_1 \in D_v, \underline{a}_1 \in D_{u_1}, \ldots, \underline{a}_n \in D_{u_n}, \Gamma, w : p(\vec{\underline{a}}) \Rightarrow u : p(\vec{\underline{a}}), \Delta}\ (id^{\mathsf{QC}})}{\mathcal{R}, v \leq u_i, \underline{a}_1 \in D_v, \ldots, \underline{a}_n \in D_{u_n}, \Gamma, w : p(\vec{\underline{a}}) \Rightarrow u : p(\vec{\underline{a}}), \Delta}\ (nd) \quad \rightsquigarrow$$

$$\cfrac{}{\mathcal{R}, v \leq u_i, \underline{a}_1 \in D_v, \ldots, \underline{a}_n \in D_{u_n}, \Gamma, w : p(\vec{\underline{a}}) \Rightarrow u : p(\vec{\underline{a}}), \Delta}\ (id^{\mathsf{QC}})$$

By the side condition of (id^{QC}), we know that there exists a propagation path $\pi(u_1, w)$ in the premise of the top left derivation such that $s_\pi(u_1, w) \in L_{S5}(a)$. By prefixing the propagation path $\pi(u_1, w)$ with v, a, u_i (obtained from the relational atom $v \leq u_i$), we obtain a propagation path $\pi'(v, w)$. If $s_\pi(u_1, w) \neq \varepsilon$, then $\pi(u_1, w)$ is either of the form $u_1, a, \pi_0(z, w)$ or $u_1, \overline{a}, \pi_0(z, w)$ for some label z occurring in the premise of the top derivation above. As explained in Lem. 40 above, $a \twoheadrightarrow^*_{S5} aa$, and by Def. 73, $\overline{a} \longrightarrow a\overline{a} \in S5$. If the leading character of $s_\pi(u_1, w)$ is a, then applying the former derivation to this occurrence of a gives the string $s_{\pi'}(v,w)$, and if the leading character of $s_{\pi}(u_1,w)$ is $\overline{a}$, then applying the aforementioned production rule gives the string $s_{\pi'}(v,w)$. In either case, $s_{\pi}(v,w) \in L_{S5}(a)$. If $s_{\pi}(u_1,w) = \varepsilon$, then $s_{\pi'}(v,w) = a$, and since $a \in L_{S5}(a)$, it follows that $s_{\pi'}(v,w) \in L_{S5}(a)$. Hence, the conclusion of the top left derivation is an instance of (id^{QC}) in its own right.

Inductive step. By Lem. 38, we need not consider the $(\supset_l)$, $(\exists_r)$, or $(\forall_l)$ cases. With the exception of the $(\exists^{\mathsf{QC}}_{r1})$ and $(\forall^{\mathsf{QC}}_{l1})$ cases, all remaining cases are resolved by invoking IH and then applying the corresponding rule. We consider the $(\exists^{\mathsf{QC}}_{r1})$ case as the $(\forall^{\mathsf{QC}}_{l1})$ case is similar.

$(\exists^{\mathsf{QC}}_{r1})$. If the domain atom $\underline{a} \in D_u$ deleted via the (nd) inference in the top left proof below is not active in the $(\exists^{\mathsf{QC}}_{r1})$ inference, then the two rules freely permute. Let us suppose then that the domain atom $\underline{a} \in D_u$ deleted via the (nd) inference is active in the $(\exists^{\mathsf{QC}}_{r1})$ inference. By the side condition on $(\exists^{\mathsf{QC}}_{r1})$, we know that there exists a propagation path $\pi(u,w)$ such that $s_{\pi}(u,w) \in L_{S5}(a)$. By prefixing the propagation path $\pi(u,w)$ with the path v, a, u obtained from the relational atom $v \leq u$, we obtain a new propagation path $\pi'(v,w)$. If $s_{\pi}(u,w) \neq \varepsilon$, then $\pi(u,w)$ is either of the form $u, a, \pi_0(z,w)$ or $u, \overline{a}, \pi_0(z,w)$ for some label z occurring in the premise of the top derivation below. In Lem. 40 above, we found that $a \twoheadrightarrow^*_{S5} aa$, and by Def. 73, $\overline{a} \longrightarrow a\overline{a} \in S5$. If the leading character of $s_{\pi}(u,w)$ is a, then applying the former derivation to this occurrence of a gives the string $s_{\pi'}(v,w)$, and if the leading character of $s_{\pi}(u,w)$ is $\overline{a}$, then applying the former production rule gives the string $s_{\pi'}(v,w)$. Hence, regardless of which case holds $s_{\pi'}(v,w) \in L_{S5}(a)$. If, on the other hand, $s_{\pi}(u,w) = \varepsilon$, then $s_{\pi'}(v,w) = a$, and since $a \in L_{S5}(a)$, we have that $s_{\pi'}(v,w) \in L_{S5}(a)$. Hence, the (nd) rule may be applied before the $(\exists^{\mathsf{QC}}_{r1})$ rule, showing that the two rules may be permuted.

$$\cfrac{\cfrac{\mathcal{R}, v \leq u, \underline{a} \in D_v, \underline{a} \in D_u, \Gamma \Rightarrow \Delta, w : \phi(\underline{a}/x), w : \exists x \phi}{\mathcal{R}, v \leq u, \underline{a} \in D_v, \underline{a} \in D_u, \Gamma \Rightarrow \Delta, w : \exists x \phi}\,(\exists^{\mathsf{QC}}_{r1})}{\mathcal{R}, v \leq u, \underline{a} \in D_v, \Gamma \Rightarrow \Delta, w : \exists x \phi}\,(nd) \quad \rightsquigarrow$$

$$\cfrac{\cfrac{\mathcal{R}, v \leq u, \underline{a} \in D_v, \underline{a} \in D_u, \Gamma \Rightarrow \Delta, w : \phi(\underline{a}/x), w : \exists x \phi}{\mathcal{R}, v \leq u, \underline{a} \in D_v, \Gamma \Rightarrow \Delta, w : \phi(\underline{a}/x), w : \exists x \phi}\,(nd)}{\mathcal{R}, v \leq u, \underline{a} \in D_v, \Gamma \Rightarrow \Delta, w : \phi(\underline{a}/x)}\,(\exists^{\mathsf{QC}}_{r1})$$

□

Lemma 42.

(i) The rule (ned) is eliminable in $\mathsf{G3IntQ} + \mathrm{R}(\mathsf{Q}) - \{(ref), (tra), (nd)\}$.

(ii) The rule (ned) is eliminable in $\mathsf{G3IntQC} + \mathrm{R}(\mathsf{QC}) - \{(ref), (tra), (nd), (cd)\}$.

Proof. We prove the result by induction on the height of the given derivation and show (ii) as it subsumes (i).

Base case. Any application of (ned) to $(\bot_l)$, (id), or (id^{QC}) yields another instance of rule.

Inductive step. By Lem. 38, we need not consider the $(\supset_l)$, $(\exists_r)$, or $(\forall_l)$ cases. With the exception of the $(\exists^{\mathsf{QC}}_{r1})$ and $(\forall^{\mathsf{QC}}_{l1})$ cases, all remaining cases are resolved by applying IH and then the corresponding rule. We show the non-trivial $(\exists^{\mathsf{QC}}_{r1})$ and $(\forall^{\mathsf{QC}}_{l1})$ cases below:

$$\dfrac{\dfrac{\mathcal{R}, \underline{a} \in D_u, \Gamma \Rightarrow \Delta, w : \phi(\underline{a}/x), w : \exists x \phi}{\mathcal{R}, \underline{a} \in D_u, \Gamma \Rightarrow \Delta, w : \exists x \phi}\,(\exists^{\mathsf{QC}}_{r1})}{\mathcal{R}, \Gamma \Rightarrow \Delta, w : \exists x \phi}\,(ned) \quad \rightsquigarrow$$

$$\dfrac{\mathcal{R}, \underline{a} \in D_u, \Gamma \Rightarrow \Delta, w : \phi(\underline{a}/x), w : \exists x \phi}{\mathcal{R}, \Gamma \Rightarrow \Delta, w : \exists x \phi}\,(\exists^{\mathsf{QC}}_{r2})$$

$$\dfrac{\dfrac{\mathcal{R}, \underline{a} \in D_u, w : \phi(\underline{a}/x), w : \forall x \phi, \Gamma \Rightarrow \Delta}{\mathcal{R}, \underline{a} \in D_u, w : \forall x \phi, \Gamma \Rightarrow \Delta}\,(\forall^{\mathsf{QC}}_{l1})}{\mathcal{R}, w : \forall x \phi, \Gamma \Rightarrow \Delta}\,(ned) \quad \rightsquigarrow$$

$$\dfrac{\mathcal{R}, \underline{a} \in D_u, w : \phi(\underline{a}/x), w : \forall x \phi, \Gamma \Rightarrow \Delta}{\mathcal{R}, w : \forall x \phi, \Gamma \Rightarrow \Delta}\,(\forall^{\mathsf{QC}}_{l1})$$

□

Lemma 43. *The rule (cd) is eliminable in* $\mathsf{G3IntQC}+\mathrm{R}(\mathsf{QC})-\{(ref),(tra),(nd),(ned)\}$.

Proof. The proof is similar to the proof of Lem. 41. □

Theorem 25.

(i) The rules $\{(id), (\supset_l), (\exists_r), (\forall_l), (ref), (tra), (nd), (ned)\}$ *are admissible in the calculus* $\mathsf{G3IntQ} + \mathrm{R}(\mathsf{Q})$.

(ii) The rules $\{(id), (\supset_l), (\exists_r), (\forall_l), (ref), (tra), (nd), (ned), (cd)\}$ *are admissible in the calculus* $\mathsf{G3IntQC} + \mathrm{R}(\mathsf{QC})$.

Proof. Follows from Lem. 38 – 43. □

We have formally confirmed that our conducive rules $\mathrm{R}(\mathsf{Q})$ and $\mathrm{R}(\mathsf{QC})$ are sufficient to allow for structural rule elimination in $\mathsf{G3IntQ}$ and $\mathsf{G3IntQC}$, respectively. Removing all structural rules, along with all unnecessary rules (e.g. $(\supset_l)$ which is subsumed by $(Pr_{\supset})$), from $\mathsf{G3IntQ}$ and $\mathsf{G3IntQC}$ gives us our quasi-refined labelled calculi, defined below:

Definition 74. *We define the quasi-refined labelled calculi* IntQL^* *and* IntQCL^* *as follows:*

$$\mathsf{IntQL}^* := \{(id^{\mathsf{Q}}), (\bot_l), (\vee_l), (\vee_r), (\wedge_l), (\wedge_r), (Pr_{\supset}), (\supset_r), (\exists^{\mathsf{Q}}_{r1}), (\exists^{\mathsf{Q}}_{r2}), (\forall^{\mathsf{Q}}_{l1}), (\forall^{\mathsf{Q}}_{l2})\}$$

$$\mathsf{IntQCL}^* := \{(id^{\mathsf{QC}}), (\bot_l), (\vee_l), (\vee_r), (\wedge_l), (\wedge_r), (Pr_{\supset}), (\supset_r), (\exists^{\mathsf{QC}}_{r1}), (\exists^{\mathsf{QC}}_{r2}), (\forall^{\mathsf{QC}}_{l1}), (\forall^{\mathsf{QC}}_{l2})\}$$

We could, at this stage, directly prove that our quasi-refined labelled calculi IntQL^* and IntQCL^* possess desirable proof-theoretic properties, as we did for $\mathsf{G3IntQ}$ and $\mathsf{G3IntQC}$ in Sect. 3.3 (i.e. without relying on proof transformations). However, we will take a separate route—similar to what was done in Sect. 4.1 for grammar logics—and show that proofs in our quasi-refined labelled calculi IntQL^* and IntQCL^* can be algorithmically transformed into proofs in $\mathsf{G3IntQ}$ and $\mathsf{G3IntQC}$ (resp.), and vice-versa. This approach shows that our quasi-refined labelled calculi inherit proof-theoretic properties from their parental labelled calculi, and also establishes a relationship between proofs within the different settings. To accomplish this aim, we first show the following lemma:

Lemma 44. *Let $\Lambda := \mathcal{R}, \Gamma \Rightarrow \Delta$. Suppose $\mathcal{R}, w \leq u, \Gamma \Rightarrow \Delta$ is derivable in* $\mathsf{G3IntQ}$ *or* $\mathsf{G3IntQC}$*, and that $\pi(w,u)$ is a propagation path (potentially empty) in $PG(\Lambda)$ such that $s_{\pi}(w,u) \in L_{S4}(a)$. Then, $\mathcal{R}, \Gamma \Rightarrow \Delta$ is derivable in* $\mathsf{G3IntQ}$ *or* $\mathsf{G3IntQC}$*, respectively.*

Proof. We prove the result for $\mathsf{G3IntQ}$ as the proof for $\mathsf{G3IntQC}$ is similar. Since $s_{\pi}(w,u) \in L_{S4}(a)$, we know that $a \twoheadrightarrow^{*}_{S4} s_{\pi}(w,u)$. The lemma is shown by induction on the length (Def. 9) of the derivation $a \twoheadrightarrow^{*}_{S4} s_{\pi}(w,u)$.

Base case. For the base case, we consider (i) a derivation of length 0, meaning that $a \twoheadrightarrow^{*}_{S} a$, and (ii) a derivation of length 1, meaning that $a \longrightarrow s_{\pi}(w,u) \in S4$, and therefore, either (ii.1) $a \longrightarrow s_{\pi}(w,u) = a \longrightarrow \varepsilon$, or (ii.2) $a \longrightarrow s_{\pi}(w,u) = a \longrightarrow aa$. (NB. It cannot be the case that $a \longrightarrow s_{\pi}(w,u) \in S4$ is the production rule $\overline{a} \longrightarrow \varepsilon$ or $\overline{a} \longrightarrow \overline{aa}$ since both have the character $\overline{a}$ as their head.) Cases (i), (ii.1), and (ii.2) are respectively shown below, where $\mathcal{R} := \mathcal{R}', w \leq u$ in case (i) and $\mathcal{R} := \mathcal{R}', w \leq u, u \leq v$ in case (ii.2).

$$\cfrac{\Pi_1}{\mathcal{R}', w \leq u, w \leq u, \Gamma \Rightarrow \Delta} \quad \rightsquigarrow \quad \cfrac{\cfrac{\Pi_1}{\mathcal{R}', w \leq u, w \leq u, \Gamma \Rightarrow \Delta}}{\mathcal{R}', w \leq u, \Gamma \Rightarrow \Delta}\ (ctr_{\mathcal{R}})$$

$$\cfrac{\Pi_2}{\mathcal{R}, w \leq w, \Gamma \Rightarrow \Delta} \quad \rightsquigarrow \quad \cfrac{\cfrac{\Pi_2}{\mathcal{R}, w \leq w, \Gamma \Rightarrow \Delta}}{\mathcal{R}, \Gamma \Rightarrow \Delta}\ (ref)$$

$$\cfrac{\Pi_3}{\mathcal{R}', w \leq u, u \leq v, w \leq v, \Gamma \Rightarrow \Delta} \quad \rightsquigarrow \quad \cfrac{\cfrac{\Pi_2}{\mathcal{R}', w \leq u, u \leq v, w \leq v, \Gamma \Rightarrow \Delta}}{\mathcal{R}', w \leq u, u \leq v, \Gamma \Rightarrow \Delta}\ (tra)$$

Inductive step. Let $\Lambda := \mathcal{R}, \Gamma \Rightarrow \Delta$ and assume that we have a proof Π of $\mathcal{R}, w \leq u, \Gamma \Rightarrow \Delta$. Suppose our derivation $a \twoheadrightarrow^{*}_{S4} s_{\pi}(w,u)$ is of length $n+1$. Recall that the length of a derivation is the minimal number of one-step derivations necessary to derive the output string from the input string in the CFCST system (Def. 9). Since we are making use of the CFCST system $S4$, it must be the case that our derivation consists of a sequence of production rules of the form $a \longrightarrow aa$ starting with the forward character $a \in \sum^{+}$. Also, because our derivation is of length $n+1$, we know that it consists of a derivation

$a \twoheadrightarrow_{S4}^{*} t$ of length n followed by a one-step derivation $t \twoheadrightarrow_{S4} s_{\pi}(w,u)$. Hence, there exist strings $r_0, r_1 \in (\sum^{+})^{*}$ such that $t = r_0 a r_1$ and $r_0 a a r_1 = s_{\pi}(w,u)$. This implies the existence of a propagation path of the form $\pi_{r_0}(w, v_1), a, z, a, \pi_{r_1}(v_2, u)$ in $PG(\Lambda)$, where $v_1, v_2, z \in \text{Lab}(\Lambda)$. It follows that $\mathcal{R}$ must be of the form $\mathcal{R}', v_1 \leq z, z \leq v_2$. Using this fact, we derive the desired conclusion as follows:

$$\begin{array}{cl}
\Pi & \\
\hline
\mathcal{R}', v_1 \leq z, z \leq v_2, w \leq u, \Gamma \Rightarrow \Delta & \\
\hdashline
\mathcal{R}', v_1 \leq z, z \leq v_2, v_1 \leq v_2, w \leq u, \Gamma \Rightarrow \Delta & (wk) \\
\hdashline
\mathcal{R}', v_1 \leq z, z \leq v_2, v_1 \leq v_2, \Gamma \Rightarrow \Delta & \text{IH} \\
\hline
\mathcal{R}', v_1 \leq z, z \leq v_2, \Gamma \Rightarrow \Delta & (tra) \\
\cdots\cdots\cdots\cdots\cdots\cdots & = \\
\mathcal{R}, \Gamma \Rightarrow \Delta &
\end{array}$$

We know that a propagation path of the form $\pi_{r_0}(w, v_1), a, z, a, \pi_{r_1}(v_2, u)$ exists in $\mathcal{R}$, so after applying (wk) to introduce the relational atom $v_1 \leq v_2$, we know that a propagation path of the form $\pi'(w,u) := \pi_{r_0}(w, v_1), a, \pi_{r_1}(v_2, u)$ exists in $\mathcal{R}, v_1 \leq v_2$ (since we can take a detour through the path v_1, a, v_w instead of v_1, a, z, a, v_2). Observe that the string $s_{\pi'}(w,u) = t$ has a derivation of length n, which means that we may invoke IH to delete the relational atom $w \leq u$. The remaining steps of the proof are self-explanatory, and thus, we have resolved the inductive step. □

Theorem 26. *Let* $\mathsf{X} \in \{\mathsf{Q}, \mathsf{QC}\}$*. Every proof of a labelled sequent* Λ *in* IntXL^{*} *can be algorithmically transformed into a proof in* $\mathsf{G3IntX}$*.*

Proof. We prove the result for IntQCL^{*} and $\mathsf{G3IntQC}$ as the proof for IntQL^{*} and $\mathsf{G3IntQ}$ is similar. All rules common to both IntQCL and $\mathsf{G3IntQC}$ straightforwardly translate, so we only show how to translate each rule from $\mathrm{R}(\mathsf{QC})$, and consider each in turn below.

(id^{QC}). Let us consider an instance of (id^{QC}):

$$\dfrac{}{\mathcal{R}, \underline{a}_1 \in D_{u_1}, \ldots, \underline{a}_n \in D_{u_n}, \Gamma, w : p(\vec{\underline{a}}) \Rightarrow u : p(\vec{\underline{a}}), \Delta} \ (id^{\mathsf{QC}})$$

By the side condition imposed on (id^{QC}), we know that there exist propagation paths $\pi_i(u_i, w)$ for $i \in \{1, \ldots, n\}$ such that $s_{\pi_i}(u_i, w) \in L_{S5}(a)$ and a propagation path $\pi(w,u)$ such that $s_{\pi}(w,u) \in L_{S4}(a)$. We may assume w.l.o.g. that the propagation paths $\pi_i(u_i, w)$ are *minimal*, that is, no propagation paths $\pi_i'(u_i, w)$ exist such that $|s_{\pi_i'}(u_i, w)| < |s_{\pi_i}(u_i, w)|$. Since each propagation path $\pi_i(u_i, w)$ is minimal, we know that no label occurs more than once in each propagation path, which is important in applying the (nd) and (cd) rules below. For $i \in \{1, \ldots, n\}$, we let $\underline{a}_i \in D_{\pi_i} := \underline{a}_i \in D_{u_i}, \underline{a}_i \in D_{v_1^i}, \ldots, \underline{a}_i \in D_{v_{k_i}^i}$, where $\pi_i(u_i, w) := u_i, x_0^i, v_1^i, \ldots, v_{k_i}^i, x_{k_i}^i, w$ and $x_j^i \in \sum = \{a, \overline{a}\}$ with $j \in \{1, \ldots, k_i\}$. Let $n_i(a)$ and $n_i(\overline{a})$ be the respective number of occurrences of a and $\overline{a}$ in the string $s_{\pi_i}(u_i, w)$.

To prove (id^{QC}) admissible in $\mathsf{G3IntQC}$, we first weaken in the domain atoms $\underline{a}_i \in D_{\pi_i}$ for each $i \in \{1, \ldots, n\}$ as shown in the derivation below. Then, we perform a *domain atom* *deletion procedure* and successively delete domain atoms along the propagation paths $\pi_i(u_i, w)$ by first deleting the domain atom $\underline{a}_i \in D_w$ at w and working our way backward toward the initial node of the path u_i through applications of (nd) and (cd). We let $(d_i) := (nd) \times n_i(a) + (cd) \times n_i(\overline{a})$ represent this sequence of (nd) and (cd) applications; note that by the definition of $n_i(a)$ and $n_i(\overline{a})$, there will be $n_i(a)$ applications of (nd) and $n_i(\overline{a})$ applications of (cd). Also, the minimality of each propagation path $\pi_i(u_i, w)$ is significant here as it implies the non-existence of repetitions of labels in $\pi_i(u_i, w)$, which, if present, could potentially block the deletion of domain atoms within the domain atom deletion procedure described above, thus causing the procedure to halt prematurely without outputting the desired result.[6] Last, we invoke Lem. 44 to delete the relational atom $w \leq u$, which is permitted due to the side condition of (id^{QC}).

$$
\begin{array}{cl}
\hline
\mathcal{R}, w \leq u, \underline{a}_1 \in D_w, \ldots, \underline{a}_n \in D_w, \Gamma, w : p(\underline{\vec{a}}) \Rightarrow u : p(\underline{\vec{a}}), \Delta & (id) \\
\hdashline
\mathcal{R}, w \leq u, \underline{a}_1 \in D_{\pi_1}, \ldots, \underline{a}_n \in D_{\pi_n}, \underline{a}_1 \in D_w, \ldots, \underline{a}_n \in D_w, \Gamma, w : p(\underline{\vec{a}}) \Rightarrow u : p(\underline{\vec{a}}), \Delta & (wk) \\
\hline
\mathcal{R}, w \leq u, \underline{a}_1 \in D_{u_1}, \ldots, \underline{a}_n \in D_{\pi_n}, \underline{a}_2 \in D_w, \ldots, \underline{a}_n \in D_w, \Gamma, w : p(\underline{\vec{a}}) \Rightarrow u : p(\underline{\vec{a}}), \Delta & (d_1) \\
\hline
\vdots & \\
\hline
\mathcal{R}, w \leq u, \underline{a}_1 \in D_{u_1}, \ldots, \underline{a}_n \in D_{\pi_n}, \underline{a}_n \in D_w, \Gamma, w : p(\underline{\vec{a}}) \Rightarrow u : p(\underline{\vec{a}}), \Delta & \\
\hline
\mathcal{R}, w \leq u, \underline{a}_1 \in D_{u_1}, \ldots, \underline{a}_n \in D_{u_n}, \Gamma, w : p(\underline{\vec{a}}) \Rightarrow u : p(\underline{\vec{a}}), \Delta & (d_n) \\
\hdashline
\mathcal{R}, \underline{a}_1 \in D_{u_1}, \ldots, \underline{a}_n \in D_{u_n}, \Gamma, w : p(\underline{\vec{a}}) \Rightarrow u : p(\underline{\vec{a}}), \Delta & \text{Lem. } 44
\end{array}
$$

$(Pr_{\supset})$. Suppose that the side condition of a $(Pr_{\supset})$ instance holds, i.e. a propagation path $\pi(w, u)$ exists such that $s_{\pi}(w, u) \in L_{S4}(a)$. Using this fact, we may invoke Lem. 44 below, allowing us to obtain our desired conclusion:

$$
\dfrac{
\dfrac{\mathcal{R}, w : \phi \supset \psi, \Gamma \Rightarrow \Delta, u : \phi}{\mathcal{R}, w \leq u, w : \phi \supset \psi, \Gamma \Rightarrow \Delta, u : \phi}\ (wk)
\qquad
\dfrac{\mathcal{R}, w : \phi \supset \psi, u : \psi, \Gamma \Rightarrow \Delta}{\mathcal{R}, w \leq u, w : \phi \supset \psi, u : \psi, \Gamma \Rightarrow \Delta}\ (wk)
}{
\dfrac{\mathcal{R}, w \leq u, w : \phi \supset \psi, \Gamma \Rightarrow \Delta}{\mathcal{R}, w : \phi \supset \psi, \Gamma \Rightarrow \Delta}\ \text{Lem. } 44
}\ (\supset_l)
$$

$(\exists^{\mathsf{QC}}_{r1})$. Suppose that the side condition of an $(\exists^{\mathsf{QC}}_{r1})$ inference holds, that is, there exists a propagation path $\pi(u, w)$ such that $s_{\pi}(u, w) \in L_{S5}(a)$. We assume w.l.o.g. that $\pi(u, w)$ is

[6] To demonstrate the significance of minimality, suppose for the sake of argument that a path v, a, v', a, v (with a repetition of v) occurs within a propagation path $\pi_i(u_i, w)$. Then, at some stage of the domain atom deletion procedure, we will have derived a labelled sequent of the form $\Lambda := \mathcal{R}', v \leq v', v' \leq v, \underline{a}_i \in D_{v'}, \underline{a}_i \in D_v, \Gamma' \Rightarrow \Delta'$. Since we delete domain atoms in reverse from the end node w toward the initial node u_i of the propagation path $\pi_i(u_i, w)$, the rule (nd) will be applied to Λ deleting the domain atom $\underline{a}_i \in D_v$, and yielding the labelled sequent $\Lambda' := \mathcal{R}', v \leq v', v' \leq v, \underline{a}_i \in D_{v'}, \Gamma' \Rightarrow \Delta'$ due to the path v', a, v occurring in $\pi_i(u_i, w)$. At this point, following the domain atom deletion procedure, we should apply (nd) to delete the domain atom $\underline{a}_i \in D_{v'}$ from Λ' (as the path v', a, v of v, a, v', a, v in $\pi_i(u_i, w)$ was just processed, and so, we must process the path v, a, v' of v, a, v', a, v in $\pi_i(u_i, w)$), but since we are not guaranteed the existence of a domain atom $\underline{a}_i \in D_v$ in Λ' (as the only domain atom of such a shape and that was guaranteed to exist was just deleted), we are not necessarily permitted to apply the (nd) rule. Hence, the procedure halts without the desired result being obtained.

minimal (as in the base case), let $\pi(u, w) := u, x_0, v_1, \ldots v_n, x_n, w$ (with $x_i \in \sum = \{a, \overline{a}\}$ for $i \in \{1, \ldots, n\}$), and let $n(a)$ and $n(\overline{a})$ be the number of occurrences of a and $\overline{a}$ in $s_\pi(u, w)$, respectively. To prove the rule admissible in $\mathsf{G3IntQC}$, we first apply hp-admissibility of (wk) to the premise of the rule (as shown below) and add domain atoms along the path $\pi(u, w)$. Then, we apply $(\exists_r)$ a single time, and last apply the (nd) and (cd) rules to delete the weakened in domain atoms by starting at w and working backward toward u in $\pi(u, w)$. The (nd) rule will be applied $n(a)$ times and the (cd) rule will be applied $n(\overline{a})$ times, and we use $(d) := (nd) \times n(a) + (cd) \times n(\overline{a})$ to denote these inferences.

$$
\begin{array}{cl}
\mathcal{R}, \underline{a} \in D_u, \Gamma \Rightarrow \Delta, w : \phi(\underline{a}/x), w : \exists x\phi & \\
\hdashline
\mathcal{R}, \underline{a} \in D_u, \underline{a} \in D_{v_1}, \ldots, \underline{a} \in D_{v_n}, \underline{a} \in D_w, \Gamma \Rightarrow \Delta, w : \phi(\underline{a}/x), w : \exists x\phi & (wk) \\
\hline
\mathcal{R}, \underline{a} \in D_u, \underline{a} \in D_{v_1}, \ldots, \underline{a} \in D_{v_n}, \underline{a} \in D_w, \Gamma \Rightarrow \Delta, w : \exists x\phi & (\exists_r) \\
\hline
\mathcal{R}, \underline{a} \in D_u, \Gamma \Rightarrow \Delta, w : \exists x\phi & (d)
\end{array}
$$

$(\forall^{\mathsf{QC}}_{l1})$. Similar to the proof of $(\exists^{\mathsf{QC}}_{r1})$ above.

$(\exists^{\mathsf{QC}}_{r2})$. Suppose that the side condition holds, i.e. there exists a propagation path $\pi(u, w)$ such that $s_\pi(u, w) \in L_{S5}(a)$. We assume w.l.o.g. that $\pi(u, w)$ is minimal (as in the base case), let $\pi(u, w) := u, x_0, v_1, \ldots, v_n, x_n, w$ (with $x_i \in \sum = \{a, \overline{a}\}$ for $i \in \{1, \ldots, n\}$), and let $n(a)$ and $n(\overline{a})$ be the number of occurrences of a and $\overline{a}$ in $s_\pi(u, w)$, respectively. To prove $(\exists^{\mathsf{QC}}_{r2})$ admissible, we first apply hp-admissibility of (wk) to add in domain atoms along the propagation path $\pi(u, w)$, and use the domain atom $\underline{a} \in D_w$ to apply $(\exists_r)$. After this, we successively delete each of the weakened in domain atoms by applying (nd) and (cd) starting at w and working our way backward toward the label u in the propagation path $\pi(u, w)$. The rule (nd) will be applied $n(a)$ times and (cd) will be applied $n(\overline{a})$ times; we let $(d) := (nd) \times n(a) + (cd) \times n(\overline{a})$ represent the successive applications of (nd) and (cd) that are applied to delete the weakened in domain atoms. Last, we apply (ned) to derive the desired conclusion, thus showing the rule admissible in $\mathsf{G3IntQC}$.

$$
\begin{array}{cl}
\mathcal{R}, \underline{a} \in D_u, \Gamma \Rightarrow w : \phi(\underline{a}/x), w : \exists x\phi, \Delta & \\
\hdashline
\mathcal{R}, \underline{a} \in D_u, \underline{a} \in D_{v_1}, \ldots, \underline{a} \in D_{v_n}, \underline{a} \in D_w, \Gamma \Rightarrow \Delta, w : A(\underline{a}/x), w : \exists xA & (wk) \\
\hline
\mathcal{R}, \underline{a} \in D_u, \underline{a} \in D_{v_1}, \ldots, \underline{a} \in D_{v_n}, \underline{a} \in D_w, \Gamma \Rightarrow \Delta, w : \exists xA & (\exists_r) \\
\hline
\mathcal{R}, \underline{a} \in D_u, \Gamma \Rightarrow \Delta, w : \exists xA & (d) \\
\hline
\mathcal{R}, \Gamma \Rightarrow w : \exists x\phi, \Delta & (ned)
\end{array}
$$

$(\forall^{\mathsf{QC}}_{l2})$. Similar to the proof of $(\exists^{\mathsf{QC}}_{r2})$ above.

□

We may now leverage Thm. 25 and 26 to show that our quasi-refined labelled calculi inherit proof-theoretic properties from their parental labelled calculi:

Theorem 27 (Proof-theoretic Properties of IntQL^* and IntQCL^*)**.** *Let* $\mathsf{X} \in \{\mathsf{Q}, \mathsf{QC}\}$*.*

(i) All rules from $\mathrm{Str}(\mathsf{Int})$ *are admissible in* IntXL^**.*

(ii) All rules in IntXL^* *are invertible.*

(iii) If $\vdash_{\mathsf{IntXL}^*} \Lambda$*,* $\models_{\mathsf{IntX}} \Lambda$*.*

(iv) If $\vdash_{\mathsf{IntX}} \phi$*, then* $\vdash_{\mathsf{IntXL}^*} \epsilon \Rightarrow w : \phi$*.*

Proof. Claims (i) – (iv) follow from Thm. 9, 10, 11, 25, 26, and Lem. 14, 15, 16, 17, and 18. □

Before moving on to the next section on domain atom removal, we show that our quasi-refined labelled calculi are sound and complete relative to labelled tree derivations with the fixed root property. The proof of this result is stated below, and although the proof is sufficient to confirm that our quasi-refined labelled calculi are 'simpler' than their parental labelled calculi—as the latter are incomplete relative to labelled tree derivations (Thm. 12)—it does not explain *why* structural rule elimination yields calculi that only require labelled tree sequents.

To answer such a question, we recognize observations made in [CLR18, Lyo20a, Lyo20b]: The rules (ref) and (tra) allow for labelled sequents to occur in derivations which are not labelled tree sequents. To demonstrate this fact, observe the following derivation:

$$\dfrac{\dfrac{w \leq v, v \leq v, v : p \Rightarrow v : p}{w \leq v, v : p \Rightarrow v : p}\ (ref)}{\Rightarrow w : p \supset p}\ (\supset_r)$$

As we can see in the example above, the initial sequent is *not* a labelled tree sequent due to the existence of the relational atom $v \leq v$ (one can check its sequent graph via Def. 50). Once (ref) is applied however, a labelled tree sequent results. Most importantly, we can see that the derivation consists of two fragments: a top fragment that does not use labelled tree sequents (i.e. the initial sequent), and a bottom fragment that does use labelled tree sequents (i.e. the bottom two sequents).

It turns out that the above observation—i.e. that a labelled derivation in IntQL^* or IntQCL^* can be partitioned into a 'treelike' and 'non-treelike' fragment—always holds, if we consider derivations that derive a labelled formula $w : \phi$. Due to the fact that the end sequent of such a derivation is of the form $\epsilon \Rightarrow w : \phi$, there will necessarily be a bottom fragment containing labelled tree sequents. If we consider the derivation in a bottom-up manner, then all rules—with the exception of (ref) and (tra)—will either preserve relational atoms or add new ones (e.g. $(\supset_r)$ and $(\forall_r)$), which constructs a tree emanating from $w : \phi$ due to the eigenvariable condition. Yet, once an application of (ref) or (tra) is applied (bottom-up) either a loop or undirected cycle will be added, breaking the 'treelike' structure and beginning the 'non-treelike' fragment of the proof. Therefore, if one imagines permuting instances of (ref) and (tra) upward in a IntQL^*

or IntQCL^* derivation (making use of the elimination algorithms explained in Lem. 39 and 40), then one can see that the bottom 'treelike' fragment grows and the top 'non-treelike' fragment diminishes. Once all (ref) and (tra) inferences have been removed, the derivation becomes a labelled tree derivation, explaining how we transition to a setting where only labelled tree derivations are required for completeness.

Theorem 28. *Every derivation in* IntQL^* *and* IntQCL^* *of a labelled formula* $w : \phi$ *is a labelled tree derivation with the fixed root property.*

Proof. Consider a proof of $\epsilon \Rightarrow w : \phi$ in IntQL^* or IntQCL^* in a bottom-up manner. The only rules that introduce relational atoms of the form $u \leq v$ are the $(\supset_r)$ and $(\forall_r)$. Hence, bottom-up applications of rules will either preserve relational structure or will construct a tree emanating from (the root) w. □

5.3 Refinement Part III: Removal of Domain Atoms

We introduce our refined labelled calculi IntQL and IntQCL (shown in Fig. 5.6 below) which are obtained from the quasi-refined calculi IntQL^* and IntQCL^* derived in the previous section. The extraction of IntQL and IntQCL from IntQL^* and IntQCL^* (resp.) results from two modifications made to the quasi-refined labelled calculi: (i) principal and auxiliary domain atoms are removed from all relevant inference rules, and (ii) the $(\exists_{r1}^{\mathsf{X}})$ and $(\exists_{r2}^{\mathsf{X}})$ rules, and the $(\forall_{l1}^{\mathsf{X}})$ and $(\forall_{l2}^{\mathsf{X}})$ rules, are combined to make the single rules $(\exists_{r}^{\mathsf{X}})$ and $(\forall_{l}^{\mathsf{X}})$ (given in Fig. 5.6 below) with a more complex side condition (expressed in Fig. 5.7). This section will be devoted to showing that the refined labelled calculi IntQL and IntQCL possess fundamental proof-theoretic properties, and in the subsequent section, we will show that the systems are notational variants of nested systems.

The removal of principal and auxiliary domain atoms from (id^{X}), $(\exists_l)$, $(\exists_{r1}^{\mathsf{X}})$, $(\exists_{r2}^{\mathsf{X}})$, $(\forall_r)$, $(\forall_{l1}^{\mathsf{X}})$, and $(\forall_{l2}^{\mathsf{X}})$ (yielding the rules (id_*), $(\exists_{l}^{\mathsf{X}})$, $(\exists_{r}^{\mathsf{X}})$, $(\forall_{r}^{\mathsf{X}})$, and $(\forall_{l}^{\mathsf{X}})$ in Fig. 5.6 below, resp.) is the central difference between our quasi-refined and refined labelled calculi. The rules (id_*), $(\exists_{l}^{\mathsf{X}})$, and $(\forall_{r}^{\mathsf{X}})$ are straightforwardly obtained from (id^{X}), $(\exists_l)$, and $(\forall_r)$, respectively, by simply omitting the principal and auxiliary domain atoms. By contrast, since applications of the $(\exists_{r1}^{\mathsf{X}})$, $(\exists_{r2}^{\mathsf{X}})$, $(\forall_{l1}^{\mathsf{X}})$, and $(\forall_{l2}^{\mathsf{X}})$ rules depend on side conditions that further depend on the existence of domain atoms, the extraction of $(\exists_{r}^{\mathsf{X}})$ and $(\forall_{l}^{\mathsf{X}})$ requires more insight. The key to transforming the $(\exists_{r1}^{\mathsf{X}})$ and $(\exists_{r2}^{\mathsf{X}})$ rules into $(\exists_{r}^{\mathsf{X}})$, and the $(\forall_{l1}^{\mathsf{X}})$ and $(\forall_{l2}^{\mathsf{X}})$ rules into $(\forall_{l}^{\mathsf{X}})$, is to re-write the side condition so that it no longer relies on the existence of a domain atom $\underline{a} \in D_u$, but rather, relies on the existence of a labelled formula $u : \psi(\vec{\underline{a}})$ with $\underline{a}$ occurring in $\vec{\underline{a}}$. Breaking the dependence of the rules on the existence of domain atoms effectively renders such syntactic structures expendable—a fact which is formally established later on (Prop. 3).

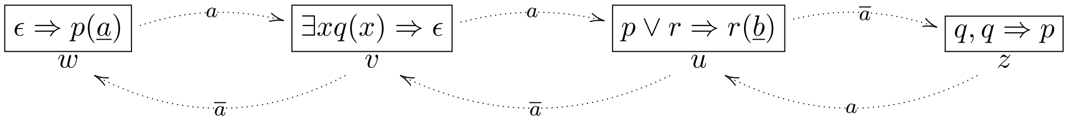
Recognizing that the reliance on domain atoms can be replaced by a reliance on labelled formulae, came about by comparing the side conditions of the $(\exists_{r1}^{\mathsf{X}})$, $(\exists_{r2}^{\mathsf{X}})$, $(\forall_{l1}^{\mathsf{X}})$, and $(\forall_{l2}^{\mathsf{X}})$ rules with the side conditions imposed by Fitting on his $R\exists$ and $L\forall$ rules employed in his nested calculus for first-order intuitionistic logic proper [Fit14, p. 59]. Whereas applications of our $(\exists^{\mathsf{X}}_{r1})$ and $(\forall^{\mathsf{X}}_{l1})$ rules rely on the existence of a propagation path connecting the label u of the active domain atom to the auxiliary label (see Fig. 5.3 and 5.4), Fitting's $R\exists$ and $L\forall$ rules rely (in part) on the existence of a path connecting a formula with a parameter $\underline{a}$ to the auxiliary formula of the inference—when such a state of affairs holds, Fitting calls the parameter $\underline{a}$ *available*. We impose similar side conditions in our refined labelled calculi, where a rule's applicability is dictated by a parameter being $S4$-*available* or $S5$-*available*:

Definition 75 ($S4$-available, $S5$-available)**.** *Let $\Lambda := \mathcal{R}, \Gamma \Rightarrow \Delta$. We say that a parameter $\underline{a}$ is $S4$-*available *($S5$-*available*) in Λ for a label w iff there exists a labelled formula $u : \phi(\underline{\vec{a}}) \in \Gamma, \Delta$ with $\underline{a}$ in $\underline{\vec{a}}$ such that $\exists \pi(s_{\pi}(u,w) \in L_{S4}(a))$ ($\exists \pi(s_{\pi}(u,w) \in L_{S5}(a))$, resp.).*

When making use of the fact that a parameter $\underline{a}$ is $S4$-available or $S5$-available, we will abuse notation and often denote the labelled formula $u : \phi(\underline{\vec{a}}) \in \Gamma, \Delta$ with $\underline{a}$ in $\underline{\vec{a}}$—which gives rise to the $S4$-availability or $S5$-availability of $\underline{a}$—as $u : \phi(\underline{a})$. This will simplify discussions and proofs that concern the side conditions of reachability rules.

Example 15. *We give an example of $S4$- and $S5$-available labels in the labelled sequent Λ shown below. Since such labels are determined by considering a labelled sequent's propagation graph, we also include a pictorial representation of $PG(\Lambda)$.*

$$\Lambda := w \leq v, v \leq u, u \leq z, v : \exists x q(x), u : p \vee r, z : q, z : q \Rightarrow w : p(\underline{a}), u : r(\underline{b}), z : p$$



$\epsilon \Rightarrow p(\underline{a})$
w
a
$\exists x q(x) \Rightarrow \epsilon$
v
a
$p \vee r \Rightarrow r(\underline{b})$
u
$\overline{a}$
$q, q \Rightarrow p$
z
$\overline{a}$
$\overline{a}$
a

In the above example, the parameter $\underline{a}$ is both $S4$- and $S5$-available for w, v, and u since there exist propagation paths $\lambda(w,w) = w$, $\pi_1(w,v) := w,a,v$, and $\pi_2(w,u) := w,a,v,a,u$ such that $s_{\lambda}(w,w) \in L_{S4}(a)$, $s_{\lambda}(w,w) \in L_{S5}(a)$, $s_{\pi_1}(w,v) \in L_{S4}(a)$, $s_{\pi_1}(w,v) \in L_{S5}(a)$, $s_{\pi_2}(w,u) \in L_{S4}(a)$, and $s_{\pi_2}(w,u) \in L_{S5}(a)$. However, note that $\underline{a}$ is not $S4$-available for z since none of the strings in $L_{S4}(a)$ contain the converse character $\overline{a}$, and any propagation path from w to z must contain such a character. On the other hand, the label z is $S5$-available since there exists a propagation path $\pi_3(w,z) := w,a,v,a,u,\overline{a},z$ such that $s_{\pi_3}(w,z) \in L_{S5}(a)$.

Our refined labelled calculi IntQL and IntQCL are given in Fig. 5.6 with the side conditions of the reachability and propagation rules given in Fig. 5.7. As explained previously, the calculi are obtained from IntQL^{*} and IntQCL^{*} by deleting principal and auxiliary domain atoms from rules as well as combining $(\exists^{\mathsf{X}}_{r1})$ and $(\exists^{\mathsf{X}}_{r2})$, and $(\forall^{\mathsf{X}}_{l1})$ and $(\forall^{\mathsf{X}}_{l2})$, yielding the rules $(\exists^{\mathsf{X}}_{r})$ and $(\forall^{\mathsf{X}}_{l})$, respectively. Furthermore, the side conditions of the $(\exists^{\mathsf{X}}_{r})$ and $(\forall^{\mathsf{X}}_{l})$

$$\frac{}{\mathcal{R}, \Gamma, w : p(\vec{\underline{a}}) \Rightarrow u : p(\vec{\underline{a}}), \Delta}\ (id_*)^{\dagger_1(\mathsf{X})} \qquad \frac{}{\mathcal{R}, \Gamma, w : \bot \Rightarrow \Delta}\ (\bot_l)$$

$$\frac{\mathcal{R}, \Gamma, w : \phi, w : \psi \Rightarrow \Delta}{\mathcal{R}, \Gamma, w : \phi \wedge \psi \Rightarrow \Delta}\ (\wedge_l) \qquad \frac{\mathcal{R}, \Gamma \Rightarrow w : \phi, \Delta \qquad \mathcal{R}, \Gamma \Rightarrow w : \psi, \Delta}{\mathcal{R}, \Gamma \Rightarrow w : \phi \wedge \psi, \Delta}\ (\wedge_r)$$

$$\frac{\mathcal{R}, \Gamma, w : \phi \Rightarrow \Delta \qquad \mathcal{R}, \Gamma, w : \psi \Rightarrow \Delta}{\mathcal{R}, \Gamma, w : \phi \vee \psi \Rightarrow \Delta}\ (\vee_l)$$

$$\frac{\mathcal{R}, w \leq u, \Gamma, u : \phi \Rightarrow u : \psi, \Delta}{\mathcal{R}, \Gamma \Rightarrow w : \phi \supset \psi, \Delta}\ (\supset_r)^{\dagger_2(\mathsf{X})} \qquad \frac{\mathcal{R}, \Gamma \Rightarrow w : \phi, w : \psi, \Delta}{\mathcal{R}, \Gamma \Rightarrow w : \phi \vee \psi, \Delta}\ (\vee_r)$$

$$\frac{\mathcal{R}, w : \phi \supset \psi, \Gamma \Rightarrow \Delta, u : \phi \qquad \mathcal{R}, w : \phi \supset \psi, u : \psi, \Gamma \Rightarrow \Delta}{\mathcal{R}, w : \phi \supset \psi, \Gamma \Rightarrow \Delta}\ (Pr_{\supset})^{\dagger_1(\mathsf{X})}$$

$$\frac{\mathcal{R}, \Gamma \Rightarrow \Delta, w : \phi(\underline{a}/x), w : \exists x \phi}{\mathcal{R}, \Gamma \Rightarrow \Delta, w : \exists x \phi}\ (\exists^{\mathsf{X}}_r)^{\dagger_3(\mathsf{X})} \qquad \frac{\mathcal{R}, w : \forall x \phi, v : \phi(\underline{a}/x), \Gamma \Rightarrow \Delta}{\mathcal{R}, w : \forall x \phi, \Gamma \Rightarrow \Delta}\ (\forall^{\mathsf{X}}_l)^{\dagger_4(\mathsf{X})}$$

$$\frac{\mathcal{R}, w \leq u, \Gamma \Rightarrow u : \phi(\underline{a}/x), \Delta}{\mathcal{R}, \Gamma \Rightarrow w : \forall x \phi, \Delta}\ (\forall^{\mathsf{X}}_r)^{\dagger_5(\mathsf{X})} \qquad \frac{\mathcal{R}, \Gamma, w : \phi(\underline{a}/x) \Rightarrow \Delta}{\mathcal{R}, \Gamma, w : \exists x \phi \Rightarrow \Delta}\ (\exists^{\mathsf{X}}_l)^{\dagger_6(\mathsf{X})}$$

Figure 5.6: The refined labelled calculus IntQL consists of all rules with $\mathsf{X} = \mathsf{Q}$, and the refined labelled calculus IntQCL consists of all rules with $\mathsf{X} = \mathsf{QC}$. The side conditions $\dagger_1(\mathsf{X})$ – $\dagger_6(\mathsf{X})$ are given in Fig. 5.7.

rules make use of the notion of $S4$- and $S5$-availability depending on if $\mathsf{X} = \mathsf{Q}$ or $\mathsf{X} = \mathsf{QC}$. As can be witnessed, none of the rules make reference to domain atoms, implying their superfluity. We formally establish this fact with the following proposition:

Proposition 3. *If $\mathcal{R}, \underline{a} \in D_w, \Gamma \Rightarrow \Delta$ is derivable in* IntQL *or* IntQCL*, then $\mathcal{R}, \Gamma \Rightarrow \Delta$ is derivable in* IntQL *or* IntQCL*, respectively.*

Proof. We prove the result by induction on the height of the given derivation and show the result for IntQCL as the proof for IntQL is similar.

Base case. The base case follows from the fact that removing any domain atoms from an instance of (id_*) or $(\bot_l)$ yields another instance of (id_*) or $(\bot_l)$, respectively.

Inductive step. All cases follow by invoking IH and then applying the corresponding rule. □

Let us now establish the soundness of our refined labelled calculi. Although we could attempt to prove soundness by supplying a semantics for our calculi and then showing that each rule of IntQL and IntQCL preserves validity, we opt for another strategy and show that each derivation in IntQL and IntQCL can be transformed into a derivation in

Name	Side Condition	Name	Side Condition
$\dagger_1$(Q)	$\exists \pi (s_\pi(w, u) \in L_{S4}(a))$	$\dagger_1$(QC)	$\exists \pi (s_\pi(w, u) \in L_{S4}(a))$
$\dagger_2$(Q)	u is an eigenvariable	$\dagger_2$(QC)	u is an eigenvariable
$\dagger_3$(Q)	$\underline{a}$ is $S4$-available for w or $\underline{a}$ is an eigenvariable	$\dagger_3$(QC)	$\underline{a}$ $S5$-available for w or $\underline{a}$ is an eigenvariable
$\dagger_4$(Q)	$\underline{a}$ is $S4$-available for v or $\underline{a}$ is an eigenvariable, and $\exists \pi (s_\pi(w, v) \in L_{S4}(a))$	$\dagger_4$(QC)	$\underline{a}$ $S5$-available for v or $\underline{a}$ is an eigenvariable, and $\exists \pi (s_\pi(w, v) \in L_{S4}(a))$
$\dagger_5$(Q)	$\underline{a}$ and u are eigenvariables	$\dagger_5$(QC)	$\underline{a}$ and u are eigenvariables
$\dagger_6$(Q)	$\underline{a}$ is an eigenvariable	$\dagger_6$(QC)	$\underline{a}$ is an eigenvariable

Figure 5.7: Side conditions for rules in IntQL and IntQCL.

IntQL* and IntQCL*, respectively. In order to transform derivations of the former systems into derivations in the latter systems, we introduce the *domain closure* function DomCL, which ensures that each parameter $\underline{a}$ occurring in any labelled formula $w : \phi(\vec{\underline{a}})$ of a labelled sequent Λ is accompanied by a corresponding domain atom $\underline{a} \in D_w$ (specifying the domain in which the parameter is included). We define the domain closure function below, followed by an example to demonstrate its functionality.

Definition 76 (The Function DomCL)). *Let $\Lambda := \mathcal{R}, \Gamma \Rightarrow \Delta$ be a labelled sequent and define*

$$\mathrm{Par}(\Lambda, w) := \{\underline{a} \mid \textit{there exists a } w : \phi(\vec{\underline{a}}) \in \Gamma, \Delta \textit{ with } \underline{a} \textit{ in } \vec{\underline{a}}\}.$$

We define $\mathtt{DomCL}(\Lambda) := \mathcal{R}, \mathcal{R}', \Gamma \Rightarrow \Delta$*, where*

$$\mathcal{R}' := \{\underline{a} \in D_w \mid w \in \mathrm{Lab}(\Lambda) \textit{ and } \underline{a} \in \mathrm{Par}(\Lambda, w)\}.$$

Example 16. *Suppose we are given the following labelled sequent:*

$$\Lambda := w \leq u, w \leq v, v : p(\underline{c}) \supset q(\underline{c}), w : r(\underline{a}) \Rightarrow w : \forall x p(x)$$

An application of $\mathtt{DomCL}(\cdot)$ *to* Λ *would output the following:*

$$\mathtt{DomCL}(\Lambda) := w \leq u, w \leq v, \underline{a} \in D_w, \underline{c} \in D_v, v : p(\underline{c}) \supset q(\underline{c}), w : r(\underline{a}) \Rightarrow w : \forall x p(x)$$

As discussed above, certain inference rules in IntQL* and IntQCL* depend on the existence of domain atoms, whereas all inference rules in IntQL and IntQCL are independent of domain atoms (see Prop. 3). Therefore, the significance of the domain closure function DomCL is that it adds a sufficient number of domain atoms to inferences within an IntQL or IntQCL proof to ensure that the proof is a valid derivation in IntQL* or IntQCL*, respectively. The following lemma demonstrates that the domain closure function permits the desired proof transformation (thus securing soundness, as will be subsequently proven).

Lemma 45.

(i) Every proof of a labelled sequent Λ in IntQL can be algorithmically transformed into a proof of $\mathtt{DomCL}(\Lambda)$ in IntQL^{*}.

(ii) Every proof of a labelled sequent Λ in IntQCL can be algorithmically transformed into a proof of $\mathtt{DomCL}(\Lambda)$ in IntQCL^{*}.

Proof. We defer the proof to Appendix C (p. 211). □

Theorem 29 (Soundness of IntQL and IntQCL)**.**

(i) If $\vdash_{\mathsf{IntQL}} \epsilon \Rightarrow w : \phi$, then $\Vdash_{\mathsf{IntQ}} \phi$.

(ii) If $\vdash_{\mathsf{IntQCL}} \epsilon \Rightarrow w : \phi$, then $\Vdash_{\mathsf{IntQC}} \phi$.

Proof. Follows from Thm. 27, Lem. 45, and Def. 48. □

We now show that IntQL and IntQCL possess desirable proof-theoretic properties such as hp-admissibility of structural rules (e.g. (wk) and (ctr_r)), hp-invertibility of all rules, and syntactic cut-elimination. After establishing these results, we harness the proof-theoretic properties of IntQL and IntQCL to prove completeness. The completeness result also relies on the following lemma, which essentially states that generalized instances of (id_*) are always derivable.

Lemma 46. *Let $\mathsf{XL} \in \{\mathsf{IntQL}, \mathsf{IntQCL}\}$ and $\Lambda := \mathcal{R}, \Gamma, w : \phi(\underline{\vec{a}}) \Rightarrow u : \phi(\underline{\vec{a}}), \Delta$. If there exists a propagation path $\pi(w, u)$ in $PG(\Lambda)$ such that $s_\pi(w, u) \in L_{S4}(a)$, then $\vdash_{\mathsf{XL}} \Lambda$.*

Proof. We prove the result by induction on the complexity of $\phi(\underline{\vec{a}})$. The base cases are trivial, so we only consider the inductive step.

Inductive step. We only show the cases where $\phi(\underline{\vec{a}})$ is of the form $\exists x \psi(\underline{\vec{a}})$ and $\forall x \psi(\underline{\vec{a}})$. Both cases are resolved as follows:

$$\begin{prooftree}
\AxiomC{}
\dashedLine
\RightLabel{IH}
\UnaryInfC{$\mathcal{R}, \Gamma, w : \psi(\underline{\vec{a}})(\underline{b}/x) \Rightarrow u : \psi(\underline{\vec{a}})(\underline{b}/x), u : \exists x \psi(\underline{\vec{a}}), \Delta$}
\RightLabel{$(\exists^{\mathsf{X}}_{r})$}
\UnaryInfC{$\mathcal{R}, \Gamma, w : \psi(\underline{\vec{a}})(\underline{b}/x) \Rightarrow u : \exists x \psi(\underline{\vec{a}}), \Delta$}
\RightLabel{$(\exists^{\mathsf{X}}_{l})$}
\UnaryInfC{$\mathcal{R}, \Gamma, w : \exists x \psi(\underline{\vec{a}}) \Rightarrow u : \exists x \psi(\underline{\vec{a}}), \Delta$}
\end{prooftree}$$

$$\begin{prooftree}
\AxiomC{}
\dashedLine
\RightLabel{IH}
\UnaryInfC{$\mathcal{R}, u \leq v, \Gamma, w : \forall x \psi(\underline{\vec{a}}), w : \psi(\underline{\vec{a}})(\underline{b}/x) \Rightarrow v : \psi(\underline{\vec{a}})(\underline{b}/x), \Delta$}
\RightLabel{$(\forall^{\mathsf{X}}_{l})$}
\UnaryInfC{$\mathcal{R}, u \leq v, \Gamma, w : \forall x \psi(\underline{\vec{a}}) \Rightarrow v : \psi(\underline{\vec{a}})(\underline{b}/x), \Delta$}
\RightLabel{$(\forall^{\mathsf{X}}_{r})$}
\UnaryInfC{$\mathcal{R}, \Gamma, w : \forall x \psi(\underline{\vec{a}}) \Rightarrow u : \forall x \psi(\underline{\vec{a}}), \Delta$}
\end{prooftree}$$

□

Next, we confirm the hp-admissibility of the substitution rules (lsb) and (psb), which serves as a foundation for all remaining results. Recall that label and parameter substitutions were defined in Def. 49. Additionally, we note that all structural rules shown hp-admissible in this section are the same as those found in Fig. 3.4. Also, most of the remaining proofs of this section will be deferred to the Appendix C (starting on p. 211) to improve readability and because such results are similar to the hp-admissibility, hp-invertibility, and elimination results proven in Ch. 3. Unless stated otherwise, all omitted proofs (i.e. proofs deferred to Appendix C) are shown by induction on the height of the given derivation.

Lemma 47. *The rule* (lsb) *is hp-admissible in* IntQL *and* IntQCL*.*

Lemma 48. *The rule* (psb) *is hp-admissible in* IntQL *and* IntQCL*.*

Lemma 49. *The rule* (wk) *is hp-admissible in* IntQL *and* IntQCL*.*

Lemma 50. *The* $(\wedge_l)$, $(\wedge_r)$, $(\vee_l)$, *and* $(\vee_r)$ *rules are hp-invertible in* IntQL *and* IntQCL*.*

Lemma 51. *The* $(Pr_{\supset})$ *and* $(\supset_r)$ *rules are hp-invertible in* IntQL *and* IntQCL*.*

Lemma 52.

(i) The $(\exists_l^{\mathsf{Q}})$, $(\exists_r^{\mathsf{Q}})$, $(\forall_l^{\mathsf{Q}})$, *and* $(\forall_r^{\mathsf{Q}})$ *rules are hp-invertible in* IntQL*.*

(ii) The $(\exists_l^{\mathsf{QC}})$, $(\exists_r^{\mathsf{QC}})$, $(\forall_l^{\mathsf{QC}})$, *and* $(\forall_r^{\mathsf{QC}})$ *rules are hp-invertible in* IntQCL*.*

Lemma 53. *All rules are hp-invertible in* IntQL *and* IntQCL*.*

Proof. Follows from Lem. 50 – 52 above. □

Lemma 54. *The rule* $(ctr_{\mathcal{R}})$ *is hp-admissible in* IntQL *and* IntQCL*.*

Proof. We proceed by induction on the height of the given derivation.

Base case. Any application of $(ctr_{\mathcal{R}})$ to (id_*) or $(\bot_l)$ yields another instance of the rule, which confirms the base case.

Inductive step. All cases are resolved by applying IH followed by the corresponding rule. □

Lemma 55. *The rule* (ctr_l) *is hp-admissible in* IntQL *and* IntQCL*.*

Lemma 56. *The rule* (ctr_r) *is hp-admissible in* IntQL *and* IntQCL*.*

Proof. The result is shown by induction on the height of the given derivation and is similar to the proof of Lem. 55 above. □

Lemma 57. *The* (ref) *and* (tra) *rules are elimimable in* IntQL *and* IntQCL*.*

Proof. The lemma is proven similarly to Lem. 39 and 40. □

Lemma 58. *Let* $\Lambda := \mathcal{R}, \Gamma \Rightarrow \Delta$*. Suppose* $\mathcal{R}, w \leq u, \Gamma \Rightarrow \Delta$ *is derivable in* IntQL *or* IntQCL*, and that* $\pi(w, u)$ *is a propagation path (potentially empty) in* $PG(\Lambda)$ *such that* $s_{\pi}(w, u) \in L_{S4}(a)$*. Then,* $\mathcal{R}, \Gamma \Rightarrow \Delta$ *is derivable in* IntQL *or* IntQCL*, respectively.*

Proof. Similar to the proof of Lem. 44, but relies on Lem. 57 above. □

Theorem 30. *The rule* (cut) *is eliminable in* IntQL *and* IntQCL*.*

Proof. The result is proven by induction on the lexicographic ordering of pairs $(|\phi|, h_1+h_2)$, where $|\phi|$ is the complexity of the cut formula ϕ, h_1 is the height of the derivation of the left premise of (cut), and h_2 is the height of the derivation of the right premise of (cut). We defer the proof to Appendix C (starting on p. 211). □

We are now in a position to prove the completeness of our refined labelled calculi IntQL and IntQCL. The proof is as follows:

Theorem 31 (Completeness)**.**

(i) If $\vdash_{\mathsf{IntQ}} \phi$*, then* $\vdash_{\mathsf{IntQL}} \epsilon \Rightarrow w : \phi$*.*

(ii) If $\vdash_{\mathsf{IntQC}} \phi$*, then* $\vdash_{\mathsf{IntQCL}} \epsilon \Rightarrow w : \phi$*.*

Proof. We show that our calculi IntQL and IntQCL can derive axioms A9 – A12 and A9 – A13, respectively, as well as simulate the inference rules R0 and R1 (see Def. 20 for the axioms of IntQ and IntQC). Proofs of axioms A0 – A8, i.e. the axioms for propositional intuitionistic logic (cf. Def. 20 and [GSS09, p. 6]), are straightforward.

We provide derivations for axioms A9 – A13 below, and note that the proof of axiom A13 (the constant domain axiom) requires the rule $(\forall_l^{\mathsf{QC}})$, i.e. the proof does not go through with the weaker rule $(\forall_l^{\mathsf{Q}})$. For all other cases we let $\mathsf{X} \in \{\mathsf{Q}, \mathsf{QC}\}$ to prove the results uniformly.

Axiom A9.

$$\cfrac{\cfrac{\overline{w \leq u, u : \forall x\phi, u : \phi(\underline{a}/x) \Rightarrow u : \phi(\underline{a}/x)}\ \text{Lem. 46}}{w \leq u, u : \forall x\phi \Rightarrow u : \phi(\underline{a}/x)}\ (\forall_l^{\mathsf{X}})}{\epsilon \Rightarrow w : \forall x\phi \supset \phi(\underline{a}/x)}\ (\supset_r)$$

Axiom A10.

$$\cfrac{\cfrac{\overline{w \leq u, u : \phi(\underline{a}/x) \Rightarrow u : \exists x\phi, u : \phi(\underline{a}/x)}\ \text{Lem. 46}}{w \leq u, u : \phi(\underline{a}/x) \Rightarrow u : \exists x\phi}\ (\exists_r^{\mathsf{X}})}{\epsilon \Rightarrow w : \phi(\underline{a}/x) \supset \exists x\phi}\ (\supset_r)$$

Axiom A11. To save space and improve readability, we let $\mathcal{R} := w \leq u, u \leq v, v \leq z$. Also, by assumption we know that x does not occur in ψ (see Def. 20).

$$\Pi_1 \quad = \left\{ \quad \dfrac{}{\mathcal{R}, u : \forall x(\psi \supset \phi), v : \psi, z : \psi \supset \phi(\underline{a}/x) \Rightarrow z : \psi, z : \phi(\underline{a}/x)} \ \text{Lem. 46} \right.$$

$$\Pi_2 \quad = \left\{ \quad \dfrac{}{\mathcal{R}, u : \forall x(\psi \supset \phi), v : \psi, z : \psi \supset \phi(\underline{a}/x), z : \phi(\underline{a}/x) \Rightarrow z : \phi(\underline{a}/x)} \ \text{Lem. 46} \right.$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Pi_1 \qquad \Pi_2}{w \leq u, u \leq v, v \leq z, u : \forall x(\psi \supset \phi), v : \psi, z : \psi \supset \phi(\underline{a}/x) \Rightarrow z : \phi(\underline{a}/x)} \ (Pr_{\supset})}{w \leq u, u \leq v, v \leq z, u : \forall x(\psi \supset \phi), v : \psi \Rightarrow z : \phi(\underline{a}/x)} \ (\forall_l^{\mathsf{X}})}{w \leq u, u \leq v, u : \forall x(\psi \supset \phi), v : \psi \Rightarrow v : \forall x \phi} \ (\forall_r^{\mathsf{X}})}{w \leq u, u : \forall x(\psi \supset \phi) \Rightarrow u : \psi \supset \forall x \phi} \ (\supset_r)}{\epsilon \Rightarrow w : \forall x(\psi \supset \phi) \supset (\psi \supset \forall x \phi)} \ (\supset_r)$$

Axiom A12. To save space and improve readability, we let $\mathcal{R} := w \leq u, u \leq v$. Also, by assumption we know that x does not occur in ϕ (see Def. 20).

$$\Pi_1 \quad = \left\{ \quad \dfrac{}{\mathcal{R}, u : \forall x(\psi \supset \phi), v : \psi(\underline{a}/x), v : \psi(\underline{a}/x) \supset \phi \Rightarrow v : \psi(\underline{a}/x), v : \phi} \ \text{Lem. 46} \right.$$

$$\Pi_2 \quad = \left\{ \quad \dfrac{}{\mathcal{R}, u : \forall x(\psi \supset \phi), v : \psi(\underline{a}/x), v : \psi(\underline{a}/x) \supset \phi, v : \phi \Rightarrow v : \phi} \ \text{Lem. 46} \right.$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Pi_1 \qquad \Pi_2}{w \leq u, u \leq v, u : \forall x(\psi \supset \phi), v : \psi(\underline{a}/x), v : \psi(\underline{a}/x) \supset \phi \Rightarrow v : \phi} \ (Pr_{\supset})}{w \leq u, u \leq v, u : \forall x(\psi \supset \phi), v : \psi(\underline{a}/x) \Rightarrow v : \phi} \ (\forall_l^{\mathsf{X}})}{w \leq u, u \leq v, u : \forall x(\psi \supset \phi), v : \exists x \psi \Rightarrow v : \phi} \ (\exists_l^{\mathsf{X}})}{w \leq u, u : \forall x(\psi \supset \phi) \Rightarrow u : \exists x \psi \supset \phi} \ (\supset_r)}{\epsilon \Rightarrow w : \forall x(\psi \supset \phi) \supset (\exists x \psi \supset \phi)} \ (\supset_r)$$

Axiom A13.

$$\Pi_1 \quad = \left\{ \quad \dfrac{}{w \leq u, u \leq v, u : \forall x(\phi \vee \psi), u : \psi \Rightarrow v : \phi(\underline{a}/x), u : \psi} \ \text{Lem. 46} \right.$$

$$\Pi_2 \quad = \left\{ \quad \dfrac{}{w \leq u, u \leq v, u : \forall x(\phi \vee \psi), u : \phi(\underline{a}/x) \Rightarrow v : \phi(\underline{a}/x), u : \psi} \ \text{Lem. 46} \right.$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Pi_1 \qquad \Pi_2}{w \leq u, u \leq v, u : \forall x(\phi \lor \psi), u : \phi(\underline{a}/x) \lor \psi \Rightarrow v : \phi(\underline{a}/x), u : \psi}\,(\lor_l)}{w \leq u, u \leq v, u : \forall x(\phi \lor \psi) \Rightarrow v : \phi(\underline{a}/x), u : \psi}\,(\forall_l^{\mathsf{QC}})}{w \leq u, u : \forall x(\phi \lor \psi) \Rightarrow u : \forall x\phi, u : \psi}\,(\forall_r^{\mathsf{QC}})}{w \leq u, u : \forall x(\phi \lor \psi) \Rightarrow u : \forall x\phi \lor \psi}\,(\lor_r)}{\epsilon \Rightarrow w : \forall x(\phi \lor \psi) \supset \forall x\phi \lor \psi}\,(\supset_r)$$

Rule R0. We invoke Lem. 57 and Thm. 30 below to apply the admissible rules (ref) and (cut).

$$\dfrac{\epsilon \Rightarrow w : \phi \qquad \dfrac{\dfrac{\dfrac{\dfrac{\epsilon \Rightarrow w : \phi \supset \psi}{w \leq u, u : \phi \Rightarrow u : \psi}\,\text{Lem. 51}}{(w \leq u)(w/u), (u : \phi)(w/u) \Rightarrow (u : \psi)(w/u)}\,(lsb)}{w \leq w, w : \phi \Rightarrow w : \psi}\,=}{w : \phi \Rightarrow w : \psi}\,(ref)}{\epsilon \Rightarrow w : \psi}\,(cut)$$

Rule R1.

$$\dfrac{\dfrac{\dfrac{\epsilon \Rightarrow w : \phi(\underline{a}/x)}{u \leq w \Rightarrow w : \phi(\underline{a}/x)}\,(wk)}{\epsilon \Rightarrow u : \forall x\phi}\,(\forall_r^{\mathsf{X}})}{\epsilon \Rightarrow w : \forall x\phi}\,(lsb)$$

□

Due to the similarity of the refined labelled calculi IntQL and IntQCL, and the quasi-refined labelled calculi IntQL* and IntQCL*, it should not be surprising that IntQL and IntQCL are complete relative to labelled tree derivations with the fixed root property. The following theorem confirms this fact. Yet, since we have removed all dependencies on domain atoms within our refined labelled calculi, such systems can be proven notational variants of nested sequent systems, which we demonstrate in the following section.

Theorem 32. *Every derivation in* IntQL *and* IntQCL *of a labelled formula* $w : \phi$ *is a labelled tree derivation with the fixed root property.*

Proof. Similar to the proof of Thm. 28. □

5.4 Relationship to Nested Sequent Formalism

We show that our refined labelled calculi IntQL and IntQCL can be viewed as nested systems. The systems we put forth are distinct from Fitting's nested calculi for IntQ and IntQC (which have been provided in Appendix A on p. 207), as we employ reachability and propagation rules such as (id_*), $(Pr_{\supset})$, $(\exists^{\mathsf{X}}_{r})$, and $(\forall^{\mathsf{X}}_{l})$ (shown in Fig. 5.8 below), whereas Fitting's calculi utilize different versions of these rules and adds a rule called *lift* (which is a reformulation of the propagation rules introduced by Postniece in [Pos09, Pos10]). As discussed previously, making use of reachability and propagation rules (which take a CFCST system as a parameter) is advantageous, as it lets us easily transform our calculi into systems for new logics by taking alternative CFCST systems as parameters. We first define nested sequents, and afterward, define their sequent graphs, which facilitate translations between labelled and nested notation.

Definition 77 (Nested Sequents for First-Order Intuitionistic Logics)**.** *We define a* nested sequent for first-order intuitionistic logics *to be a syntactic object X defined via the grammars in BNF shown below:*

$$X ::= Y \Rightarrow Y \mid Y \Rightarrow Y, \{X\}, \ldots, \{X\} \qquad Y ::= \varepsilon \mid \phi \mid Y, Y$$

where $\phi \in \mathcal{L}_{\mathsf{Int}}$.

We denote nested sequents with X, Y, Z, ... (possibly annotated), and as usual, ϵ represents the empty string, which is an identity element for comma, where comma associates and commutes. As before, we use the notation $X[Y]$ (and $X[Y][Z]$) to mean that the nested sequent Y (Y and Z, resp.) occurs (occur, resp.) at some depth in the nestings of X. For example, if the nested sequent X is $p(\underline{a}) \Rightarrow q, \{\forall x p(x) \Rightarrow q \supset q\}, \{\epsilon \Rightarrow q\}$, then $X[p(\underline{a}) \Rightarrow q]$, $X[p(\underline{a}) \Rightarrow q, \{\forall x p(x) \Rightarrow q \supset q\}]$, and $X[p(\underline{a}) \Rightarrow q][\epsilon \Rightarrow q]$ are all valid representations of X.

Definition 78 (Sequent Graph of a Nested Sequent for IntQ and IntQC)**.** *We define the* sequent graph of a nested sequent X *inductively on the depth of the nestings of X as shown below. As in Def. 62, we make use of sequences of natural numbers, and represent such sequences as $n_1.n_2.\ldots.n_{k-1}.n_k$, using σ (possible annotated) to denote them. Our inductive definition of $G(X) := G_0(X)$ is as follows:*

- *If $X = \phi_1, \ldots, \phi_n \Rightarrow \psi_1, \ldots, \psi_k$, then $G_\sigma(X) := (V_\sigma, E_\sigma, L_\sigma)$, where*

  $$(i) \quad V_\sigma := \{\sigma\} \qquad (ii) \quad E_\sigma := \emptyset \qquad (iii) \quad L_\sigma := \{(\sigma, \phi_1, \ldots, \phi_n \Rightarrow \psi_1, \ldots, \psi_k)\}$$

- *Let $X := \phi_1, \ldots, \phi_n \Rightarrow \psi_1, \ldots, \psi_k, \{Y_1\}, \ldots, \{Y_m\}$ and suppose that each $G_{\sigma.i}(Y_i) := (V_{\sigma.i}, E_{\sigma.i}, L_{\sigma.i})$ (with $i \in \{1, \ldots m\}$ and $m \in \mathbb{N}$) is already defined. We define $G_\sigma(X) := (V_\sigma, E_\sigma, L_\sigma)$ as shown below:*

$$\frac{}{X[Y_1, p(\underline{\vec{a}}) \Rightarrow Z_1]_w[Y_2 \Rightarrow p(\underline{\vec{a}}), Z_2]_u}\ (id_*)^{\dagger_1(\mathsf{X})} \qquad \frac{}{X[Y, \bot \Rightarrow Z]}\ (\bot_l)$$

$$\frac{X[Y, \phi, \psi \Rightarrow Z]}{X[Y, \phi \land \psi \Rightarrow Z]}\ (\land_l) \qquad \frac{X[Y \Rightarrow \phi, \psi, Z]}{X[Y \Rightarrow \phi \lor \psi, Z]}\ (\lor_r) \qquad \frac{X[Y \Rightarrow Z, \{\phi \Rightarrow \psi\}]}{X[Y \Rightarrow \phi \supset \psi, Z]}\ (\supset_r)$$

$$\frac{X[Y, \phi \Rightarrow Z] \qquad X[Y, \psi \Rightarrow Z]}{X[Y, \phi \lor \psi \Rightarrow Z]}\ (\lor_l) \qquad \frac{X[Y \Rightarrow \phi, Z] \qquad X[Y \Rightarrow \psi, Z]}{X[Y \Rightarrow \phi \land \psi, Z]}\ (\land_r)$$

$$\frac{X[Y_1, \phi \supset \psi \Rightarrow Z_1]_w[Y_2 \Rightarrow \phi, Z_2]_u \qquad X[Y_1, \phi \supset \psi \Rightarrow Z_1]_w[Y_2, \psi \Rightarrow Z_2]_u}{X[Y_1, \phi \supset \psi \Rightarrow Z_1]_w[Y_2 \Rightarrow Z_2]_u}\ (Pr_{\supset})^{\dagger_1(\mathsf{X})}$$

$$\frac{X[Y, \phi(\underline{a}/x) \Rightarrow Z]}{X[Y, \exists x\phi \Rightarrow Z]}\ (\exists_l^{\mathsf{X}})^{\dagger_2(\mathsf{X})} \qquad \frac{X[Y_1, \forall x\phi \Rightarrow Z_1]_w[Y_2, \phi(\underline{a}/x) \Rightarrow Z_2]_v}{X[Y_1, \forall x\phi \Rightarrow Z_1]_w[Y_2 \Rightarrow Z_2]_v}\ (\forall_l^{\mathsf{X}})^{\dagger_4(\mathsf{X})}$$

$$\frac{X[Y \Rightarrow \phi(\underline{a}/x), \exists x\phi, Z]_w}{X[Y \Rightarrow \exists x\phi, Z]_w}\ (\exists_r^{\mathsf{X}})^{\dagger_3(\mathsf{X})} \qquad \frac{X[Y \Rightarrow Z, \{\epsilon \Rightarrow \phi(\underline{a}/x)\}]}{X[Y \Rightarrow \forall x\phi, Z]}\ (\forall_r^{\mathsf{X}})^{\dagger_2(\mathsf{X})}$$

Figure 5.8: The nested calculus DIntQ consists of all rules with X = Q. The nested calculus DIntQC consists of all rules with X = QC. The side conditions of the propagation and reachability rules are given in Fig. 5.9 below.

- $V_\sigma := \{\sigma\} \cup \bigcup_{1 \leq i \leq m} V_{\sigma.i}$
- $E_\sigma := \{(\sigma, \sigma.i, a) \mid 1 \leq i \leq m\} \cup \bigcup_{1 \leq i \leq m} E_{\sigma.i}$
- $L_\sigma := \{(\sigma, \phi_1, \ldots, \phi_n \Rightarrow \psi_1, \ldots, \psi_k)\} \cup \bigcup_{1 \leq i \leq m} L_{\sigma.i}$

Note that when $n = 0$ or $k = 0$, the multiset $\phi_1, \ldots, \phi_n$ and $\psi_1, \ldots, \psi_k$ is taken to be the empty string ϵ. We will often use w, u, v, ... to represent vertices as opposed to sequences of natural numbers.

For a nested sequent $X[Y \Rightarrow Z]$ or $X[Y_1 \Rightarrow Z_1][Y_2 \Rightarrow Z_2]$, we also use the notation $X[Y \Rightarrow Z]_w$ and $X[Y_1 \Rightarrow Z_1]_w[Y_2 \Rightarrow Z_2]_u$ to denote that $Y \Rightarrow Z$ is associated with the vertex w (meaning that $L(w) = Y \Rightarrow Z$ in $G(X) = (V, E, L)$), and to denote that $Y_1 \Rightarrow Z_1$ and $Y_2 \Rightarrow Z_2$ are associated with the vertices w and u (meaning that $L(w) = Y_1 \Rightarrow Z_1$ and $L(u) = Y_2 \Rightarrow Z_2$ in $G(X) = (V, E, L)$), respectively.

As with our refined labelled calculi, applications of the reachability and propagation rules depend on the notion of a propagation graph, $S4$-availability, and $S5$-availability. We define these concepts below, and subsequently provide an example of a sequent graph and propagation graph for a nested sequent, as well as give an example of a parameter which is $S4$-available and $S5$-available.

Definition 79 (Propagation Graphs for DIntQ and DIntQC)**.** *Let X be a nested sequent for* IntQ *and* IntQC *with the sequent graph $G(X) = (V, E, L)$. We define the* propagation graph *$PG(X) = (\mathrm{V}', \mathrm{E}')$ to be the directed graph such that*

- $\mathrm{V}' := V$*;*
- $\mathrm{E}' := \{(w, u, a), (u, w, \overline{a}) \mid (w, u, a) \in E\}$.

We will often write $w \in PG(X)$ to mean $w \in \mathrm{V}'$, and $(w, u, x) \in PG(X)$ to mean $(w, u, x) \in \mathrm{E}'$, for $x \in \sum = \{a, \overline{a}\}$.

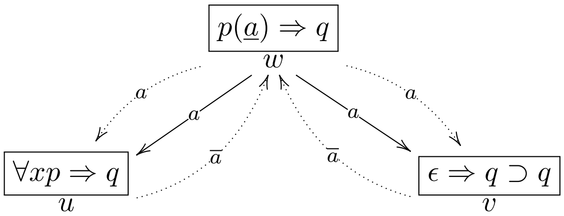
Definition 80 ($S4$-available, $S5$-available)**.** *Let $X := X[Y_1 \Rightarrow Z_1]_u[Y_2 \Rightarrow Z_2]_w$ be a nested sequent. We say that a parameter $\underline{a}$ is* $S4$-available *(*$S5$-available*) in X for w iff there exists a formula $\psi(\vec{\underline{a}}) \in Y_1, Z_1$ with $\underline{a}$ in $\vec{\underline{a}}$ and a propagation path $\pi(u, w)$ in $PG(X)$ such that $s_\pi(u, w) \in L_{S4}(a)$ ($s_\pi(u, w) \in L_{S5}(a)$, resp.).*

Propagation paths and strings of propagation paths (along with their converses) are defined as in Def. 59, so we do not repeat these definitions here. However, we do provide examples of these objects below along with our example:

Example 17. *Let our nested sequent be the following:*

$$X := p(\underline{a}) \Rightarrow q, \{\forall x p(x) \Rightarrow q \supset q\}, \{\epsilon \Rightarrow q\}$$

The sequent graph $G(X)$ is obtained by deleting the dotted edges in the graph below, and the propagation graph $PG(X)$ is obtained by deleting the solid edges.


$p(\underline{a}) \Rightarrow q$
w
a
a
$\overline{a}$
$\overline{a}$
a
a
$\forall xp \Rightarrow q$
u
$\epsilon \Rightarrow q \supset q$
v

Examples of propagation paths include the propagation path $\pi(w, u) = w, a, u$ and the propagation path $\pi'(u, w) := u, \overline{a}, w, a, v, \overline{a}, w$. The string of each propagation path is $s_\pi(w, u) = a$ and $s_{\pi'}(u, w) = \overline{a}a\overline{a}$, respectively. Also, since $p(\underline{a})$ occurs in the antecedent at w and there exists a propagation path $\pi(w, u) := w, a, u$ with $a \in L_{S4}(a)$ and $a \in L_{S5}(a)$, we know that $\underline{a}$ is both $S4$- and $S5$-available for u.

Making use of the above notions, we define our nested calculi DIntQ and DIntQC for the first-order intuitionistic logics IntQ and IntQC, respectively, which are presented in Fig. 5.8 with the side conditions of reachability and propagation rules given in Fig. 5.9.

Name	Side Condition	Name	Side Condition
$\dagger_1(\mathsf{Q})$	$\exists\pi(s_\pi(w,u) \in L_{S4}(a))$	$\dagger_1(\mathsf{QC})$	$\exists\pi(s_\pi(w,u) \in L_{S4}(a))$
$\dagger_2(\mathsf{Q})$	$\underline{a}$ is an eigenvariable	$\dagger_2(\mathsf{QC})$	$\underline{a}$ is an eigenvariable
$\dagger_3(\mathsf{Q})$	$\underline{a}$ is $S4$-available for w or $\underline{a}$ is an eigenvariable	$\dagger_3(\mathsf{QC})$	$\underline{a}$ $S5$-available for w or $\underline{a}$ is an eigenvariable
$\dagger_4(\mathsf{Q})$	$\underline{a}$ is $S4$-available for v or $\underline{a}$ is an eigenvariable, and $\exists\pi(s_\pi(w,v) \in L_{S4}(a))$	$\dagger_4(\mathsf{QC})$	$\underline{a}$ $S5$-available for v or $\underline{a}$ is an eigenvariable, and $\exists\pi(s_\pi(w,v) \in L_{S4}(a))$

Figure 5.9: Side conditions for rules in IntQL and IntQCL.

We now define our translations functions $\mathfrak{N}$ and $\mathfrak{L}$, which translate from labelled to nested notation, and from nested to labelled notation, respectively. The definition depends on the notation of the downward closure of a sequent graph, defined in Def. 65. After presenting these definitions, we prove that IntQL and IntQCL are the respective notational variants of the calculi DIntQ and DIntQC, and give an example of translating derivations between the calculi.

Definition 81 (The Translation $\mathfrak{N}$). *Let $\Lambda := \mathcal{R}, \Gamma \Rightarrow \Delta$ be the a labelled tree sequent with $G(\Lambda) = (V, E, L)$ and $w \in V$ the root. We define the translation $\mathfrak{N}(\Lambda) := \mathfrak{N}(G_w(\Lambda))$ inductively as follows:*

- ▶ *If $G(\Lambda) = (V, E, L)$ with $V = \{w\}$, $E = \emptyset$, and $L = \{(w, \Gamma \restriction w \Rightarrow \Delta \restriction w)\}$, then*

$$\mathfrak{N}(G_w(\Lambda)) := \Gamma \restriction w \Rightarrow \Delta \restriction w$$

- ▶ *If $G(\Lambda) = (V, E, L)$ with $w, u_1, \ldots, u_n \in V$, $(w, u_i) \in E$ (for $i \in \{1, \ldots, n\}$), then*

$$\mathfrak{N}(G_w(\Lambda)) := L(w), \{\mathfrak{N}(G_{u_1}(\Lambda))\}, \ldots, \{\mathfrak{N}(G_{u_n}(\Lambda))\}$$

Example 18. *We show how to translate the labelled tree sequent*

$$\Lambda := w \leq u, u \leq v, w \leq z, w : q, v : p \supset q, v : r \Rightarrow, w : \forall x q, u : p \lor p, z : c, z : p$$

into a nested sequent via the computation below:

$$\begin{aligned}
\mathfrak{N}(\Lambda) \quad &:= \mathfrak{N}(G_w(\Lambda)) \\
&= q \Rightarrow \forall x q, \{\mathfrak{N}(G_u(\Lambda))\}, \{\mathfrak{N}(G_z(\Lambda))\} \\
&= q \Rightarrow \forall x q, \{\epsilon \Rightarrow p \lor p, \{\mathfrak{N}(G_v(\Lambda))\}\}, \{\epsilon \Rightarrow c, p\} \\
&= q \Rightarrow \forall x q, \{\epsilon \Rightarrow p \lor p, \{p \supset q, r \Rightarrow \epsilon\}\}, \{\epsilon \Rightarrow c, p\}
\end{aligned}$$

Definition 82 (The Translation $\mathfrak{L}$)**.** *Let X be the a nested sequent. We define the translation $\mathfrak{L}(X) := \mathfrak{L}(G_0(X))$ inductively as follows:*

- *If $G_\sigma(X) = (V, E, L)$ with $V = \{\sigma\}$, $E = \emptyset$, and $L = \{(\sigma, Y \Rightarrow Z)\}$, then*

$$\mathfrak{L}(G_\sigma(X)) := w_\sigma : Y \Rightarrow w_\sigma : Z$$

- *If $G_\sigma(X) = (V, E, L)$ with $\sigma, \sigma.1, \ldots, \sigma.n \in V$, $(\sigma, \sigma.1), \ldots, (\sigma, \sigma.n) \in E$, and $(\sigma, Y \Rightarrow Z) \in L$, then*

$$\mathfrak{L}(G_\sigma(X)) := (w_\sigma \leq w_{\sigma.1}, \ldots, w_\sigma \leq w_{\sigma.n}, w_\sigma : Y \Rightarrow w_\sigma : Z) \circ \mathfrak{L}(G_{\sigma.1}(X)) \circ \cdots \circ \mathfrak{L}(G_{\sigma.n}(X))$$

In practice, we will often use labels such as w, u, v, … as opposed to labels indexed with sequences of natural numbers for simplicity.

Example 19. *Below, we show how to translate the nested sequent*

$$X := q \Rightarrow \forall xq, \{\epsilon \Rightarrow p \vee p, \{p \supset q, r \Rightarrow \epsilon\}\}, \{\epsilon \Rightarrow c, p\}$$

into a labelled (tree) sequent:

$$\begin{aligned}
\mathfrak{L}(X) &:= \mathfrak{L}(G_0(X)) \\
&= (w \leq u, w \leq z, w : q \Rightarrow w : \forall xq) \circ \mathfrak{L}(G_u(X)) \circ \mathfrak{L}(G_z(X)) \\
&= (w \leq u, w \leq z, w : q \Rightarrow w : \forall xq) \circ (u \leq v \Rightarrow u : p \vee p) \circ \\
&\quad \mathfrak{L}(G_v(X)) \circ (\epsilon \Rightarrow z : c, z : p) \\
&= (w \leq u, w \leq z, w : q \Rightarrow w : \forall xq) \circ (u \leq v \Rightarrow u : p \vee p) \circ \\
&\quad (v : p \supset q, v : r \Rightarrow \epsilon) \circ (\epsilon \Rightarrow z : c, z : p) \\
&= w \leq u, u \leq v, w \leq z, w : q, v : p \supset q, v : r \Rightarrow, w : \forall xq, u : p \vee p, z : c, z : p
\end{aligned}$$

As with our translations for grammar logics, it is not difficult to confirm that the sequent graph of a labelled tree sequent or nested sequent is isomorphic to the sequent graph of its translatee under $\mathfrak{N}$ and $\mathfrak{L}$, respectively. This fact is expressed in the following lemma:

Lemma 59. *Let Λ be a labelled tree sequent and X be a nested sequent (for first-order intuitionistic logics). Then,*

(i) $G(\Lambda) \cong G(\mathfrak{N}(\Lambda))$

(ii) $G(X) \cong G(\mathfrak{L}(X))$

Theorem 33. *Let $\mathsf{X} \in \{\mathsf{Q}, \mathsf{QC}\}$. Every derivation in IntXL is algorithmically translatable to a derivation in DIntX.*

Proof. We prove the result by induction on the height of the given derivation and show the claim for IntQCL and DIntQC, as translating from IntQL to DIntQ is similar.

Base case. Let the initial sequent shown below left be Λ, and assume that the nested sequent shown below right is $\mathfrak{N}(\Lambda)$. Due to the application of (id_*) (bottom left), we know that there exists a propagation path $\pi(w, u)$ in $PG(\Lambda)$ such that $s_\pi(w, u) \in L_{S4}(a)$. By Lem. 59 above, the same propagation path will exist in $PG(\mathfrak{N}(\Lambda))$, and hence, the application of the (id_*) rule (bottom-right) deriving $\mathfrak{N}(\Lambda)$ is valid.

$$\frac{}{\mathcal{R}, \Gamma, w : p(\vec{\underline{a}}) \Rightarrow u : p(\vec{\underline{a}}), \Delta}\,(id_*) \quad \rightsquigarrow \quad \frac{}{X[Y_1, p(\vec{\underline{a}}) \Rightarrow Z_1]_w[Y_2 \Rightarrow p(\vec{\underline{a}}), Z_2]_u}\,(id_*)$$

Inductive step. We consider the $(Pr_\supset)$, $(\exists^{\mathsf{QC}}_l)$, and $(\exists^{\mathsf{QC}}_r)$ cases; all other cases are straightforward or similar.

$(Pr_\supset)$. Let the left premise of the $(Pr_\supset)$ inference shown below top be Λ_1 and the right premise be Λ_2. We assume that the left premise of the $(Pr_\supset)$ inference shown below bottom is $\mathfrak{N}(\Lambda_1)$ and the right premise is $\mathfrak{N}(\Lambda_2)$. Due to the application of $(Pr_\supset)$ shown below top, we know that there exists a propagation path $\pi(w, u)$ in $PG(\Lambda_1)$ and $PG(\Lambda_2)$ such that $s_\pi(w, u) \in L_{S4}(a)$. By Lem. 59, we know that the propagation path exists in $PG(\mathfrak{N}(\Lambda_1))$ and $PG(\mathfrak{N}(\Lambda_2))$ as well, and so, $(Pr_\supset)$ may be applied to $\mathfrak{N}(\Lambda_1)$ and $\mathfrak{N}(\Lambda_2)$ as shown below bottom.

$$\frac{\mathcal{R}, w : \phi \supset \psi, \Gamma \Rightarrow \Delta, u : \phi \qquad \mathcal{R}, w : \phi \supset \psi, u : \psi, \Gamma \Rightarrow \Delta}{\mathcal{R}, w : \phi \supset \psi, \Gamma \Rightarrow \Delta}\,(Pr_\supset) \quad \rightsquigarrow$$

$$\frac{X[Y_1, \phi \supset \psi \Rightarrow Z_1]_w[Y_2 \Rightarrow \phi, Z_2]_u \qquad X[Y_1, \phi \supset \psi \Rightarrow Z_1]_w[Y_2, \psi \Rightarrow Z_2]_u}{X[Y_1, \phi \supset \psi \Rightarrow Z_1]_w[Y_2 \Rightarrow Z_2]_u}\,(Pr_\supset)$$

$(\exists^{\mathsf{QC}}_l)$. Let Λ be the premise of the left inference below, and $\mathfrak{N}(\Lambda)$ be the premise of the right inference. By assumption, we know that $\underline{a}$ is an eigenvariable in Λ, and so, by Def. 81, $\underline{a}$ is an eigenvariable in $\mathfrak{N}(\Lambda)$. Hence, we may apply the $(\exists^{\mathsf{QC}}_l)$ rule to $\mathfrak{N}(\Lambda)$, giving the desired conclusion as shown below right.

$$\frac{\mathcal{R}, \Gamma, w : \phi(\underline{a}/x) \Rightarrow \Delta}{\mathcal{R}, \Gamma, w : \exists x \phi \Rightarrow \Delta}\,(\exists^{\mathsf{QC}}_l) \quad \rightsquigarrow \quad \frac{X[Y, \phi(\underline{a}/x) \Rightarrow Z]}{X[Y, \exists x \phi \Rightarrow Z]}\,(\exists^{\mathsf{QC}}_l)$$

$(\exists^{\mathsf{QC}}_r)$. Let the premise of the left inference be Λ and the premise of the right inference be $\mathfrak{N}(\Lambda)$. By the side condition on the left $(\exists^{\mathsf{QC}}_l)$ inference, we know that either $\underline{a}$ is an eigenvariable, or $\underline{a}$ is $S5$-available for w. In the first case, by the definition of $\mathfrak{N}$ (Def. 81), we know that $\underline{a}$ will be an eigenvariable in $\mathfrak{N}(\Lambda)$ as well. In the second case, by Lem. 59, $\underline{a}$ will be $S5$-available. Regardless of the case then, the side condition holds, and we may apply the rule $(\exists^{\mathsf{QC}}_l)$ to $\mathfrak{N}(\Lambda)$ as shown below right, giving the desired conclusion.

$$\frac{\mathcal{R}, \Gamma \Rightarrow \Delta, w : \phi(\underline{a}/x), w : \exists x \phi}{\mathcal{R}, \Gamma \Rightarrow \Delta, w : \exists x \phi} (\exists_r^{\mathsf{QC}}) \quad \rightsquigarrow \quad \frac{X[Y \Rightarrow \phi(\underline{a}/x), \exists x \phi, Z]_w}{X[Y \Rightarrow \exists x \phi, Z]_w} (\exists_r^{\mathsf{QC}})$$

□

Theorem 34. *Let* $\mathsf{X} \in \{\mathsf{Q}, \mathsf{QC}\}$. *Every derivation in* DIntX *is algorithmically translatable to a derivation in* IntXL.

Proof. The theorem is proven in a similar fashion as Thm. 33 above. One proves by induction on the height of the given derivation that each rule in DIntQ and DIntQC can be translated into an instance of the corresponding rule in IntQL and IntQCL, respectively. Similar to the proof of Thm. 33, each rule straightforwardly translates (under the $\mathfrak{L}$ function). □

Example 20. *If the top derivation below (in* IntQCL*) is input into the translation algorithm of Thm. 33, then we obtain the derivation shown below bottom. On the other hand, the derivation below bottom (in* DIntQC*) translates via Thm. 34 to the derivation shown below top.*

Also, obverse that the $(\forall_l^{\mathsf{QC}})$ *rule may be applied in the top derivation because* $\underline{a}$ *is* $S5$*-available for* u*, that is, we have the labelled formula* $v : p(\underline{a})$ *and the propagation path* $\pi(v, u) := v, \overline{a}, u$ *such that* $\overline{a} \in L_{S5}(a)$*. Similarly, the side condition holds in the* DIntQC *derivation, so the* $(\forall_l^{\mathsf{QC}})$ *rule may be applied there as well. It should be pointed out that in the non-constant domain setting, the theorem shown below (which is an instance of the constant domain axiom A13) is not derivable because the instance of* $(\forall_l^{\mathsf{QC}})$ *cannot be replaced by an instance of* $(\forall_l^{\mathsf{Q}})$*. The inability to substitute* $(\forall_l^{\mathsf{Q}})$ *for* $(\forall_l^{\mathsf{QC}})$ *is due to the fact that the string of every propagation path from* v *to* u *must contain at least one occurrence of the character* $\overline{a}$*, and so, the string will never be in* $L_{S4}(a)$.

$$\Lambda_1 := w \leq u, u \leq v, u : \forall x(p(x) \lor q), u : p(\underline{a}) \Rightarrow v : p(\underline{a}), u : q$$

$$\Lambda_2 := w \leq u, u \leq v, u : \forall x(p(x) \lor q), u : q \Rightarrow v : p(\underline{a}), u : q$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Lambda_1}(id_*) \quad \dfrac{}{\Lambda_2}(id_*)}{w \leq u, u \leq v, u : \forall x(p(x) \lor q), u : p(\underline{a}) \lor q \Rightarrow v : p(\underline{a}), u : q}(\lor_l)}{w \leq u, u \leq v, u : \forall x(p(x) \lor q) \Rightarrow v : p(\underline{a}), u : q}(\forall_l^{\mathsf{QC}})}{w \leq u, u : \forall x(p(x) \lor q) \Rightarrow u : \forall x p(x), u : q}(\forall_r^{\mathsf{QC}})}{w \leq u, u : \forall x(p(x) \lor q) \Rightarrow u : \forall x p(x) \lor q}(\lor_r)}{\epsilon \Rightarrow w : \forall x(p(x) \lor q) \supset \forall x p(x) \lor q}(\supset_r)$$

$$X_1 := \epsilon \Rightarrow \epsilon, \{\forall x(p(x) \lor q), p(\underline{a}) \Rightarrow q, \{\epsilon \Rightarrow p(\underline{a})\}\}$$

$$X_2 := \epsilon \Rightarrow \epsilon, \{\forall x(p(x) \lor q), q \Rightarrow q, \{\epsilon \Rightarrow p(\underline{a})\}\}$$

$$\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{}{X_1}\,(id_*) \qquad \cfrac{}{X_2}\,(id_*)}{\epsilon \Rightarrow \epsilon, \{\forall x(p(x) \vee q), p(\underline{a}) \vee q \Rightarrow q, \{\epsilon \Rightarrow p(\underline{a})\}\}}\,(\vee_l)}{\epsilon \Rightarrow \epsilon, \{\forall x(p(x) \vee q) \Rightarrow q, \{\epsilon \Rightarrow p(\underline{a})\}\}}\,(\forall_l^{\mathsf{QC}})}{\epsilon \Rightarrow \epsilon, \{\forall x(p(x) \vee q) \Rightarrow \forall x p(x), q\}}\,(\forall_r^{\mathsf{QC}})}{\epsilon \Rightarrow \epsilon, \{\forall x(p(x) \vee q) \Rightarrow \forall x p(x) \vee q\}}\,(\vee_r)}{\epsilon \Rightarrow \forall x(p(x) \vee q) \supset \forall x p(x) \vee q}\,(\supset_r)$$

CHAPTER 6

Applications: Decidability and Interpolation

In this chapter, we put our refined labelled calculi for deontic STIT logics and grammar logics to work. In Sect. 6.1, we build off of the author's work in [LvB19] and show how each calculus $\mathsf{DS}_0^k\mathsf{L}$ for the single-agent deontic STIT logic DS_0^k (with $n = 0$, i.e. $\mathrm{Ag} = \{0\}$) can be utilized in a proof-search procedure to decide the validity of formulae. As is typical when proving decidability via proof-search, we write an algorithm that applies the rules of $\mathsf{DS}_0^k\mathsf{L}$ in reverse, attempting to construct a proof of the input formula. As a corollary, we will not only obtain decidability for each logic DS_0^k, but will also confirm that each logic DS_0^k is in possession of the *finite model property*, i.e. any formula that is satisfiable, is satisfiable on a finite model. We note that we do not provide proof-search procedures for first-order intuitionistic logics since they are undecidable and do not provide proof-search algorithms for grammar logics either since such algorithms were already provided for a decidable sub-class of such logics in [TIG12]. (NB. Context-free grammar logics with converse are undecidable in general.)

In Sect. 6.2, the syntactic method of interpolation for nested sequent systems introduced in [LTGC20] is presented and applied to show that each context-free grammar logic with converse possesses the effective Lyndon interpolation property. At the beginning of Sect. 6.2, we define (Lyndon) interpolation, discuss the importance of the property, and also describe the state of the art in the proof-theoretic approach to confirming the property. Last, we note that the new results of the second section (i.e. Sect. 6.2) generalize those of [GN05] from the class of regular grammar logics to the class of context-free grammar logics with converse (and from effective Craig interpolation to effective Lyndon interpolation).

6.1 Proof-Search and Decidability for Deontic STIT Logics

Proof-search algorithms are special procedures defined relative to proof systems that construct or find the proof of an input formula (or, sequent) by applying the rules of the proof system in reverse. Such algorithms are often used to prove logics decidable, but are also indispensable in the domain of automated reasoning—having been used to decide knowledge representation languages [Rad12], to provide complexity-optimal decision procedures [LP15, Vig00], and to automate the extraction of counter-models [LvB19, TIG12], among other things.

The proof-search algorithms we introduce in this section leverage refined labelled calculi from the class $\{\mathsf{DS}^{k}_{0}\mathsf{L} \mid k \in \mathbb{N}\}$ (see Fig. 4.4). The algorithms operate by taking a labelled sequent Λ as input, and check (i) if a labelled formula $w : \phi$ and its negation $w : \neg\phi$ occur in Λ, and if not, then check (ii) if a complex formula exists in Λ that has not yet been analyzed. Regarding point (i), if a labelled formula and its negation occur in the input labelled sequent, then by Lem. 19 and Thm. 22, we know the sequent is provable, meaning that the algorithm no longer needs to search for a proof. Regarding point (ii), if a complex formula has yet to be analyzed, then this leaves open the possibility that a proof of the labelled sequent may be found by applying the appropriate rule (bottom-up) to the input labelled sequent which analyzes the complex formula into the simpler auxiliary formulae from which it may be formed. It is important to point out that in our setting, our proof-search algorithms are performing two tasks simultaneously—searching for a proof of the input labelled sequent and constructing a counter-model of the input labelled sequent in case a proof is not found. To demonstrate this dualistic functionality of our proof-search procedure, we consider a concrete example of proof-search using a refined labelled calculus for a deontic STIT logic.

Let us suppose for the sake of the example that we have a single-agent (i.e. $n = 0$ and $Ag := \{0\}$) and that the agent's choices are not bounded (that is, $k = 0$). We want to search for a proof or counter-model of the labelled sequent $\epsilon \Rightarrow w : [0] \otimes_0 (p \vee \neg q)$ using the calculus $\mathsf{DS}^{0}_{0}\mathsf{L}$. In attempt to construct a proof, we apply relevant rules to the sequent in reverse, yielding the derivation shown below. Also, note that when applying rules in reverse we preserve the analyzed (principal) formula of the inference into the premise forcing us to invoke the admissibility of (ctr_r) (Cor. 4) before each inference. The reason for doing this is that if a counter-model is to be constructed from failed proof-search, then having the formula present simplifies the proof confirming that the counter-model of the end sequent is indeed a counter-model; this will become apparent when showing the correctness of our proof-search procedures in Thm. 35 later on.

$$\begin{array}{cl}
R_{[0]}wu, I_{\otimes_0}v \Rightarrow w : [0] \otimes_0 (p \lor \neg q), u : \otimes_0(p \lor \neg q), v : p \lor \neg q, v : p, v : \neg q & \\
\hline
R_{[0]}wu, I_{\otimes_0}v \Rightarrow w : [0] \otimes_0 (p \lor \neg q), u : \otimes_0(p \lor \neg q), v : p \lor \neg q, v : p \lor \neg q & (\lor_r) \\
\hdashline
R_{[0]}wu, I_{\otimes_0}v \Rightarrow w : [0] \otimes_0 (p \lor \neg q), u : \otimes_0(p \lor \neg q), v : p \lor \neg q & (ctr_r) \\
\hline
R_{[0]}wu \Rightarrow w : [0] \otimes_0 (p \lor \neg q), u : \otimes_0(p \lor \neg q), u : \otimes_0(p \lor \neg q) & (\otimes_0) \\
\hdashline
R_{[0]}wu \Rightarrow w : [0] \otimes_0 (p \lor \neg q), u : \otimes_0(p \lor \neg q) & (ctr_r) \\
\hline
\epsilon \Rightarrow w : [0] \otimes_0 (p \lor \neg q), w : [0] \otimes_0 (p \lor \neg q) & ([0]) \\
\hdashline
\epsilon \Rightarrow w : [0] \otimes_0 (p \lor \neg q) & (ctr_r)
\end{array}$$

Despite the fact that each principal formula is preserved upward, it happens to be sufficient (for building a proof or counter-model of the end sequent) to analyze (i.e. apply an inference rule bottom-up to) each complex labelled formula only once. By this fact, our proof-search example will terminate with the derivation above since each complex labelled formula occurring in the top sequent was analyzed at some point lower in the derivation. Since the above 'derivation' is not a valid derivation in $\mathsf{DS}^0_0\mathsf{L}$, as the top sequent is not an initial sequent, we can extract a counter-model for the end sequent from the top seqeunt, which we now explain.

A nice feature of the (refined) labelled formalism is that its close association with the semantics of the logic allows labelled sequents to be readily converted into models. In the current example then, our aim is to transform the top sequent of the derivation above into a DS^0_0-model $M = (W, R_{[0]}, I_{\otimes_0}, V)$. We define W to be the set of labels, i.e. $W := \{w, u, v\}$. To define $R_{[0]}$, we make use of the relational atom $R_{[0]}wu$ and the set of worlds W, forming a set $\{(w, w), (u, u), (v, v), (w, u)\}$ that contains a reflexive pair for each world in W (i.e. each label in the top sequent), and a pair from each relevant relational atom, which in this case just provides (w, u) due to the occurrence of $R_{[0]}wu$. We then take the transitive and symmetric closure of the set, thus giving $R_{[0]} := \{(w, w), (w, u), (u, w), (u, u), (v, v)\}$, which ensures that the relation is an equivalence relation that partitions W as dictated by condition $\mathbf{(S_1)}$ (Def. 24). We define $I_{\otimes_0}$ to be all labels in the choice-cell (from $R_{[0]}$) of an 'ideal label'; since the only 'ideal label' is v due to the $I_{\otimes_0}v$ occurring in the top sequent, and because the only label in the choice-cell of v is v, we have that $I_{\otimes_0} := \{v\}$. Last, we define the valuation function V so that it invalidates each literal at the world (i.e. label) that prefixes it; thus, we have $V(p) := \{w, u\}$ and $V(q) := \{v\}$. It is straightforward to verify that M is a DS^0_0-model and satisfies conditions $\mathbf{(S_1)}$–$\mathbf{(S_3)}$ and $\mathbf{(D_1)}$–$\mathbf{(D_3)}$.

We can see that M is a counter-model for $[0] \otimes_0 (p \lor \neg q)$ via the following calculation: Since $v \not\in V(p)$ and $v \in V(q)$, it follows that $M, v \not\Vdash p$ and $M, v \not\Vdash \neg q$, which further implies that $M, v \not\Vdash p \lor \neg q$. This fact, in conjunction with the fact that $v \in I_{\otimes_0}$, implies that $M, u \not\Vdash \otimes_0(p \lor \neg q)$, which entails that $M, w \not\Vdash [0] \otimes_0 (p \lor \neg q)$, when one takes into account that $u \in R_{[0]}(w)$. Through failed proof-search then, we have shown that the top sequent of the above derivation can be used to construct a counter-model of the end sequent. Let us now move on to show that such operations can be performed generally.

6.1.1 The Proof-Search Algorithm $\texttt{ProveDS}^{\texttt{k}}$ and its Corollaries

We provide a correct and terminating proof-search procedure $\texttt{ProveDS}^{\texttt{k}}$ (Alg. 6.1 below), which harnesses each refined labelled calculus $\mathsf{DS}_0^k\mathsf{L}$ to decide its associated logic DS_0^k. We only consider the single-agent setting where $n = 0$ (i.e. $\mathrm{Ag} = \{0\}$) and use 0 to denote the single-agent throughout the course of the section. Despite this restriction, we still allow for choice limitation, and therefore the proof-search algorithm $\texttt{ProveDS}^{\texttt{k}}$ takes a parameter k limiting the number of choices of the agent 0 to a maximum of k (when $k > 0$), and imposes no limitation when $k = 0$ (that is, the agent 0 may have an arbitrary number of choices available when $k = 0$). The single-agent setting greatly simplifies proof-search as the independence of agents rule (IOA) becomes redundant and no longer needs to be considered during proof-search as demonstrated by the following lemma:[1]

Lemma 60. *For each $k \in \mathbb{N}$, the (IOA) rule is eliminable in DS_0^k.*

Proof. Let Π be a derivation in $\mathsf{DS}_0^k\mathsf{L}$ containing some number of (IOA) inferences. We show how to eliminate topmost occurrences of the (IOA) rule, which, through repeated application yields a proof free of such inferences. Therefore, let us consider a topmost occurrence of (IOA) as shown below left. We invoke admissibility of (lsb) (Cor. 4) as shown below right to replace the eigenvariable u with the label w, followed by the eliminability of (ref_0) (Lem. 33) to obtain a derivation without the (IOA) inference. It is important to note two things: (i) the admissibility of (lsb) and the eliminability of (ref_0) do not introduce instances of (IOA), and (ii) although the resulting derivation may increase in size, since the number of (IOA) inferences has decreased, through repeated application of the explained procedure we still obtain a proof free of (IOA) inferences.

$$\frac{\mathcal{R}, R_{[0]}wu \Rightarrow \Gamma}{\mathcal{R} \Rightarrow \Gamma}\,(IOA) \quad \rightsquigarrow \quad \cfrac{\cfrac{\mathcal{R}, R_{[0]}wu \Rightarrow \Gamma}{\mathcal{R}, R_{[0]}ww \Rightarrow \Gamma}\,(lsb)}{\mathcal{R} \Rightarrow \Gamma}\,(ref_0)$$

□

As discussed at the end of Sect. 4.2, bottom-up applications of the (IOA) rule introduce edges in the sequent graph of a labelled sequent that all meet at a fresh vertex, ensuring that the resulting labelled sequent is not a labelled forest sequent. In the single-agent setting, such structures do not appear, and furthermore, the redundancy of the (IOA) rule implies that we need not consider the rule whatsoever during proof-search. As we will see (in Lem. 61 below), this has the desirable consequence that we are permitted to restrict ourselves to considering only labelled forest derivations during proof-search—an insight which proves advantageous in designing our proof-search procedure. Since labelled

[1] We note that proof-search algorithms can be given for the multi-agent case as well, but require more sophisticated methods. We only consider the single-agent case here as it simplifies our work and is closely related to the published results in [LvB19] on proof-search for single-agent (non-deontic) STIT logics.

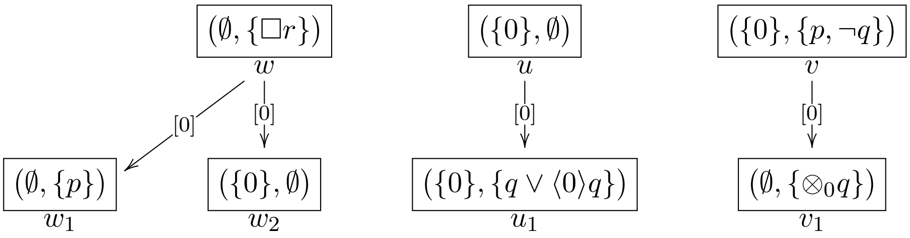
forest sequents (which compose labelled forest derivations) play a crucial role in our proof-search methodology, we introduce the relevant concept of a *choice-tree* and explain the semantic meaning of such an object in Rmk. 4 below.

Definition 83 (Choice-Tree)**.** *Let Λ be a labelled forest sequent with sequent graph $G(\Lambda) = (V, E, L)$. We refer to each tree in $G(\Lambda)$ (which is a forest since Λ is a labelled forest sequent) as a* choice-tree *and if $w \in V$, we let $\mathrm{CT}(w)$ denote the choice-tree to which w belongs. Last, we use $\mathrm{Lab}(\mathrm{CT}(w))$ to represent the set of all vertices (i.e. labels) occurring in the choice-tree $\mathrm{CT}(w)$.*

Example 21. *We provide an example of a labelled forest sequent Λ below along with a pictorial representation of its sequent graph $G(\Lambda)$.*

$$\Lambda := I_{\otimes_0} w_2, I_{\otimes_0} u, I_{\otimes_0} u_1, I_{\otimes_0} v, R_{[0]} ww_1, R_{[0]} ww_2, R_{[0]} uu_1, R_{[0]} vv_1 \Rightarrow$$

$$w : \Box r, w_1 : p, u_1 : q \vee \langle 0 \rangle q, v : p, v : \neg q, v_1 : \otimes_0 q$$



The labels w, u, and v are the roots of their respective choice-trees that make up the forest. The choice-tree above left is equally denoted by $\mathrm{CT}(w)$, $\mathrm{CT}(w_1)$, and $\mathrm{CT}(w_2)$, the choice-tree above middle is equally denoted by $\mathrm{CT}(u)$ and $\mathrm{CT}(u_1)$, and the choice-tree above right is equally denoted by $\mathrm{CT}(v)$ and $\mathrm{CT}(v_1)$. Furthermore, we have $\mathrm{Lab}(\mathrm{CT}(w)) = \mathrm{Lab}(\mathrm{CT}(w_1)) = \mathrm{Lab}(\mathrm{CT}(w_2)) = \{w, w_1, w_2\}$, $\mathrm{Lab}(\mathrm{CT}(u)) = \mathrm{Lab}(\mathrm{CT}(u_1)) = \{u, u_1\}$, and $\mathrm{Lab}(\mathrm{CT}(v)) = \mathrm{Lab}(\mathrm{CT}(v_1)) = \{v, v_1\}$.

Remark 4. *Each choice-tree that occurs in the sequent graph of a labelled forest sequent is a syntactic representation of a choice-cell for agent 0, that is, it represents an equivalence class in $R_{[0]}$ in DS^k_0-model. This insight tells us that if agent 0 is restricted to a maximum of $k > 0$ choices and there are $m > k$ choice-trees in the sequent graph of a labelled forest sequent, then at least two choice-trees must correspond to the same choice-cell. We use this observation to specify how (APC^k_0) is (bottom-up) applied in our proof-search algorithm.*

We now introduce blocking conditions (similar to those used in [TIG12]), which will be employed in our proof-search procedure. A naïve approach to proof-search would simply apply inference rules in reverse on an input formula in attempt to construct a proof. Yet, such a strategy often leads to a non-terminating proof-search algorithm, suggesting that our algorithm must be properly equipped to determine when a rule should and should not be applied; for example, the ($\langle 0 \rangle$) rule can be bottom-up applied an infinite number of times to the labelled sequent $R_{[0]}wu \Rightarrow w : \langle 0 \rangle p$. Blocking conditions are essential in this regard, as such conditions allow for a (bottom-up) rule application when false, and prohibit such an application when true. Since each blocking condition is associated with the application of a single rule, once all such conditions are true, it becomes clear that none of the inference rules are bottom-up applicable, and proof-search may safely terminate.

We introduce four sets of blocking conditions in Def. 84 – 87 below. The first set of blocking conditions are the *saturation conditions*, which determine if a derivable sequent has been reached during proof-search and if the $(\vee_r)$ and $(\wedge_r)$ rules are bottom-up applicable. The second set of blocking conditions—the *realization conditions*—dictate the applicability of the $(\Box)$, $([0])$, and $(\otimes_0)$ rules, while the third set of *propagation conditions* dictate the applicability of the $(\Diamond)$, $(Pr_{\langle 0 \rangle})$, and $(Pr^1_{\ominus_0})$ rules. Last, the notion of being $\mathbf{(D_2)}$-satisfied and the notion of being $\mathbf{(S_3)}$-satisfied (composing the fourth set of blocking conditions below) govern bottom-up applications of the $(Pr^2_{\ominus_0})$ and (APC^k_0) rules, respectively. Last, we note that the terminology and blocking conditions used here are based on and motivated by the terminology and blocking conditions of [TIG12].

Definition 84 (Saturation). *Let* $\Lambda := \mathcal{R} \Rightarrow \Gamma$ *be a labelled forest sequent with* $w \in \mathrm{Lab}(\Lambda)$*. The label* w *is* saturated *iff (i) for all* $w : \phi \in \Gamma$, $w : \neg\phi \not\in \Gamma$*, (ii) for all* $w : \phi \vee \psi \in \Gamma$*, both* $w : \phi \in \Gamma$ *and* $w : \psi \in \Gamma$*, (iii) for all* $w : \phi \wedge \psi \in \Gamma$*, either* $w : \phi \in \Gamma$ *or* $w : \psi \in \Gamma$*.*

Definition 85 ($\Box$-, $[0]$, $\otimes_0$-realization). *Let* $\Lambda := \mathcal{R} \Rightarrow \Gamma$ *be a labelled forest sequent with* $w \in \mathrm{Lab}(\Lambda)$*.*

- *The label* w *is* $\Box$-realized *iff for every* $w : \Box\phi \in \Gamma$*, there exists a label* $u \in \mathrm{Lab}(\Lambda)$ *such that* $u : \phi \in \Gamma$*.*
- *The label* w *is* $[0]$-realized *iff for every* $w : [0]\phi \in \Gamma$*, there exists a label* $u \in \mathrm{Lab}(\Lambda)$ *such that* $u \in \mathrm{CT}(w)$ *and* $u : \phi \in \Gamma$*.*
- *The label* w *is* $\otimes_0$-realized *iff for every* $w : \otimes_0\phi \in \Lambda$*, there exists a label* $u \in \mathrm{Lab}(\Lambda)$ *such that* $I_{\otimes_0}u \in \mathcal{R}$ *and* $u : \phi \in \Gamma$*.*

Definition 86 ($\Diamond$-, $\langle 0 \rangle$-, $\ominus_0$-propagated). *Let* Λ *be a labelled forest sequent with* $w \in \mathrm{Lab}(\Lambda)$*.*

- *The label* w *is* $\Diamond$-propagated *iff for every* $w : \Diamond\phi \in \Gamma$*, we have* $u : \phi \in \Gamma$ *for all* $u \in \mathrm{Lab}(\Lambda)$*.*
- *The label* w *is* $\langle 0 \rangle$-propagated *iff for every* $w : \langle 0 \rangle\phi \in \Gamma$*, we have* $u : \phi \in \Gamma$ *for all labels* $u \in \mathrm{Lab}(\Lambda)$ *such that* $w \sim^{\mathcal{R}}_0 u$*.*

- ▶ *The label w is* $\ominus_0$-propagated *iff for every $w : \ominus_0 \phi \in \Gamma$, we have $v : \phi \in \Gamma$ for all labels $v \in \mathrm{Lab}(\Lambda)$ such that there exists a label $u \in \mathrm{Lab}(\Lambda)$ with $I_{\otimes_0} u \in \mathcal{R}$ and $u \sim_0^{\mathcal{R}} v$.*

Definition 87 (**($\mathbf{D_2}$)**-satisfied, **($\mathbf{S_3}$)**-satisfied)**.** *Let $\Lambda := \mathcal{R} \Rightarrow \Gamma$ be a labelled forest sequent with $n > 0$. We say that Λ is* **($\mathbf{D_2}$)**-satisfied *iff for all $w : \ominus_0 \phi \in \Gamma$, there exists a label $u \in \mathrm{Lab}(\Lambda)$ such that $I_{\otimes_0} u, u : \phi \in \mathcal{R}, \Gamma$. We say that Λ is* **($\mathbf{S_3}$)**-satisfied *iff $G(\Lambda)$ contains at most n-many choice-trees.*

As mentioned above, if a labelled sequent satisfies all blocking conditions, then the proof-search algorithm no longer needs to continue building a proof above that sequent and can terminate on that branch of the computation. When a sequent generated during the course of proof-search enters such a state we say that the sequent is *stable* (cf. [TIG12]). The formal definition of stability is as follows:

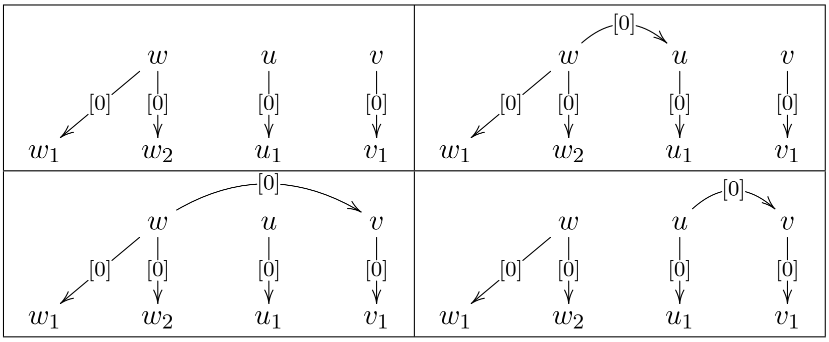
Definition 88 (Stability)**.** *A forestlike labelled sequent Λ is* stable *iff (i) all labels w in Λ are saturated, (ii) all labels are $\Box$-, $[0]$-, $\otimes_0$-realized, (iii) all labels are $\Diamond$-, $\langle 0 \rangle$-, $\ominus_0$-propagated, and (iv) Λ is **(D_2)**-satisfied, (v) Λ is **(S_3)**-satisfied.*

We have now laid the necessary groundwork to write our proof-search algorithm $\mathtt{ProveDS^k}$, i.e. Alg. 6.1 below. Due to the length of the algorithm, its instructions have been split into two parts occurring on different pages. The algorithm works by sequentially checking if an input labelled sequent satisfies each blocking condition, and if so, then the sequent will be stable by Def. 88 above, and the procedure will return `False` for that input labelled sequent. If, on the other hand, the input labelled sequent does not satisfy some blocking condition, then the labelled sequent contains a complex labelled formula that has not yet been analyzed, leaving open the possibility that a proof of the initial input formula may still be found; in such a situation, $\mathtt{ProveDS^k}$ effectively applies the relevant rule (bottom-up) yielding a new labelled sequent (or, new labelled sequents) that is (are) recursively input back into the proof-search algorithm. Note that due to instructions 10–20 and 49–57, more than one recursive call may be made, corresponding to the branching rules $(\wedge_r)$ and (APC_0^k), and which attempts to build proofs for the premises of each rule instance. If all branches of the computation return `True`, then a proof of the input labelled sequent has been found, but if a single branch returns `False`, then the input labelled sequent is not provable as is shown in the correctness theorem (Thm. 35) below. The reader should also take note of the following remark, which explains how to modify $\mathtt{ProveDS^k}$ when $k = 0$.

Remark 5. *If $k = 0$, meaning that the agent 0's choices are unbounded, then we omit instructions 49–57 from the proof-search algorithm. Recall from Fig. 4.4 (introducing each refined labelled calculus $\mathsf{DS}_n^k\mathsf{L}$) that the choice-limitation rule (APC_i^k) is omitted when $k = 0$. Since instructions 49–57 govern bottom-up applications of (APC_0^k) during proof-search, the instructions are unnecessary when $k = 0$, justifying their exclusion in this case.*

It is significant how $\texttt{ProveDS}^{\texttt{k}}$ handles the (APC^{k}_{0}) rule (see instructions 49–57 in Alg. 6.1). As explained in Rmk. 4 above, each choice-tree in the sequent graph of a labelled forest sequent is a syntactic representation of a choice-cell. If at some step of the computation $\texttt{ProveDS}^{\texttt{k}}$ generates a labelled forest sequent whose sequent graph contains $m > k$ choice-trees, then we know that at least two of the choice-trees must represent the same choice-cell. The means by which we syntactically encode that two choice-trees represent the same choice-cell is by connecting them with an edge, and in fact, this is the operation performed by instructions 49–57. However, since we do not know which two choice-trees represent the same choice-cell we must try all possible connections as dictated by the (APC^{k}_{0}) rule when applied bottom-up. It is important to point out that in order to preserve the forest structure of a labelled sequent, we cannot simply allow for two choice-trees to be arbitrarily connected in $\texttt{ProveDS}^{\texttt{k}}$. Rather, to preserve the forest structure of labelled sequents throughout the computation of $\texttt{ProveDS}^{\texttt{k}}$, we connect two choice-trees by introducing an edge from the root of one to the root of the other, which begets a new tree combining the previous two. To provide more intuition regarding bottom-up applications of the (APC^{k}_{0}) rule (corresponding to instructions 49–57 in $\texttt{ProveDS}^{\texttt{k}}$), we present an example below showing how (APC^{2}_{0}) is bottom-up applied to a labelled forest sequent.

Example 22. *Let us consider the labelled forest sequent Λ from Ex. 21 above. To improve readability and focus only on essential details, we omit the decoration of the vertices in our sequent graphs. The sequent graph $G(\Lambda)$ (without decorated vertices) is shown below top-left. If we were to apply the (APC^{2}_{0}) rule bottom-up to Λ (as dicatated by instructions 49–57 in* $\texttt{ProveDS}^{\texttt{k}}$*), then the other three labelled forest sequents (whose non-decorated sequent graphs are shown below) will result. Notice that the sequent graphs of the newly generated labelled forest sequents have two choice-trees instead of three, and that by connecting the root of one choice-tree to the root of another, we preserve the forest structure in the output.*



Since bottom-up applications of (APC^{k}_{0}) rely on our input labelled sequent being a labelled forest sequent, it is necessary to confirm that all labelled sequents generated throughout the course of the computation of $\texttt{ProveDS}^{\texttt{k}}$ are labelled forest sequents (since otherwise, instructions 49–57 will be nonsensical). We argue this result in Lem. 61 below.

Furthermore, this fact, in conjunction with the correctness and termination theorems below (Thm. 35 and 36, resp.), has the interesting consequence that each calculus $\mathsf{DS}_0^k\mathsf{L}$ is complete relative to labelled forest derivations with the rooted property (Cor. 8). This result tells us that refinement has brought about a reduction in the underlying data structure employed in labelled sequents occurring in derivations (cf. Thm. 16), though, this fact is observed and confirmed via our proof-search algorithm.

Lemma 61. *Let $\phi \in \mathcal{L}_{\mathsf{DS}_n^k}$. Every labelled sequent generated throughout the course of computing* $\texttt{ProveDS}^{\texttt{k}}(\epsilon \Rightarrow w : \phi)$ *is a labelled forest sequent.*

Proof. By Def. 56, we know that $w : \phi$ is a labelled forest sequent since it's graph is of the form $G(\epsilon \Rightarrow w : \phi) = (\{w\}, \emptyset, \{(w, \{\phi\})\})$, which is a single point. Observe that only conditional statements in $\texttt{ProveDS}^{\texttt{k}}$ that add new nodes or edges (when the condition holds true) to the sequent graph of the input labelled sequent are those given on lines 21–24, 25–28, 29–32, and 45–48 in the algorithm corresponding to bottom-up applications of the $(\Box)$, $([0])$, $(\ominus_0)$, and $(Pr^2_{\ominus_0})$ rules, respectively; with the exception of the instruction block 49–57, all other instructions preserve the forest structure inherent in the input labelled forest sequent because they only change the labeling function L of the sequent graph of the input labelled sequent (i.e. such rules only redecorate vertices). Instructions 21–24, 29–32, and 45–48 add new disconnected vertices to the sequent graph of the input labelled sequent, which serve as new roots of new choice-trees, thus preserving the forest structure. The instructions 25–28 add a new edge to a fresh vertex in a choice-tree of the sequent graph of the input labelled sequent, which preserves the forest structure because only an additional branch is added to some choice-tree. The instructions 49–57 connect the root of one choice-tree to the root of another choice-tree (as explained in Ex. 22 above), thus yielding a new choice-tree (that subsumes the original two from which it was formed) in the sequent graph of the input labelled sequent. Therefore, every labelled sequent generated throughout the course of $\texttt{ProveDS}^{\texttt{k}}(\epsilon \Rightarrow w : \phi)$ will be a labelled forest sequent since all instructions preserve the forest structure of any input labelled forest sequent. □

We are now in a position to prove the correctness and termination of our proof-search algorithm $\texttt{ProveDS}^{\texttt{k}}$. Before showing each of these results, we prove a useful lemma relating choice-trees to undirected 0-paths:

Lemma 62. *Let $\Lambda := \mathcal{R} \Rightarrow \Gamma$ be a labelled forest sequent with $w, u \in \mathrm{Lab}(\Lambda)$, then $u \in \mathrm{CT}(w)$ iff $w \sim_0^{\mathcal{R}} u$.*

Proof. The forward direction is trivial since if $u \in \mathrm{CT}(w)$, then w and u occur in the same choice-tree, implying that there is some undirected 0-path between the two labels. The backward direction is simple as well: if Λ is a labelled forest sequent, then the only way there can be an undirected 0-path between two labels w and u is if they occur in the same choice-tree. □

Algorithm 6.1: $\mathtt{ProveDS^k}$

```
Input: A Labelled Sequent: $\mathcal{R} \Rightarrow \Gamma$
Output: A Boolean: True, False
if $w : \phi, w : \neg\phi \in \Gamma$ then
    return True;
end
if $\mathcal{R} \Rightarrow \Gamma$ is stable then
    return False;
end
if there exists a $w : \phi \vee \psi \in \Gamma$, but either $w : \phi \notin \Gamma$ or $w : \psi \notin \Gamma$ then
    Let $\Gamma' := w : \phi, w : \psi, \Gamma$;
    return ProveDS$^k$($\mathcal{R} \Rightarrow \Gamma'$);
end
if there exists a $w : \phi \wedge \psi \in \Gamma$, but $w : \phi, w : \psi \notin \Gamma$ then
    Let $\Gamma_1 := w : \phi, \Gamma$;
    Let $\Gamma_2 := w : \psi, \Gamma$;
    if ProveDS$^k$($\mathcal{R} \Rightarrow \Gamma_i$) = False for some $i \in \{1, 2\}$ then
        return False;
    end
    else
        return True;
    end
end
```

Theorem 35 (Correctness)**.** *Let $\phi \in \mathcal{L}_{\mathsf{DS}^k_n}$ and $k \in \mathbb{N}$.*

(i) If $\mathtt{ProveDS^k}(\epsilon \Rightarrow w : \phi)$ *returns* `True`*, then $\epsilon \Rightarrow w : \phi$ is $\mathsf{DS}^k_0\mathsf{L}$-provable.*

(ii) If $\mathtt{ProveDS^k}(\epsilon \Rightarrow w : \phi)$ *returns* `False`*, then $\epsilon \Rightarrow w : \phi$ is not $\mathsf{DS}^k_0\mathsf{L}$-provable and this is witnessed by a finite counter-model.*

Proof. (i) Every recursive call in $\mathtt{ProveDS^k}$ is a bottom-up application of a rule from $\mathsf{DS}^k_0\mathsf{L}$, and so, if $\mathtt{ProveDS^k}(\epsilon \Rightarrow w : \phi)$ returns `True`, then we obtain a proof of $\epsilon \Rightarrow w : \phi$ where all top sequents are of the form $\mathcal{R} \Rightarrow u : \psi, u : \neg\psi, \Gamma$. By Lem. 19 and Thm. 22, we know that all such labelled sequents have a derivation in $\mathsf{DS}^k_0\mathsf{L}$. Also, it should be noted that because the principal formula of a bottom-up inference (corresponding to a recursive call) is preserved upwards, it may be necessary to apply hp-admissibility of (ctr_r) (Lem. 23) to obtain the final proof.

(ii) Suppose that $\mathtt{ProveDS^k}(\epsilon \Rightarrow w : \phi)$ returns `False`. This implies that a stable labelled forest sequent $\Lambda := \mathcal{R} \Rightarrow \Gamma$ was generated. Let $G(\Lambda) := (V, E, L)$. We use Λ to define two counter-models $M = (W, R_{[0]}, I_{\otimes_0}, V)$ and $M' = (W', R'_{[0]}, I'_{\otimes_0}, V')$ for ϕ, depending

```
if some $w \in \text{Lab}(\mathcal{R} \Rightarrow \Gamma)$ is not $\Box$-realized then
    For a labelled formula $w : \Box\phi \in \Gamma$ such that $u : \phi \not\in \Gamma$ for all $u \in \text{Lab}(\mathcal{R} \Rightarrow \Gamma)$,
      let $\Gamma' := v : \phi, \Gamma$ with $v$ fresh;
    return ProveDS$^{\mathsf{k}}$($\mathcal{R} \Rightarrow \Gamma'$);
end
if some $w \in \text{Lab}(\mathcal{R} \Rightarrow \Gamma)$ is not $[0]$-realized then
    For a labelled formula $w : [0]\phi \in \Gamma$ such that $R_{[0]}wu, u : \phi \not\in \mathcal{R}, \Gamma$ for all
      $u \in \text{Lab}(\mathcal{R} \Rightarrow \Gamma)$, let $\mathcal{R}' := R_{[0]}wv$ and $\Gamma' := v : \phi, \Gamma$ with $v$ fresh;
    return ProveDS$^{\mathsf{k}}$($\mathcal{R}' \Rightarrow \Gamma'$);
end
if some $w \in \text{Lab}(\mathcal{R} \Rightarrow \Gamma)$ is not $\otimes_0$-realized then
    For a labelled formula $w : \otimes_0\phi \in \Gamma$ such that $I_{\otimes_0}u, u : \phi \not\in \mathcal{R}, \Gamma$ for all
      $u \in \text{Lab}(\mathcal{R} \Rightarrow \Gamma)$, let $\mathcal{R}' := I_{\otimes_0}v, \mathcal{R}$ and $\Gamma' := v : \phi, \Gamma$ with $v$ fresh;
    return ProveDS$^{\mathsf{k}}$($\mathcal{R}' \Rightarrow \Gamma'$);
end
if some $w \in \text{Lab}(\mathcal{R} \Rightarrow \Gamma)$ is not $\Diamond$-propagated then
    Pick a label $u \in \text{Lab}(\mathcal{R} \Rightarrow \Gamma)$ such that $w : \Diamond\phi \in \Gamma$ and $u : \phi \not\in \Gamma$, and let
      $\Gamma' := u : \phi, \Gamma$;
    return ProveDS$^{\mathsf{k}}$($\mathcal{R} \Rightarrow \Gamma'$);
end
if some $w \in \text{Lab}(\mathcal{R} \Rightarrow \Gamma)$ is not $\langle 0 \rangle$-propagated then
    Pick a label $u \in \text{Lab}(\mathcal{R} \Rightarrow \Gamma)$ such that $w : \langle 0 \rangle\phi \in \Gamma$, $w \sim_0^{\mathcal{R}} u$, $u : \phi \not\in \Gamma$, and
      let $\Gamma' := u : \phi, \Gamma$;
    return ProveDS$^{\mathsf{k}}$($\mathcal{R} \Rightarrow \Gamma'$);
end
if some $w \in \text{Lab}(\mathcal{R} \Rightarrow \Gamma)$ is not $\ominus_0$-propagated then
    Pick labels $u, v \in \text{Lab}(\mathcal{R} \Rightarrow \Gamma)$ such that $w : \ominus_0\phi \in \Gamma$, $I_{\otimes_0}u \in \mathcal{R}$, $u \sim_0^{\mathcal{R}} v$, and
      $v : \phi \not\in \Gamma$, and let $\Gamma' := v : \phi, \Gamma$;
    return ProveDS$^{\mathsf{k}}$($\mathcal{R} \Rightarrow \Gamma'$);
end
if $\mathcal{R} \Rightarrow \Gamma$ is not (D$_2$)-satisfied then
    Pick a label $w \in \text{Lab}(\mathcal{R} \Rightarrow \Gamma)$ such that $w : \ominus_0\phi \in \Gamma$, and for all
      $u \in \text{Lab}(\mathcal{R} \Rightarrow \Gamma)$, $I_{\otimes_0}u, u : \phi \not\in \mathcal{R}, \Gamma$, and let $\mathcal{R}' := I_{\otimes_0}v, \mathcal{R}$ and $\Gamma' := v : \phi, \Gamma$
      with $v$ fresh;
    return ProveDS$^{\mathsf{k}}$($\mathcal{R}' \Rightarrow \Gamma'$);
end
if $\mathcal{R} \Rightarrow \Gamma$ is not (S$_3$)-satisfied then
    Let $\mathcal{R}_{m,j}, \mathcal{R} := R_{[0]}w_m w_j$ (with $0 \leq m \leq k-1$ and $m+1 \leq j \leq k$) where $w_m$
      and $w_j$ are distinct roots of choice-trees in $\mathcal{R} \Rightarrow \Gamma$;
    if ProveDS$^{\mathsf{k}}$($\mathcal{R}_{m,j} \Rightarrow \Gamma$) = False for some $m$ and $j$ then
        return False;
    end
    else
        return True;
    end
end
```

on if Λ contains a labelled formula of the form $u : \ominus_0\psi$, or it does not, respectively. The DS_0^k-model M is defined as follows:

- $W := \mathrm{Lab}(\Lambda)$
- For all $u, v \in \mathrm{Lab}(\Lambda)$, $v \in R_{[0]}(u)$ iff $u \sim_0^{\mathcal{R}} v$
- $I_{\otimes_0} := \{v \mid \text{exists } u \in \mathrm{Lab}(\Lambda),\ I_{\otimes_0}u \in \mathcal{R} \text{ and } u \sim_0^{\mathcal{R}} v\}$
- $V(u) := \{p \mid u : \neg p \in \Gamma\}$, for all $u \in \mathrm{Lab}(\Lambda)$

For the DS_0^k-model M', we let $W' := W$, $R'_{[0]} := R_{[0]}$, $I'_{\otimes_0} = \{u \mid w \sim_0^{\mathcal{R}} u\}$, and $V' = V$, that is, the only difference is that $I_{\otimes_0}$ is set equal to the choice-cell of w, which must be non-empty (as it must contain w at the very least). The reason for this is that we need to ensure that $I_{\otimes_0} \neq \emptyset$, i.e. the model M' satisfies the condition $\mathbf{(D_2)}$ (and is a DS_0^k-model). We let $\mathbf{M} \in \{M, M'\}$, $\mathbf{W} \in \{W, W'\}$, $\mathbf{R}_{[0]} \in \{R_{[0]}, R'_{[0]}\}$, $\mathbf{I}_{\otimes_0} \in \{I_{\otimes_0}, I'_{\otimes_0}\}$, and $\mathbf{V} \in \{V, V'\}$ to prove our results uniformly, and only distinguish between M, M', and their components when needed.

To complete the proof of claim (ii), we prove two things: (ii.1) $\mathbf{M}$ is a DS_0^k-model with $\mathrm{Ag} = \{0\}$, and (ii.2) $\mathbf{M}, w \not\Vdash \phi$, i.e. $\mathbf{M}$ is a counter-model for ϕ. Showing these two cases proves that $\epsilon \Rightarrow w : \phi$ is not $\mathsf{DS}_0^k\mathsf{L}$-provable by soundness (Thm. 13).

(ii.1) We prove that $\mathbf{M}$ is a DS_0^k-model by showing that it satisfies conditions $\mathbf{(S_1)}$–$\mathbf{(S_3)}$ and $\mathbf{(D_1)}$–$\mathbf{(D_3)}$.

$\mathbf{(S_1)}$ Follows from the definition of $\mathbf{R}_{[0]}$ and the fact that $\sim_0$ is an equivalence relation (Lem. 31).

$\mathbf{(S_2)}$ The $\mathbf{(S_2)}$ condition is trivially satisfied in the single-agent setting.

$\mathbf{(S_3)}$ Since the labelled sequent Λ is stable, we know that it is $\mathbf{(S_3)}$-satisfied, meaning that $G(\Lambda)$ contains at most k many choice-trees. By Lem. 62, we know that $u \sim_0 v$ iff $v \in \mathrm{CT}(u)$. This fact, in conjunction with the definition of $\mathbf{R}_{[0]}$ above, and the fact that $\sim_0$ is an equivalence relation (Lem. 31), implies that each choice-tree corresponds to a choice-cell in $\mathbf{R}_{[0]}$. Hence, there will be at most k many choice cells.

$\mathbf{(D_1)}$ Follows from the definition of $\mathbf{I}_{\otimes_0}$ above.

$\mathbf{(D_2)}$ There are two possible cases depending on if $\mathbf{M} = M$ or $\mathbf{M} = M'$, i.e. depending of if Λ contains a labelled formula of the form $u : \ominus_0\psi$, or if it does not. For the first case, since Λ is $\mathbf{(D_2)}$-satisfied and contains a labelled formula of the form $u : \ominus_0\psi$, we know that there exists a label $u \in \mathrm{Lab}(\Lambda)$ such that $I_{\otimes_0}u \in \mathcal{R}$ by instructions 45–48. For the second case, the non-emptiness of $I'_{\otimes_0}$ follows by the definition of $I'_{\otimes_0}$. Hence, $I_{\otimes_0}$ and $I'_{\otimes_0}$ will be non-empty, that is, $\mathbf{I}_{\otimes_0} \neq \emptyset$.

$\mathbf{(D_3)}$ There are two possible cases depending on if $\mathbf{M} = M$ or $\mathbf{M} = M'$, i.e. depending of if Λ contains a labelled formula of the form $u : \ominus_0\psi$, or if it does not. The second case is trivial, as $I'_{\otimes_0}$ satisfies $\mathbf{(D_3)}$ by definition, so we focus on the first case. Suppose that $u \in I_{\otimes_0}$ and $v \in R_{[i]}(u)$. Then, there exists a label $z \in \mathrm{Lab}(\Lambda)$ such that $I_{\otimes_0}z$ and $z \sim_0^{\mathcal{R}} u$ holds by the definition of $I_{\otimes_0}$ above. By the definition of $R_{[0]}$ above, we also know that $u \sim_0^{\mathcal{R}} v$ holds, which implies that $z \sim_0^{\mathcal{R}} v$ by Lem. 31; therefore, $v \in I_{\otimes_0}$ by the definition of $I_{\otimes_0}$ above.

(ii.2) We prove that if $u : \psi \in \Gamma$, then $\mathbf{M}, u \not\Vdash \psi$ by induction on the complexity of ψ. The desired conclusion that $\mathbf{M}, w \not\Vdash \phi$ follows from this claim since $w : \phi \in \Gamma$.

Base case. Suppose that $u : \psi$ is of the form $u : p$ or $u : \neg p$. In the former case, since Λ is stable, we know that it is saturated, implying that $u : \neg p \notin \Gamma$ because $u : p \in \Gamma$. By the definition of $\mathbf{V}$ above, $p \notin \mathbf{V}(u)$, which lets us conclude that $\mathbf{M}, u \not\Vdash p$. In the latter case, by the definition of $\mathbf{V}$, we know that $p \in \mathbf{V}(u)$, entailing that $\mathbf{M}, u \not\Vdash \neg p$.

Inductive step. We consider the main connective of ψ below and show that in each case the desired result follows.

$\vee$. Assume that $\psi := \chi \vee \theta$ and $u : \chi \vee \theta \in \Gamma$. Then, since Λ is saturated, we know that $u : \chi, u : \theta \in \Gamma$, which implies that $\mathbf{M}, u \not\Vdash \chi$ and $\mathbf{M}, u \not\Vdash \theta$ by the inductive hypothesis. Hence, $\mathbf{M}, u \not\Vdash \chi \vee \theta$.

$\wedge$. Assume that $\psi := \chi \wedge \theta$ and $u : \chi \wedge \theta \in \Gamma$. Then, since Λ is saturated, we know that either $u : \chi \in \Gamma$ or $u : \theta \in \Gamma$, which implies that either $\mathbf{M}, u \not\Vdash \chi$ or $\mathbf{M}, u \not\Vdash \theta$ by the inductive hypothesis. Regardless of the case, $\mathbf{M}, u \not\Vdash \chi \wedge \theta$.

$\Diamond$. Assume that $\psi := \Diamond\chi$ and $u : \Diamond\chi \in \Gamma$. Then, since Λ is saturated, we know that Λ is $\Diamond$-propagated. Hence, $v : \chi \in \Gamma$ for all $v \in \mathrm{Lab}(\Lambda)$. By IH and the definition of $\mathbf{W}$ then, $\mathbf{M}, v \not\Vdash \chi$ for all $v \in \mathbf{W}$. Therefore, $\mathbf{M}, u \not\Vdash \Diamond\chi$.

$\langle 0 \rangle$. Assume that $\psi := \langle 0 \rangle\chi$ and $u : \langle 0 \rangle\chi \in \Gamma$. Then, since Λ is saturated, we know that Λ is $\langle 0 \rangle$-propagated. Hence, $v : \chi \in \Gamma$ for all $v \in \mathrm{Lab}(\Lambda)$ such that $u \sim_0^{\mathcal{R}} v$. By IH and the definition of $\mathbf{R}_{[0]}$ then, $\mathbf{M}, v \not\Vdash \chi$ for all $v \in \mathbf{R}_{[0]}(u)$. Therefore, $\mathbf{M}, u \not\Vdash \langle 0 \rangle\chi$.

$\ominus_0$. If Λ does not contain a labelled formula of the form $u : \ominus_0\chi$, then the case holds vacuously. Let us suppose then that Λ contains such a labelled formula, meaning that $\mathbf{M} = M$. Assume that $\psi := \ominus_0\chi$ and $u : \ominus_0\chi \in \Gamma$. Then, since Λ is saturated, we know that Λ is $\ominus_0$-propagated. Hence, $v : \chi \in \Gamma$ for all $v \in \mathrm{Lab}(\Lambda)$ such that for some $z \in \mathrm{Lab}(\Lambda)$ with $I_{\otimes_0}z \in \mathcal{R}$ and $z \sim_0^{\mathcal{R}} v$. By IH and the definition of $I_{\otimes_0}$ then, we know that $M, v \not\Vdash \chi$ for all $v \in I_{\otimes_0}$, implying that $M, u \not\Vdash \ominus_0\chi$.

$\Box$. Assume that $\psi := \Box\chi$ and $u : \Box\chi \in \Gamma$. Then, since Λ is saturated, we know that Λ is $\Box$-realized, meaning there exists a label $v \in \mathrm{Lab}(\Lambda)$ such that $v : \chi \in \Gamma$. By the definition of $\mathbf{W}$ we know that $v \in \mathbf{W}$ and by IH we have that $\mathbf{M}, v \not\Vdash \chi$. Therefore, $\mathbf{M}, u \not\Vdash \Box\chi$.

$[0]$. Assume that $\psi := [0]\chi$ and $u : [0]\chi \in \Gamma$. Then, since Λ is saturated, we know that Λ is $[0]$-realized, meaning there exists a label $v \in \mathrm{Lab}(\Lambda)$ such that $R_{[0]}uv$ and $v : \chi \in \Gamma$. By the definition of $\mathbf{R}_{[0]}$ we know that $v \in \mathbf{R}_{[0]}(u)$ and by IH we have that $\mathbf{M}, v \not\Vdash \chi$. Therefore, $\mathbf{M}, u \not\Vdash [0]\chi$.

$\otimes_0$. Assume that $\psi := \otimes_0\chi$ and $u : \otimes_0\chi \in \Gamma$. Then, since Λ is saturated, we know that Λ is $\otimes_0$-realized, meaning there exists a label $v \in \mathrm{Lab}(\Lambda)$ such that $I_{\otimes_0}v \in \mathcal{R}$ and $v : \chi \in \Gamma$. By the definition of $\mathbf{I}_{\otimes_0}$ we know that $v \in \mathbf{I}_{\otimes_0}$ and by IH we have that $\mathbf{M}, v \not\Vdash \chi$. Therefore, $\mathbf{M}, u \not\Vdash \otimes_0\chi$. □

Theorem 36 (Termination). *For each $\phi \in \mathcal{L}_{\mathsf{DS}_0^k}$, $\mathtt{ProveDS^k}(\epsilon \Rightarrow w : \phi)$ terminates.*

Proof. Consider a run of $\mathtt{ProveDS^k}(\epsilon \Rightarrow w : \phi)$. The only rules that add new vertices in the sequent graphs of labelled forest sequents generated throughout the course of the computation are instructions 21–24, 25–28, 29–32, 45–48, which correspond to bottom-up applications of the rules $(\Box)$, $([0])$, $(\otimes_0)$, and $(Pr^2_{\ominus_0})$, respectively. The number of bottom-up applications of $(\Box)$, $([0])$, $(\otimes_0)$, and $(Pr^2_{\ominus_0})$ rules is restricted by the number of $\Box$, $[0]$, $\otimes_0$, and $\ominus_0$ subformulae of ϕ, and hence can only be applied a finite number of times. Since the aforementioned instructions/rules are the only ones that introduce new vertices in the sequent graphs of labelled forest sequents, only a finite number of vertices (i.e. labels) can be introduced throughout the computation of $\mathtt{ProveDS^k}(\epsilon \Rightarrow w : \phi)$.

Instructions 7–10 and 11–20 correspond to bottom-up applications of the $(\vee_r)$ and $(\wedge_r)$ rules, and once such instructions are applied to a disjunctive or conjunctive subformula of ϕ, only formulae of a smaller complexity are introduced and the considered disjunctive or conjunctive subformula need not be considered again. Instructions 33–36, 37–40, 41–44 correspond to bottom-up applications of the $(\Diamond)$, $(\langle 0\rangle)$, and $(\ominus_0)$ rules, and ensure that for each $\Diamond\psi$, $\langle 0\rangle\psi$, and $\ominus_0\psi$ subformula of ϕ, $u : \psi$ (which is a formula of less complexity) is added only once with some label u. Since only a finite number of labels (i.e. vertices) can be introduced throughout the course of the computation (by what was said above), this implies that the $(\Diamond)$, $(\langle 0\rangle)$, and $(\ominus_0)$ rules can only be applied a finite number of times.

Last, instructions 49–57 (corresponding to a bottom-up application of (APC^k_0)) take a labelled forest sequent whose sequent graph has $m > k$ choice-trees as input and returns a new labelled forest sequent whose sequent graph contains $m - 1 \geq k$ choice-trees. Since only a finite number of labels (and relational atoms) can be introduced throughout the computation, only a finite number of choice-trees can be introduced throughout the computation; hence, a point will always be reached where the (APC^k_0) rule need not be applied. □

Corollary 7 (Decidability and FMP). *For each $k \in \mathbb{N}$, the logic DS^k_0 is decidable and has the finite model property.*

Proof. Follows from Thm. 35 and 36. □

Corollary 8. *Let $k \in \mathbb{N}$ and $n = 0$. If a labelled formula $w : \phi$ is provable in $\mathsf{DS}^k_0\mathsf{L}$, then it has a labelled forest derivation with the rooted property.*

Proof. Follows from Lem. 61, Thm. 35, Thm. 36, and that fact that in the sequent graph of a labelled forest sequent, none of the instructions of $\texttt{ProveDS}^{\mathsf{k}}$ ever add an edge that points to the root of a choice-tree, ensuring that roots are preserved throughout the course of the computation, and hence, the output derivation has the rooted property. □

6.2 Interpolation for Context-free Grammar Logics with Converse

The Lyndon interpolation property for a logic L states that if $\phi \rightarrow \psi \in \mathsf{L}$, then there exists a formula χ (called an *interpolant*) such that $\phi \rightarrow \chi, \chi \rightarrow \psi \in \mathsf{L}$ and χ is built from logical connectives and propositional variables common to both ϕ and ψ that have the same polarity. A weaker version of Lyndon interpolation—referred to as *Craig interpolation*—only requires an interpolant to be built using propositional variables common to both ϕ and ψ, thus dropping the added condition that the propositional variables used in χ share the same polarity in ϕ and ψ (therefore, Lyndon interpolation implies Craig interpolation). Interpolation has found a variety of applications in logic and computer science; e.g. interpolation is used in computer-aided verification to divide a problem involving an implication $\phi \rightarrow \psi$ into smaller problems involving $\phi \rightarrow \chi$ and $\chi \rightarrow \psi$ [McM18], to establish Beth definability [KO10], and is used in knowledge representation to conceal or forget superfluous or private information in ontology querying [LW11]. Hence, interpolation is of practical consequence.

Both semantic or syntactic arguments may be and have been used to establish that a logic L possesses the Lyndon or Craig interpolation property. As one might expect, the semantic approach requires a semantics for the logic L, whereas the syntactic approach requires a proof calculus for the logic. The latter, syntactic approach—initiated by Maehara [Mae60], who utilized Gentzen-style sequent systems—constructs an interpolant χ for an implication $\phi \rightarrow \psi \in \mathsf{L}$ and has the advantage that the method transforms a proof of $\phi \rightarrow \psi$ into proofs of $\phi \rightarrow \chi$ and $\chi \rightarrow \psi$ witnessing that χ is an interpolant. To provide intuition regarding Maehara's interpolation methodology, which motivates the more general methodology we will employ for grammar logics, we explain the method below in the context of classical propositional logic. Afterward, we describe how Maehara's method for Gentzen-style sequent systems can be generalized and extended to nested/refined labelled systems.

As explained in Appendix B (p. 208), the rules (id), $(\vee_r)$, and $(\wedge_r)$ (from $\mathsf{G3Km}(S)$) serve as a sound and complete calculus for classical propositional logic. Consequently, we use these three rules as our calculus to demonstrate Maehara's method, and note that by the work in the Appendix B (p. 208), our calculus only needs sequents of the form $\epsilon \Rightarrow \Gamma$ where every labelled formula in Γ has the same label w. Since every labelled formula has the same label w, it is permissible to omit labels; hence, we omit the use of labels in the sequents of our example, meaning that we will utilize (one-sided) Gentzen-style sequents while demonstrating Maehara's method.

If showing Lyndon interpolation, then Maehara's method is a means of transforming a proof of a sequent $\epsilon \Rightarrow \Gamma_1, \Gamma_2$ into proofs of $\epsilon \Rightarrow \Gamma_1, \chi$ and $\epsilon \Rightarrow \neg\chi, \Gamma_2$, where χ only contains propositional variables common to both $\neg\Gamma_1$ and Γ_2 that have the same polarity.[2] If such a transformation can be performed for each derivable sequent of the form $\epsilon \Rightarrow \Gamma_1, \Gamma_2$, then it can be performed for any derivable sequent of the form $\epsilon \Rightarrow \neg\phi, \psi$, which is equivalent to $\epsilon \Rightarrow \phi \rightarrow \psi$. It follows that the proof of $\epsilon \Rightarrow \neg\phi, \psi$ can be transformed into proofs of $\epsilon \Rightarrow \neg\phi, \chi$ and $\epsilon \Rightarrow \neg\chi, \psi$, which are equivalent to $\epsilon \Rightarrow \phi \rightarrow \chi$ and $\epsilon \Rightarrow \chi \rightarrow \psi$, respectively—yielding an interpolant χ for the implication $\phi \rightarrow \psi$.

The method works by constructing an interpolant χ by induction on the proof of the derivable sequent $\epsilon \Rightarrow \Gamma_1, \Gamma_2$ we wish to interpolate. This inductive construction is encoded within inference rules that employ more expressive sequents, specifying how the context of the sequent is partitioned (with the left component of the partition representing the antecedent that implies the interpolant, and the right component of the partition representing the consequent that is implied by the interpolant) and the interpolant constructed thus far. The more expressive sequents we employ are of the form $\epsilon \Rightarrow \Gamma_1 \mid \Gamma_2 \parallel \chi$ denoting that the sequent $\epsilon \Rightarrow \Gamma_1, \Gamma_2$ can be partitioned into $\epsilon \Rightarrow \Gamma_1, \chi$ and $\epsilon \Rightarrow \neg\chi, \Gamma_2$ with interpolant χ. The inference rules we make use of are variants of the (id), $(\vee_r)$, and $(\wedge_r)$ rules where we consider all possible 'partitions' of the context and assign interpolants based on how the context is split. We first introduce the initial rules, and then introduce the variants of $(\vee_r)$ and $(\wedge_r)$.

$$\frac{}{\epsilon \Rightarrow \Gamma_1, p, \neg p \mid \Gamma_2 \parallel \bot}\,(id_1) \qquad \frac{}{\epsilon \Rightarrow \Gamma_1, p \mid \neg p, \Gamma_2 \parallel \neg p}\,(id_2)$$

$$\frac{}{\epsilon \Rightarrow \Gamma_1, \neg p \mid p, \Gamma_2 \parallel p}\,(id_3) \qquad \frac{}{\epsilon \Rightarrow \Gamma_1 \mid p, \neg p, \Gamma_2 \parallel \top}\,(id_4)$$

As shown above, there are four possible interpolants, depending on how the context of an initial sequent is partitioned. The (id_1) rule tells us that the derivable sequent $\epsilon \Rightarrow \Gamma_1, p, \neg p, \Gamma_2$ can be split into the derivable sequents $\epsilon \Rightarrow \Gamma_1, p, \neg p, \bot$ and $\epsilon \Rightarrow \neg\bot, \Gamma_2$ with interpolant $\bot$. The other three initial rules are read in a similar fashion. Our disjunction and conjunction rules are as follows:

$$\frac{\epsilon \Rightarrow \Gamma_1, \phi, \psi \mid \Gamma_2 \parallel \chi}{\epsilon \Rightarrow \Gamma_1, \phi \vee \psi \mid \Gamma_2 \parallel \chi}\,(\vee_{r1}) \qquad \frac{\epsilon \Rightarrow \Gamma_1, \phi \mid \Gamma_2 \parallel \chi \qquad \epsilon \Rightarrow \Gamma_1, \psi \mid \Gamma_2 \parallel \theta}{\epsilon \Rightarrow \Gamma_1, \phi \wedge \psi \mid \Gamma_2 \parallel \chi \vee \theta}\,(\wedge_{r1})$$

$$\frac{\epsilon \Rightarrow \Gamma_1 \mid \phi, \psi, \Gamma_2 \parallel \chi}{\epsilon \Rightarrow \Gamma_1 \mid \phi \vee \psi, \Gamma_2 \parallel \chi}\,(\vee_{r2}) \qquad \frac{\epsilon \Rightarrow \Gamma_1 \mid \phi, \Gamma_2 \parallel \chi \qquad \epsilon \Rightarrow \Gamma_1 \mid \psi, \Gamma_2 \parallel \theta}{\epsilon \Rightarrow \Gamma_1 \mid \phi \wedge \psi, \Gamma_2 \parallel \chi \wedge \theta}\,(\wedge_{r2})$$

Notice how each of the rules above constructs an interpolant of the conclusion by making use of the interpolant(s) of the premise(s). Maehara's method therefore works by

[2] If $\Gamma_1 := \phi_1, \ldots, \phi_n$, we let $\neg\Gamma_1 := \neg\phi_1, \ldots, \neg\phi_n$.

assigning interpolants to all initial sequents in the given derivation, and then inductively constructing the interpolant of the end sequent via the above rules. We demonstrate this process with an example, and consider a derivation of an implication $p \wedge q \rightarrow p \wedge q$, which is represented as $(\neg p \vee \neg q) \vee (p \wedge q)$ since we are working with formulae negation normal form.

$$\begin{prooftree}
\AxiomC{}
\RightLabel{(id)}
\UnaryInfC{$\epsilon \Rightarrow \neg p, \neg q, p$}
\AxiomC{}
\RightLabel{(id)}
\UnaryInfC{$\epsilon \Rightarrow \neg p, \neg q, q$}
\RightLabel{$(\wedge_r)$}
\BinaryInfC{$\epsilon \Rightarrow \neg p, \neg q, p \wedge q$}
\RightLabel{$(\vee_r)$}
\UnaryInfC{$\epsilon \Rightarrow \neg p \vee \neg q, p \wedge q$}
\RightLabel{$(\vee_r)$}
\UnaryInfC{$\epsilon \Rightarrow (\neg p \vee \neg q) \vee (p \wedge q)$}
\dottedLine
\RightLabel{$=$}
\UnaryInfC{$\epsilon \Rightarrow p \wedge q \rightarrow p \wedge q$}
\end{prooftree}$$

We transform the above derivation into a new derivation that makes use of the (relevant) inference rules introduced previously. In doing so, we only make use of the top portion of the above derivation down to the sequent $\epsilon \Rightarrow \neg p \vee \neg q, p \wedge q$ since Maehara's method demands the antecedent formula $\neg p \vee \neg q$ and consequent formula $p \wedge q$ be split from one another:

$$\begin{prooftree}
\AxiomC{}
\RightLabel{(id_3)}
\UnaryInfC{$\epsilon \Rightarrow \neg p, \neg q \mid p \parallel p$}
\AxiomC{}
\RightLabel{(id_3)}
\UnaryInfC{$\epsilon \Rightarrow \neg p, \neg q \mid q \parallel q$}
\RightLabel{$(\wedge_{r2})$}
\BinaryInfC{$\epsilon \Rightarrow \neg p, \neg q \mid p \wedge q \parallel p \wedge q$}
\RightLabel{$(\vee_{r1})$}
\UnaryInfC{$\epsilon \Rightarrow \neg p \vee \neg q \mid p \wedge q \parallel p \wedge q$}
\end{prooftree}$$

The derivation above may now be split into two derivations in the original calculus witnessing that $p \wedge q$ is the interpolant of $p \wedge q \rightarrow p \wedge q := (\neg p \vee \neg q) \vee (p \wedge q)$. The derivation below left is obtained by taking the data from the left components of the partitioned sequents together with the initial interpolants, and the derivation below right is obtained by taking the right components together with the negation of the initial interpolants. Observe that the use of $(\wedge_r)$ in the derivation below left builds the interpolant $p \wedge q$, and the $(\vee_r)$ rule in the derivation below right builds the interpolant $\neg p \vee \neg q$. Also, note that it may be necessary to invoke the (hp-)admissibility of structural rules such as (wk) (as shown in the derivation below right) to obtain the desired result.

$$\begin{prooftree}
\AxiomC{}
\RightLabel{(id)}
\UnaryInfC{$\epsilon \Rightarrow \neg p, \neg q, p$}
\AxiomC{}
\RightLabel{(id)}
\UnaryInfC{$\epsilon \Rightarrow \neg p, \neg q, q$}
\RightLabel{$(\wedge_r)$}
\BinaryInfC{$\epsilon \Rightarrow \neg p, \neg q, p \wedge q$}
\RightLabel{$(\vee_r)$}
\UnaryInfC{$\epsilon \Rightarrow \neg p \vee \neg q, p \wedge q$}
\end{prooftree}
\qquad
\begin{prooftree}
\AxiomC{}
\RightLabel{(id)}
\UnaryInfC{$\epsilon \Rightarrow \neg p, p$}
\dashedLine
\RightLabel{(wk)}
\UnaryInfC{$\epsilon \Rightarrow \neg q, \neg p, p$}
\AxiomC{}
\RightLabel{(id)}
\UnaryInfC{$\epsilon \Rightarrow \neg q, q$}
\dashedLine
\RightLabel{(wk)}
\UnaryInfC{$\epsilon \Rightarrow \neg p, \neg q, q$}
\RightLabel{$(\wedge_r)$}
\BinaryInfC{$\epsilon \Rightarrow \neg p, \neg q, p \wedge q$}
\RightLabel{$(\vee_r)$}
\UnaryInfC{$\epsilon \Rightarrow \neg p \vee \neg q, p \wedge q$}
\end{prooftree}$$

By applying $(\vee_r)$ to both end sequents, we obtain two derivations of the valid formula $(\neg p \vee \neg q) \vee (p \wedge q) := p \wedge q \rightarrow p \wedge q$, which witness that $p \wedge q$ is the interpolant $p \wedge q \rightarrow p \wedge q$. Note that although the implication $p \wedge q \rightarrow p \wedge q$ is rather simple to interpolate, the above example was chosen for its simplicity and because of the fact that it still gives a concrete demonstration of Maehara's method.

In Maehara's seminal paper [Mae60], the above method was provided for Gentzen-style sequent calculi. However, if our aim is to proof-theoretically (and syntactically) demonstrate that our class of grammar logics possesses the Lyndon interpolation property, then we must figure out how to generalize Maehara's method to nested (or, refined labelled) sequent calculi, since we have such calculi at our disposal for grammar logics.[3] Such a generalization was presented in [LTGC20], where Craig interpolation was established for tense logics and bi-intuitionistic logic.[4] Also, it should be noted that although the method of [LTGC20] was introduced for nested systems, refined labelled systems were leveraged in the methodology; as explained there, the labelled *notation* tends to be easier to work with and allows one to avoid the introduction of certain auxiliary concepts. We therefore follow the same strategy, and utilize our refined labelled calculi $\mathsf{Km}(S)\mathsf{L}$ as opposed to the nested calculi $\mathsf{DKm}(S)$ while proving Lyndon interpolation for grammar logics.

Extending Maehara's method to the framework of refined labelled (i.e. nested) sequents calls for generalizing two of the notions inherent in Maehara's method: (i) the notion of an interpolant, and (ii) the negation of an interpolant. We first motivate how interpolants ought to be extended in the refined labeled (i.e. nested) setting by considering a simple example, and then discuss (ii) afterward.

For our simple example, let us consider the following instance of the (id) rule (recall that we are now working in the calculus $\mathsf{Km}(S)\mathsf{L}$):

$$\frac{}{R_a wu \Rightarrow u : p, u : \neg p}\ (id)$$

As we saw in the example of Maehara's method above, each rule of the given calculus is transformed into a set of rules employing more expressive sequents that partition the contexts and assign interpolants accordingly—*where interpolants are proper logical formulae*. For the sake of the example, let us suppose that we partition the above initial sequent as follows: $R_a wu \Rightarrow u : p \mid u : \neg p \parallel \mathcal{I}$, with $\mathcal{I}$ denoting the interpolant we aim to discover. In the current context, it is nonsensical to make use of proper logical formulae as interpolants because such formulae omit labels, which encode essential information in the refined labelled setting. At the very least then, our interpolant ought to make use of labels. If we pick the interpolant $u : \neg p$ as our interpolant $\mathcal{I}$, then the aforementioned split sequent becomes $R_a wu \Rightarrow u : p \mid u : \neg p \parallel u : \neg p$. If we read this sequent in a manner analogous to how we read partitioned sequents in the context of Maehara's method, then the partitioned sequent states that $R_a wu \Rightarrow u : p, u : \neg p$ and $R_a wu \Rightarrow u : \neg\neg p, u : \neg p$ are both derivable in $\mathsf{Km}(S)\mathsf{L}$ with interpolant $u : \neg p$. Since $\neg\neg p := p$, this happens to

[3] Recall that in the context of grammar logics, nested sequent and refined labelled sequent calculi are notational variants of one another (see Sect. 4.1). Therefore, in the current context, we may use the two terms interchangeably.

[4] Although the method of [LTGC20] establishes Craig interpolation, strengthening the method to yield Lyndon interpolation is almost trivial, and follows from also considering the polarities of propositional variables.

be true. We may conclude that the generalized interpolants used in the refined labelled setting must make use of labelled formulae at the very least.

Drawing an analogy with Maehara's method, we must introduce inference rules to calculate interpolants for each inference rule of $\mathsf{Km}(S)\mathsf{L}$, which includes the two premise rule $(\wedge_r)$. In the context of classical propositional logic, we saw that the $(\wedge_{r1})$ and $(\wedge_{r2})$ rules constructed the interpolants $\chi \vee \theta$ and $\chi \wedge \theta$, respectively, from the interpolants of the premises. In the refined labelled setting with $\mathsf{Km}(S)\mathsf{L}$, where we have already agreed that interpolants ought to include labels, what would happen if we had a conjunction inference where the interpolant of one premise was of the form $w : \chi$ and the interpolant in the second premise was of the form $u : \theta$? To resolve this issue, it happens to be sufficient to define interpolants to be sets of labelled sequents, and when a rule such as $(\wedge_r)$ is applied, we union to the two sets to construct the new interpolant.

The discussion thus far has led us to confirm that interpolants ought to be sets of labelled sequents when generalizing Maehara's method to the refined labelled setting. Nevertheless, recall that Maehara's method takes a derivable sequent $\epsilon \Rightarrow \Gamma_1, \Gamma_2$ and returns a proof of $\epsilon \Rightarrow \Gamma_1, \chi$ and $\epsilon \Rightarrow \neg\chi, \Gamma_2$, where χ is the interpolant. By analogy, to show our grammar logics interpolate via $\mathsf{Km}(S)\mathsf{L}$, we aim to show that given a derivable labelled sequent $\mathcal{R} \Rightarrow \Gamma_1, \Gamma_2$, we can transform its proof into proofs of $\mathcal{R} \Rightarrow \Gamma_1, \mathcal{I}$ and $\mathcal{R} \Rightarrow \neg\mathcal{I}, \Gamma_2$, where $\mathcal{I}$ is the interpolant. Yet, if $\mathcal{I}$ is a set of labelled sequents, what should the negation $\neg\mathcal{I}$ represent?

At this point, it behooves us to recognize a general pattern underlying the relationship between an interpolant and its negation in Maehara's method, namely, the empty sequent can be derived from an interpolant and its negation by making use of (cut) (we will also allow for structural rules such as (wk) and (ctr_r) to be used in our more general setting); we say that an interpolant and its negation are *orthogonal* if such a relationship holds. In the context of classical propositional logic and Gentzen-style sequent calculi, this relationship is easy to verify; if χ is an interpolant, then a single application of (cut) between $\epsilon \Rightarrow \chi$ and $\epsilon \Rightarrow \neg\chi$ yields the empty sequent $\epsilon \Rightarrow \epsilon$. Nonetheless, in the refined labelled setting, phrasing the relationship between an interpolant and its negation in such a general manner permits us to define the negation $\neg\mathcal{I}$ of an interpolant $\mathcal{I}$ as a set of labelled sequents orthogonal to $\mathcal{I}$. By this definition, the negation of an interpolant is not necessarily unique, though we will strengthen the definition in our interpolation work below to ensure a unique negation, since it simplifies matters. To demonstrate the concept of orthogonality, observe that the interpolant $\mathcal{I}$ given below is orthogonal to the interpolant $\mathcal{I}'$ as the empty sequent $\epsilon \Rightarrow \epsilon$ can be derived with applications of (cut) and (wk) between the labelled sequents (which have been placed in between parentheses to improve readability) of the interpolants:

$$\mathcal{I} := \{(\epsilon \Rightarrow w : p), (\epsilon \Rightarrow u : \neg q)\} \qquad \mathcal{I}' := \{(\epsilon \Rightarrow w : \neg p, u : q)\}$$

Before moving on to prove the main results of the section, we remark on the relationship between the syntactic method of interpolation from [LTGC20] presented here and the closely related semantic (yet, still proof-theoretic) method of interpolation by Kuznets *et al.* [FK15, Kuz16b, Kuz16a, Kuz18, KL18] applied to modal and intermediate logics equipped with a Kripke semantics.[5] One notable difference is that the semantic method introduces meta-connectives $\owedge$ and $\ovee$ expressing meta-level conjunction and disjunction, respectively, and which are used to construct interpolants. This contrasts with the syntactic approach that uses sets of labelled (or, nested) sequents as explained above. Moreover, in the semantic method, the meta-connectives $\owedge$ and $\ovee$ are interpreted in a semantics extending the semantics of the logic under consideration. In this regard, the syntactic method can be seen as possessing a higher degree of parsimony, as the semantics of the considered logic need not be generalized to a meta-language. Additionally, the interpolants of the syntactic method are justified on purely syntactic and proof-theoretic grounds, whereas the interpolants of the semantic method are justified via semantic arguments. As pointed out in [LTGC20], defining and justifying interpolants and their negations in a purely syntactic fashion independent of the underlying logic's semantics, appears to have the advantageous consequence that the method is applicable to logics for which no Kripke semantics is known (e.g. bi-intuitionistic linear logic [CDGT13]), something which the semantic method does not appear immediately capable.

6.2.1 Lyndon Interpolation via Proof-Theoretic Methods

We leverage the syntactic method of interpolation from [LTGC20] to prove (the new result) that all context-free grammar logics with converse possess the effective Lyndon interpolation property, meaning that our proof not only shows the existence of a Lyndon interpolant for each valid implication, but also shows how to construct it (cf. [GN05]). We first define a set of concepts necessary to apply the syntactic method, and then prove the main results.

As mentioned previously, in our more general setting, an interpolant is a set of labelled sequents. In defining our interpolants however, the multiset of relational atoms $\mathcal{R}$ is left empty since $\mathcal{R}$ can be recovered from the sequents with which the interpolants are used. This gives rise to the following definitions:

Definition 89 (Flat Labelled Sequent, Interpolant)**.** *A* flat labelled sequent *is a labelled sequent of the form $\epsilon \Rightarrow \Gamma$. An* interpolant *$\mathcal{I}$ is defined to be a set of flat sequents, i.e.*

$$\mathcal{I} := \{(\epsilon \Rightarrow \Gamma_1), \ldots, (\epsilon \Rightarrow \Gamma_n)\}$$

We use $\mathcal{I}$, $\mathcal{I}'$, ... (occasionally annotated) to denote interpolants.

The relation of being *orthogonal* was informally defined above, and intuitively holds between interpolants $\mathcal{I}_1$ and $\mathcal{I}_2$ iff the empty sequent is derivable from the labelled sequents in $\mathcal{I}_1$ and $\mathcal{I}_2$ using (cut) and possibly (wk) and/or (ctr_r). This relationship is

[5] This method may be qualified as both proof-theoretic and semantic since the method inductively constructs interpolants of formulae based on the structure of a given derivation, but proves the correctness of the inductive construction via *semantic arguments*.

used to establish a generalized notion of negation, that is to say, if $\mathcal{I}_1$ is orthogonal to $\mathcal{I}_2$, then the two interpolants are negations of one another. For example, suppose that we are given the interpolant $\mathcal{I}_1 := \{(\epsilon \Rightarrow w : \phi, u : \psi), (\epsilon \Rightarrow v : \chi)\}$. Then, the following interpolants are orthogonal to $\mathcal{I}_1$ and can be considered negations:

$$\begin{array}{rcl} \mathcal{I}_2 & := & \{(\epsilon \Rightarrow w : \neg\phi), (\epsilon \Rightarrow u : \neg\psi, v : \neg\chi)\} \\ \mathcal{I}_3 & := & \{(\epsilon \Rightarrow w : \neg\phi, v : \neg\chi), (\epsilon \Rightarrow u : \neg\psi, u : \neg\psi)\} \\ \mathcal{I}_4 & := & \{(\epsilon \Rightarrow w : \neg\phi, v : \neg\chi), (\epsilon \Rightarrow u : \neg\psi, v : \neg\chi)\} \end{array}$$

In each case, one can derive the empty sequent from $\mathcal{I}_1 \cup \mathcal{I}_i$ using (cut), (wk), and (ctr_r) for each $i \in \{2, 3, 4\}$. Despite this fact, to make the negation of an interpolant deterministic, our definition below will choose $\mathcal{I}_4$ to be the negation, as it straightforward to calculate such an interpolant as a function of $\mathcal{I}_1$. Instead of using the general notion of negation presented above, we opt for a more specific notion, which we refer to as the *orthogonal*:

Definition 90 (Orthogonal). *For an interpolant* $\mathcal{I} := \{(\epsilon \Rightarrow \Gamma_1), \ldots, (\epsilon \Rightarrow \Gamma_n)\}$, *its* orthogonal $(\mathcal{I})^\perp$ *is defined as:*

$$(\mathcal{I})^\perp ::= \{(\epsilon \Rightarrow w_1 : \neg\phi_1, \ldots, w_n : \neg\phi_1) \mid \textit{for all } i \in \{1, \ldots, n\}, w_i : \phi_i \in \Gamma_i\}$$

When constructing an interpolant by induction on the height of the input derivation, the processing of $([x])$ rules requires a special interpolant to be produced. The necessity of such interpolants will become apparent in the proof of Lem. 66 below. Such interpolants are defined as follows:

Definition 91. *Suppose we have an interpolant of the following form:*

$$\mathcal{I} := \{(\epsilon \Rightarrow \Gamma_1, u : \phi_{1,1}, \ldots, u : \phi_{1,k_1}), \ldots, (\epsilon \Rightarrow \Gamma_n, u : \phi_{n,1}, \ldots, u : \phi_{n,k_n})\}$$

where u *does not occur in* $\Gamma_1, \ldots, \Gamma_n$. *Let* $x \in \sum$, *that is, let* x *be a character in our alphabet* $\sum$, *and define*

$$[x]\mathcal{I}_u^w := \{(\epsilon \Rightarrow \Gamma_1, w : [x] \bigvee_{i=1}^{k_1} \phi_{1,i}), \ldots, (\epsilon \Rightarrow \Gamma_n, w : [x] \bigvee_{i=1}^{k_n} \phi_{n,i})\}$$

Let us now define the types of sequents that will be used in confirming the effective Lyndon interpolation property:

Definition 92 (Interpolation Sequent). *An* interpolation sequent *is defined to be a syntactic object of the form* $\mathcal{R} \Rightarrow \Gamma \mid \Delta \parallel \mathcal{I}$, *where* $\mathcal{R}$ *is a set of relational atoms,* Γ *and* Δ *are labelled formulae, and* $\mathcal{I}$ *is an interpolant.*

The vertical bar ‘ | ’ occurring in an interpolation sequent partitions the sequent into a left part serving as the antecedent in the interpolation statement, and a right part serving as the consequent in the interpolation statement. To illustrate this point, the interpolation sequent shown below top partitions into the two labelled sequents shown below bottom:

$$\mathcal{R} \Rightarrow \Gamma_1 \mid w : p, w : \neg p, \Gamma_2 \parallel \{(\epsilon \Rightarrow w : \top)\}$$

$$\Lambda_1 := (\mathcal{R} \Rightarrow \Gamma_1, w : \top) \quad \Lambda_2 := (\mathcal{R} \Rightarrow w : \neg\top, w : p, w : \neg p, \Gamma_2)$$

In the above example, the left component Γ_1 of the partition is placed in Λ_1, and the right component $w : p, w : \neg p, \Gamma_2$ is placed in Λ_2. We think of the interpolant $w : \top$ as being *implied by* the left component of the partition, and so, we place it in Λ_1, and we think of the interpolant as *implying* the right component of the partition, so we place the negation of the interpolant in Λ_2. Observe that we can derive the labelled sequent $\mathcal{R} \Rightarrow \Gamma_1, w : p, w : \neg p, \Gamma_2$ by applying (cut) (possibly with (wk)) to Λ_1 and Λ_2, which *syntactically establishes* that the interpolant is indeed an interpolant (given that the interpolant satisfies certain other properties; see Lem. 65 below).

Each interpolation calculus $\mathsf{Km}(S)\mathsf{LI}$, for each grammar logic $\mathsf{Km}(S)$, is given in Fig. 6.1. The derivability relation is defined as follows:

Definition 93. *We write* $\vdash_{\mathsf{Km}(S)\mathsf{LI}} \mathcal{R} \Rightarrow \Gamma \mid \Delta \parallel \mathcal{I}$ *to indicate that an interpolation sequent* $\mathcal{R} \Rightarrow \Gamma \mid \Delta \parallel \mathcal{I}$ *is derivable in the interpolation calculus* $\mathsf{Km}(S)\mathsf{LI}$*.*

Since each interpolation calculus is obtained from a corresponding refined labelled calculus $\mathsf{Km}(S)\mathsf{L}$, each propagation rule $(Pr_{\langle x \rangle})$ relies on the notion of a propagation graph, which we define for interpolation sequents below. The notion of a propagation path, the string of a propagation path, and their converses are defined as in Def. 59, so we do not repeat them here.

Definition 94 (Propagation Graph of Interpolation Sequent)**.** *Let* $\Lambda := \mathcal{R} \Rightarrow \Gamma \mid \Delta \parallel \mathcal{I}$ *and* $\Lambda' := \mathcal{R} \Rightarrow \Gamma, \Delta$*. We define the* propagation graph *of an interpolation sequent* Λ *as follows:* $PG(\Lambda) := PG(\Lambda')$*.*

Perhaps the most unique feature of each interpolation calculus $\mathsf{Km}(S)\mathsf{LI}$ is the $(orth)$ rule. The rule lets us cut the number of rules needed in our calculus in half by permitting the components of the partition in an interpolation sequent to be 'flipped'. The key to the correctness of the rule, is given in the lemma below, which shows that applying the orthogonal operation twice to an interpolant always retains the sequents of the original interpolant.

Lemma 63. *If* $(\epsilon \Rightarrow \Gamma) \in ((\mathcal{I})^{\perp})^{\perp}$*, then there exists a* $(\epsilon \Rightarrow \Delta) \in \mathcal{I}$ *such that* $\Delta \subseteq \Gamma$*.*

Proof. We prove the lemma by contradiction, and assume that there exists a $(\epsilon \Rightarrow \Gamma) \in ((\mathcal{I})^{\perp})^{\perp}$ such that for all $(\epsilon \Rightarrow \Delta) \in \mathcal{I}$, $\Delta \not\subseteq \Gamma$. Let $\mathcal{I} := \{(\epsilon \Rightarrow \Gamma_1), \ldots, (\epsilon \Rightarrow \Gamma_n)\}$. By our assumption, we know that for each $i \in \{1, \ldots, n\}$, there exists a formula $w_i : \phi_i \in \Gamma_i$ such that $w_i : \phi_i \notin \Gamma$. Let $\Gamma' := w_1 : \phi_1, \ldots, w_n : \phi_n$ and observe that by construction $\Gamma \cap \Gamma' = \emptyset$. However, by Def. 90, we know that $(\epsilon \Rightarrow \Gamma') \in (\mathcal{I})^{\perp}$, and since $(\epsilon \Rightarrow \Gamma) \in ((\mathcal{I})^{\perp})^{\perp}$, it follows that $\Gamma \cap \Gamma' \neq \emptyset$, giving a contradiction. □

$$\frac{}{\mathcal{R} \Rightarrow \Gamma, w : \neg p \mid w : p, \Delta \parallel \{(\Rightarrow w : p)\}}\ (id_1) \qquad \frac{\mathcal{R} \Rightarrow \Gamma \mid \Delta \parallel \mathcal{I}}{\mathcal{R} \Rightarrow \Delta \mid \Gamma \parallel (\mathcal{I})^{\perp}}\ (orth)$$

$$\frac{}{\mathcal{R} \Rightarrow \Gamma \mid w : \neg p, w : p, \Delta \parallel \{(\Rightarrow w : \top)\}}\ (id_2) \qquad \frac{\mathcal{R} \Rightarrow \Gamma \mid w : \phi, w : \psi, \Delta \parallel \mathcal{I}}{\mathcal{R} \Rightarrow \Gamma \mid w : \phi \vee \psi, \Delta \parallel \mathcal{I}}\ (\vee_r)$$

$$\frac{\mathcal{R} \Rightarrow \Gamma \mid w : \phi, \Delta \parallel \mathcal{I}_1 \qquad \mathcal{R} \Rightarrow \Gamma \mid w : \psi, \Delta \parallel \mathcal{I}_2}{\mathcal{R} \Rightarrow \Gamma \mid w : \phi \wedge \psi, \Delta \parallel \mathcal{I}_1 \cup \mathcal{I}_2}\ (\wedge_r)$$

$$\frac{\mathcal{R}, R_x wu \Rightarrow \Gamma \mid u : \phi, \Delta \parallel \mathcal{I}}{\mathcal{R} \Rightarrow \Gamma \mid w : [x]\phi, \Delta \parallel [x]\mathcal{I}^{w}_{u}}\ ([x])^{\dagger_1} \qquad \frac{\mathcal{R} \Rightarrow \Gamma \mid w : \langle x \rangle \phi, u : \phi, \Delta \parallel \mathcal{I}}{\mathcal{R} \Rightarrow \Gamma \mid w : \langle x \rangle \phi, \Delta \parallel \mathcal{I}}\ (Pr_{\langle x \rangle})^{\dagger_2}$$

Figure 6.1: The calculus $\mathsf{Km}(S)\mathsf{LI}$ for constructing interpolants for the grammar logic $\mathsf{Km}(S)$, for a CFCST system S. The calculus contains a $([x])$ and $(Pr_{\langle x \rangle})$ rule for each $x \in \sum$, i.e. for each character in our alphabet. The side condition $\dagger_1$ states that the rule is applicable only if u is an eigenvariable, and $\dagger_2$ states that the rule is applicable only if $s_{\pi}(w, u) \in L_S(x)$.

Enough groundwork has been laid for us to prove our main results, though before we do so, we formally define the literal function $\mathrm{Lit}(\cdot)$ and the Lyndon interpolation property:

Definition 95 (Literal Function). *For a formula $\phi \in \mathcal{L}_{\mathsf{Km}}(S)$, we define $\mathrm{Lit}(\phi)$ to be the set of all literals occurring in ϕ. If the multiset $\Gamma := w_1 : \phi_1, \ldots, w_n : \phi_n$ and the interpolant $\mathcal{I} := \{(\epsilon \Rightarrow \Gamma_1), \ldots, (\epsilon \Rightarrow \Gamma_n)\}$, then we respectively let*

$$\mathrm{Lit}(\Gamma) := \bigcup_{1 \leq i \leq n} \mathrm{Lit}(\phi_i) \qquad \mathrm{Lit}(\mathcal{I}) := \bigcup_{1 \leq i \leq n} \mathrm{Lit}(\Gamma_i)$$

Definition 96 (Lyndon Interpolation Property). *A grammar logic $\mathsf{Km}(S)$ has the* Lyndon interpolation property *iff for every implication $\phi \to \psi \in \mathcal{L}_{\mathsf{Km}}(S)$ such that $\vdash_{\mathsf{Km}(S)} \phi \to \psi$, there exists a formula χ such that (i) $\mathrm{Lit}(\chi) \subseteq \mathrm{Lit}(\phi) \cap \mathrm{Lit}(\psi)$ and (ii) $\vdash_{\mathsf{Km}(S)} \phi \to \chi$ and $\vdash_{\mathsf{Km}(S)} \chi \to \psi$.*

Lemma 64. *If $\vdash_{\mathsf{Km}(S)\mathsf{L}} \mathcal{R} \Rightarrow \Gamma, \Delta$, then $\vdash_{\mathsf{Km}(S)\mathsf{LI}} \mathcal{R} \Rightarrow \Gamma \mid \Delta \parallel \mathcal{I}$, for some interpolant $\mathcal{I}$.*

Proof. Straightforward, by induction on the height of the proof of $\mathcal{R} \Rightarrow \Gamma, \Delta$. □

Lemma 65. *If $\vdash_{\mathsf{Km}(S)\mathsf{LI}} \mathcal{R} \Rightarrow \Gamma \mid \Delta \parallel \mathcal{I}$, then $\mathrm{Lit}(\mathcal{I}) \subseteq \mathrm{Lit}(\neg\Gamma) \cap \mathrm{Lit}(\Delta)$, and all labels occurring in $\mathcal{I}$ also occur in $\mathcal{R}, \Gamma$ or Δ.*

Proof. By induction on the height of the given derivation.

Base case. The base case can be confirmed by observing the (id_1) and (id_2) rules.

Inductive step. We consider the $(orth)$ and $(\wedge_r)$ rules, as the $(\vee_r)$, $([x])$, and $(Pr_{\langle x \rangle})$ cases follow directly from the inductive hypothesis. In all cases, it is straightforward to verify that all labels occurring in $\mathcal{I}$ also occur in $\mathcal{R}, \Gamma$ or Δ.

$(orth)$. Let $l \in \mathrm{Lit}(\mathcal{I})$. By IH, we know that $l \in \mathrm{Lit}(\neg\Gamma) \cap \mathrm{Lit}(\Delta)$. By Def. 90, it follows that $\neg l \in \mathrm{Lit}((\mathcal{I})^{\perp})$, which implies that $\neg l \in \mathrm{Lit}(\neg\Delta) \cap \mathrm{Lit}(\Gamma)$ by the fact expressed in the previous sentence.

$(\wedge_r)$. Let $l \in \mathrm{Lit}(\mathcal{I}_1 \cup \mathcal{I}_2)$. Then, either $l \in \mathrm{Lit}(\mathcal{I}_1)$ or $l \in \mathrm{Lit}(\mathcal{I}_2)$. Without loss of generality, we assume the former case holds. This entails that $l \in \mathrm{Lit}(\neg\Gamma) \cap \mathrm{Lit}(w : \phi, \Delta)$, and since $\mathrm{Lit}(w : \phi, \Delta) \subseteq \mathrm{Lit}(w : \phi \wedge \psi, \Delta)$, we have that $l \in \mathrm{Lit}(\neg\Gamma) \cap \mathrm{Lit}(w : \phi \wedge \psi, \Delta)$. □

The following lemma establishes that each interpolation calculus $\mathsf{Km}(S)\mathsf{LI}$ correctly constructs interpolants, and is vital in proving our main theorem below (Thm. 37).

Lemma 66. *For all $\mathcal{R}$, Γ, Δ, and $\mathcal{I}$, if $\vdash_{\mathsf{Km}(S)\mathsf{LI}} \mathcal{R} \Rightarrow \Gamma \mid \Delta \parallel \mathcal{I}$, then*

(i) for all $(\epsilon \Rightarrow \Xi) \in \mathcal{I}$, we have $\vdash_{\mathsf{Km}(S)\mathsf{L}} \mathcal{R} \Rightarrow \Gamma, \Xi$ and

(ii) for all $(\epsilon \Rightarrow \Theta) \in (\mathcal{I})^{\perp}$, we have $\vdash_{\mathsf{Km}(S)\mathsf{L}} \mathcal{R} \Rightarrow \Theta, \Delta$.

Proof. By induction on the height of the given derivation of $\mathcal{R} \Rightarrow \Gamma \mid \Delta \parallel \mathcal{I}$.

Base case. If the interpolation sequent was derived by an instance of (id_1), then both (i) and (ii) follow by using the (id) rule of $\mathsf{Km}(S)\mathsf{L}$. If the interpolation sequent was derived by an instance of (id_2), then (i) and (ii) are resolved as shown below, and make use of the definitions $\top := q \vee \neg q$ and $\bot := q \wedge \neg q$.

$$\dfrac{\dfrac{\dfrac{}{\mathcal{R} \Rightarrow \Gamma, w : q, w : \neg q}(id)}{\mathcal{R} \Rightarrow \Gamma, w : q \vee \neg q}(\vee_r)}{\mathcal{R} \Rightarrow \Gamma, w : \top} = \qquad \dfrac{\dfrac{}{\mathcal{R} \Rightarrow w : q \wedge \neg q, w : p, w : \neg p, \Delta}(id)}{\mathcal{R} \Rightarrow w : \bot, w : p, w : \neg p, \Delta} =$$

Inductive step. We consider the (non-trivial) $(orth)$, $(\wedge_r)$, and $([x])$ cases; the $(\vee_r)$ and $(Pr_{\langle x \rangle})$ cases are straightforward.

$(orth)$. Suppose our derivation ends with and $(orth)$ inference of the following form:

$$\dfrac{\mathcal{R} \Rightarrow \Gamma \mid \Delta \parallel \mathcal{I}}{\mathcal{R} \Rightarrow \Delta \mid \Gamma \parallel (\mathcal{I})^{\perp}}(orth)$$

(i) Let $(\epsilon \Rightarrow \Xi) \in (\mathcal{I})^{\perp}$. To resolve the case, we need to show that $\vdash_{\mathsf{Km}(S)\mathsf{L}} \mathcal{R} \Rightarrow \Xi, \Delta$; however, this follows immediately from IH.

(ii) Let $(\epsilon \Rightarrow \Theta) \in ((\mathcal{I})^{\perp})^{\perp}$. We need to show that $\vdash_{\mathsf{Km}(S)\mathsf{L}} \mathcal{R} \Rightarrow \Gamma, \Theta$. By Lem. 63, we know there exists a $(\epsilon \Rightarrow \Theta') \in \mathcal{I}$ such that $\Theta' \subseteq \Theta$. By IH, $\vdash_{\mathsf{Km}(S)\mathsf{L}} \mathcal{R} \Rightarrow \Gamma, \Theta'$, and by the admissibility of (wk) (Cor. 1), it follows that $\vdash_{\mathsf{Km}(S)\mathsf{L}} \mathcal{R} \Rightarrow \Gamma, \Theta$.

$(\wedge_r)$. Suppose our derivation ends with the following $(\wedge_r)$ inference:

$$\dfrac{\mathcal{R} \Rightarrow \Gamma \mid w : \phi, \Delta \parallel \mathcal{I}_1 \qquad \mathcal{R} \Rightarrow \Gamma \mid w : \psi, \Delta \parallel \mathcal{I}_2}{\mathcal{R} \Rightarrow \Gamma \mid w : \phi \land \psi, \Delta \parallel \mathcal{I}_1 \cup \mathcal{I}_2} \ (\land_r)$$

(i) Let $(\epsilon \Rightarrow \Xi) \in \mathcal{I}_1 \cup \mathcal{I}_2$. By IH, we have that for any $(\epsilon \Rightarrow \Xi_1) \in \mathcal{I}_1$ and $(\epsilon \Rightarrow \Xi_2) \in \mathcal{I}_2$, $\vdash_{\mathsf{Km}(S)\mathsf{L}} \mathcal{R} \Rightarrow \Gamma, \Xi_1$ and $\vdash_{\mathsf{Km}(S)\mathsf{L}} \mathcal{R} \Rightarrow \Gamma, \Xi_2$. Therefore, the conclusion follows regardless of if $(\epsilon \Rightarrow \Xi_1) \in \mathcal{I}_1$ or $(\epsilon \Rightarrow \Xi_2) \in \mathcal{I}_2$.

(ii) Let $(\epsilon \Rightarrow \Theta) \in (\mathcal{I}_1 \cup \mathcal{I}_2)^{\perp}$. By IH, for all $(\epsilon \Rightarrow \Theta_1) \in (\mathcal{I}_1)^{\perp}$ and $(\epsilon \Rightarrow \Theta_2) \in (\mathcal{I}_2)^{\perp}$, $\vdash_{\mathsf{Km}(S)\mathsf{L}} \mathcal{R} \Rightarrow \Theta_1, w : \phi, \Delta$ and $\vdash_{\mathsf{Km}(S)\mathsf{L}} \mathcal{R} \Rightarrow \Theta_2, w : \psi, \Delta$. By Def. 90, there exists a $(\epsilon \Rightarrow \Theta_1) \in (\mathcal{I}_1)^{\perp}$ and a $(\epsilon \Rightarrow \Theta_2) \in (\mathcal{I}_2)^{\perp}$ such that $(\epsilon \Rightarrow \Theta) = (\epsilon \Rightarrow \Theta_1, \Theta_2)$. Therefore, by the admissibility of (wk) (Cor. 1), we can derive $\vdash_{\mathsf{Km}(S)\mathsf{L}} \mathcal{R} \Rightarrow \Theta_1, \Theta_2, w : \phi, \Delta$ and $\vdash_{\mathsf{Km}(S)\mathsf{L}} \mathcal{R} \Rightarrow \Theta_1, \Theta_2, w : \psi, \Delta$. A single application of $(\land_r)$ gives the desired conclusion.

$([x])$. Suppose our derivation ends with an inference of the following form:

$$\dfrac{\mathcal{R}, R_x wu \Rightarrow \Gamma \mid u : \phi, \Delta \parallel \mathcal{I}}{\mathcal{R} \Rightarrow \Gamma \mid w : [x]\phi, \Delta \parallel [x]\mathcal{I}^w_u} \ ([x])$$

(i) Let $(\epsilon \Rightarrow \Xi) \in [x]\mathcal{I}^w_u$. By Def. 91, we know that $\Xi = \Xi', w : [x](\phi_1 \lor \cdots \lor \phi_n)$ with $(\epsilon \Rightarrow, \Xi', u : \phi_1, \ldots, u : \phi_n) \in \mathcal{I}$. We derive the desired conclusion as follows:

$$\begin{array}{ll}
\hdashline \mathcal{R}, R_x wu \Rightarrow \Gamma, \Xi', u : \phi_1, \ldots, u : \phi_n & \text{IH} \\
\hline \mathcal{R}, R_x wu \Rightarrow \Gamma, \Xi', u : \phi_1 \lor \cdots \lor \phi_n & (\lor_r) \times (n-1) \\
\hline \mathcal{R} \Rightarrow \Gamma, \Xi', u : [x](\phi_1 \lor \cdots \lor \phi_n) & ([x])
\end{array}$$

(ii) Let $(\epsilon \Rightarrow \Theta) \in ([x]\mathcal{I}^w_u)^{\perp}$. We need to establish that $\vdash_{\mathsf{Km}(S)\mathsf{L}} \mathcal{R} \Rightarrow \Theta, w : [x]\phi, \Delta$. By Def. 90, Θ may contain zero or more formulae of the form $w : \neg[x](\psi_1 \lor \cdots \lor \psi_n)$ such that there exists a $(\epsilon \Rightarrow \Theta') \in \mathcal{I}$ with $\{u : \psi_1, \ldots, u : \psi_n\} \subseteq \Theta'$. We refer to such formulae as *modal-interpolant* formluae, and for the sake of simplicity we assume that one modal-interpolant formula exists in Θ with $n = 2$, that is, $w : \neg[x](\psi_1 \lor \psi_2)$; we prove the result for this simplifies case as the general case is analogous. We let $\Theta := \Theta', w : \neg[x](\psi_1 \lor \psi_2)$.

By assumption, we know that there exists a $(\epsilon \Rightarrow \Xi) \in \mathcal{I}$ such that $\{u : \psi_1, u : \psi_2\} \subseteq \Xi$, which implies that there exist $(\epsilon \Rightarrow \Xi_1), (\epsilon \Rightarrow \Xi_2) \in (\mathcal{I})^{\perp}$ of the form $\Xi_1 = u : \neg\psi_1, \Theta'$ and $\Xi_2 = u : \neg\psi_2, \Theta'$. By IH , we have:

$$\vdash_{\mathsf{Km}(S)\mathsf{L}} \mathcal{R}, R_x wu \Rightarrow u : \neg\psi_1, \Theta', u : \phi, \Delta$$

$$\vdash_{\mathsf{Km}(S)\mathsf{L}} \mathcal{R}, R_x wu \Rightarrow u : \neg\psi_2, \Theta', u : \phi, \Delta$$

By invoking the admissibility of (wk) (Cor. 1), we obtain:

$$\vdash_{\mathsf{Km}(S)\mathsf{L}} \mathcal{R}, R_x wu \Rightarrow w : \langle x \rangle(\neg\psi_1 \land \neg\psi_2), u : \neg\psi_1, \Theta', u : \phi, \Delta$$

$$\vdash_{\mathsf{Km}(S)\mathsf{L}} \mathcal{R}, R_x wu \Rightarrow w : \langle x \rangle(\neg\psi_1 \land \neg\psi_2), u : \neg\psi_2, \Theta', u : \phi, \Delta$$

Applying the $(\wedge_r)$ rule between the above to labelled sequents gives:

$$\vdash_{\mathsf{Km}(S)\mathsf{L}} \mathcal{R}, R_x wu \Rightarrow w : \langle x\rangle(\neg\psi_1 \wedge \neg\psi_2), u : \neg\psi_1 \wedge \neg\psi_2, \Theta', u : \phi, \Delta$$

Last, the desired conclusion is derived as follows:

$$\dfrac{\dfrac{\dfrac{\mathcal{R}, R_x wu \Rightarrow w : \langle x\rangle(\neg\psi_1 \wedge \neg\psi_2), u : \neg\psi_1 \wedge \neg\psi_2, \Theta', u : \phi, \Delta}{\mathcal{R}, R_x wu \Rightarrow w : \langle x\rangle(\neg\psi_1 \wedge \neg\psi_2), \Theta', u : \phi, \Delta}\,(Pr_{\langle x\rangle})}{\mathcal{R} \Rightarrow w : \langle x\rangle(\neg\psi_1 \wedge \neg\psi_2), \Theta', w : [x]\phi, \Delta}\,([x])}{\mathcal{R} \Rightarrow w : \neg[x](\psi_1 \vee \psi_2), \Theta', w : [x]\phi, \Delta}\,=$$

□

To prove (effective) Lyndon interpolation, we need to be able to construct interpolants that are *formulae* from our more general notion of an interpolant. This construction is possible if all labelled formulae in the interpolant share the same label. In such a case, the interpolant is of the form $\mathcal{I} := \{(\epsilon \Rightarrow \Xi_1), \ldots, (\epsilon \Rightarrow \Xi_n)\}$ with $\Xi_i := \{w : \phi_{i,1}, \ldots, w : \phi_{i,k_i}\}$ for each $1 \leq i \leq n$, and its corresponding formula is $\bigwedge_{i=1}^{n} \bigvee_{j=1}^{k_i} \phi_{i,j}$. Given such an interpolant $\mathcal{I}$, we write $\bigwedge\bigvee\mathcal{I}$ to mean its corresponding formula. The following two lemmata show how to construct the corresponding formula of an interpolant (in the form just described) proof-theoretically, which will inevitably let us construct Lyndon interpolants effectively:

Lemma 67. *Let $\mathcal{I} := \{(\epsilon \Rightarrow \Xi_1), \ldots, (\epsilon \Rightarrow \Xi_n)\}$ with $\Xi_i := \{w : \phi_{i,1}, \ldots, w : \phi_{i,k_i}\}$ for each $1 \leq i \leq n$. For any multiset of relational atoms $\mathcal{R}$ and multiset of labelled formulae Γ, if $\vdash_{\mathsf{Km}(S)\mathsf{L}} \mathcal{R} \Rightarrow \Gamma, \Xi$ for all $(\epsilon \Rightarrow \Xi) \in \mathcal{I}$, then $\vdash_{\mathsf{Km}(S)\mathsf{L}} \mathcal{R} \Rightarrow \Gamma, w : \bigwedge\bigvee\mathcal{I}$.*

Proof. Assume that $\vdash_{\mathsf{Km}(S)\mathsf{L}} \mathcal{R} \Rightarrow \Gamma, \Xi_i$, for all $\Xi_i \in \mathcal{I}$. By repeated application of the $(\vee_r)$ rule, we obtain

$$\vdash_{\mathsf{Km}(S)\mathsf{L}} \mathcal{R} \Rightarrow \Gamma, w : \bigvee_{j=1}^{k_i} \phi_{i,j}$$

for each $i \in \{1, \ldots, n\}$. Repeated application of the $(\wedge_r)$ rule gives us

$$\vdash_{\mathsf{Km}(S)\mathsf{L}} \mathcal{R} \Rightarrow \Gamma, w : \bigwedge_{i=1}^{n} \bigvee_{j=1}^{k_i} \phi_{i,j}$$

which is the desired conclusion since $\bigwedge\bigvee\mathcal{I} = \bigwedge_{i=1}^{n} \bigvee_{j=1}^{k_i} \phi_{i,j}$. □

Lemma 68. *Let $\mathcal{I} := \{(\epsilon \Rightarrow \Xi_1), \ldots, (\epsilon \Rightarrow \Xi_n)\}$ be an interpolant with $\Xi_i := \{w : \phi_{i,1}, \ldots, w : \phi_{i,k_i}\}$ for each $1 \leq i \leq n$. For any multiset of relational atoms $\mathcal{R}$ and multiset of labelled formulae Δ, if $\vdash_{\mathsf{Km}(S)\mathsf{L}} \mathcal{R} \Rightarrow \Theta, \Delta$ for all $(\epsilon \Rightarrow \Theta) \in (\mathcal{I})^{\perp}$, then $\vdash_{\mathsf{Km}(S)\mathsf{L}} \mathcal{R} \Rightarrow w : \neg\bigwedge\bigvee\mathcal{I}, \Delta$.*

Proof. We prove that $\vdash_{\mathsf{Km}(S)\mathsf{LI}} \mathcal{R}w : \neg\bigwedge\bigvee\mathcal{I}, \Delta$ by induction on the cardinality of $\mathcal{I}$.

Base case. If $\mathcal{I}$ is a singleton, then by assumption $\vdash_{\mathsf{Km}(S)\mathsf{L}} \mathcal{R} \Rightarrow w : \neg\phi_{1,j}, \Delta$ is derivable for all $j \in \{1, \ldots, k_1\}$. The conclusion follows by $k_1 - 1$ applications of the $(\wedge_r)$ rule.

Inductive step. Suppose that $\mathcal{I} := \{(\epsilon \Rightarrow \Theta_1), \ldots, (\epsilon \Rightarrow \Theta_{n+1})\}$ contains $n+1$ elements and assume that $\vdash_{\mathsf{Km}(S)\mathsf{L}} \mathcal{R} \Rightarrow \Theta, \Delta$ for all $(\epsilon \Rightarrow \Theta) \in (\mathcal{I})^{\perp}$. It follows that for each $j \in \{1, \ldots, k_{n+1}\}$, $\vdash_{\mathsf{Km}(S)\mathsf{L}} \mathcal{R} \Rightarrow \Theta', w : \neg\phi_{n+1,j}, \Delta$ for all $(\epsilon \Rightarrow \Theta') \in (\mathcal{I} - \{(\epsilon \Rightarrow \Theta_{n+1})\})^{\perp}$, which implies that

$$\vdash_{\mathsf{Km}(S)\mathsf{L}} \mathcal{R} \Rightarrow w : \bigvee\bigwedge\neg(\mathcal{I} - \{(\epsilon \Rightarrow \Theta_{n+1})\}), w : \neg\phi_{n+1,j}, \Delta$$

for each $j \in \{1, \ldots, k_{n+1}\}$. By applying the $(\wedge_r)$ rule between each of the $k_{n+1} - 1$ labelled sequents above, followed by a single application of the $(\vee_r)$ rule, we obtain

$$\vdash_{\mathsf{Km}(S)\mathsf{L}} \mathcal{R} \Rightarrow w : \bigvee\bigwedge\neg(\mathcal{I} - \{(\epsilon \Rightarrow \Theta_{n+1})\}) \vee \bigwedge_{1 \leq j \leq k_{n+1}} \neg\phi_{n+1,j}, \Delta$$

which gives our desired conclusion since

$$\neg\bigwedge\bigvee\mathcal{I} = \bigvee\bigwedge\neg(\mathcal{I} - \{(\epsilon \Rightarrow \Theta_{n+1})\}) \vee \bigwedge_{1 \leq j \leq k_{n+1}} \neg\phi_{n+1,j}.$$

□

Let us now state and prove our main theorem, which entails that each grammar logic in our class possesses the effective Lyndon interpolation property.

Theorem 37. *If* $\vdash_{\mathsf{Km}(S)\mathsf{L}} w : \phi \rightarrow \psi$*, then there exists a* χ *such that (i)* $\mathrm{Lit}(\chi) \subseteq \mathrm{Lit}(\phi) \cap \mathrm{Lit}(\psi)$*, and (ii)* $\vdash_{\mathsf{Km}(S)\mathsf{L}} \epsilon \Rightarrow w : \phi \rightarrow \chi$ *and* $\vdash_{\mathsf{Km}(S)\mathsf{L}} \epsilon \Rightarrow w : \chi \rightarrow \psi$*.*

Proof. Suppose that $\vdash_{\mathsf{Km}(S)\mathsf{L}} \epsilon \Rightarrow w : \phi \rightarrow \psi$, that is, $\vdash_{\mathsf{Km}(S)\mathsf{L}} \epsilon \Rightarrow w : \neg\phi \vee \psi$. By the invertibility of $(\vee_r)$ (Cor. 2), we have $\vdash_{\mathsf{Km}(S)\mathsf{L}} \epsilon \Rightarrow w : \neg\phi, w : \psi$. Therefore, by Lem. 64 and 65, there exists an interpolant $\mathcal{I} := \{(\epsilon \Rightarrow \Xi_1), \ldots, (\epsilon \Rightarrow \Xi_n)\}$ with $\Xi_i = \{w : \chi_{i,1}, \ldots, w : \chi_{i,k_i}\}$ for $i \in \{1, \ldots, n\}$ such that $\vdash_{\mathsf{Km}(S)\mathsf{LI}} \epsilon \Rightarrow w : \neg\phi \mid w : \psi \parallel \mathcal{I}$ and $\mathrm{Lit}(\mathcal{I}) \subseteq \mathrm{Lit}(w : \neg\neg\phi) \cap \mathrm{Lit}(w : \psi) = \mathrm{Lit}(\phi) \cap \mathrm{Lit}(\psi)$. By Lem. 66, we know that:

(a) For all $(\epsilon \Rightarrow \Xi) \in \mathcal{I}$, $\vdash_{\mathsf{Km}(S)\mathsf{L}} \epsilon \Rightarrow w : \neg\phi, \Xi$, and

(b) for all $(\epsilon \Rightarrow \Theta) \in (\mathcal{I})^{\perp}$, $\vdash_{\mathsf{Km}(S)\mathsf{L}} \epsilon \Rightarrow \Theta, w : \psi$.

By Lem. 67, claim (a) implies that $\vdash_{kmsl} \epsilon \Rightarrow w : \neg\phi, \bigwedge\bigvee\mathcal{I}$, and by Lem. 68, claim (b) implies that $\vdash_{kmsl} \epsilon \Rightarrow w : \neg\bigwedge\bigvee\mathcal{I}, w : \psi$. The truth of claim (ii) is confirmed by the following derivations in $\mathsf{Km}(S)\mathsf{L}$:

$$\dfrac{\dfrac{\epsilon \Rightarrow w : \neg\phi, \bigwedge\bigvee\mathcal{I}}{\epsilon \Rightarrow w : \neg\phi \vee \bigwedge\bigvee\mathcal{I}}\,(\vee_r)}{\epsilon \Rightarrow w : \phi \rightarrow \bigwedge\bigvee\mathcal{I}}\,= \qquad \dfrac{\dfrac{\epsilon \Rightarrow w : \neg\bigwedge\bigvee\mathcal{I}, w : \psi}{\epsilon \Rightarrow w : \neg\bigwedge\bigvee\mathcal{I} \vee \psi}\,(\vee_r)}{\epsilon \Rightarrow w : \bigwedge\bigvee\mathcal{I} \rightarrow \psi}\,=$$

The truth of claim (i) follows from the fact that $\mathrm{Lit}(\bigwedge\bigvee\mathcal{I}) = \mathrm{Lit}(\mathcal{I})$. □

Corollary 9. *Every context-free grammar logic with converse has the effective Lyndon interpolation property.*

Proof. Follows from the soundness and completeness of $\mathsf{Km}(S)\mathsf{L}$ (Cor. 3) and Thm. 37 above. Also, the property is effective because Thm. 37 gives a procedure showing how to construct the interpolant of any valid implication. □

CHAPTER 7

Conclusion and Future Work

This thesis began by introducing labelled sequent calculi for a diverse class of logics, and then presented the method of refinement as a means of 'simplifying' the systems. Since labelled calculi are straightforwardly obtainable from a logic's semantics, the method of refinement can be seen as a strategy for transforming the relational (i.e. Kripke) semantics of a modal or constructive logic into a proof system with an economical amount of sequential structure and which encodes semantic information within the functionality of logical rules. The method leverages results from the labelled paradigm to obtain labelled sequent calculi for modal and constructive logics via their semantics, and then performs structural rule elimination to reduce the syntactic structures utilized in the proof systems. As was seen, nested calculi regularly result as the output of this method suggesting that the method serves as a theoretical basis underpinning a variety of nested sequent systems. Nevertheless, the work in Sect. 4.2 on refining labelled systems for deontic STIT logics showed that refinement can still be performed in the presence of certain structural rules which do not appear immediately eliminable via the method. The last technical chapter of the thesis (Ch. 6) harnessed refined labelled calculi for deontic STIT logics to provide proof-search algorithms and made use of the refined labelled calculi for context-free grammar logics with converse (along with the syntactic method of interpolation from [LTGC20]) to show that all such grammar logics possess the effective Lyndon interpolation property.

A promising avenue of future research concerns the development of a general theory of nested sequent systems for large classes of logics characterized by relational semantics. Not only would such a theory be useful in generating nested systems on demand for modal, constructive, and related logics—which have found significant applications in philosophy [BP90, vBL21], legal theory [Bro11b, LS15], artificial intelligence [MH81], verification [CES86, SBA20], and distributed computing [HM90]—but would allow for automated reasoning techniques to be applied to these logics by harnessing their nested systems (e.g. [FK15, LvB19, LTGC20, TIG12]). Also, although refined calculi are often notational variants of nested systems, it seems plausible that through the implementation of additional methods, one can potentially simplify the refined systems even further to obtain calculi within the paradigm of hypersequents [Avr96], linear nested sequents [Lel15], or Gentzen-style sequents [PRL19].

Moreover, we would like to apply the refinement method to modal and constructive logics outside the classes discussed. As was hinted at and briefly discussed in Ch. 5, the use of grammar theoretic machinery in the refined labelled systems for first-order intuitionistic logics should allow for the systems to be straightforwardly transformed into systems for a diverse class of propositional and first-order (sub-)(bi-)intuitionistic logics—given that logical rules for the exclusion connective $-\!\!<$ (the dual of intuitionistic implication) are added in the bi-intuitionistic cases. (NB. See [Cor87, Res94] for information on sub-intuitionistic logics and [GPT08, PU18, Rau80] for information on bi-intuitionistic logic.) Other fruitful candidates for applying refinement concern modal intuitionistic logics and bi-intuitionistic tense logics. Logics within both classes possess known nested sequent calculi (e.g. see [Str13] for modal intuitionistic nested systems and [GPT08] for a bi-intuitionistic tense nested system), and since such calculi are typically the output of refinement, it is reasonable to conjecture that labelled sequent systems for the logics are refinable. Moreover, Simpson [Sim94] defined labelled sequent systems for modal intuitionistic logics and defined propagation rules for such systems, which further strengthens the possibility that such systems can be refined due to the central role played by propagation rules. We are also interested in applying refinement to hybrid logics (e.g. [BdRV01, Sect. 7.3]) and relevance logics (e.g. [Vig00, Ch. 3]).

Another promising direction for future research regards refinement in the presence of structural rules that do not appear eliminable. In Sect. 4.2, where the refinement of labelled systems for deontic STIT logics was discussed, the (IOA) and (APC_i^k) rules were not eliminated, but allowed for the elimination of the other structural rules. Both rules had the property that the conclusion was free of active relational atoms. This suggests that rules of a similar shape ought to allow for other structural rules to be eliminated in a labelled sequent system, allowing for the calculus to be refined to a degree. We aim to investigate what sets of rules allow for refinement to be performed and the degree to which structural rules can be eliminated and replaced by propagation or reachability rules. Furthermore, as was seen in Sect. 4.2, the class $\{\mathsf{DS}_n^0\mathsf{L} \mid n \in \mathbb{N}\}$ of refined labelled calculi for deontic STIT logics only required labelled DAG derivations, showing that such calculi are close relatives of indexed-nested sequent systems (cf. [Fit15, MS17]). Therefore, the question arises: can refinement serve as a foundation for producing large classes of indexed-nested sequent calculi as well? As explained in [MS17], it seems that some logics cannot be given a cut-free treatment in the nested sequent formalism (e.g. modal logics extended with Scott-Lemmon axioms [LS77]), and so, indexed-nested sequents were introduced as a slight extension of nested sequents to capture such logics in a cut-free manner [Fit15]. Such systems employ sequents whose underlying data structure is a directed acyclic graph, demonstrating that such systems exhibit a higher degree of parsimony than their labelled counterparts (which use general graphs). As with nested sequent systems, this reduction in structure could compress the size of proofs (relative to labelled proofs of the same theorems), and lead to a savings in space as well as ease the confirmation of termination for associated proof-search algorithms.

Last, it seems worthwhile to investigate if labelled sequent systems can be bypassed altogether and if refined labelled systems using propagation and reachability rules can be straightforwardly obtained from the semantics of a logic. If so, then perhaps general results could be obtained for modal, constructive, and related logics demonstrating that (indexed-)nested sequent calculi are immediately obtainable from a logic's relational semantics. It is conceivable that such results could yield algorithms which transform frame conditions or axioms into propagation and reachability rules, thus allowing for refined labelled, nested, or indexed-nested systems to be obtained in an automated fashion. Such work is reminiscent of the algorithms in [CGT08, CMS13, CST09, Lah13], which transform axioms into structural rules, rather than propagation and reachability rules, in order to obtain classes of cut-free calculi for sizable classes of logics.

Appendix

A Fitting's Nested Calculi for First-Order Intuitionistic Logics

In this section of the appendix we introduce Fitting's nested calculi for the first-order intuitionistic logics IntQ and IntQC (see Def. 20). We first define Fitting's notion of an *available parameter*, which serves as a side condition on the $(\exists_r)$ and $(\forall_l)$ rule in the nested calculus for first-order intuitionistic logic with non-constant domains; afterward, we define his nested calculi.

Definition 97 (Available Parameter [Fit14]). *Let $X[Y \Rightarrow Z, \{X_1\}, \ldots, \{X_n\}]$ be a nested sequent. If there exists a formula $A(\underline{a}) \in Y, Z$, then the parameter $\underline{a}$ is available in $Y \Rightarrow Z$ and in all boxed subsequents X_i (with $i \in \{1, \ldots, n\}$).*

Definition 98 (Fitting's Nested Calculi for IntQ and IntQC [Fit14]). *The rules for Fitting's nested calculi are shown below. We define the nested calculus for IntQ and IntQC, and note that the side condition $\dagger_1$ states that the parameter $\underline{a}$ is either available or is an eigenvariable, and $\dagger_2$ states that $\underline{a}$ is an eigenvariable.*

1. *The nested calculus for first-order intuitionistic logic with non-constant domains (IntQ) consists of (id), $(\wedge_l)$, $(\wedge_r)$, $(\vee_r)$, $(\vee_l)$, $(\neg_l)$, $(\neg_r)$, $(\supset_r)$, $(\supset_l)$, $(lift)$, $(\exists_l)$, $(\exists_r)$, $(\forall_l)$, and $(\forall_r^{\mathsf{Q}})$.*

2. *The nested calculus first-order intuitionistic logic with constant domains (IntQC) consists of (id), $(\wedge_l)$, $(\wedge_r)$, $(\vee_r)$, $(\vee_l)$, $(\neg_l)$, $(\neg_r)$, $(\supset_r)$, $(\supset_l)$, $(lift)$, $(\exists_l)$, $(\exists_r)$, $(\forall_l)$, and $(\forall_r^{\mathsf{QC}})$ and omits the side condition $\dagger_1$ on the $(\exists_r)$ and $(\forall_l)$ rules.*

$$\frac{}{X[Y, p(\vec{\underline{a}}) \Rightarrow p(\vec{\underline{a}}), Z]}\,(id) \qquad \frac{X[Y, A, B \Rightarrow Z]}{X[Y, A \wedge B \Rightarrow Z]}\,(\wedge_l) \qquad \frac{X[Y, A(\underline{a}/x) \Rightarrow Z]}{X[Y, \forall x A \Rightarrow Z]}\,(\forall_l)^{\dagger_1}$$

$$\frac{X[Y, A \Rightarrow Z] \qquad X[Y, B \Rightarrow Z]}{X[Y, A \vee B \Rightarrow Z]}\,(\vee_l) \qquad \frac{X[Y \Rightarrow A, Z] \qquad X[Y \Rightarrow B, Z]}{X[Y \Rightarrow A \wedge B, Z]}\,(\wedge_r)$$

$$\frac{X[Y \Rightarrow Z, \{A \Rightarrow \epsilon\}]}{X[Y \Rightarrow Z, \neg A]}\,(\neg_r) \qquad \frac{X[Y \Rightarrow A, Z]}{X[Y, \neg A \Rightarrow Z]}\,(\neg_l) \qquad \frac{X[Y \Rightarrow A, B, Z]}{X[Y \Rightarrow A \vee B, Z]}\,(\vee_r)$$

$$\frac{X[Y \Rightarrow Z, \{A \Rightarrow B\}]}{X[Y \Rightarrow A \supset B, Z]}\ (\supset_r) \qquad \frac{X[Y \Rightarrow A, Z] \qquad X[Y, B \Rightarrow Z]}{X[Y, A \supset B \Rightarrow Z]}\ (\supset_l)$$

$$\frac{X[Y \Rightarrow A(\underline{a}/x), Z]}{X[Y \Rightarrow \exists x A, Z]}\ (\exists_r)^{\dagger_1} \qquad \frac{X[Y, A(\underline{a}/x) \Rightarrow Z]}{X[Y, \exists x A \Rightarrow Z]}\ (\exists_l)^{\dagger_2} \qquad \frac{X[Y \Rightarrow A(\underline{a}/x), Z]}{X[Y \Rightarrow \forall x A, Z]}\ (\forall_r^{\mathsf{QC}})^{\dagger_2}$$

$$\frac{X[Y \Rightarrow Z, \{\epsilon \Rightarrow A(\underline{a}/x)\}]}{X[Y \Rightarrow \forall x A, Z]}\ (\forall_r^{\mathsf{Q}})^{\dagger_2} \qquad \frac{X[Y \Rightarrow Z, \{Y', A \Rightarrow Z'\}]}{X[Y, A \Rightarrow Z, \{Y' \Rightarrow Z'\}]}\ (lift)$$

B Classical Completeness of $\mathsf{G3Km}(S)$ and $\mathsf{G3DS}_n^k$

In this appendix, we show that all instances of classical propositional tautologies in $\mathcal{L}_{\mathsf{Km}}(\sum)$ and $\mathcal{L}_{\mathsf{DS}_n^k}$ are derivable in $\mathsf{G3Km}(S)$ and $\mathsf{G3DS}_n^k$, respectively. To prove this, we first show that the rules $\{(id), (\vee_r), (\wedge_r)\}$—all of which are contained in $\mathsf{G3Km}(S)$ and $\mathsf{G3DS}_n^k$—are complete relative to classical propositional logic. Second, we show how a proof of a classical propositional tautology can be proof-theoretically transformed into a proof of any *instance* of the tautology in $\mathcal{L}_{\mathsf{Km}}(\sum)$ and $\mathcal{L}_{\mathsf{DS}_n^k}$.

Definition 99 (The Language $\mathcal{L}$)**.** *We define the* classical propositional language $\mathcal{L}$ *as follows:*

$$\phi ::= p \mid \neg p \mid (\phi \vee \phi) \mid (\phi \wedge \phi)$$

where $p \in$ Prop*.*

Definition 100 (Valuation, Tautology)**.** *We say that a function* $\mathbb{V} : \mathcal{L} \mapsto \{0, 1\}$ *mapping formulae of* $\mathcal{L}$ *to the truth values in* $\{0, 1\}$ *is a* valuation *iff the following hold:*

- $\mathbb{V}(p) \in \{0, 1\}$*, for each* $p \in$ Prop
- $\mathbb{V}(\neg p) = 1 - \mathbb{V}(p)$*, for each* $p \in$ Prop
- $\mathbb{V}(\phi \vee \psi) = max\{\mathbb{V}(\phi), \mathbb{V}(\psi)\}$
- $\mathbb{V}(\phi \wedge \psi) = min\{\mathbb{V}(\phi), \mathbb{V}(\psi)\}$

We say that a formula $\phi \in \mathcal{L}$ *is a* classical propositional tautology *iff* $\mathbb{V}(\phi) = 1$ *for all valuations* $\mathbb{V}$*.*

Definition 101 (Saturation)**.** *Let* $\Lambda := \epsilon \Rightarrow \Gamma$ *be a labelled sequent. We say that* Λ *is* saturated *iff*

- *If* $w : \phi \in \Gamma$*, then* $w : \neg\phi \notin \Gamma$*;*
- *If* $w : \phi \vee \psi \in \Gamma$*, then* $w : \phi, w : \psi \in \Gamma$*;*

Algorithm 7.1: `ProveCPL`

```
Input: A Labelled Sequent: $\epsilon \Rightarrow \Gamma$
Output: A Boolean: True, False
if $w : p, w : \neg p \in \Gamma$ then
    return True;
end
if $\epsilon \Rightarrow \Gamma$ is saturated then
    return False;
end
if $w : \phi \vee \psi \in \Gamma$, but either $w : \phi \notin \Gamma$ or $w : \psi \notin \Gamma$ then
    Let $\Gamma' := w : \phi, w : \psi, \Gamma$;
    return ProveCPL($\epsilon \Rightarrow \Gamma'$);
end
if $w : \phi \wedge \psi \in \Gamma$, but $w : \phi, w : \psi \notin \Gamma$ then
    Let $\Gamma_1 := w : \phi, \Gamma$;
    Let $\Gamma_2 := w : \psi, \Gamma$;
    if ProveCPL($\mathcal{R} \Rightarrow \Gamma_i$) = False for some $i \in \{1, 2\}$ then
        return False;
    end
    else
        return True;
    end
end
```

- *If $w : \phi \wedge \psi \in \Gamma$, then either $w : \phi \in \Gamma$ or $w : \psi \in \Gamma$.*

Lemma 69 (Termination of `ProveCPL`). *Let $\phi \in \mathcal{L}$. Then,* `ProveCPL`*($w : \phi$) terminates.*

Proof. Let $\Lambda := \epsilon \Rightarrow \Gamma$. We first define what it means for a labelled formula to be *active* in $\epsilon \Rightarrow \Gamma$ below:

- For all $p \in \text{Prop}$, neither $w : p$ nor $w : \neg p$ are *active* in $\epsilon \Rightarrow \Gamma$;
- Consider a formula ψ of the form $\chi \vee \xi$. Then, $w : \psi$ is *active* in $\epsilon \Rightarrow \Gamma$ iff $w : \psi \in \Gamma$, and either $w : \chi \notin \Gamma$ or $w : \xi \notin \Gamma$;
- Consider a formula ψ of the form $\chi \wedge \xi$. Then, $w : \psi$ is *active* in $\epsilon \Rightarrow \Gamma$ iff $w : \psi \in \Gamma$, and $w : \chi, w : \xi \notin \Gamma$.

Next, we define the *complexity* of a labelled sequent $\Lambda = \epsilon \Rightarrow \Gamma$ (written $|\Lambda| = |\epsilon \Rightarrow \Gamma|$) as follows:

$$|\Lambda| := \sum_{\psi \in \mathrm{X}} |\psi|$$

with

$$\mathrm{X} := \{\psi \mid w : \psi \in \Gamma \text{ and } w : \psi \text{ is active in } \epsilon \Rightarrow \Gamma.\}$$

The recursive call of line 7 "deactivates" a formula of the form $\psi \vee \chi$, and introduces ψ and χ into the sequent, while the recursive call of line 11 "deactivates" a formula of the form $\psi \wedge \chi$ and either introduces ψ or χ to a sequent. In the former case, $|\psi| + |\chi| < |\psi \vee \chi|$, and in the latter case $|\psi|, |\chi| < |\psi \wedge \chi|$, meaning that the complexity continually decreases—implying termination. □

Lemma 70 (Correctness of `ProveCPL`). *Let S be a CFCST system with alphabet $\sum$, and $n, k \in \mathbb{N}$*

(i) If `ProveCPL`*($w : \phi$) = `True`, then $w : \phi$ is provable in* $\mathsf{G3Km}(S)$ *and* $\mathsf{G3DS}^k_n$.

(ii) If `ProveCPL`*($w : \phi$) = `False`, then there exists a valuation $\mathbb{V}$ such that $\mathbb{V}(\phi) = 0$.*

Proof. Claim (i) follows from the fact that each recursive call (lines 7 and 11) to `ProveCPL` corresponds to a bottom-up application of (ctr_r), followed by an instance of $(\vee_r)$ or $(\wedge_r)$. Since (ctr_r) is hp-admissible in both $\mathsf{G3Km}(S)$ and $\mathsf{G3DS}^k_n$ (Lem. 10 and Lem. 23, resp.), we know that if `ProveCPL`($w : \phi$) = `True`, then a proof of $w : \phi$ was constructed in both $\mathsf{G3Km}(S)$ and $\mathsf{G3DS}^k_n$.

Let us now prove claim (ii). Since `ProveCPL`($\epsilon \Rightarrow w : \phi$) = `False`, we know that a saturated sequent $\epsilon \Rightarrow \Gamma$ was generated. We will make use of $\epsilon \Rightarrow \Gamma$ to construct a valuation $\mathbb{V}$ such that $\mathbb{V}(\phi) = 0$. We define $\mathbb{V}$ as follows: $\mathbb{V}(p) = 1$ iff $w : \neg p \in \Gamma$. Let us now prove the following claim by induction on the complexity of ψ: if $w : \psi \in \Gamma$, then $\mathbb{V}(\psi) = 0$.

Base case. Let $p \in$ Prop. Suppose that ψ is of the form p, and assume that $w : p \in \Gamma$. Since Γ is saturated, it follows that $w : \neg p \notin \Gamma$. Hence, by the definition of $\mathbb{V}$, we know that $\mathbb{V}(p) = 0$. If ψ is of the form $\neg p$, then the claim follows directly from the assumption that $w : \neg p \in \Gamma$ and the definition of $\mathbb{V}$.

Inductive step. We consider the case where ψ is of the form $\chi \vee \xi$ and the case were ψ is of the form $\chi \wedge \xi$.

$\vee$. Suppose that $w : \psi = w : \chi \vee \xi \in \Gamma$. Since $\epsilon \Rightarrow \Gamma$ is saturated, we know that $w : \chi, w : \xi \in \Gamma$ by the recursive call on line 7. By IH it follows that $\mathbb{V}(\chi) = \mathbb{V}(\xi) = 0$, implying that $\mathbb{V}(\psi) = 0$.

$\wedge$. Suppose that $w : \psi = w : \chi \wedge \xi \in \Gamma$. Since $\epsilon \Rightarrow \Gamma$ is saturated, we know that either $w : \chi \in \Gamma$ or $w : \xi \in \Gamma$ by the recursive call on line 11. By IH it follows that either $\mathbb{V}(\chi) = 0$ or $\mathbb{V}(\xi) = 0$, implying that $\mathbb{V}(\psi) = 0$. □

Lemma 12. *All instances of classical propositional tautologies in $\mathcal{L}_{\mathsf{Km}}(\sum)$ are derivable in* $\mathsf{G3Km}(S)$.

Proof. Let ϕ be a classical propositional tautology. Then, $\mathbb{V}(\phi) = 1$ for all valuations $\mathbb{V}$, implying that `ProveCPL`($w : \phi$) $\neq$ `False` by Lem. 70 above. By Lem. 69, this implies that $\texttt{ProveCPL}(w : \phi) = \texttt{True}$, which implies that $\epsilon \Rightarrow w : \phi$ has a proof in $\mathsf{G3Km}(S)$ by Lem. 70. Let us call this proof Π. Observe that Π is of the following form:

$$\Pi \quad = \left\{ \begin{array}{ccc} \dfrac{}{\Rightarrow w : p_0, w : \neg p_0, \Gamma_0}\,(id) & \cdots & \dfrac{}{\Rightarrow w : p_n, w : \neg p_n, \Gamma_n}\,(id) \\ & \vdots & \\ & \epsilon \Rightarrow w : \phi & \end{array} \right.$$

In order to show that all *instances* of classical propositional tautologies in $\mathcal{L}_{\mathsf{Km}}(\sum)$ are derivable, we have to show that for any substitution $\sigma = (\psi_0/p_0, \ldots, \psi_n/p_n)$ of formulae $\psi_i \in \mathcal{L}_{\mathsf{Km}}(\sum)$ for the propositional variables p_i (with $0 \leq i \leq n$), $w : \phi\sigma$ is derivable in $\mathsf{G3Km}(S)$. To show this, we first apply the substitution σ to the entire derivation Π, i.e. to every labelled formula of every sequent in Π, yielding the following derivation $\Pi\sigma$:

$$\Pi\sigma \quad = \left\{ \begin{array}{ccc} \dfrac{}{\Rightarrow w : \psi_0, w : \neg\psi_0, \Gamma_0\sigma}\,(id) & \cdots & \dfrac{}{\Rightarrow w : \psi_n, w : \neg\psi_n, \Gamma_n\sigma}\,(id) \\ & \vdots & \\ & \Rightarrow w : \phi\sigma & \end{array} \right.$$

Next, we think of each instance of (id) as an *interface*, and employ Lem. 9, which implies that every sequent of the form $\Rightarrow w : \psi_i, w : \neg\psi_i, \Gamma_i\sigma$ (for $0 \leq i \leq n$) is derivable in $\mathsf{G3Km}(S)$. Let us call the derivation of each such sequent Π_i. By attaching these derivations to the leaves, as shown below, we obtain a proof Π' of the desired conclusion:

$$\Pi' \quad = \left\{ \begin{array}{ccc} \Pi_1 & & \Pi_n \\ \Rightarrow w : \psi_0, w : \neg\psi_0, \Gamma_0\sigma & \cdots & \Rightarrow w : \psi_n, w : \neg\psi_n, \Gamma_n\sigma \\ & \vdots & \\ & \Rightarrow w : \phi\sigma & \end{array} \right.$$

□

Lemma 24. *All instances of classical propositional tautologies in $\mathcal{L}_{\mathsf{DS}^k_n}$ are derivable in $\mathsf{G3DS}^k_n$.*

Proof. Similar to the proof of Lem. 12. □

C Additional Proofs

Lemma 38.

(i) The rule (id) is an instance of (id^{Q}) and (id^{QC}).

(ii) The $(\supset_l)$ rule is an instance of the $(Pr_{\supset})$ rule.

(iii) The $(\exists_r)$ rule is an instance of the rules $(\exists^{\mathsf{Q}}_{r1})$ and $(\exists^{\mathsf{QC}}_{r1})$.

(iv) The $(\forall_l)$ rule is an instance of the rules $(\forall^{\mathsf{Q}}_{l1})$ and $(\forall^{\mathsf{QC}}_{l1})$.

Proof. We prove claims (ii)–(iv) below:

(ii) In the $(Pr_{\supset})$ inference below, observe that due to the $w \leq u$ relational atom occurring in the premises, we know that the propagation path $\pi(w,u) := w, a, u$ occurs in the propagation graphs of both premises. Since $a \in L_{S4}(a)$, and because $s_{\pi}(w,u) = a$, it follows that $s_{\pi}(w,u) \in L_{S4}(a)$ showing that the side condition is satisfied, and therefore, the $(Pr_{\supset})$ rule may be applied to the premises to derive the desired conclusion.

$$\frac{\mathcal{R}, w \leq u, \Gamma, w : \phi \supset \psi \Rightarrow \Delta, u : \phi \qquad \mathcal{R}, w \leq u, \Gamma, w : \phi \supset \psi, u : \psi \Rightarrow \Delta}{\mathcal{R}, w \leq u, \Gamma, w : \phi \supset \psi \Rightarrow \Delta}\ (Pr_{\supset})$$

(iii) We prove that $(\exists_{r})$ is an instance of $(\exists^{\mathsf{QC}}_{r1})$ since showing that $(\exists_{r})$ is an instance of $(\exists^{\mathsf{Q}}_{r1})$ is similar. By Def. 73, we know that $\varepsilon \in L_{S5}(a)$. We also know that $s_{\lambda}(w,w) = \varepsilon$ by Def. 59, implying that $s_{\lambda}(w,w) \in L_{S5}(a)$. Consequently, the side condition of $(\exists^{\mathsf{QC}}_{r1})$ is satisfied, and the following inference may be performed:

$$\frac{\mathcal{R}, \underline{a} \in D_{w}, \Gamma \Rightarrow w : \phi(\underline{a}/x), w : \exists x \phi, \Delta}{\mathcal{R}, \underline{a} \in D_{w}, \Gamma \Rightarrow w : \exists x \phi, \Delta}\ (\exists^{\mathsf{QC}}_{r1})$$

(iv) We show that the $(\forall_{l})$ rule is an instance of $(\forall^{\mathsf{QC}}_{l1})$ since the proof for $(\forall^{\mathsf{Q}}_{l1})$ is similar. By Def. 73, we know that $\varepsilon \in L_{S5}(a)$, and since there is a propagation path $\lambda(u,u) = u$ from u to itself with $s_{\lambda}(u,u) = \varepsilon$, it follows that $s_{\lambda}(u,u) \in L_{S5}(a)$. Furthermore, by Def. 9, we know that $a \twoheadrightarrow^{*}_{S5} a$ holds, implying that $a \in L_{S4}(a)$. Since there is a propagation path $\pi(w,u) = w, a, u$ and $s_{\pi}(w,u) = a$, we have that $s_{\pi}(w,u) \in L_{S4}(a)$. Therefore, the side condition of $(\forall^{\mathsf{QC}}_{l1})$ holds and the rule can simulate $(\forall_{l})$ as shown below.

$$\frac{\mathcal{R}, w \leq u, \underline{a} \in D_{u}, \Gamma, u : \phi(\underline{a}/x), w : \forall x \phi \Rightarrow \Delta}{\mathcal{R}, w \leq u, \underline{a} \in D_{u}, \Gamma, w : \forall x \phi \Rightarrow \Delta}\ (\forall^{\mathsf{QC}}_{l1})$$

□

Lemma 40.

(i) The rule (tra) *is eliminable in* $\mathsf{G3IntQ} + \mathrm{R}(\mathsf{Q}) - \{(ref), (nd), (ned)\}$.

(ii) The rule (tra) *is eliminable in* $\mathsf{G3IntQC} + \mathrm{R}(\mathsf{QC}) - \{(ref), (nd), (cd), (ned)\}$.

Proof. We prove the result by induction on the height of the given derivation and only show claim (ii) as it subsumes claim (i).

Base case. The $(\bot_{l})$ case is simple to verify and due to Lem. 38, we do not need to consider the (id) case, so we only consider the (id^{QC}) case below:

$$\dfrac{\dfrac{}{\mathcal{R}, w \leq u, u \leq v, w \leq v, \underline{a}_{1} \in D_{u_{1}}, \ldots, \underline{a}_{n} \in D_{u_{n}}, \Gamma, w : p(\vec{\underline{a}}) \Rightarrow u : p(\vec{\underline{a}}), \Delta}\ (id^{\mathsf{QC}})}{\mathcal{R}, w \leq u, u \leq v, \underline{a}_{1} \in D_{u_{1}}, \ldots, \underline{a}_{n} \in D_{u_{n}}, \Gamma, w : p(\vec{\underline{a}}) \Rightarrow u : p(\vec{\underline{a}}), \Delta}\ (tra) \quad \rightsquigarrow$$

$$\frac{}{\mathcal{R}, w \leq u, u \leq v, \underline{a}_1 \in D_{u_1}, \ldots, \underline{a}_n \in D_{u_n}, \Gamma, w : p(\vec{\underline{a}}) \Rightarrow u : p(\vec{\underline{a}}), \Delta} \; (id^{\mathsf{QC}})$$

By the side condition on (id^{QC}), we know that there exist propagation paths $\pi_i(u_i, w)$ for each $i \in \{1, \ldots, n\}$ and a propagation path $\pi(w, u)$ in the premise of the top derivation such that $s_{\pi_i}(u_i, w) \in L_{S5}(a)$ and $s_\pi(w, u) \in L_{S4}(a)$. If the relational atom $w \leq v$ active in the (tra) inference occurs along a propagation path $\pi_i(u_i, w)$ or along the propagation path $\pi(w, u)$, then by replacing each occurrence of w, a, v and $v, \overline{a}, w$ in $\pi_i(u_i, w)$ and $\pi(w, u)$ with w, a, u, a, v and $v, \overline{a}, u, \overline{a}, w$ (respectively), we obtain new propagation paths $\pi_i'(u_i, w)$ and $\pi'(w, u)$ that do not go rely on the relational atom $w \leq v$. We now argue that $s_{\pi_i'}(u_i, w) \in L_{S5}(a)$ and $s_{\pi'}(w, u) \in L_{S4}(a)$. The derivations below show that $a \twoheadrightarrow^*_{S5} aa$ and $\overline{a} \twoheadrightarrow^*_{S5} \overline{aa}$, and we note that $a \longrightarrow aa, \overline{a} \longrightarrow \overline{aa} \in S4$ (by Def. 73).

$$a \longrightarrow \overline{a}a \longrightarrow a\overline{a}a \longrightarrow aa \qquad \overline{a} \longrightarrow a\overline{a} \longrightarrow \overline{a}a\overline{a} \longrightarrow \overline{aa}$$

Since $s_{\pi_i}(u_i, w) \in L_{S5}(a)$, $s_\pi(w, u) \in L_{S4}(a)$, $a \twoheadrightarrow^*_{S5} aa$, $\overline{a} \twoheadrightarrow^*_{S5} \overline{aa}$, and $a \longrightarrow aa, \overline{a} \longrightarrow \overline{aa} \in S4$, by applying $a \twoheadrightarrow^*_{S5} aa$ and $\overline{a} \twoheadrightarrow^*_{S5} \overline{aa}$ to $s_{\pi_i}(u_i, w)$, and $a \longrightarrow aa$ and $\overline{a} \longrightarrow \overline{aa}$ to $s_\pi(w, u)$ for each occurrence of a and $\overline{a}$ obtained from the path w, a, v and $v, \overline{a}, w$ (respectively), we obtain $s_{\pi_i'}(u_i, w)$ and $s_{\pi'}(w, u)$, respectively; the former of which is in $L_{S5}(a)$ and the latter of which is in $L_{S4}(a)$. The confirms that the side condition of (id^{QC}) holds, showing that the second inference above is allowed.

Inductive step. By Lem. 38 we need not consider the $(\supset_l)$, $(\exists_r)$, or $(\forall_l)$ cases. With the exception of $(Pr_\supset)$, $(\exists^{\mathsf{QC}}_{r1})$, and $(\forall^{\mathsf{QC}}_{l1})$ cases, all remaining cases are resolved by invoking IH and then applying the corresponding rule. We consider the $(Pr_\supset)$ and $(\forall^{\mathsf{QC}}_{l1})$ cases; the $(\exists^{\mathsf{QC}}_{r1})$ case is similar.

$(Pr_\supset)$. Due to the application of $(Pr_\supset)$, we know that a propagation path $\pi(w_1, w_2)$ exists in the premises Λ_1 and Λ_2 (shown below) such that $s_\pi(w_1, w_2) \in L_{S4}(a)$. If the active relational atom $w \leq v$ occurs along the propagation path $\pi(w_1, w_2)$, then by replacing each occurrence of w, a, v and $v, \overline{a}, w$ (obtained from the relational atom $w \leq v$) in $\pi(w_1, w_2)$ with w, a, u, a, v and $v, \overline{a}, u, \overline{a}, w$ (respectively), we obtain a new propagation path $\pi'(w_1, w_2)$ that does not rely on $w \leq v$ (but instead, on $w \leq u, u \leq v$). Recall that $a \longrightarrow aa, \overline{a} \longrightarrow \overline{aa} \in S4$, and since $s_\pi(w_1, w_2) \in L_{S4}(a)$, we have that $s_{\pi'}(w_1, w_2) \in L_{S4}(a)$ as we can replace each occurrence of a that corresponds to the path w, a, v and $\overline{a}$ that corresponds to the path $v, \overline{a}, w$ in $s_\pi(w_1, w_2)$ with aa and $\overline{aa}$ by means of the production rules $a \longrightarrow aa$ and $\overline{a} \longrightarrow \overline{aa}$, respectively. This shows that the side condition continues to hold if (tra) is applied first, and so, the rules may be permuted.

$$\Lambda_1 := \mathcal{R}, w \leq u, u \leq v, w \leq v, \Gamma, w_1 : \phi \supset \psi \Rightarrow \Delta, w_2 : \phi$$

$$\Lambda_2 := \mathcal{R}, w \leq u, u \leq v, w \leq v, \Gamma, w_1 : \phi \supset \psi, w_2 : \psi \Rightarrow \Delta$$

$$\cfrac{\cfrac{\Lambda_1 \qquad \Lambda_2}{\mathcal{R}, w \leq u, u \leq v, w \leq v, \Gamma, w_1 : \phi \supset \psi \Rightarrow \Delta} \; (Pr_\supset)}{\mathcal{R}, w \leq u, u \leq v, \Gamma, w_1 : \phi \supset \psi \Rightarrow \Delta} \; (tra) \quad \rightsquigarrow$$

$$\dfrac{\dfrac{\Lambda_1}{\Lambda_1'}\,(tra) \qquad \dfrac{\Lambda_2}{\Lambda_2'}\,(tra)}{\mathcal{R}, w \leq u, u \leq v, \Gamma, w_1 : \phi \supset \psi \Rightarrow \Delta}\,(Pr_{\supset})$$

$$\Lambda_1' := \mathcal{R}, w \leq u, u \leq v, \Gamma, w_1 : \phi \supset \psi \Rightarrow \Delta, w_2 : \phi$$

$$\Lambda_2' := \mathcal{R}, w \leq u, u \leq v, \Gamma, w_1 : \phi \supset \psi, w_2 : \psi \Rightarrow \Delta$$

$(\forall^{\mathsf{QC}}_{l1})$. By the application of $(\forall^{\mathsf{QC}}_{l1})$, we know that propagation paths $\pi_1(w_1, w_3)$ and $\pi_1(w_2, w_3)$ exist in the premise of the top left derivation below such that $s_{\pi_1}(w_1, w_3) \in L_{S5}(a)$ and $s_{\pi_2}(w_2, w_3) \in L_{S4}(a)$. If the relational atom $w \leq v$ occurs along the propagation path $\pi_1(w_1, w_3)$ or $\pi_2(w_2, w_3)$, then by replacing each path w, a, v and $v, \overline{a}, w$ (obtained from the relational atom $w \leq v$) with w, a, u, a, v and $v, \overline{a}, u, \overline{a}, w$ (respectively) in $\pi_1(w_1, w_3)$ and $\pi_2(w_2, w_3)$, we obtain new propagation paths $\pi_1'(w_1, w_3)$ and $\pi_2'(w_2, w_3)$ that do not rely on the relational atom $w \leq v$. Since $s_{\pi_1}(w_1, w_3) \in L_{S5}(a)$, $s_{\pi_2}(w_2, w_3) \in L_{S4}(a)$, $a \twoheadrightarrow^*_{S5} aa$ and $\overline{a} \twoheadrightarrow^*_{S5} \overline{aa}$ (as explained in the base case), and $a \longrightarrow aa, \overline{a} \longrightarrow \overline{aa} \in S4$, we know that $s_{\pi_1'}(w_1, w_3) \in L_{S5}(a)$ and $s_{\pi_2'}(w_2, w_3) \in L_{S4}(a)$, as we can replace each occurrence of a arising from the path w, a, u with aa and each occurrence of $\overline{a}$ arising from the path $u, \overline{a}, w$ with $\overline{aa}$ in $\pi_1(w_1, w_3)$ and $\pi_2(w_2, w_3)$. Therefore, if we apply the (tra) rule first, the side condition of $(\forall^{\mathsf{QC}}_{l1})$ still holds, and so, we may permute the two rules as shown below.

$$\dfrac{\dfrac{\mathcal{R}, w \leq u, u \leq v, w \leq v, \underline{a} \in D_{w_1}, w_3 : \phi(\underline{a}/x), w_2 : \forall x \phi, \Gamma \Rightarrow \Delta}{\mathcal{R}, w \leq u, u \leq v, w \leq v, \underline{a} \in D_{w_1}, w_2 : \forall x \phi, \Gamma \Rightarrow \Delta}\,(\forall^{\mathsf{QC}}_{l1})}{\mathcal{R}, w \leq u, u \leq v, \underline{a} \in D_{w_1}, w_2 : \forall x \phi, \Gamma \Rightarrow \Delta}\,(tra) \quad \rightsquigarrow$$

$$\dfrac{\dfrac{\mathcal{R}, w \leq u, u \leq v, w \leq v, \underline{a} \in D_{w_1}, w_3 : \phi(\underline{a}/x), w_2 : \forall x \phi, \Gamma \Rightarrow \Delta}{\mathcal{R}, w \leq u, u \leq v, \underline{a} \in D_{w_1}, w_3 : \phi(\underline{a}/x), w_2 : \forall x \phi, \Gamma \Rightarrow \Delta}\,(tra)}{\mathcal{R}, w \leq u, u \leq v, \underline{a} \in D_{w_1}, w_2 : \forall x \phi, \Gamma \Rightarrow \Delta}\,(\forall^{\mathsf{QC}}_{l1})$$

□

Lemma 45.

(i) Every proof of a labelled sequent Λ in IntQL *can be algorithmically transformed into a proof of* DomCL(Λ) *in* IntQL*.

(ii) Every proof of a labelled sequent Λ in IntQCL *can be algorithmically transformed into a proof of* DomCL(Λ) *in* IntQCL*.

Proof. We prove the result by induction on the height of the given derivation and consider (ii) as (i) is similar.

Base case. The $(\bot_l)$ case is easy to verify. For the (id_*) inference, due to the application of DomCL, we know that domain atoms $\vec{a} \in D_w$ exist in the output labelled sequent, ensuring that it is in fact an instance of (id^{QC}).

$$\frac{}{\mathcal{R},\Gamma, w : p(\underline{\vec{a}}) \Rightarrow u : p(\underline{\vec{a}}), \Delta}\,(id_*) \quad \rightsquigarrow \quad \frac{}{\texttt{DomCL}(\mathcal{R},\Gamma, w : p(\underline{\vec{a}}) \Rightarrow u : p(\underline{\vec{a}}), \Delta)}\,(id^{\mathsf{QC}})$$

Inductive step. We show the $(\wedge_r)$, $(\supset_r)$, $(\exists_r^{\mathsf{QC}})$, $(\exists_l^{\mathsf{QC}})$, $(\forall_l^{\mathsf{QC}})$, and $(\forall_r^{\mathsf{QC}})$ cases as all other cases are simple or similar. We consider each of the aforementioned cases below.

$(\wedge_r)$. Let our given $(\wedge_r)$ inference be as shown below top-left. Also, let $\underline{\vec{a}} := \underline{a}_1, \ldots, \underline{a}_n$ be all free parameters in $\phi, \Gamma \restriction w, \Delta \restriction w$ not occurring in $\psi, \Gamma \restriction w, \Delta \restriction w$, and let $\underline{\vec{b}} := \underline{b}_1, \ldots, \underline{b}_k$ be all free parameters in $\psi, \Gamma \restriction w, \Delta \restriction w$ not occurring in $\phi, \Gamma \restriction w, \Delta \restriction w$. To obtain the desired conclusion, we first apply the admissibility of (wk) (Thm. 27) in order to ensure that the contexts of the premises match, followed by an application of the $(\wedge_r)$ rule.

$$\frac{\mathcal{R},\Gamma \Rightarrow w : \phi, \Delta \qquad \mathcal{R},\Gamma \Rightarrow w : \psi, \Delta}{\mathcal{R},\Gamma \Rightarrow w : \phi \wedge \psi, \Delta}\,(\wedge_r) \quad \rightsquigarrow$$

$$\cfrac{\cfrac{\cfrac{\cfrac{}{\texttt{DomCL}(\mathcal{R},\Gamma \Rightarrow w : \phi, \Delta)}\,\text{IH}}{\mathcal{R}', \underline{\vec{a}} \in D_w, \Gamma \Rightarrow w : \phi, \Delta}\,=}{\mathcal{R}', \underline{\vec{a}} \in D_w, \underline{\vec{b}} \in D_w, \Gamma \Rightarrow w : \phi, \Delta}\,(wk) \qquad \cfrac{\cfrac{\cfrac{}{\texttt{DomCL}(\mathcal{R},\Gamma \Rightarrow w : \psi, \Delta)}\,\text{IH}}{\mathcal{R}', \underline{\vec{b}} \in D_w, \Gamma \Rightarrow w : \psi, \Delta}\,=}{\mathcal{R}', \underline{\vec{a}} \in D_w, \underline{\vec{b}} \in D_w, \Gamma \Rightarrow w : \psi, \Delta}\,(wk)}{\cfrac{\mathcal{R}', \underline{\vec{a}} \in D_w, \underline{\vec{b}} \in D_w, \Gamma \Rightarrow w : \phi \wedge \psi, \Delta}{\texttt{DomCL}(\mathcal{R},\Gamma \Rightarrow w : \phi \wedge \psi, \Delta)}\,=}\,(\wedge_r)$$

$(\supset_r)$. There are two cases to consider: either (i) the auxiliary formulae $u : \phi$ and $u : \psi$ contain free occurrences of parameters $\underline{\vec{a}} = \underline{a}_1, \ldots, \underline{a}_n$, or (ii) they do not. Case (ii) is easily resolved by applying IH and then the $(\supset_r)$ rule, so we focus on case (i) which is shown below. Let $\mathcal{R}', \underline{\vec{a}} \in D_u$ be the multiset of relational atoms occurring within the premise of the bottom right inference below (that is, the multiset of relational atoms occurring in the labelled sequent obtained by applying `DomCL` to the premise of the top left inference). To derive the desired conclusion, we first apply admissibility of (wk) (Lem. 27) to add in the domain atoms $\underline{\vec{a}}' \in D_w := \underline{a}_{i_1} \in D_w, \ldots, \underline{a}_{i_k} \in D_w$ such that $\underline{a}_{i_j} \in D_w \not\in \mathcal{R}'$ for $i_j \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, k\}$, i.e. for each parameter $\underline{a}_{i_j}$ in $\underline{\vec{a}}$ we add a new domain atom $\underline{a}_{i_j} \in D_w$ given that it does not already occur in $\mathcal{R}'$. After (wk) is applied, we apply the admissible (nd) rule (Thm. 25) n times to delete all domain atoms $\underline{\vec{a}} \in D_u$, thus making u an eigenvarible, and allowing for $(\supset_r)$ to be applied, giving the desired result.

$$\frac{\mathcal{R}, w \leq u, \Gamma, u : \phi \Rightarrow u : \psi, \Delta}{\mathcal{R},\Gamma \Rightarrow w : \phi \supset \psi, \Delta}\,(\supset_r) \quad \rightsquigarrow$$

$$
\begin{array}{cl}
\hdashline
\texttt{DomCL}(\mathcal{R}, w \leq u, \Gamma, u : \phi \Rightarrow u : \psi, \Delta) & \text{IH} \\
\hdashline[1pt/2pt]
\mathcal{R}', \underline{\vec{a}} \in D_u, w \leq u, \Gamma, u : \phi \Rightarrow u : \psi, \Delta & = \\
\hdashline
\mathcal{R}', \underline{\vec{a}}' \in D_w, \underline{\vec{a}} \in D_u, w \leq u, \Gamma, u : \phi \Rightarrow u : \psi, \Delta & (wk) \\
\hdashline
\mathcal{R}', \underline{\vec{a}}' \in D_w, w \leq u, \Gamma, u : \phi \Rightarrow u : \psi, \Delta & (nd) \times n \\
\hline
\mathcal{R}', \underline{\vec{a}}' \in D_w, \Gamma \Rightarrow w : \phi \supset \psi, \Delta & (\supset_r) \\
\hdashline[1pt/2pt]
\texttt{DomCL}(\mathcal{R}, \Gamma \Rightarrow w : \phi \supset \psi, \Delta) & =
\end{array}
$$

$(\exists_r^{\mathsf{QC}})$-(i). In case (i), we assume that $\underline{a}$ is not an eigenvariable. By the side condition imposed on $(\exists_r^{\mathsf{QC}})$ then, we know there exists a labelled formula $u : \psi(\underline{a}) \in \Gamma, \Delta$ and propagation path $\pi(u, w)$ such that $s_\pi(u, w) \in L_{S5}(a)$. Therefore, after applying IH which applies DomCL, there will be a domain atom of the form $\underline{a} \in D_u$ occurring in the output labelled sequent. We may then apply the $(\exists_{r1}^{\mathsf{QC}})$ rule since the side condition is satisfied, giving the desired conclusion.

$$
\dfrac{\mathcal{R}, \Gamma \Rightarrow \Delta, w : \phi(\underline{a}/x), w : \exists x \phi}{\mathcal{R}, \Gamma \Rightarrow \Delta, w : \exists x \phi}\ (\exists_r^{\mathsf{QC}}) \quad \rightsquigarrow
$$

$$
\begin{array}{cl}
\hdashline
\texttt{DomCL}(\mathcal{R}, \Gamma \Rightarrow \Delta, w : \phi(\underline{a}/x), w : \exists x \phi) & \text{IH} \\
\hdashline
\texttt{DomCL}(\mathcal{R}, \Gamma \Rightarrow \Delta, w : \exists x \phi) & (\exists_{r1}^{\mathsf{QC}})
\end{array}
$$

$(\exists_r^{\mathsf{QC}})$-(ii). In case (ii), we assume that $\underline{a}$ is an eigenvariable. Then, after applying IH there will be a *single* domain atom of the form $\underline{a} \in D_w$ (and no other domain atoms will contain the parameter $\underline{a}$). This follows from the definition of DomCL and the fact that $w : \phi(\underline{a}/x)$ is the only labelled formula containing $\underline{a}$ (by the eigenvariable condition). We then apply the $(\exists_{r2}^{\mathsf{QC}})$ rule to obtain the desired conclusion.

$$
\dfrac{\mathcal{R}, \Gamma \Rightarrow \Delta, w : \phi(\underline{a}/x), w : \exists x \phi}{\mathcal{R}, \Gamma \Rightarrow \Delta, w : \exists x \phi}\ (\exists_r^{\mathsf{QC}}) \quad \rightsquigarrow
$$

$$
\begin{array}{cl}
\hdashline
\texttt{DomCL}(\mathcal{R}, \Gamma \Rightarrow \Delta, w : \phi(\underline{a}/x), w : \exists x \phi) & \text{IH} \\
\hdashline
\texttt{DomCL}(\mathcal{R}, \Gamma \Rightarrow \Delta, w : \exists x \phi) & (\exists_{r2}^{\mathsf{QC}})
\end{array}
$$

$(\exists_l^{\mathsf{QC}})$. After invoking IH we know that a *single* domain atom of the form $\underline{a} \in D_w$ occurs within the output labelled sequent (and no other domain atoms contain the parameter $\underline{a}$). This fact follows from the fact that $\underline{a}$ is an eigenvariable, and so, by the definition of DomCL only a single domain atom uniquely containing the parameter $\underline{a}$ will be introduced on the basis of the labelled formula $w : \phi(\underline{a}/x)$. Hence, we may directly apply $(\exists_l)$ after applying the inductive hypothesis.

$$
\dfrac{\mathcal{R}, \Gamma, w : \phi(\underline{a}/x) \Rightarrow \Delta}{\mathcal{R}, \Gamma, w : \exists x \phi \Rightarrow \Delta}\ (\exists_l^{\mathsf{QC}}) \quad \rightsquigarrow \quad
\begin{array}{cl}
\hdashline
\texttt{DomCL}(\mathcal{R}, \Gamma, w : \phi(\underline{a}/x) \Rightarrow \Delta) & \text{IH} \\
\hline
\texttt{DomCL}(\mathcal{R}, \Gamma, w : \exists x \phi \Rightarrow \Delta) & (\exists_l)
\end{array}
$$

$(\forall_l^{\mathsf{QC}})$-(i). In case (i), we assume that $\underline{a}$ is not an eigenvariable. By the side condition imposed on $(\forall_l^{\mathsf{QC}})$, we know that $\underline{a}$ is $S5$-available for v and that there exists a propagation path $\pi(w, v)$ such that $s_\pi(w, v) \in L_{S4}(a)$. The former implies that there exists a labelled formula $u : \psi(\underline{a}) \in \Gamma, \Delta$ and propagation path $\pi'(u, v)$ such that $s_{\pi'}(u, v) \in L_{S5}(a)$. After invoking IH, a domain atom of the form $\underline{a} \in D_u$ will occur in the output labelled sequent (since applying IH applies DomCL), showing that the side condition of $(\forall_{l1}^{\mathsf{QC}})$ is satisfied, and hence, the rule $(\forall_{l1}^{\mathsf{QC}})$ may be applied.

$$\dfrac{\mathcal{R}, w : \forall x\phi, v : \phi(\underline{a}/x), \Gamma \Rightarrow \Delta}{\mathcal{R}, w : \forall x\phi, \Gamma \Rightarrow \Delta}\ (\forall_l^{\mathsf{QC}}) \quad \rightsquigarrow$$

$$\begin{array}{c} \hdashline \texttt{DomCL}(\mathcal{R}, w : \forall x\phi, v : \phi(\underline{a}/x), \Gamma \Rightarrow \Delta) \\ \hline \texttt{DomCL}(\mathcal{R}, w : \forall x\phi, \Gamma \Rightarrow \Delta) \end{array} \begin{array}{l} \text{IH} \\ (\forall_{l1}^{\mathsf{QC}}) \end{array}$$

$(\forall_l^{\mathsf{QC}})$-(ii). In case (ii), we assume that $\underline{a}$ is an eigenvariable and let $\Lambda := \texttt{DomCL}(\mathcal{R}, w : \forall x\phi, v : \phi(\underline{a}/x), \Gamma \Rightarrow \Delta)$. By the side condition imposed on $(\forall_l^{\mathsf{QC}})$, we know that there exists a propagation path $\pi(w, v)$ in the premise of the top left derivation such that $s_\pi(w, v) \in L_{S4}(a)$. Since $\underline{a}$ is an eigenvariable, we know that after invoking IH (which applies DomCL), there will be a domain atom of the form $\underline{a} \in D_v$ occurring in Λ. Since $\varepsilon \in L_{S5}(a)$ and the empty path $\lambda(v, v) = v$ occurs in Λ, we know that $s_\lambda(v, v) = \varepsilon \in L_{S5}(a)$. Therefore, we have that $\underline{a}$ is an eigenvarible, $s_\lambda(v, v) \in L_{S5}(a)$, and $s_\pi(w, v) \in L_{S4}(a)$ holds for Λ, meaning that $(\forall_{l2}^{\mathsf{QC}})$ may be applied, giving the desired result.

$$\dfrac{\mathcal{R}, w : \forall x\phi, v : \phi(\underline{a}/x), \Gamma \Rightarrow \Delta}{\mathcal{R}, w : \forall x\phi, \Gamma \Rightarrow \Delta}\ (\forall_l^{\mathsf{QC}}) \quad \rightsquigarrow$$

$$\begin{array}{c} \hdashline \texttt{DomCL}(\mathcal{R}, w : \forall x\phi, v : \phi(\underline{a}/x), \Gamma \Rightarrow \Delta) \\ \hline \texttt{DomCL}(\mathcal{R}, w : \forall x\phi, \Gamma \Rightarrow \Delta) \end{array} \begin{array}{l} \text{IH} \\ (\forall_{l2}^{\mathsf{QC}}) \end{array}$$

$(\forall_r^{\mathsf{QC}})$. Let $\Lambda := \texttt{DomCL}(\mathcal{R}, w \leq u, \Gamma \Rightarrow u : \phi(\underline{a}/x), \Delta)$. We have two cases to consider: either (i) $u : \phi(\underline{a}/x)$ contains no other parameters other than possibly $\underline{a}$, or (ii) $u : \phi(\underline{a}/x)$ is of the form $u : \phi(\underline{\vec{a}})(\underline{a}/x)$ and contains the parameters $\underline{\vec{a}} := \underline{a}_1, \ldots, \underline{a}_n$. Case (i) is simple to resolve, so we show how to resolve case (ii). By the assumption of case (ii), Λ is of the form $\mathcal{R}', w \leq u, \underline{\vec{a}} \in D_u, \underline{a} \in D_u, \Gamma \Rightarrow u : \phi(\underline{\vec{a}})(\underline{a}/x), \Delta$. To resolve the case, as shown below, we first apply admissibility of (wk) (Lem. 27) to add in the domain atoms $\underline{\vec{a}}' \in D_w := \underline{a}_{i_1} \in D_w, \ldots, \underline{a}_{i_k} \in D_w$ such that $\underline{a}_{i_j} \in D_w \not\in \mathcal{R}'$ for $i_j \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, k\}$, i.e. for each parameter $\underline{a}_{i_j}$ in $\underline{\vec{a}}$ we add a new domain atom $\underline{a}_{i_j} \in D_w$ given that it does not already occur in $\mathcal{R}'$. We then apply the admissible (nd) rule (Thm. 25) n times to delete all domain atoms $\underline{\vec{a}} \in D_u$, yielding a labelled sequent where u and $\underline{a}$ are eigenvariables. One application of $(\forall_r^{\mathsf{X}})$ gives the desired conclusion.

$$
\frac{\mathcal{R}, w \leq u, \Gamma \Rightarrow u : \phi(\underline{\vec{a}})(\underline{a}/x), \Delta}{\mathcal{R}, \Gamma \Rightarrow w : \forall x \phi(\underline{\vec{a}}), \Delta}\ (\forall_r^{\mathsf{QC}}) \quad \rightsquigarrow
$$

$$
\begin{array}{cl}
\hdashline
\mathtt{DomCL}(\mathcal{R}, w \leq u, \Gamma \Rightarrow u : \phi(\underline{\vec{a}})(\underline{a}/x), \Delta) & \mathrm{IH} \\
\hdashline[1pt/3pt]
\mathcal{R}', w \leq u, \underline{\vec{a}} \in D_u, \underline{a} \in D_u, \Gamma \Rightarrow u : \phi(\underline{\vec{a}})(\underline{a}/x), \Delta & = \\
\hdashline
\mathcal{R}', w \leq u, \underline{\vec{a}}' \in D_w, \underline{\vec{a}} \in D_u, \underline{a} \in D_u, \Gamma \Rightarrow u : \phi(\underline{\vec{a}})(\underline{a}/x), \Delta & (wk) \\
\hdashline
\mathcal{R}', w \leq u, \underline{\vec{a}}' \in D_w, \underline{a} \in D_u, \Gamma \Rightarrow u : \phi(\underline{\vec{a}})(\underline{a}/x), \Delta & (nd) \times n \\
\hline
\mathcal{R}', \underline{\vec{a}}' \in D_w, \Gamma \Rightarrow w : \forall x \phi(\underline{\vec{a}}), \Delta & (\forall_r) \\
\hdashline[1pt/3pt]
\mathtt{DomCL}(\mathcal{R}, \Gamma \Rightarrow w : \forall x \phi(\underline{\vec{a}}), \Delta) & =
\end{array}
$$

□

Lemma 47. *The rule* (lsb) *is hp-admissible in* IntQL *and* IntQCL*.*

Proof. We prove the result by induction on the height of the given derivation.

Base case. The base case follows from the fact that any application of (lsb) to (id_*) or $(\bot_l)$ yields another instance of the rule.

Inductive step. With the exception of the $(\supset_r)$ and $(\forall_r^{\mathsf{X}})$ rules, all cases follow by invoking IH and then applying the corresponding rule. In the $(\supset_r)$ and $(\forall_r^{\mathsf{X}})$ cases it may be necessary to invoke IH twice (before applying each respective rule) to ensure that the eigenvariable condition is met, similar to what was done in the proof of Lem. 14. □

Lemma 48. *The rule* (psb) *is hp-admissible in* IntQL *and* IntQCL*.*

Proof. We show the result for IntQL as the proof for IntQCL is similar, and prove the claim by induction on the height of the given derivation.

Base case. Any application of (psb) to (id_*) or $(\bot_l)$ yields another instance of the rule, resolving the base case.

Inductive step. With the exception of the $(\exists_r^{\mathsf{Q}})$, $(\forall_l^{\mathsf{Q}})$, $(\exists_l^{\mathsf{Q}})$, and $(\forall_r^{\mathsf{Q}})$ rules, all cases follow by invoking IH and then applying the corresponding rule. The non-trivial $(\exists_r^{\mathsf{Q}})$ and $(\forall_l^{\mathsf{Q}})$ cases occur when (psb) substitutes a parameter $\underline{b}$ for the parameter $\underline{a}$ (occurring in the auxiliary formula) or introduces the parameter $\underline{a}$ of the auxiliary labelled formula when the parameter is an eigenvariable. The non-trivial $(\exists_l^{\mathsf{Q}})$ and $(\forall_r^{\mathsf{Q}})$ cases occur when (psb) introduces the eigenvariable $\underline{a}$. Let us consider each non-trivial case in turn:

$(\exists_r^{\mathsf{Q}})$. Either (i) $\underline{a}$ is $S4$-available for w or (ii) $\underline{a}$ is an eigenvariable. In case (i), the non-trivial (psb) instance applies a substitution of the form $(\underline{b}/\underline{a})$. By the assumption of the case, we know there exists a labelled formula $u : \psi(\underline{a}) \in \Gamma, \Delta$ and a propagation path $\pi(u, w)$ such that $s_\pi(u, w) \in L_{S4}(a)$. Observe that since (psb) applies $(\underline{b}/\underline{a})$ to all labelled formulae of the sequent, the auxiliary formula will become $w : \phi(\underline{b}/x)$, and we will know that a labelled formula $u : \psi(\underline{b}) \in \Gamma, \Delta$ exists with a propagation path $\pi(u, w)$

such that $s_{\pi}(u,w) \in L_{S4}(a)$. Hence, the side condition of the inference will continue to hold if IH is applied first, allowing for the two rules to be permuted. In case (ii), where $\underline{a}$ is an eigenvariable, the non-trivial case occurs when (psb) is of the form $(\underline{a}/\underline{b})$. In such a case we invoke IH twice to ensure that the eigenvariable condition is met (similar to the non-trivial case of $(\exists^{\mathsf{Q}}_{l})$ below), and then apply $(\exists^{\mathsf{Q}}_{r})$.

$(\forall^{\mathsf{Q}}_{l})$. By the side condition of the rule, either (i) $\underline{a}$ is $S4$-available for v or (ii) $\underline{a}$ is an eigenvariable, and there exists a propagation path $\pi(w,v)$ such that $s_{\pi}(w,v) \in L_{S4}(a)$. Let us consider case (i) first, where the non-trivial (psb) instance applies a substitution of the form $(\underline{b}/\underline{a})$. Since $\underline{a}$ is $S4$-available for v, we know that there exists a labelled formula $u : \psi(\underline{a}) \in \Gamma, \Delta$ and a propagation path $\pi(u,v)$ such that $s_{\pi}(u,v) \in L_{S4}(a)$. If we invoke IH and apply (psb) first, then the auxiliary formula will be of the form $v : \phi(\underline{b}/x)$, and by the side condition there will be a labelled formula $u : \psi(\underline{b}) \in \Gamma, \Delta$ and a propagation path $\pi(u,v)$ such that $s_{\pi}(u,v) \in L_{S4}(a)$. Hence, the side condition continues to hold after the invocation of IH, allowing for $(\forall^{\mathsf{Q}}_{l})$ to be applied an the two rules to be permuted. In case (ii), the non-trivial case occurs when (psb) introduces the eigenvariable $\underline{a}$. In such a case we invoke IH twice to ensure that the eigenvariable condition is met (similar to the non-trivial case of $(\forall^{\mathsf{Q}}_{r})$ below), and then apply $(\forall^{\mathsf{Q}}_{l})$.

$(\exists^{\mathsf{Q}}_{l})$. The non-trivial case arises when IH (i.e. (psb)) introduces the eigenvariable $\underline{a}$, and is resolved as shown below:

$$\dfrac{\dfrac{\mathcal{R},\Gamma,w:\phi(\underline{a}/x)\Rightarrow\Delta}{\mathcal{R},\Gamma,w:\exists x\phi\Rightarrow\Delta}\ (\exists^{\mathsf{X}}_{l})}{\mathcal{R}(\underline{a}/\underline{b}),\Gamma(\underline{a}/\underline{b}),(w:\exists x\phi)(\underline{a}/\underline{b})\Rightarrow\Delta(\underline{a}/\underline{b})}\ (psb) \quad \rightsquigarrow$$

$$\begin{array}{cl}
\mathcal{R},\Gamma,w:\phi(\underline{a}/x)\Rightarrow\Delta & \\ \hdashline
\mathcal{R}(\underline{c}/\underline{a}),\Gamma(\underline{c}/\underline{a}),(w:\phi(\underline{a}/x))(\underline{c}/\underline{a})\Rightarrow\Delta(\underline{c}/\underline{a}) & \text{IH} \\
\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots & = \\
\mathcal{R},\Gamma,w:\phi(\underline{c}/x)\Rightarrow\Delta & \\ \hdashline
\mathcal{R}(\underline{a}/\underline{b}),\Gamma(\underline{a}/\underline{b}),(w:\phi(\underline{c}/x))(\underline{a}/\underline{b})\Rightarrow\Delta(\underline{a}/\underline{b}) & \text{IH} \\
\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots & = \\
\mathcal{R}(\underline{a}/\underline{b}),\Gamma(\underline{a}/\underline{b}),w:\phi(\underline{c}/x)(\underline{a}/\underline{b})\Rightarrow\Delta(\underline{a}/\underline{b}) & \\ \hline
\mathcal{R}(\underline{a}/\underline{b}),\Gamma(\underline{a}/\underline{b}),(w:\exists x\phi)(\underline{a}/\underline{b})\Rightarrow\Delta(\underline{a}/\underline{b}) & (\exists^{\mathsf{Q}}_{l})
\end{array}$$

$(\forall^{\mathsf{Q}}_{r})$. The non-trivial case arises when IH introduces the eigenvariable $\underline{a}$, and is resolved as shown below:

$$\dfrac{\dfrac{\mathcal{R},w\leq u,\Gamma\Rightarrow u:\phi(\underline{a}/x),\Delta}{\mathcal{R},\Gamma\Rightarrow w:\forall x\phi,\Delta}\ (\forall^{\mathsf{Q}}_{r})}{\mathcal{R}(\underline{a}/\underline{b}),\Gamma(\underline{a}/\underline{b})\Rightarrow(w:\forall x\phi)(\underline{a}/\underline{b}),\Delta(\underline{a}/\underline{b})}\ (psb) \quad \rightsquigarrow$$

$$\begin{array}{cl}
\mathcal{R}, w \leq u, \Gamma \Rightarrow u : \phi(\underline{a}/x), \Delta & \\ \hdashline
\mathcal{R}(\underline{c}/\underline{a}), (w \leq u)(\underline{c}/\underline{a}), \Gamma(\underline{c}/\underline{a}) \Rightarrow (u : \phi(\underline{a}/x))(\underline{c}/\underline{a}), \Delta(\underline{c}/\underline{a}) & \text{IH} \\ \hdashline
\mathcal{R}, w \leq u, \Gamma \Rightarrow u : \phi(\underline{c}/x), \Delta & = \\ \hdashline
\mathcal{R}(\underline{a}/\underline{b}), (w \leq u)(\underline{a}/\underline{b}), \Gamma(\underline{a}/\underline{b}) \Rightarrow (u : \phi(\underline{c}/x))(\underline{a}/\underline{b}), \Delta(\underline{a}/\underline{b}) & \text{IH} \\ \hdashline
\mathcal{R}(\underline{a}/\underline{b}), w \leq u, \Gamma(\underline{a}/\underline{b}) \Rightarrow u : \phi(\underline{c}/x)(\underline{a}/\underline{b}), \Delta(\underline{a}/\underline{b}) & = \\ \hline
\mathcal{R}(\underline{a}/\underline{b}), \Gamma(\underline{a}/\underline{b}) \Rightarrow (w : \forall x \phi)(\underline{a}/\underline{b}), \Delta(\underline{a}/\underline{b}) & (\forall^{\mathsf{Q}}_{r})
\end{array}$$

□

Lemma 49. *The rule* (wk) *is hp-admissible in* IntQL *and* IntQCL.

Proof. We show the result for IntQL as the proof for IntQCL is similar. We prove the result by induction on the height of the given derivation.

Base case. The base case is straightforward since any application of (wk) to (id_{*}) or $(\bot_{l})$ yields another instance of the rule.

Inductive step. The only non-trivial cases concern $(\supset_{r})$, $(\exists^{\mathsf{Q}}_{r})$, $(\forall^{\mathsf{Q}}_{l})$, $(\exists^{\mathsf{Q}}_{l})$, and $(\forall^{\mathsf{Q}}_{r})$, and arise when (wk) introduces a parameter or label identical to an eigenvariable of one of the aforementioned inferences. The cases involving the other rules are resolved by applying IH followed by the corresponding rule. We show how to resolve the non-trivial $(\supset_{r})$, $(\exists^{\mathsf{Q}}_{l})$, $(\forall^{\mathsf{Q}}_{r})$ cases below; all other cases are similar or simple. We let $\underline{c}$ and z be a fresh parameter and label, respectively.

$$\begin{array}{cl}
\mathcal{R}, w \leq u, \Gamma, u : \phi \Rightarrow u : \psi, \Delta & \\ \hline
\mathcal{R}, \Gamma \Rightarrow w : \phi \supset \psi, \Delta & (\supset_{r}) \\ \hline
\mathcal{R}, \mathcal{R}', \Gamma, \Gamma' \Rightarrow w : \phi \supset \psi, \Delta, \Delta' & (wk)
\end{array} \quad \rightsquigarrow$$

$$\begin{array}{cl}
\mathcal{R}, w \leq u, \Gamma, u : \phi \Rightarrow u : \psi, \Delta & \\ \hdashline
\mathcal{R}(z/u), (w \leq u)(z/u), \Gamma(z/u), (u : \phi)(z/u) \Rightarrow (u : \psi)(z/u), \Delta(z/u) & (lsb) \\ \hdashline
\mathcal{R}, w \leq z, \Gamma, z : \phi \Rightarrow z : \psi, \Delta & = \\ \hdashline
\mathcal{R}, \mathcal{R}', w \leq z, \Gamma, \Gamma', z : \phi \Rightarrow z : \psi, \Delta, \Delta' & \text{IH} \\ \hline
\mathcal{R}, \mathcal{R}', \Gamma, \Gamma' \Rightarrow w : \phi \supset \psi, \Delta, \Delta' & (\supset_{r})
\end{array}$$

$$\begin{array}{cl}
\mathcal{R}, \Gamma, w : \phi(\underline{a}/x) \Rightarrow \Delta & \\ \hline
\mathcal{R}, \Gamma, w : \exists x \phi \Rightarrow \Delta & (\exists^{\mathsf{Q}}_{l}) \\ \hline
\mathcal{R}, \mathcal{R}', \Gamma, \Gamma', w : \exists x \phi \Rightarrow \Delta, \Delta' & (wk)
\end{array} \quad \rightsquigarrow$$

$$\begin{array}{cl}
\mathcal{R}, \Gamma, w : \phi(\underline{a}/x) \Rightarrow \Delta & \\ \hdashline
\mathcal{R}(\underline{c}/\underline{a}), \Gamma(\underline{c}/\underline{a}), (w : \phi(\underline{a}/x))(\underline{c}/\underline{a}) \Rightarrow \Delta(\underline{c}/\underline{a}) & (psb) \\ \hdashline
\mathcal{R}, \Gamma, w : \phi(\underline{c}/x) \Rightarrow \Delta & = \\ \hdashline
\mathcal{R}, \mathcal{R}', \Gamma, \Gamma', w : \phi(\underline{c}/x) \Rightarrow \Delta, \Delta' & \text{IH} \\ \hline
\mathcal{R}, \mathcal{R}', \Gamma, \Gamma', w : \exists x \phi \Rightarrow \Delta, \Delta' & (\exists^{\mathsf{Q}}_{l})
\end{array}$$

$$\cfrac{\cfrac{\mathcal{R}, w \leq u, \Gamma \Rightarrow u : \phi(\underline{a}/x), \Delta}{\mathcal{R}, \Gamma \Rightarrow w : \forall x\phi, \Delta}\,(\forall^{\mathsf{Q}}_{r})}{\mathcal{R}, \mathcal{R}', \Gamma, \Gamma' \Rightarrow w : \forall x\phi, \Delta, \Delta'}\,(wk) \quad \rightsquigarrow$$

$$\begin{array}{cl}
\mathcal{R}, w \leq u, \Gamma \Rightarrow u : \phi(\underline{a}/x), \Delta & \\ \hdashline
\mathcal{R}(z/u), (w \leq u)(z/u), \Gamma(z/u) \Rightarrow (u : \phi(\underline{a}/x))(z/u), \Delta(z/u) & (lsb) \\ \hdashline[1pt/2pt]
\mathcal{R}, w \leq z, \Gamma \Rightarrow z : \phi(\underline{a}/x), \Delta & = \\ \hdashline
\mathcal{R}(\underline{c}/\underline{a}), (w \leq z)(\underline{c}/\underline{a}), \Gamma(\underline{c}/\underline{a}) \Rightarrow (z : \phi(\underline{a}/x))(\underline{c}/\underline{a}), \Delta(\underline{c}/\underline{a}) & (psb) \\ \hdashline[1pt/2pt]
\mathcal{R}, w \leq z, \Gamma \Rightarrow z : \phi(\underline{c}/x), \Delta & = \\ \hdashline
\mathcal{R}, \mathcal{R}', w \leq z, \Gamma, \Gamma' \Rightarrow z : \phi(\underline{c}/x), \Delta, \Delta' & \mathrm{IH} \\ \hline
\mathcal{R}, \mathcal{R}', \Gamma, \Gamma' \Rightarrow w : \forall x\phi, \Delta, \Delta' & (\forall^{\mathsf{Q}}_{r})
\end{array}$$

□

Lemma 50. *The* $(\wedge_l)$*,* $(\wedge_r)$*,* $(\vee_l)$*, and* $(\vee_r)$ *rules are hp-invertible in* IntQL *and* IntQCL*.*

Proof. We prove the result by induction on the height of the given derivation, and consider only the $(\wedge_l)$ case since the $(\wedge_r)$, $(\vee_l)$, and $(\vee_r)$ cases are similar.

Base case. The base cases are resolved as shown below:

$$\cfrac{}{\mathcal{R}, \Gamma, v : \phi \wedge \psi, w : p(\underline{\vec{a}}) \Rightarrow u : p(\underline{\vec{a}}), \Delta}\,(id_*) \quad \rightsquigarrow$$

$$\cfrac{}{\mathcal{R}, \Gamma, v : \phi, v : \psi, w : p(\underline{\vec{a}}) \Rightarrow u : p(\underline{\vec{a}}), \Delta}\,(id)$$

$$\cfrac{}{\mathcal{R}, \Gamma, w : \bot, u : \phi \wedge \psi \Rightarrow \Delta}\,(\bot_l) \quad \rightsquigarrow \quad \cfrac{}{\mathcal{R}, \Gamma, w : \bot, u : \phi, u : \psi \Rightarrow \Delta}\,(\bot_l)$$

Inductive step. With the exception of the $(\wedge_l)$ case, all cases are resolved by invoking IH followed by the corresponding rule. If the last rule applied in the derivation is $(\wedge_l)$, then there are two cases to consider: either (i) the conjunction we aim to invert is principal, or (ii) it is not. Case (i) is shown below top—the derivation of the premise gives the desired conclusion—and case (ii) is shown below bottom.

$$\cfrac{\mathcal{R}, \Gamma, w : \phi, w : \psi \Rightarrow \Delta}{\mathcal{R}, \Gamma, w : \phi \wedge \psi \Rightarrow \Delta}\,(\wedge_l) \quad \rightsquigarrow \quad \mathcal{R}, \Gamma, w : \phi, w : \psi \Rightarrow \Delta$$

$$\cfrac{\mathcal{R}, \Gamma, u : \chi, u : \theta, w : \phi \wedge \psi \Rightarrow \Delta}{\mathcal{R}, \Gamma, u : \chi \wedge \theta, w : \phi \wedge \psi \Rightarrow \Delta}\,(\wedge_l) \quad \rightsquigarrow \quad \begin{array}{cl}
\mathcal{R}, \Gamma, u : \chi, u : \theta, w : \phi \wedge \psi \Rightarrow \Delta & \\ \hdashline
\mathcal{R}, \Gamma, u : \chi, u : \theta, w : \phi, w : \psi \Rightarrow \Delta & \mathrm{IH} \\ \hline
\mathcal{R}, \Gamma, u : \chi \wedge \theta, w : \phi, w : \psi \Rightarrow \Delta & (\wedge_l)
\end{array}$$

□

Lemma 51. *The* $(Pr_{\supset})$ *and* $(\supset_r)$ *rules are hp-invertible in* IntQL *and* IntQCL*.*

Proof. The hp-invertibility of $(Pr_{\supset})$ follows from Lem. 49. We argue that the rule $(\supset_r)$ is hp-invertible by induction on the height of the given derivation.

Base case. The base cases are resolved as shown below:

$$\frac{}{\mathcal{R}, \Gamma, w : p(\underline{\vec{a}}) \Rightarrow v : \phi \supset \psi, u : p(\underline{\vec{a}}), \Delta}\ (id_*) \quad \rightsquigarrow$$

$$\frac{}{\mathcal{R}, v \leq z, \Gamma, w : p(\underline{\vec{a}}), z : \phi \Rightarrow z : \psi, u : p(\underline{\vec{a}}), \Delta}\ (id_*)$$

$$\frac{}{\mathcal{R}, \Gamma, w : \bot \Rightarrow v : \phi \supset \psi, \Delta}\ (\bot_l) \quad \rightsquigarrow \quad \frac{}{\mathcal{R}, v \leq z, \Gamma, w : \bot, z : \phi \Rightarrow z : \psi, \Delta}\ (\bot_l)$$

Inductive step. With the exception of the $(\supset_r)$ rule, all cases are resolved by invoking IH followed by the corresponding rule. If the last inference of the derivation is an instance of $(\supset_r)$, then there are two cases to consider: either (i) the implication we aim to invert is principal, or (ii) it is not. Case (i) is resolved below top—the derivation of the premise gives the desired conclusion—and case (ii) is resolved as shown below bottom.

$$\frac{\mathcal{R}, w \leq u, \Gamma, u : \phi \Rightarrow u : \psi, \Delta}{\mathcal{R}, \Gamma \Rightarrow w : \phi \supset \psi, \Delta}\ (\supset_r) \quad \rightsquigarrow \quad \mathcal{R}, w \leq u, \Gamma, u : \phi \Rightarrow u : \psi, \Delta$$

$$\frac{\mathcal{R}, v \leq z, \Gamma, z : \chi \Rightarrow z : \theta, w : \phi \supset \psi, \Delta}{\mathcal{R}, \Gamma \Rightarrow v : \chi \supset \theta, w : \phi \supset \psi, \Delta}\ (\supset_r) \quad \rightsquigarrow$$

$$\frac{\dfrac{\mathcal{R}, v \leq z, \Gamma, z : \chi \Rightarrow z : \theta, w : \phi \supset \psi, \Delta}{\mathcal{R}, w \leq u, v \leq z, \Gamma, z : \chi, u : \phi \Rightarrow u : \psi, z : \theta, \Delta}\ \text{IH}}{\mathcal{R}, w \leq u, \Gamma, u : \phi \Rightarrow u : \psi, v : \chi \supset \theta, \Delta}\ (\supset_r)$$

□

Lemma 52.

(i) The $(\exists^{\mathsf{Q}}_l)$, $(\exists^{\mathsf{Q}}_r)$, $(\forall^{\mathsf{Q}}_l)$, *and* $(\forall^{\mathsf{Q}}_r)$ *rules are hp-invertible in* IntQL.

(ii) The $(\exists^{\mathsf{QC}}_l)$, $(\exists^{\mathsf{QC}}_r)$, $(\forall^{\mathsf{QC}}_l)$, *and* $(\forall^{\mathsf{QC}}_r)$ *rules are hp-invertible in* IntQCL.

Proof. We prove both results simultaneously and let $\mathsf{X} \in \{\mathsf{Q}, \mathsf{QC}\}$. Hp-invertibility of $(\exists^{\mathsf{X}}_r)$ and $(\forall^{\mathsf{X}}_l)$ follows from Lem. 49. Hp-invertibility of $(\exists^{\mathsf{X}}_l)$ and $(\forall^{\mathsf{X}}_r)$ is shown by induction on the height of the given derivation; we prove the $(\forall^{\mathsf{X}}_r)$ case as the $(\exists^{\mathsf{X}}_l)$ case is similar.

Base case. The base cases are resolved as shown below:

$$\frac{}{\mathcal{R}, \Gamma, w : p(\underline{\vec{a}}) \Rightarrow u : p(\underline{\vec{a}}), v : \forall x \phi, \Delta}\ (id_*) \quad \rightsquigarrow$$

$$\frac{}{\mathcal{R}, v \leq z, \Gamma, w : p(\underline{\vec{a}}) \Rightarrow u : p(\underline{\vec{a}}), z : \phi(\underline{b}/y), \Delta}\ (id_*)$$

$$\dfrac{}{\mathcal{R},\Gamma,v:\bot\Rightarrow w:\forall x\phi,\Delta}\,(\bot_l)\quad\rightsquigarrow\quad\dfrac{}{\mathcal{R},w\leq u,\Gamma,v:\bot\Rightarrow u:\phi(\underline{a}/x),\Delta}\,(\bot_l)$$

Inductive step. With the exception of the $(\forall^{\mathsf{Q}}_{r})$ rule, all cases are resolved by invoking IH followed by the corresponding rule. If the last inference of the derivation is an instance of $(\forall^{\mathsf{Q}}_{r})$, then there are two cases to consider: either (i) the labelled formula we aim to invert is principal, or (ii) it is not. Case (i) is resolved below top—the derivation of the premise gives the desired conclusion—and case (ii) is resolved as shown below bottom.

$$\dfrac{\mathcal{R},w\leq u,\Gamma\Rightarrow u:\phi(\underline{a}/x),\Delta}{\mathcal{R},\Gamma\Rightarrow w:\forall x\phi,\Delta}\,(\forall^{\mathsf{X}}_{r})\quad\rightsquigarrow\quad\mathcal{R},w\leq u,\Gamma\Rightarrow u:\phi(\underline{a}/x),\Delta$$

$$\dfrac{\mathcal{R},v\leq z,\Gamma\Rightarrow z:\psi(\underline{b}/y),w:\forall x\phi,\Delta}{\mathcal{R},\Gamma\Rightarrow v:\forall x\psi,w:\forall x\phi,\Delta}\,(\forall^{\mathsf{X}}_{r})\quad\rightsquigarrow$$

$$\begin{array}{cl}
\mathcal{R},v\leq z,\Gamma\Rightarrow z:\psi(\underline{b}/y),w:\forall x\phi,\Delta & \\ \hdashline
\mathcal{R},v\leq z,w\leq u,\Gamma\Rightarrow z:\psi(\underline{b}/y),u:\phi(\underline{a}/x),\Delta & \text{IH} \\ \hline
\mathcal{R},w\leq u,\Gamma\Rightarrow w:\forall x\psi,u:\phi(\underline{a}/x),\Delta & (\forall^{\mathsf{X}}_{r})
\end{array}$$

□

Lemma 55. *The rule* (ctr_l) *is hp-admissible in* IntQL *and* IntQCL.

Proof. We prove both results simultaneously and let $\mathsf{X}\in\{\mathsf{Q},\mathsf{QC}\}$. We prove the result by induction on the height of the given derivation.

Base case. Any application of (ctr_l) an instance of (id_*) or $(\bot_l)$ yields another instance of the rule, which confirms the base case.

Inductive step. For the inductive step, we assume that the derivation ends with an instance of a rule (r) (from IntQL or IntQCL) followed by an instance of (ctr_l). If the principal formula of (r) is not active in (ctr_l), then the case is handled by invoking IH followed by an instance of (r). We therefore assume that the principal formula of (r) is active in the (ctr_l) inference. This assumption also implies that we need only consider the cases where (r) is an instance of $(\wedge_l)$, $(\vee_l)$, $(Pr_{\supset})$, $(\exists^{\mathsf{X}}_{l})$, or $(\forall^{\mathsf{X}}_{l})$. We show how to resolve the $(\wedge_l)$, $(\exists^{\mathsf{X}}_{l})$, and $(\forall^{\mathsf{X}}_{l})$ cases below; the $(\vee_l)$ and $(Pr_{\supset})$ cases are similar or simple to verify.

$$\begin{array}{cl}
\mathcal{R},\Gamma,w:\phi\wedge\psi,w:\phi,w:\psi\Rightarrow\Delta & \\ \hline
\mathcal{R},\Gamma,w:\phi\wedge\psi,w:\phi\wedge\psi\Rightarrow\Delta & (\wedge_l) \\ \hline
\mathcal{R},\Gamma,w:\phi\wedge\psi\Rightarrow\Delta & (ctr_l)
\end{array}\quad\rightsquigarrow$$

$$\begin{array}{cl}
\mathcal{R},\Gamma,w:\phi\wedge\psi,w:\phi,w:\psi\Rightarrow\Delta & \\ \hdashline
\mathcal{R},\Gamma,w:\phi,w:\psi,w:\phi,w:\psi\Rightarrow\Delta & \text{Lem. 50} \\ \hdashline
\mathcal{R},\Gamma,w:\phi,w:\psi\Rightarrow\Delta & \text{IH}\times 2 \\ \hline
\mathcal{R},\Gamma,w:\phi\wedge\psi\Rightarrow\Delta & (\wedge_l)
\end{array}$$

$$\dfrac{\dfrac{\mathcal{R},\Gamma,w:\exists x\phi,w:\phi(\underline{a}/x)\Rightarrow\Delta}{\mathcal{R},\Gamma,w:\exists x\phi,w:\exists x\phi\Rightarrow\Delta}(\exists^{\mathsf{X}}_{l})}{\mathcal{R},\Gamma,w:\exists x\phi\Rightarrow\Delta}(ctr_{l}) \quad \rightsquigarrow$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\mathcal{R},\Gamma,w:\exists x\phi,w:\phi(\underline{a}/x)\Rightarrow\Delta}{\mathcal{R},\Gamma,w:\phi(\underline{a}/x),w:\phi(\underline{b}/x)\Rightarrow\Delta}\text{Lem. 52}}{\mathcal{R}(\underline{a}/\underline{b}),\Gamma(\underline{a}/\underline{b}),(w:\phi(\underline{a}/x))(\underline{a}/\underline{b}),(w:\phi(\underline{b}/x))(\underline{a}/\underline{b})\Rightarrow\Delta(\underline{a}/\underline{b})}(psb)}{\mathcal{R},\Gamma,w:\phi(\underline{a}/x),w:\phi(\underline{a}/x)\Rightarrow\Delta}=}{\mathcal{R},\Gamma,w:\phi(\underline{a}/x)\Rightarrow\Delta}\text{IH}}{\mathcal{R},\Gamma,w:\exists x\phi\Rightarrow\Delta}(\exists^{\mathsf{X}}_{l})$$

$$\dfrac{\dfrac{\mathcal{R},\Gamma,u:\phi(\underline{a}/x),w:\forall x\phi,w:\forall x\phi\Rightarrow\Delta}{\mathcal{R},\Gamma,w:\forall x\phi,w:\forall x\phi\Rightarrow\Delta}(\forall^{\mathsf{X}}_{l})}{\mathcal{R},\Gamma,w:\forall x\phi\Rightarrow\Delta}(ctr_{l}) \quad \rightsquigarrow$$

$$\dfrac{\dfrac{\mathcal{R},\Gamma,u:\phi(\underline{a}/x),w:\forall x\phi,w:\forall x\phi\Rightarrow\Delta}{\mathcal{R},\Gamma,u:\phi(\underline{a}/x),w:\forall x\phi\Rightarrow\Delta}\text{IH}}{\mathcal{R},\Gamma,w:\forall x\phi\Rightarrow\Delta}(\forall^{\mathsf{X}}_{l})$$

□

Theorem 30. *The rule* (*cut*) *is eliminable in* IntQL *and* IntQCL*.*

Proof. We show both results simultaneously and let $\mathsf{X} \in \{\mathsf{Q}, \mathsf{QC}\}$. We proceed by induction on the lexicographic ordering of pairs $(|\phi|, h_1 + h_2)$, where $|\phi|$ is the complexity of the cut formula ϕ, h_1 is the height of the derivation of the left premise of (*cut*), and h_2 is the height of the derivation of the right premise of (*cut*). We assume w.l.o.g. that (*cut*) is the last rule used in our given derivation, and that no other instances of (*cut*) appear in the given derivation. The general result follows by repeatedly applying the algorithm described below to successively eliminate topmost instances of (*cut*) until the derivation is free of (*cut*) instances.

As usual, we separate the proof into a large variety of cases, and additionally, we explicitly write out the assumption being made in each case.

1. The complexity of the cut formula is 0*, that is,* $|\phi| = 0$*.*

1.1 Both premises of (*cut*) *are an instance of* (id_*) *or* $(\bot_l)$*.*

1.1.1 Both premises of (*cut*) *are instances of* (id_*) *with the cut formula principal in both premises.* By the side conditions of the (id_*) instances (in the top left derivation), we know that there exists a propagation path $\pi(w, u)$ and a propagation path $\pi'(u, v)$ such that $s_\pi(w, u), s_{\pi'}(u, v) \in L_{S4}(a)$. This implies the existence of a propagation path $\pi''(w, v)$ such that $s_{\pi''}(w, v) \in L_{S4}(a)$, showing that the output instance of (id_*) is valid.

$$\dfrac{\dfrac{}{\mathcal{R}, \Gamma, w : p(\underline{\vec{a}}) \Rightarrow u : p(\underline{\vec{a}}), \Delta}(id_*) \qquad \dfrac{}{\mathcal{R}, \Gamma, u : p(\underline{\vec{a}}) \Rightarrow v : p(\underline{\vec{a}}), \Delta}(id_*)}{\mathcal{R}, \Gamma, w : p(\underline{\vec{a}}) \Rightarrow v : p(\underline{\vec{a}}), \Delta}(cut) \quad \rightsquigarrow$$

$$\dfrac{}{\mathcal{R}, \Gamma, w : p(\underline{\vec{a}}) \Rightarrow v : p(\underline{\vec{a}}), \Delta}(id_*)$$

1.1.2 Both premises of (cut) *are instances of* (id_*) *with the cut formula principal in the left premise and not the right.* Then, the conclusion of (cut) is an instance of (id_*).

1.1.3 Both premises of (cut) *are instances of* (id_*) *with the cut formula principal in the right premise and not the left.* Then, the conclusion of (cut) is an instance of (id_*).

1.1.4 Both premises of (cut) *are instances of* (id_*) *with the cut formula principal in neither premise.* Then, the conclusion of (cut) is an instance of (id_*).

1.1.5 The left premise of (cut) *is an instance of* $(\bot_l)$*.* Then, the conclusion of (cut) is an instance of $(\bot_l)$.

1.1.6 The right premise of (cut) *is an instance of* $(\bot_l)$*.* Due to our assumption in case 1.1.5 above, we may assume that the left premise is an instance of (id_*), which gives rise to two possibilities: either (i) the principal occurrence of $\bot$ in $(\bot_l)$ is the cut formula, or (ii) it is not. In case (i), the conclusion is an instance of (id_*), and in case (ii), the conclusion is an instance of $(\bot_l)$.

1.2 The left premise of (cut) *is an instance of* (id_*) *or* $(\bot_l)$ *and the right premise is not.* Note that by assumption 1, the principal formula of the right premise of (cut) is not the cut-formula. We split case 1.2 into two further cases, which are resolved as explained below:

1.2.1 The right premise of (cut) *is derived with an instance of* $(\supset_r)$*,* $(\exists^{\mathsf{X}}_r)$*,* $(\forall^{\mathsf{X}}_l)$*,* $(\exists^{\mathsf{X}}_l)$*, or* $(\forall^{\mathsf{X}}_r)$*.* The result follows by (i) potentially invoking the hp-admissibility of (lsb) (Lem. 47) and (psb) (Lem. 48) on the premise of $(\supset_r)$, $(\exists^{\mathsf{X}}_r)$, $(\forall^{\mathsf{X}}_l)$, $(\exists^{\mathsf{X}}_l)$, or $(\forall^{\mathsf{X}}_r)$ (which gives the right premise of (cut)) to ensure the eigenvariable condition is met (if required), followed by the hp-invertibility of $(\supset_r)$, $(\exists^{\mathsf{X}}_r)$, $(\forall^{\mathsf{X}}_l)$, $(\exists^{\mathsf{X}}_l)$, or $(\forall^{\mathsf{X}}_r)$ (Lem. 53) on the left premise of (cut) to ensure that its contexts match those of the sequent obtained from step (i). Then, (iii) we invoke IH between the proofs of our two newly obtained sequents, followed by (iv) an application of the corresponding $(\supset_r)$, $(\exists^{\mathsf{X}}_r)$, $(\forall^{\mathsf{X}}_l)$, $(\exists^{\mathsf{X}}_l)$, or $(\forall^{\mathsf{X}}_r)$ rule.

1.2.2 The right premise of (cut) *is derived with a rule other than* $(\supset_r)$*,* $(\exists^{\mathsf{X}}_r)$*,* $(\forall^{\mathsf{X}}_l)$*,* $(\exists^{\mathsf{X}}_l)$*, or* $(\forall^{\mathsf{X}}_r)$ *(and by assumption 1.2 above, cannot be an instance of* (id_*) *or* $(\bot_l)$*).* Let (r) be the rule used to derive the right premise of (cut). The result follows by (i) applying the hp-invertibility of (r) (Lem. 53) to the left premise of (cut) to ensure its contexts match the contexts of the premise(s) of (r), followed by (ii) an application of IH between the proof(s) of the sequent(s) obtained from step (i) and the premise(s) of (r), and last, (iii) an application of (r) gives the desired conclusion.

1.3 The right premise of (cut) *is an instance of* (id_*) *or* $(\bot_l)$ *and the left premise is not.* This case is argued similarly to case 1.2 above.

1.4 Neither premise of (cut) *is an instance of* (id_{*}) *or* $(\bot_{l})$. Let (r_1) be the rule used to derive the left premise of (cut) and (r_2) be the rule used to derive the right premise of (cut). Since the cut formula is of the form $p(\underline{\vec{a}})$ or $\bot$ by assumption 1 above, we know that the cut formula is not principal in (r_1) and (r_2). To eliminate (cut), we perform the following: (i) We potentially apply hp-admissibility of (lsb) (Lem. 47) and/or (psb) (Lem. 48) to the premise(s) of (r_1) and (r_2) to ensure that the eigenvariable condition is met, given that (r_1) or (r_2) is an instance of $(\supset_r)$, $(\exists^{\mathsf{X}}_{r})$, $(\forall^{\mathsf{X}}_{l})$, $(\exists^{\mathsf{X}}_{l})$, or $(\forall^{\mathsf{X}}_{r})$. (ii) We apply hp-invertibility of (r_2) to the proof(s) of the sequent(s) obtained from the premise(s) of (r_1) in step (i) and hp-invertibility of (r_1) to the proof(s) of the sequent(s) obtained from the premise(s) of (r_2) in step (i) (thus invoking Lem. 53). (iii) We invoke IH between the proofs of the sequents obtained from step (ii), followed by (iv) applications of (r_1) and (r_2) to obtain a cut-free proof of the desired conclusion.

2. The complexity of the cut formula is greater than 0*, that is,* $|\phi| > 0$.

2.1 The cut formula is not principal in either premise of (cut). Similar to the proof of case 1.4 above. Note that in case 1.4, our assumptions implied that the cut formula was not principal in the left or right premise of (cut). Here, in case 2.1, this fact holds by assumption, and so, we may argue similarly.

2.2 The cut formula is principal in the left, but not the right, premise of (cut). Let (r) be the rule used to derive the right premise of (cut). Below top, we show how to resolve the case were (r) is a unary rule, and below bottom, we show how to resolve the case where (r) is a binary rule. In the unary case below top, (r) may be an instance of $(\supset_r)$, $(\exists^{\mathsf{X}}_{r})$, $(\forall^{\mathsf{X}}_{l})$, $(\exists^{\mathsf{X}}_{l})$, or $(\forall^{\mathsf{X}}_{r})$, and so, in such a case it may be necessary to apply the hp-admissibility of (lsb) (Lem. 47) and/or (psb) (Lem. 48) to ensure the eigenvariable condition is met after the application of IH. We note that in the cases below, IH may be applied since the sum of the heights $h_1 + h_2$ has decreased by 1.

$$\dfrac{\mathcal{R},\Gamma \Rightarrow w:\phi,\Delta \qquad \dfrac{\mathcal{R}',\Gamma',w:\phi \Rightarrow \Delta'}{\mathcal{R},\Gamma,w:\phi \Rightarrow \Delta}(r)}{\mathcal{R},\Gamma \Rightarrow \Delta}(cut) \quad \rightsquigarrow$$

$$\dfrac{\dfrac{\dfrac{\mathcal{R},\Gamma \Rightarrow w:\phi,\Delta}{\mathcal{R}',\Gamma' \Rightarrow w:\phi,\Delta'}\text{Lem. 53} \qquad \mathcal{R}',\Gamma',w:\phi \Rightarrow \Delta'}{\mathcal{R}',\Gamma' \Rightarrow \Delta'}\text{IH}}{\mathcal{R},\Gamma \Rightarrow \Delta}(r)$$

$$\dfrac{\mathcal{R},\Gamma \Rightarrow w:\phi,\Delta \qquad \dfrac{\mathcal{R}',\Gamma',w:\phi \Rightarrow \Delta' \qquad \mathcal{R}'',\Gamma'',w:\phi \Rightarrow \Delta''}{\mathcal{R},\Gamma,w:\phi \Rightarrow \Delta}(r)}{\mathcal{R},\Gamma \Rightarrow \Delta}(cut) \quad \rightsquigarrow \quad \dfrac{\Pi_1 \qquad \Pi_2}{\mathcal{R},\Gamma \Rightarrow \Delta}(r)$$

$$\Pi_1 \quad = \left\{ \quad \dfrac{\dfrac{\mathcal{R},\Gamma \Rightarrow w:\phi,\Delta}{\mathcal{R}',\Gamma' \Rightarrow w:\phi,\Delta'}\text{Lem. 53} \qquad \mathcal{R}',\Gamma',w:\phi \Rightarrow \Delta'}{\mathcal{R}',\Gamma' \Rightarrow \Delta'}\text{IH} \right.$$

$$\Pi_2 \quad = \left\{ \dfrac{\dfrac{\mathcal{R},\Gamma \Rightarrow w:\phi,\Delta}{\mathcal{R}'',\Gamma'' \Rightarrow w:\phi,\Delta''}\text{Lem. 53} \qquad \mathcal{R}'',\Gamma'',w:\phi \Rightarrow \Delta''}{\mathcal{R}'',\Gamma'' \Rightarrow \Delta''}\text{IH} \right.$$

2.3 The cut formula is principal in the right, but not the left, premise of (*cut*). Similar to the previous case 2.2.

2.4 The cut formula is principal in both premises of (*cut*). By the cut-elimination theorem for G3IntQ and G3IntQC (Thm. 10), we need only consider the cases where the cut formula is of the form $\forall x \psi$ or $\exists x \psi$, as the propositional cases (where the cut formula is of the form $\psi \vee \chi$, $\psi \wedge \chi$, or $\phi \supset \chi$) are resolved in a similar manner.

2.4.1 The cut formula is of the form $\exists x \psi$*.* The case is resolved as shown below. Observe that we may the first use of IH as the sum of the heights $h_1 + h_2$ is one less than the original (*cut*), and we may invoke the second use of IH since the cut formula (namely, $w : \psi(\underline{a}/x)$) is of a smaller complexity, i.e. $|\psi(\underline{a}/x)| < |\exists x \psi| = |\phi|$.

$$\dfrac{\dfrac{\mathcal{R},\Gamma \Rightarrow \Delta, w:\psi(\underline{a}/x), w:\exists x\psi}{\mathcal{R},\Gamma \Rightarrow \Delta, w:\exists x\psi}(\exists^{\mathsf{X}}_r) \qquad \dfrac{\mathcal{R}, w:\psi(\underline{b}/x),\Gamma \Rightarrow \Delta}{\mathcal{R}, w:\exists x\psi,\Gamma \Rightarrow \Delta}(\exists^{\mathsf{X}}_l)}{\mathcal{R},\Gamma \Rightarrow \Delta}(cut) \quad \rightsquigarrow$$

$$\dfrac{\dfrac{\Pi_1 \qquad \dfrac{\dfrac{\mathcal{R}, w:\psi(\underline{b}/x),\Gamma \Rightarrow \Delta}{\mathcal{R}, w:\psi(\underline{b}/x),\Gamma \Rightarrow w:\psi(\underline{a}/x),\Delta}(wk)}{\mathcal{R}, w:\exists x\psi,\Gamma \Rightarrow w:\psi(\underline{a}/x),\Delta}(\exists^{\mathsf{X}}_l)}{\mathcal{R},\Gamma \Rightarrow w:\psi(\underline{a}/x),\Delta}\text{IH} \qquad \dfrac{\mathcal{R}, w:\psi(\underline{b}/x),\Gamma \Rightarrow \Delta}{\mathcal{R}, w:\psi(\underline{a}/x),\Gamma \Rightarrow \Delta}(psb)}{\mathcal{R},\Gamma \Rightarrow \Delta}\text{IH}$$

$$\Pi_1 = \left\{ \quad \mathcal{R},\Gamma \Rightarrow \Delta, w:\psi(\underline{a}/x), w:\exists x\psi \right.$$

2.4.2 The cut formula is of the form $\forall x \psi$*.* The case is resolved as shown below. We my invoke the first use of IH since the sum of the heights $h_1 + h_2$ is one less than the original (*cut*), and we may invoke the second use of IH since the cut formula $v : \psi(\underline{a}/x)$ is of a smaller complexity, that is $|\psi(\underline{a}/x)| < |\forall x \psi| = |\phi|$. We note that the use of Lem. 58 is justified since there exists a propagation path $\pi(w, v)$ in the propagation graph of $\mathcal{R},\Gamma \Rightarrow \Delta$ such that $s_\pi(w, v) \in L_{S4}(a)$ due to the application of $(\forall^{\mathsf{X}}_l)$.

$$\dfrac{\dfrac{\mathcal{R}, w \leq u,\Gamma \Rightarrow \Delta, u:\psi(\underline{b}/x)}{\mathcal{R},\Gamma \Rightarrow \Delta, w:\forall x\psi}(\forall^{\mathsf{X}}_r) \qquad \dfrac{\mathcal{R}, v:\psi(\underline{a}/x), w:\forall x\psi,\Gamma \Rightarrow \Delta}{\mathcal{R}, w:\forall x\psi,\Gamma \Rightarrow \Delta}(\forall^{\mathsf{X}}_l)}{\mathcal{R},\Gamma \Rightarrow \Delta}(cut) \quad \rightsquigarrow$$

$$
\begin{array}{cl}
\begin{array}{cc}
\begin{array}{cl}
\mathcal{R}, w \leq u, \Gamma \Rightarrow \Delta, u : \psi(\underline{b}/x) & \\ \hdashline
\mathcal{R}, w \leq v, \Gamma \Rightarrow \Delta, v : \psi(\underline{b}/x) & (lsb) \\ \hdashline
\mathcal{R}, w \leq v, \Gamma \Rightarrow \Delta, v : \psi(\underline{a}/x) & (psb)
\end{array}
&
\begin{array}{cl}
\Pi_1 \qquad \Pi_2 & \\ \hdashline
\mathcal{R}, w \leq v, v : \psi(\underline{a}/x), \Gamma \Rightarrow \Delta & \mathrm{IH}
\end{array}
\end{array} & \\ \hdashline
\mathcal{R}, w \leq v, \Gamma \Rightarrow \Delta & \mathrm{IH} \\ \hdashline
\mathcal{R}, \Gamma \Rightarrow \Delta & \text{Lem. 58}
\end{array}
$$

$$
\Pi_1 \quad = \left\{ \quad
\begin{array}{cl}
\dfrac{\mathcal{R}, w \leq u, \Gamma \Rightarrow \Delta, u : \psi(\underline{b}/x)}{\mathcal{R}, \Gamma \Rightarrow \Delta, w : \forall x \psi} (\forall_r^{\mathsf{X}}) & \\ \hdashline
\mathcal{R}, w \leq v, v : \psi(\underline{a}/x), \Gamma \Rightarrow \Delta, w : \forall x \psi & (wk)
\end{array}
\right.
$$

$$
\Pi_2 \quad = \left\{ \quad
\begin{array}{cl}
\mathcal{R}, v : \psi(\underline{a}/x), w : \forall x \psi, \Gamma \Rightarrow \Delta & \\ \hdashline
\mathcal{R}, w \leq v, v : \psi(\underline{a}/x), w : \forall x \psi, \Gamma \Rightarrow \Delta & (wk)
\end{array}
\right.
$$

□

Index

Bibliography

[Åqv84] Lennart Åqvist. Deontic logic. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic: Volume II: Extensions of Classical Logic*, pages 605–714. Springer Netherlands, Dordrecht, 1984.

[Avr96] Arnon Avron. The method of hypersequents in the proof theory of propositional non-classical logics. In Wilfrid Hodges, Martin Hyland, Charles Steinhorn, and John Truss, editors, *Logic: From Foundations to Applications: European Logic Colloquium*, page 1–32. Clarendon Press, USA, 1996.

[Bal00] Matteo Baldoni. Normal multimodal logics with interaction axioms. In David Basin, Marcello D'Agostino, Dov M. Gabbay, Seán Matthews, and Luca Viganò, editors, *Labelled Deduction*, pages 33–57. Springer Netherlands, Dordrecht, 2000.

[BdRV01] Patrick Blackburn, Maarten de Rijke, and Yde Venema. *Modal Logic*, volume 53 of *Cambridge Tracts in Theoretical Computer Science.* Cambridge University Press, 2001.

[Bel82] Nuel D Belnap. Display logic. *Journal of philosophical logic*, 11(4):375–417, 1982.

[BG97] Bernhard Beckert and Rajeev Goré. Free variable tableaux for propositional modal logics. In Didier Galmiche, editor, *Automated Reasoning with Analytic Tableaux and Related Methods*, pages 91–106, Berlin, Heidelberg, 1997. Springer Berlin Heidelberg.

[BGM98] Matteo Baldoni, Laura Giordano, and Alberto Martelli. A tableau calculus for multimodal logics and some (un)decidability results. In Harrie de Swart, editor, *Automated Reasoning with Analytic Tableaux and Related Methods*, volume 17 of *Lecture Notes in Computer Science*, pages 44–59, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg.

[BH75] Luitzen Egbertus Jan Brouwer and Arend Heyting. *L.E.J. Brouwer: Collected Works, Volume 1: Philosophy and Foundations of Mathematics.* North-Holland Publishing Company; New York: American Elsevier Publishing Company, 1975.

[BHT08] Philippe Balbiani, Andreas Herzig, and Nicolas Troquard. Alternative axiomatics and complexity of deliberative stit theories. *Journal of Philosophical Logic*, 37(4):387–406, 2008.

[BMV97] David Basin, Seán Matthews, and Luca Vigano. Labelled propositional modal logics: Theory and practice. *Journal of Logic and Computation*, 7(6):685–717, 1997.

[BP90] Nuel Belnap and Michael Perloff. Seeing to it that: A canonical form for agentives. In Henry E. Kyburg, Ronald P. Loui, and Greg N. Carlson, editors, *Knowledge Representation and Defeasible Reasoning*, pages 167–190. Springer Netherlands, Dordrecht, 1990.

[BPX01] Nuel D Belnap, Michael Perloff, and Ming Xu. *Facing the future: agents and choices in our indeterminist world.* Oxford University Press, 2001.

[Bro11a] Jan Broersen. Deontic epistemic stit logic distinguishing modes of mens rea. *Journal of Applied Logic*, 9(2):137–152, 2011.

[Bro11b] Jan M Broersen. Making a start with the stit logic analysis of intentional action. *Journal of philosophical logic*, 40(4):499–530, 2011.

[Brü09] Kai Brünnler. Deep sequent systems for modal logic. *Arch. Math. Log.*, 48(6):551–577, 2009.

[Bul92] Robert A. Bull. Cut elimination for propositional dynamic logic without *. *Z. Math. Logik Grundlag. Math.*, 38(2):85–100, 1992.

[Bus98] Samuel R Buss. An introduction to proof theory. *Handbook of proof theory*, 137:1–78, 1998.

[CdCGH97] Marcos A Castilho, Luis Farinas del Cerro, Olivier Gasquet, and Andreas Herzig. Modal tableaux with propagation rules and structural rules. *Fundamenta Informaticae*, 32(3, 4):281–297, 1997.

[CDGT13] Ranald Clouston, Jeremy Dawson, Rajeev Goré, and Alwen Tiu. Annotation-Free Sequent Calculi for Full Intuitionistic Linear Logic. In Simona Ronchi Della Rocca, editor, *Computer Science Logic 2013 (CSL 2013)*, volume 23 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 197–214, Dagstuhl, Germany, 2013. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[CES86] E. M. Clarke, E. A. Emerson, and A. P. Sistla. Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Trans. Program. Lang. Syst.*, 8(2):244–263, April 1986.

[CGT08] A. Ciabattoni, N. Galatos, and K. Terui. From axioms to analytic rules in nonclassical logics. In *2008 23rd Annual IEEE Symposium on Logic in Computer Science*, pages 229–240, 2008.

[CLR18] Agata Ciabattoni, Tim Lyon, and Revantha Ramanayake. From display to labelled proofs for tense logics. In Sergei N. Artëmov and Anil Nerode, editors, *Logical Foundations of Computer Science - International Symposium, LFCS 2018, Deerfield Beach, FL, USA, January 8-11, 2018, Proceedings*, volume 10703 of *Lecture Notes in Computer Science*, pages 120–139. Springer, 2018.

[CLRT20] Agata Ciabattoni, Tim Lyon, Revantha Ramanayake, and Alwen Tiu. Display to labelled proofs and back again for tense logics. *ACM Trans. Comput. Log.*, 2020. Submitted.

[CMS13] Agata Ciabattoni, Paolo Maffezioli, and Lara Spendier. Hypersequent and labelled calculi for intermediate logics. In Didier Galmiche and Dominique Larchey-Wendling, editors, *Automated Reasoning with Analytic Tableaux and Related Methods*, volume 8123 of *Lecture Notes in Computer Science*, pages 81–96, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[Cor87] Giovanna Corsi. Weak logics with strict implication. *Mathematical Logic Quarterly*, 33(5):389–406, 1987.

[Cor89] Giovanna Corsi. A cut-free calculus for dummett's lc quantified. *Mathematical Logic Quarterly*, 35(4):289–301, 1989.

[CS02] Claudio Castellini and Alan Smaill. A systematic presentation of quantified modal logics. *Logic Journal of the IGPL*, 10(6):571–599, 2002.

[CST09] Agata Ciabattoni, Lutz Straßburger, and Kazushige Terui. Expanding the realm of systematic proof theory. In Erich Grädel and Reinhard Kahle, editors, *Computer Science Logic, 23rd international Workshop, CSL 2009, 18th Annual Conference of the EACSL, Coimbra, Portugal, September 7-11, 2009. Proceedings*, volume 5771 of *Lecture Notes in Computer Science*, pages 163–178. Springer, 2009.

[Cur52] Haskell B. Curry. The elimination theorem when modality is present. *The Journal of Symbolic Logic*, 17(4):249–265, 1952.

[dCH95] Luis Fariñas del Cerro and Andreas Herzig. Modal deduction with applications in epistemic and temporal logics. In Dov M. Gabbay, C. J. Hogger, and J. A. Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming (Vol. 4): Epistemic and Temporal Reasoning*, page 499–594. Oxford University Press, Inc., USA, 1995.

[dCP88] Luis Fariñas del Cerro and Martti Penttonen. Grammar logics. *Logique et Analyse*, 31(121/122):123–134, 1988.

[DdN05] Stéphane Demri and Hans de Nivelle. Deciding regular grammar logics with converse through first-order logic. *Journal of Logic, Language and Information*, 14(3):289–329, 2005.

[Dem01] Stephane Demri. The complexity of regularity in grammar logics and related modal logics. *Journal of Logic and Computation*, 11(6):933–960, 2001.

[DN12] Roy Dyckhoff and Sara Negri. Proof analysis in intermediate logics. *Archive for Mathematical Logic*, 51(1-2):71–92, 2012.

[Doš88] Kosta Došen. Sequent-systems and groupoid models. i. *Studia Logica*, 47(4):353–385, 1988.

[Dyc92] Roy Dyckhoff. Contraction-free sequent calculi for intuitionistic logic. *The Journal of Symbolic Logic*, 57(3):795–807, 1992.

[Fit72] Melvin Fitting. Tableau methods of proof for modal logics. *Notre Dame Journal of Formal Logic*, 13(2):237–247, 1972.

[Fit83] Melvin Fitting. *Proof methods for modal and intuitionistic logics*, volume 169. Springer Science & Business Media, 1983.

[Fit14] Melvin Fitting. Nested sequents for intuitionistic logics. *Notre Dame Journal of Formal Logic*, 55(1):41–61, 2014.

[Fit15] Melvin Fitting. Cut-free proof systems for geach logics. *IfCoLog Journal of Logics and their Applications*, 2(2):17–64, 2015.

[FK15] Melvin Fitting and Roman Kuznets. Modal interpolation via nested sequents. *Annals of pure and applied logic*, 166(3):274–305, 2015.

[FM99] Camillo Fiorentini and Pierangelo Miglioli. A cut-free sequent calculus for the logic of constant domains with a limited amount of duplications. *Logic Journal of IGPL*, 7(6):733–753, 1999.

[FMHV95] Ronald Fagin, Yoram Moses, Joseph Y Halpern, and Moshe Y Vardi. *Reasoning about knowledge.* MIT press, 1995.

[Fre79] Gottlob Frege. *Begriffsschrift, eine der arithmetischen nachgebildete Formelsprache des reinen Denkens.* Nebert, 1879.

[Gab96] Dov M Gabbay. *Labelled deductive systems*, volume 33 of *Oxford Logic guides.* Clarendon Press/Oxford Science Publications, 1996.

[Gar13] James W. Garson. *Modal Logic for Philosophers.* Cambridge University Press, 2 edition, 2013.

[Gen35a] Gerhard Gentzen. Untersuchungen über das logische schließen. i. *Mathematische zeitschrift*, 39(1):176–210, 1935.

[Gen35b] Gerhard Gentzen. Untersuchungen über das logische schließen. ii. *Mathematische Zeitschrift*, 39(1):405–431, 1935.

[Gir87] Jean-Yves Girard. Linear logic. *Theoretical computer science*, 50(1):1–101, 1987.

[Gli29] Valery Glivenko. Sur quelques points de la logique de m. brouwer. *Bulletins de la classe des sciences*, 15(5):183–188, 1929.

[GN05] Rajeev Goré and Linh Anh Nguyen. A tableau calculus with automaton-labelled formulae for regular grammar logics. In Bernhard Beckert, editor, *Automated Reasoning with Analytic Tableaux and Related Methods*, volume 3702 of *Lecture Notes in Computer Science*, pages 138–152, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[Gör71] Sabine Görnemann. A logic stronger than intuitionism. *The Journal of Symbolic Logic*, 36(2):249–261, 1971.

[GPT08] Rajeev Goré, Linda Postniece, and Alwen Tiu. Cut-elimination and proof-search for bi-intuitionistic logic using nested sequents. In Carlos Areces and Robert Goldblatt, editors, *Advances in Modal Logic 7, papers from the seventh conference on "Advances in Modal Logic," held in Nancy, France, 9-12 September 2008*, pages 43–66. College Publications, 2008.

[GPT11] Rajeev Goré, Linda Postniece, and Alwen Tiu. On the correspondence between display postulates and deep inference in nested sequent calculi for tense logics. *Log. Methods Comput. Sci.*, 7(2), 2011.

[GR12] Rajeev Goré and Revantha Ramanayake. Labelled tree sequents, tree hypersequents and nested (deep) sequents. In Thomas Bolander, Torben Braüner, Silvio Ghilardi, and Lawrence S. Moss, editors, *Advances in Modal Logic 9, papers from the ninth conference on "Advances in Modal Logic," held in Copenhagen, Denmark, 22-25 August 2012*, pages 279–299. College Publications, 2012.

[Grz64] Andrzej Grzegorczyk. A philosophically plausible formal interpretation of intuitionistic logic. *Indagationes Mathematicae*, 26(5):596–601, 1964.

[GSS09] D. Gabbay, V. Shehtman, and D. Skvortsov. *Quantification in Non-classical Logics*. Studies in Logic and Foundations of Mathematics. Elsevier, 2009.

[Hey30] Arend Heyting. Die formalen regeln der intuitionistischen logik. *Sitzungsbericht PreuBische Akademie der Wissenschaften Berlin, physikalisch-mathematische Klasse II*, pages 42–56, 1930.

[HM90] Joseph Y. Halpern and Yoram Moses. Knowledge and common knowledge in a distributed environment. *J. ACM*, 37(3):549–587, July 1990.

[HM92] Joseph Y Halpern and Yoram Moses. A guide to completeness and complexity for modal logics of knowledge and belief. *Artificial intelligence*, 54(3):319–379, 1992.

[Hor01] John F Horty. *Agency and deontic logic.* Oxford University Press, 2001.

[How80] William A Howard. The formulae-as-types notion of construction. *To HB Curry: essays on combinatory logic, lambda calculus and formalism*, 44:479–490, 1980.

[HS04] Ian Horrocks and Ulrike Sattler. Decidability of shiq with complex role inclusion axioms. *Artificial Intelligence*, 160(1-2):79–104, 2004.

[IK07] Ryo Ishigaki and Kentaro Kikuchi. Tree-sequent methods for subintuitionistic predicate logics. In Nicola Olivetti, editor, *Automated Reasoning with Analytic Tableaux and Related Methods*, volume 4548 of *Lecture Notes in Computer Science*, pages 149–164, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[Kan57] Stig Kanger. *Provability in logic.* Almqvist & Wiksell, 1957.

[Kas94] Ryo Kashima. Cut-free sequent calculi for some tense logics. *Studia Logica*, 53(1):119–135, 1994.

[KL18] Roman Kuznets and Björn Lellmann. Interpolation for intermediate logics via hyper- and linear nested sequents. In Guram Bezhanishvili, Giovanna D'Agostino, George Metcalfe, and Thomas Studer, editors, *Advances in Modal Logic 12, proceedings of the 12th conference on "Advances in Modal Logic," held in Bern, Switzerland, August 27-31, 2018*, pages 473–492. College Publications, 2018.

[Kle52] Stephen Cole Kleene. *Introduction to Metamathematics.* American Elsevier Publishing Company, INC. - New York, 1952.

[Kle69] Dieter Klemke. *Ein vollständiger Kalkül für die Folgerungsbeziehung der Grzegorczyk-Semantik.* PhD thesis, Freiburg University, Freiburg, Germany, 1969.

[KO03] H Kushida and Mitsuhiro Okada. A proof-theoretic study of the correspondence of classical logic and modal logic. *Journal of Symbolic Logic*, 68(4):1403–1414, 2003.

[KO10] Hitoshi Kihara and Hiroakira Ono. Interpolation properties, beth definability properties and amalgamation properties for substructural logics. *Journal of Logic and Computation*, 20(4):823–875, 2010.

[Kol67] A.N. Kolmogorov. On the principle of tertium non datur. mathematics of the ussr, sbornik 32, 646–667 (1925). In Jean Van Heijenoort, editor, *From Frege to Gödel: a source book in mathematical logic, 1879-1931*, volume 9. Harvard University Press, 1967.

[Kri65] Saul A. Kripke. Semantical analysis of intuitionistic logic i. In J.N. Crossley and M.A.E. Dummett, editors, *Formal Systems and Recursive Functions*, volume 40 of *Studies in Logic and the Foundations of Mathematics*, pages 92 – 130. Elsevier, 1965.

[KS94] Ryo Kashima and Tatsuya Shimura. Cut-elimination theorem for the logic of constant domains. *Mathematical Logic Quarterly*, 40(2):153–172, 1994.

[Kuz16a] Roman Kuznets. Craig interpolation via hypersequents. *Concepts of Proof in Mathematics, Philosophy, and Computer Science. Ontos Mathematical Logic*, 6:193–214, 2016.

[Kuz16b] Roman Kuznets. Proving craig and lyndon interpolation using labelled sequent calculi. In Loizos Michael and Antonis Kakas, editors, *Logics in Artificial Intelligence*, volume 10021 of *Lecture Notes in Computer Science*, pages 320–335, Cham, 2016. Springer International Publishing.

[Kuz18] Roman Kuznets. Multicomponent proof-theoretic method for proving interpolation properties. *Annals of Pure and Applied Logic*, 169(12):1369–1418, 2018.

[Lah13] Ori Lahav. From frame properties to hypersequent rules in modal logics. In *28th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2013, New Orleans, LA, USA, June 25-28, 2013*, pages 408–417, 2013.

[Lel15] Björn Lellmann. Linear nested sequents, 2-sequents and hypersequents. In Hans De Nivelle, editor, *Automated Reasoning with Analytic Tableaux and Related Methods*, volume 9323 of *Lecture Notes in Computer Science*, pages 135–150, Cham, 2015. Springer International Publishing.

[Lew18] Clarence Irving Lewis. *A survey of symbolic logic.* University of California Press, 1918.

[LIEG17] Tim Lyon, Christian Ittner, Timo Eckhardt, and Norbert Gratzl. The basics of display calculi. *Kriterion – Journal of Philosophy*, 31(2):55–100, 2017.

[Lor13] Emiliano Lorini. Temporal logic and its application to normative reasoning. *Journal of Applied Non-Classical Logics*, 23(4):372–399, 2013.

[LP15] Björn Lellmann and Elaine Pimentel. Proof search in nested sequent calculi. In Martin Davis, Ansgar Fehnker, Annabelle McIver, and Andrei Voronkov, editors, *Logic for Programming, Artificial Intelligence, and Reasoning -*

20th International Conference, LPAR-20 2015, Suva, Fiji, November 24-28, 2015, Proceedings, volume 9450 of *Lecture Notes in Computer Science*, page 558–574, Berlin, Heidelberg, 2015. Springer-Verlag.

[LS77] Edward J Lemmon and Dana S Scott. *An Introduction to Modal Logic: the Lemmon Notes*. Blackwell, 1977.

[LS15] Emiliano Lorini and Giovanni Sartor. Influence and responsibility: A logical analysis. In Antonino Rotolo, editor, *28th International Conference on Legal Knowledge and Information Systems (JURIX 2015)*, volume 279 of *Frontiers in Artificial Intelligence and Applications*, pages 51–60. IOS Press, 2015.

[LTGC20] Tim Lyon, Alwen Tiu, Rajeev Goré, and Ranald Clouston. Syntactic interpolation for tense logics and bi-intuitionistic logic via nested sequents. In Maribel Fernández and Anca Muscholl, editors, *28th EACSL Annual Conference on Computer Science Logic, CSL 2020, January 13-16, 2020, Barcelona, Spain*, volume 152 of *LIPIcs*, pages 28:1–28:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[LvB19] Tim Lyon and Kees van Berkel. Automating agential reasoning: Proof-calculi and syntactic decidability for stit logics. In Matteo Baldoni, Mehdi Dastani, Beishui Liao, Yuko Sakurai, and Rym Zalila Wenkstern, editors, *PRIMA 2019: Principles and Practice of Multi-Agent Systems - 22nd International Conference, Turin, Italy, October 28-31, 2019, Proceedings*, volume 11873 of *Lecture Notes in Computer Science*, pages 202–218, Cham, 2019. Springer International Publishing.

[LW11] Carsten Lutz and Frank Wolter. Foundations for uniform interpolation and forgetting in expressive description logics. In Toby Walsh, editor, *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, volume 2. AAAI Press/International Joint Conferences on Artificial Intelligence, 2011.

[Lyo20a] Tim Lyon. On deriving nested calculi for intuitionistic logics from semantic systems. In Sergei N. Artëmov and Anil Nerode, editors, *Logical Foundations of Computer Science - International Symposium, LFCS 2020, Deerfield Beach, FL, USA, January 4-7, 2020, Proceedings*, volume 11972 of *Lecture Notes in Computer Science*, pages 177–194. Springer, 2020.

[Lyo20b] Tim Lyon. On the correspondence between nested calculi and semantic systems for intuitionistic logics. *Journal of Logic and Computation*, 2020.

[Lyo20c] Tim Lyon. Syntactic cut-elimination for intuitionistic fuzzy logic via linear nested sequents. In Sergei N. Artëmov and Anil Nerode, editors, *Logical Foundations of Computer Science - International Symposium, LFCS 2020, Deerfield Beach, FL, USA, January 4-7, 2020, Proceedings*, volume 11972 of *Lecture Notes in Computer Science*, pages 156–176. Springer, 2020.

[Mae60] Shoji Maehara. On the interpolation theorem of craig. *Sûgaku*, 12(4):235–237, 1960.

[Mas92] Andrea Masini. 2-sequent calculus: A proof theory of modalities. *Annals of Pure and Applied Logic*, 58(3):229–246, 1992.

[Mas93] Andrea Masini. 2-sequent calculus: Intuitionism and natural deduction. *Journal of Logic and Computation*, 3(5):533–562, 1993.

[McM18] Kenneth L. McMillan. Interpolation and model checking. In Edmund M. Clarke, Thomas A. Henzinger, Helmut Veith, and Roderick Bloem, editors, *Handbook of Model Checking*, pages 421–446. Springer International Publishing, Cham, 2018.

[MH81] J. McCarthy and P.J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. In Bonnie Lynn Webber and Nils J. Nilsson, editors, *Readings in Artificial Intelligence*, pages 431 – 450. Morgan Kaufmann, 1981.

[Min97] Grigori Mints. Indexed systems of sequents and cut-elimination. *Journal of Philosophical Logic*, 26(6):671–696, 1997.

[MS14] Sonia Marin and Lutz Straßburger. Label-free Modular Systems for Classical and Intuitionistic Modal Logics. In *Advances in Modal Logic 10*, Groningen, Netherlands, August 2014.

[MS17] Sonia Marin and Lutz Straßburger. Proof theory for indexed nested sequents. In Renate A. Schmidt and Cláudia Nalon, editors, *Automated Reasoning with Analytic Tableaux and Related Methods*, volume 10501 of *Lecture Notes in Computer Science*, pages 81–97, Cham, 2017. Springer International Publishing.

[Mur04] Yuko Murakami. Utilitarian deontic logic. In Renate A. Schmidt, Ian Pratt-Hartmann, Mark Reynolds, and Heinrich Wansing, editors, *Advances in Modal Logic 5, papers from the fifth conference on "Advances in Modal logic," held in Manchester, UK, 9-11 September 2004*, pages 211–230. King's College Publications, 2004.

[Neg05] Sara Negri. Proof analysis in modal logic. *Journal of Philosophical Logic*, 34(5-6):507, 2005.

[Ner91] Anil Nerode. Some lectures on modal logic. In Friedrich L. Bauer, editor, *Logic, Algebra, and Computation*, volume 79, pages 281–334, Berlin, Heidelberg, 1991. Springer Berlin Heidelberg.

[NP20] Sara Negri and E Pavlović. Proof-theoretic analysis of the logics of agency: The deliberative stit. *Studia Logica*, pages 1–35, 2020.

[NS09] Linh Anh Nguyen and Andrzej Szałas. A tableau calculus for regular grammar logics with converse. In Renate A. Schmidt, editor, *Automated Deduction – CADE-22*, volume 1397 of *Lecture Notes in Computer Science*, pages 421–436, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[NS11] Linh Anh Nguyen and Andrzej Szałas. Exptime tableau decision procedures for regular grammar logics with converse. *Studia Logica*, 98(3):387, 2011.

[NVP11] Sara Negri and Jan Von Plato. *Proof analysis: a contribution to HIlbert's last problem.* Cambridge University Press, 2011.

[ONA04] Mauricio Osorio, Juan A. Navarro, and José Arrazola. Applications of intuitionistic logic in answer set programming. *Theory Pract. Log. Program.*, 4(3):325–354, May 2004.

[Orl28] Ivan E Orlov. The calculus of compatibility of propositions. *Mathematics of the USSR, Sbornik*, 35:263–286, 1928.

[OW18] Grigory K Olkhovikov and Heinrich Wansing. An axiomatic system and a tableau calculus for stit imagination logic. *Journal of Philosophical Logic*, 47(2):259–279, 2018.

[Pea89] Giuseppe Peano. *Arithmetices principia: Nova methodo exposita.* Fratres Bocca, 1889.

[Pim18] Elaine Pimentel. A semantical view of proof systems. In Lawrence S. Moss, Ruy J. G. B. de Queiroz, and Maricarmen Martínez, editors, *Logic, Language, Information, and Computation - 25th International Workshop, WoLLIC 2018, Bogota, Colombia, July 24-27, 2018, Proceedings*, volume 10944 of *Lecture Notes in Computer Science*, pages 61–76. Springer, 2018.

[Pog09] Francesca Poggiolesi. The method of tree-hypersequents for modal propositional logic. In David Makinson, Jacek Malinowski, and Heinrich Wansing, editors, *Towards Mathematical Philosophy*, volume 28 of *Trends in logic*, pages 31–51. Springer, 2009.

[Pos47] Emil L Post. Recursive unsolvability of a problem of thue. *The Journal of Symbolic Logic*, 12(1):1–11, 1947.

[Pos09] Linda Postniece. Deep inference in bi-intuitionistic logic. In Hiroakira Ono, Makoto Kanazawa, and Ruy de Queiroz, editors, *Logic, Language, Information and Computation*, pages 320–334, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[Pos10] Linda Postniece. *Proof Theory and Proof Search of Bi-Intuitionistic and Tense Logic.* PhD thesis, The Australian National University, 2010.

[Pot83] Garrel Pottinger. Uniform, cut-free formulations of t, s4 and s5. *Journal of Symbolic Logic*, 48(3):900, 1983.

[Pra65] Dag Prawitz. *Natural deduction : a proof-theoretical study.* PhD thesis, Stockholm University, Department of Philosophy, 1965.

[PRL19] Elaine Pimentel, Revantha Ramanayake, and Björn Lellmann. Sequentialising nested systems. In Serenella Cerrito and Andrei Popescu, editors, *Automated Reasoning with Analytic Tableaux and Related Methods*, volume 11714 of *Lecture Notes in Computer Science*, pages 147–165, Cham, 2019. Springer International Publishing.

[PU18] Luís Pinto and Tarmo Uustalu. A proof-theoretic study of bi-intuitionistic propositional sequent calculus. *Journal of Logic and Computation*, 28(1):165–202, 2018.

[Rad12] Alexandre Rademaker. *A proof theory for description logics.* Springer Science & Business Media, 2012.

[Rag65] Andrés R. Raggio. Gentzen's hauptsatz for the systems ni and nk. *Logique et Analyse*, 8(30):91–100, 1965.

[Rau80] Cecylia Rauszer. An algebraic and kripke-style approach to a certain extension of intuitionistic logic, dissertations mathematicae, 1980.

[Res94] Greg Restall. Subintuitionistic logics. *Notre Dame Journal of Formal Logic*, 35(1):116–129, 1994.

[Res06] Greg Restall. Comparing modal sequent systems. *Unpublished manuscript*, pages 1–13, 2006.

[Rus06] Bertrand Russell. The theory of implication. *American Journal of Mathematics*, 28(2):159–202, 1906.

[SBA20] Colin Shea-Blymyer and Houssam Abbas. A deontic logic analysis of autonomous systems' safety. In *Proceedings of the 23rd International Conference on Hybrid Systems: Computation and Control*, HSCC '20, New York, NY, USA, 2020. Association for Computing Machinery.

[Sim94] Alex K Simpson. *The proof theory and semantics of intuitionistic modal logic.* PhD thesis, University of Edinburgh. College of Science and Engineering. School of Informatics, 1994.

[Sip12] Michael Sipser. *Introduction to the Theory of Computation.* Cengage learning, 2012.

[Sla97] John K. Slaney. Minlog: A minimal logic theorem prover. In William McCune, editor, *Automated Deduction - CADE-14, 14th International Conference on Automated Deduction, Townsville, North Queensland, Australia, July 13-17, 1997, Proceedings*, volume 1249 of *Lecture Notes in Computer Science*, pages 268–271. Springer, 1997.

[Str13] Lutz Straßburger. Cut elimination in nested sequents for intuitionistic modal logics. In Frank Pfenning, editor, *Foundations of Software Science and Computation Structures*, volume 7794 of *Lecture Notes in Computer Science*, pages 209–224, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[Tak13] Gaisi Takeuti. *Proof theory*, volume 81. Courier Corporation, 2013.

[Ten87] Neil Tennant. Natural deduction and sequent calculus for intuitionistic relevant logic. *The Journal of symbolic logic*, 52(3):665–680, 1987.

[TIG12] Alwen Tiu, Egor Ianovski, and Rajeev Goré. Grammar logics in nested sequent calculus: Proof theory and decision procedures. In Thomas Bolander, Torben Braüner, Silvio Ghilardi, and Lawrence S. Moss, editors, *Advances in Modal Logic 9, papers from the ninth conference on "Advances in Modal Logic," held in Copenhagen, Denmark, 22-25 August 2012*, pages 516–537. College Publications, 2012.

[TvD88] A.S. Troelstra and D. van Dalen. *Constructivism in Mathematics*, volume 1. Elsevier Science, 1988.

[Vak87] Dimiter Vakarelov. Abstract characterization of some knowledge representation systems and the logic nil of nondeterministic information. In Ph. Jorrand and V. Sgurev, editors, *Artificial Intelligence II*, pages 255–260, Amsterdam, 1987. North-Holland.

[Var97] Moshe Y. Vardi. Why is modal logic so robustly decidable? In Neil Immerman and Phokion G. Kolaitis, editors, *Descriptive Complexity and Finite Models: Proceedings of a DIMACS Workshop, January 14-17, 1996, Princeton University*, volume 31, page 149. American Mathematical Soc., 1997.

[vBL19a] Kees van Berkel and Tim Lyon. Cut-free calculi and relational semantics for temporal stit logics. In Francesco Calimeri, Nicola Leone, and Marco Manna, editors, *Logics in Artificial Intelligence*, pages 803–819, Cham, 2019. Springer International Publishing.

[vBL19b] Kees van Berkel and Tim Lyon. A neutral temporal deontic STIT logic. In Patrick Blackburn, Emiliano Lorini, and Meiyun Guo, editors, *Logic, Rationality, and Interaction - 7th International Workshop, LORI 2019, Chongqing, China, October 18-21, 2019, Proceedings*, volume 11813 of *Lecture Notes in Computer Science*, pages 340–354. Springer, 2019.

[vBL21] Kees van Berkel and Tim Lyon. The varieties of ought-implies-can and deontic stit logic. In Fenrong Liu, Alessandra Marra, Paul Portner, and Frederik Van De Putte, editors, *Deontic Logic and Normative Systems: 15th International Conference (DEON2020/2021, Munich)*. College Publications, 2021.

[vD04] Dirk van Dalen. *Logic and Structure*. Springer-Verlag Berlin Heidelberg, 2004.

[Vig00] Luca Viganò. *Labelled Non-Classical Logics*. Springer Science & Business Media, 2000.

[vP18] Jan von Plato. The development of proof theory. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, winter 2018 edition, 2018.

[Vra07] Peter BM Vranas. I ought, therefore i can. *Philosophical studies*, 136(2):167–216, 2007.

[vW51] Georg Henrik von Wright. Deontic logic. *Mind*, 60(237):1–15, 1951.

[Wan94] Heinrich Wansing. Sequent calculi for normal modal propositional logics. *Journal of Logic and Computation*, 4(2):125–142, 1994.

[Wan02] Heinrich Wansing. Sequent systems for modal logics. In D. M. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic: Volume 8*, pages 61–145. Springer Netherlands, Dordrecht, 2002.

[Wan06] Heinrich Wansing. Tableaux for multi-agent deliberative-stit logic. In Guido Governatori, Ian M. Hodkinson, and Yde Venema, editors, *Advances in Modal Logic 6, papers from the sixth conference on "Advances in Modal Logic," held in Noosa, Queensland, Australia, on 25-28 September 2006*, pages 503–520, 2006.