Decidable Query Entailment with Existential Rules of Bounded-Treewidth via Proof-theoretic Characteristics*

First Author[1[0000−1111−2222−3333]] and Second Author[1[1111−2222−3333−4444]]

[1] Princeton University, Princeton NJ 08544, USA
[2] Springer Heidelberg, Tiergartenstr. 17, 69121 Heidelberg, Germany
`lncs@springer.com`
http://www.springer.com/gp/computer-science/lncs
[3] ABC Institute, Rupert-Karls-University Heidelberg, Heidelberg, Germany
`{abc,lncs}@uni-heidelberg.de`

Abstract. tim: TODO

Keywords: one · two · ...

1 Introduction

The formalism of existential rules has come to prominence as an effective approach for both specifying and querying knowledge. Within this context, a knowledge base takes the form $\mathcal{K} = (\mathcal{D}, \mathcal{R})$, where $\mathcal{D}$ is a finite collection of atomic facts (called a *database*) and $\mathcal{R}$ is a finite set of *existential rules* (called a *rule set*), which are first-order formulae of the form $\forall \mathbf{x}\mathbf{y}(\varphi(\mathbf{x}, \mathbf{y}) \to \exists \mathbf{z}\psi(\mathbf{y}, \mathbf{z}))$. Although existential rules are written in a relatively simple language, they are expressive enough to generalize many important languages used in knowledge representation, such as datalog [**?**] and description logics [**?**]. Moreover, existential rules have meaningful applications within the domain of ontology-based query answering [2], data exchange and integration [9], and have proven beneficial in the study of general decidability criteria [10].

The *query entailment problem* consists of taking a knowledge base $\mathcal{K}$, a query q, and determining if $\mathcal{K} \models q$. As this problem is known to be undecidable for arbitrary rule sets [7], much work has gone into identifying existential rule fragments for which decidability can be reclaimed. Typically, such classes of rule sets are identified in one of two ways: either, decidable query entailment is established on *syntactic* grounds, i.e. a rule set satisfies a set of *syntactic properties* (such classes are called *concrete classes*), or on *abstract* grounds, i.e. a rule set satisfies an abstract property which may not be obvious (such classes are called *abstract classes*). Examples of concrete classes include functional/inclusion dependencies [11], datalog [**?**], and guarded rules [6]. Examples of abstract classes

* Work supported by the European Research Council (ERC) Consolidator Grant 771779 (DeciGUT).

include finite expansion sets [4], finite unification sets [3], and bounded-treewidth sets (**bts**) [6].

Yet, there is a third means of establishing the decidability of query entailment: only limited work has gone into identifying classes of rule sets with decidable query entailment based on their *proof-theoretic characteristics*, that is, based on the derivations such rules produce. To the best of the authors' knowledge, only the class of *greedy bounded treewidth sets* (**gbts**) has been identified in such a manner (see [14]). A rule set qualifies as **gbts** when every derivation it produces is *greedy*, i.e. the frontier of every rule application in the derivation only contains constants from the knowledge base or nulls occurring in the head of a single, previous rule application. The utility of this property is that it lets one straightforwardly construct a model with finite treewidth for the knowledge base under consideration, thus establishing the decidability of query entailment [6].

In this paper, we investigate the **gbts** class as well as three new classes of rule sets where decidability is determined proof-theoretically. First, we define a weakened version of **gbts**, dubbed **wgbts**, where the rule set need only produce *at least one greedy derivation* relative to any given database. Second, we investigate two new classes of rule sets, dubbed *cycle-free derivation graph sets* (**cdgs**) and *weakly cycle-free derivation graph sets* (**wcdgs**), which are defined relative to the notion of a *derivation graph*. Derivation graphs were introduced by Baget et al. [5] and are directed acyclic graphs encoding *how* certain facts are derived throughout the course of a derivation. The utility of such objects is that through the application of *reduction operations* a derivation graph may be reduced to a tree, which serves as a tree decomposition of a model of the considered knowledge base. Such objects were used to establish the subsumption of (weakly) frontier guarded rule sets under bounded-treewidth sets [5].

Our contributions are as follows: (1) We investigate how proof-theoretic structures gives rise to decidable query entailment and propose three new classes of rule sets. (2) We show that **gbts** = **cdgs** and **wgbts** = **wcdgs** classes, establishing a correspondence between greedy derivations and reducible derivation graphs. (3) We show that **wgbts** *properly subsumes* **gbts** via a novel proof transformation argument. Therefore, by the former point, we also find that **wcdgs** properly subsumes **cdgs**. (4) We show that the purported proof-theoretical characterization of **bts** fails to subsume **fes**, and thus, fails to coincide with **bts**.

This paper is organized accordingly: In Section 2, we define preliminary notions, and in Section 3, we discuss issues surrounding a proof-theoretic characterization of **bts**. Subsequently, we study **gbts** and **wgbts** in Section 4, and show that the latter class properly subsumes the former via an intricate proof transformation argument. In Section 5, we define **cdgs** and **wcdgs** as well as show that **gbts** = **cdgs** and **wgbts** = **wcdgs**. Last, in Section 6, we conclude and discuss future work.

2 Preliminaries

Syntax and formulae. We let Ter be a set of *terms*, which is the the union of three countably infinite, pairwise disjoint sets, namely, the set of *constants* Con, the set of *variables* Var, and the set of *nulls* Nul. We use a, b, c, ... (occasionally annotated) to denote constants, and x, y, z, ... (occasionally annotated) to denote both variables and nulls. A *signature* Σ is a set of *predicates* p, q, r, ... (which may be annotated) such that for each $p \in \Sigma$, $ar(p) \in \mathbb{N}$ is the *arity* of p. For simplicity, we assume a fixed signature Σ throughout the course of the paper.

An *atom* over Σ is defined to be a formula of the form $p(t_1, \ldots, t_n)$, where $p \in \Sigma$, $ar(p) = n$, and $t_i \in \text{Ter}$ for each $i \in \{1, \ldots, n\}$. A *ground atom* over Σ is defined to be an atom $p(a_1, \ldots, a_n)$ such that $a_i \in \text{Con}$ for each $i \in \{1, \ldots, n\}$. We will often use $\mathbf{t}$ to denote a tuple $(t_1, \ldots, t_n)$ of terms and $p(\mathbf{t})$ to denote a (ground) atom $p(t_1, \ldots, t_n)$. An *instance* over Σ is defined to be a (potentially infinite) set $\mathcal{I}$ of atoms over constants and nulls, and a *database* $\mathcal{D}$ is defined to be a finite set of ground atoms. We let $\mathcal{X}$, $\mathcal{Y}$, ... (occasionally annotated) denote (potentially infinite) sets of atoms with $\text{Ter}(\mathcal{X})$, $\text{Con}(\mathcal{X})$, $\text{Var}(\mathcal{X})$, and $\text{Nul}(\mathcal{X})$ denoting the set of terms, constants, variables, and nulls occurring in the atoms of $\mathcal{X}$, respectively.

Substitutions and homomorphisms. A *substitution* is defined to be a partial function over the set of terms Ter. A *homomorphism* h from a set $\mathcal{X}$ of atoms to a set $\mathcal{Y}$ of atoms, is a substitution such that (i) $p(h(t_1), \ldots, h(t_n)) \in \mathcal{Y}$, if $p(t_1, \ldots, t_n) \in \mathcal{X}$, and (ii) $h(a) = a$ for each $a \in \text{Con}$. If h is a homomorphism from $\mathcal{X}$ to $\mathcal{Y}$, we say that h *homomorphically maps* $\mathcal{X}$ to $\mathcal{Y}$. A set $\mathcal{X}$ of atoms and a set $\mathcal{Y}$ of atoms are *homomorphically equivalent*, written $\mathcal{X} \equiv \mathcal{Y}$, *iff* a homomorphism exists from $\mathcal{X}$ to $\mathcal{Y}$, and vice versa. A homomorphism h is an *isomorphism iff* h is bijective and h^{-1} is a homomorphism.

Existential rules. Whereas databases encode assertional knowledge, ontologies are built in the current setting by means of *existential rules*, which we will frequently refer to as *rules* more simply. An existential rule is a first-order sentence of the form:

$$\rho = \forall \mathbf{x}\mathbf{y}(\varphi(\mathbf{x}, \mathbf{y}) \rightarrow \exists \mathbf{z}\psi(\mathbf{y}, \mathbf{z}))$$

where $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$ are pairwise disjoint collections of variables, $\varphi(\mathbf{x}, \mathbf{y})$ is a conjunction of atoms over constants and the variables $\mathbf{x}, \mathbf{y}$, and $\psi(\mathbf{y}, \mathbf{z})$ is a conjunction of atoms over constants and the variables $\mathbf{y}, \mathbf{z}$. We define $body(\rho) = \varphi(\mathbf{x}, \mathbf{y})$ to be the *body* of ρ, and $head(\rho) = \psi(\mathbf{y}, \mathbf{z})$ to be the *head* of ρ. For convenience, we will often interpret a conjunction $p_1(\mathbf{t}_1) \wedge \cdots \wedge p_n(\mathbf{t}_n)$ of atoms (such as the body or head of a rule) as a set $\{p_1(\mathbf{t}_1), \cdots, p_n(\mathbf{t}_n)\}$ of atoms; if h is a homomorphism, then $h(p_1(\mathbf{t}_1) \wedge \cdots \wedge p_n(\mathbf{t}_n)) := \{p_1(h(\mathbf{t}_1)), \cdots, p_n(h(\mathbf{t}_n))\}$ with h applied componentwise to each tuple $\mathbf{t}_i$ of terms. The *frontier* of ρ, written $fr(\rho)$, is the set of variables $\mathbf{y}$ that the body and head of ρ have in common, that is, $fr(\rho) = \text{Var}(body(\rho)) \cap \text{Var}(head(\rho))$. We define a *frontier atom* in a rule ρ to be an atom containing at least one frontier variable. We use ρ and annotated versions thereof to denote rules, as well as $\mathcal{R}$ and annotated versions thereof to denote finite sets of rules (called *rule sets*).

Models. We note that sets of atoms (which include instances and databases) may be seen as first-order interpretations, and so, we may use $\models$ to represent the satisfaction of formulae on such structures. A set of atoms $\mathcal{X}$ satisfies a set of atoms $\mathcal{Y}$ (or, equivalently, $\mathcal{X}$ is a model of $\mathcal{Y}$), written $\mathcal{X} \models \mathcal{Y}$, *iff* there exists a homomorphic mapping from $\mathcal{Y}$ to $\mathcal{X}$. A set of atoms $\mathcal{X}$ satisfies a rule ρ (or, equivalently, $\mathcal{X}$ is a model of ρ), written $\mathcal{X} \models \rho$, *iff* for any homomorphism h, if h is a homomorphism from $body(\rho)$ to $\mathcal{X}$, then it can be extended to a homomorphism $\overline{h}$ from $head(\rho)$ to $\mathcal{X}$. A set of atoms $\mathcal{X}$ satisfies a rule set $\mathcal{R}$ (or, equivalently, $\mathcal{X}$ is a model of $\mathcal{R}$), written $\mathcal{X} \models \mathcal{R}$, *iff* $\mathcal{X} \models \rho$ for every rule $\rho \in \mathcal{R}$. If a set of atoms $\mathcal{X}$ homomorphically maps into *every* model of a set of atoms, a rule, or a rule set, then we refer to $\mathcal{X}$ as a *universal model* of the set of atoms, rule, or rule set [8].

Knowledge bases and querying. A *knowledge base (KB)* $\mathcal{K}$ is defined to be a pair $(\mathcal{D}, \mathcal{R})$, where $\mathcal{D}$ is a database and $\mathcal{R}$ is a rule set. An instance $\mathcal{I}$ is a *model* of $\mathcal{K} = (\mathcal{D}, \mathcal{R})$ *iff* $\mathcal{D} \subseteq \mathcal{I}$ and $\mathcal{I} \models \mathcal{R}$. We consider querying knowledge bases with *conjunctive queries (CQs)*, that is, with formulae of the form $q(\mathbf{y}) = \exists \mathbf{x} \varphi(\mathbf{x}, \mathbf{y})$, where $\varphi(\mathbf{x}, \mathbf{y})$ is a non-empty conjunction of atoms over the variables $\mathbf{x}, \mathbf{y}$ and constants. We refer to the variables $\mathbf{y}$ in $q(\mathbf{y})$ as *free* and define a *Boolean conjunctive query (BCQ)* to be a CQ without free variables, i.e. a BCQ is a CQ of the form $q = \exists \mathbf{x} \varphi(\mathbf{x})$. A knowledge base $\mathcal{K} = (\mathcal{D}, \mathcal{R})$ *entails* a CQ $q(\mathbf{y}) = \exists \mathbf{x} \varphi(\mathbf{x}, \mathbf{y})$, written $\mathcal{K} \models q(\mathbf{y})$, *iff* $\varphi(\mathbf{x}, \mathbf{y})$ homomorphically maps into every model $\mathcal{I}$ of $\mathcal{K}$; we note that this is equivalent to $\varphi(\mathbf{x}, \mathbf{y})$ homomorphically mapping into a universal model of $\mathcal{D}$ and $\mathcal{R}$.

As we are interested in extracting implicit knowledge from the explicit knowledge presented in a knowledge base $\mathcal{K} = (\mathcal{D}, \mathcal{R})$, we are interested in deciding the *BCQ entailment problem*:[4]

(BCQ Entailment) Given a KB $\mathcal{K}$ and a BCQ q, is it the case that $\mathcal{K} \models q$?

While it is well-known that the BCQ entailment problem is undecidable in general [7], restricting oneself to certain classes of rule sets (e.g. datalog or finite unification sets [5]) may recover decidability of the above problem. We refer to classes of rule sets for which BCQ entailment is decidable as *query-decidable classes*.

Derivations. One means by which we can extract implicit knowledge from a given KB is through the use of *derivations*, that is, sequences of instances obtained by sequentially applying rules to given data. We say that a rule $\rho = \forall \mathbf{x}\mathbf{y}(\varphi(\mathbf{x}, \mathbf{y}) \rightarrow \exists \mathbf{z} \psi(\mathbf{y}, \mathbf{z}))$ is *triggered* in an instance $\mathcal{I}$ via a homomorphism h, written succinctly as $\tau(\rho, \mathcal{I}, h)$, *iff* h homomorphically maps $\varphi(\mathbf{x}, \mathbf{y})$ to $\mathcal{I}$. In this case, we define $\mathbf{Ch}(\mathcal{I}, \rho, h) = \mathcal{I} \cup \overline{h}(\psi(\mathbf{y}, \mathbf{z}))$, where $\overline{h}$ is an extension

[4] We recall that entailment of non-Boolean CQs or even query answering can all be reduced to BCQ entailment.

of h mapping every variable z in $\mathbf{z}$ to fresh a null. Consequently, we define an $\mathcal{R}$-*derivation* to be a sequence $\mathcal{I}_0, (\rho_1, h_1, \mathcal{I}_1), \ldots, (\rho_n, h_n, \mathcal{I}_n)$ such that (i) $\rho_i \in \mathcal{R}$ for each $i \in \{1, \ldots, n\}$, (ii) $\tau(\rho_i, \mathcal{I}_{i-1}, h_i)$ holds for $i \in \{1, \ldots, n\}$, and (iii) $\mathcal{I}_i = \mathbf{Ch}(\mathcal{I}_{i-1}, \rho, h_i)$ for $i \in \{1, \ldots, n\}$. We will use δ and annotations thereof to denote $\mathcal{R}$-derivations, and we define the length of an $\mathcal{R}$-derivation $\delta = \mathcal{I}_0, (\rho_1, h_1, \mathcal{I}_1), \ldots, (\rho_n, h_n, \mathcal{I}_n)$, denoted $|\delta|$, to be n. Furthermore, for instances $\mathcal{I}$ and $\mathcal{I}'$, we write $\mathcal{I} \xrightarrow{\delta}_{\mathcal{R}} \mathcal{I}'$ to mean that there exists an $\mathcal{R}$-derivation δ of $\mathcal{I}'$ from $\mathcal{I}$. Also, if $\mathcal{I}''$ can be derived from $\mathcal{I}'$ by means of a rule $\rho \in \mathcal{R}$ and homomorphism h, we abuse notation and write $\mathcal{I} \xrightarrow{\delta}_{\mathcal{R}} \mathcal{I}', (\rho, h, \mathcal{I}'')$ to indicate that $\mathcal{I} \xrightarrow{\delta}_{\mathcal{R}} \mathcal{I}'$ and $\mathcal{I}' \xrightarrow{\delta'}_{\mathcal{R}} \mathcal{I}''$ with $\delta' = \mathcal{I}', (\rho, h, \mathcal{I}'')$. Derivations play a fundamental role in this paper as we aim to identify (and analyze the relationships between) query-decidable classes of rule sets based on *how* such rule sets derive information, i.e. we are interested in classes of rule sets that may be *proof-theoretically characterized*.

Chase. A tool that will prove useful in the current work is the *chase*, which in our setting is a procedure that (in essence) simultaneously constructs all $\mathcal{K}$-derivations in a breadth-first manner. Although many variants of the chase exist [5, 9, 12], we utilize the chase procedure (also called the *k-Saturation*) from Baget et al. [5]. We use the chase in the current work as a purely technical tool for obtaining universal models of knowledge bases, proving useful in separating certain query-decidable classes of rule sets.

We define the *one-step application* of all triggered rules of a rule set $\mathcal{R}$ in $\mathcal{I}$ accordingly:

$$\mathbf{Ch}_1(\mathcal{I}, \mathcal{R}) = \bigcup_{\rho \in \mathcal{R}, \tau(\rho, \mathcal{I}, h)} \mathbf{Ch}(\mathcal{I}, \rho, h).$$

We let $\mathbf{Ch}_0(\mathcal{I}, \mathcal{R}) = \mathcal{I}$, as well as let $\mathbf{Ch}_{i+1}(\mathcal{I}, \mathcal{R}) = \mathbf{Ch}_1(\mathbf{Ch}_i(\mathcal{I}, \mathcal{R}), \mathcal{R})$, and define the *chase* to be

$$\mathbf{Ch}_\infty(\mathcal{I}, \mathcal{R}) = \bigcup_{i \in \mathbb{N}} \mathbf{Ch}_i(\mathcal{I}, \mathcal{R}).$$

For any KB $\mathcal{K} = (\mathcal{D}, \mathcal{R})$, the chase $\mathbf{Ch}_\infty(\mathcal{D}, \mathcal{R})$ always serves as a universal model of $\mathcal{K}$, that is, $\mathcal{D} \subseteq \mathbf{Ch}_\infty(\mathcal{D}, \mathcal{R})$, $\mathbf{Ch}_\infty(\mathcal{D}, \mathcal{R}) \models \mathcal{R}$, and $\mathbf{Ch}_\infty(\mathcal{D}, \mathcal{R})$ homomorphically maps into every model of $\mathcal{D}$ and $\mathcal{R}$.

Rule dependence. Let ρ and ρ' be rules. We say that ρ' *depends* on ρ *iff* there exists an instance $\mathcal{I}$ such that (i) ρ' is not triggered in $\mathcal{I}$ via any homomorphism, (ii) ρ is triggered in $\mathcal{I}$ via a homomorphism h, and (iii) ρ' is triggered in $\mathbf{Ch}(\mathcal{I}, \rho, h')$ via a homomorphism h'. We define the *graph of rule dependencies* [1] of a set $\mathcal{R}$ of rules to be $G(\mathcal{R}) = (V, E)$ such that (i) $V = \mathcal{R}$ and (ii) $(\rho, \rho') \in E$ *iff* ρ' depends on ρ.

Treewidth. A *tree decomposition* of an instance $\mathcal{I}$ is defined to be a tree $T = (V, E)$ such that $V \subseteq 2^{\mathrm{Ter}(\mathcal{I})}$ (where each element of V is called a *bag*) and $E \subseteq V \times V$, satisfying the following three conditions: (i) $\bigcup_{X \in V} X = \mathrm{Ter}(\mathcal{I})$, (ii) for each $p(t_1, \ldots, t_n) \in \mathcal{I}$, there is an $X \in V$ such that $\{t_1, \ldots, t_n\} \subseteq X$, and (iii) for each $t \in \text{Ter}(\mathcal{I})$, the subgraph of T induced by the bags $X \in V$ with $t \in X$ is connected. (NB. Condition (iii) is referred to as the *connectedness condition*.) We define the *width* of a tree decomposition $T = (V, E)$ of an instance $\mathcal{I}$ as follows:

$$w(T) := \max\{|X| : X \in V\} - 1$$

i.e. the width is equal to the cardinality of the largest node in T minus 1. We let $w(T) = \infty$ *iff* for all $n \in \mathbb{N}$, $n \leq \max\{|X| : X \in V\}$. We define the *treewidth* of an instance $\mathcal{I}$, written $tw(\mathcal{I})$, as follows:

$$tw(\mathcal{I}) := \min\{w(T) : \ T \text{ is a tree decomposition of } \mathcal{I}\}$$

i.e. the treewidth of an instance is equal to the minimal width among all its tree decompositions, which is set to ∞ when no tree decomposition of $\mathcal{I}$ has a finite width.

3 Finite-Expansion and Bounded-Treewidth

In this section, we accomplish two goals: first, we discuss two query-decidable classes of existential rules that are of particular importance in this paper, namely, *proof-theoretic bounded treewidth sets* (**pbts**) and *finite treewidth sets* (**fts**). The former class admits a syntactic definition, containing those rule sets for which the treewidth of every step of the chase is uniformly bounded by a natural number, whereas the latter admits a semantic definition, containing those rule sets which possess a universal model of finite treewidth, relative to any given database (both classes are formally defined in Definition 1 below).[5]

Since classes of rule sets introduced in subsequent sections will be defined in a syntactic or proof-theoretic manner (meaning, they only contain rule sets that produce specific types of derivations), it will prove straightforward to relate such definitions with the syntactic definition of **pbts**, ultimately showing that such classes are subsumed by **pbts**. Since **pbts** is subsumed by **fts** – which is known to admit decidable query entailment – **pbts** will serve as a technical tool and bridge connecting the classes of rule sets we consider later on to **fts**, thus demonstrating that such classes admit decidable query entailment.

The second goal we achieve in this section concerns the relationship between **pbts**, **fts**, and the class of *finite expansion sets* (**fes**), which admits a semantic definition as well (see Definition 1 below), containing those rule sets which possess a finite universal model. It is trivial to confirm that **fes** is subsumed by **fts** as any finite universal model is a universal model of finite treewidth, yet, despite claims to the contrary, we will prove that **fes** is *not* subsumed by **pbts**.

Let us now formally define **fes**, **pbts**, and **fts**. Afterward, we give a proposition stating the subsumption of **fts** over the former two classes of rule sets.

[5] We note that both **pbts** and **fts** have been referred to as *bounded treewidth sets* in the literature (cf. [5]). However, as these two classes are distinct from one another, we apply unique names to distinguish them.

Definition 1 (fes, pbts, fts). *Let $\mathcal{R}$ be a rule set. $\mathcal{R}$ is defined to be a* finite expansion set (**fes**) *iff for every database $\mathcal{D}$, $(\mathcal{D}, \mathcal{R})$ has a finite universal model. $\mathcal{R}$ is defined to be a* proof-theoretic bounded treewidth set (**pbts**) *iff for every database $\mathcal{D}$, there exists an $n \in \mathbb{N}$ such that for every $k \in \mathbb{N}$, $tw(\mathbf{Ch}_k(\mathcal{D}, \mathcal{R})) \leq n$. $\mathcal{R}$ is defined to be a* finite treewidth set (**fts**) *iff for every database $\mathcal{D}$, there exists an $n \in \mathbb{N}$ and a universal model $\mathcal{I}^*$ of $(\mathcal{D}, \mathcal{R})$ such that $tw(\mathcal{I}^*) \leq n$.*

Proposition 1. *Let $\mathcal{R}$ be a rule set.*

1. *If $\mathcal{R}$ is* **fes**, *then $\mathcal{R}$ is* **fts**;
2. *if $\mathcal{R}$ is* **pbts**, *then $\mathcal{R}$ is* **fts**.

We now provide a rule set, denoted $\mathcal{R}_1$, that falls within the **fes** class, but outside of the **pbts** class, that is, we establish the following theorem:

Theorem 1. *$\mathcal{R}_1$ is* **fes**, *but is not* **pbts**, *and thus,* **pbts** *does not subsume* **fes**.

The above theorem is a consequence of two lemmata, each of which we argue in turn below: first, we argue that the rule set $\mathcal{R}_1$ is not **pbts** (Lemma 1), and second, we argue that it is **fes** (Lemma 2). We will dedicate the remainder of the section to justifying these two lemmata by means of an example, and defer the formal proofs to the appendix.

We define $\mathcal{R}_1$ to be the rule set $\{\rho_1, \rho_2, \rho_3\}$ where

$$\rho_1 = r(x, y) \rightarrow \exists z.r(y, z)$$
$$\rho_2 = r(x, y) \wedge r(y, z) \rightarrow r(x, z)$$
$$\rho_3 = r(x, y) \rightarrow r(x, x)$$

To provide intuition as to why $\mathcal{R}_1$ is not **pbts**, let us consider the instances obtained via the chase on the database $\mathcal{D}_* = \{r(a, b)\}$. The database $\mathcal{D}_*$, $\mathbf{Ch}_1(\mathcal{D}_*, \mathcal{R}_1)$, and $\mathbf{Ch}_2(\mathcal{D}_*, \mathcal{R}_1)$ can be viewed graphically as shown in Figure 1.

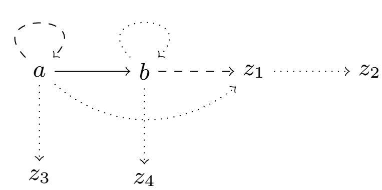

Fig. 1. An example of applying the chase twice to the database $\mathcal{D}_* = \{r(a, b)\}$ with solid, dashed, and dotted lines depicting the predicate r. The constants a and b connected by the solid line depict the original database $\mathcal{D}_*$, the dashed lines show how $\mathcal{D}_*$ is extended after one step of the chase with the null z_1 being introduced, and the dotted lines show how $\mathcal{D}_*$ is extended after a second step of the chase with the nulls z_2, z_3, and z_4 being introduced.

Observe that ρ_1 extends the terminal vertex of a binary edge with a binary edge to a fresh null, ρ_2 introduces transitive edges, and ρ_3 introduces loops at the source vertex of a binary edge. Therefore, one can see via the example in Figure 1 that as k increases, $\mathbf{Ch}_k(\mathcal{D}_*, \mathcal{R}_1)$ will contain ever longer lines that are transitively (and reflexively) closed. Due to the connectedness condition imposed on tree decompositions, all terms in such a line must occur within the same bag, meaning that since such a line grows (i.e. includes increasingly many terms) in $\mathbf{Ch}_k(\mathcal{D}_*, \mathcal{R}_1)$ as k increases, the width of any tree decomposition must also increase, and hence, $tw(\mathbf{Ch}_k(\mathcal{D}_*, \mathcal{R}_1))$ will increase as k increases. Therefore, we have found a database (viz. $\mathcal{D}_*$) for which $tw(\mathbf{Ch}_k(\mathcal{D}_*, \mathcal{R}_1))$ is not uniformly bounded for each $k \in \mathbb{N}$, showing that $\mathcal{R}_1$ is not **pbts**. Thus, we have:

Lemma 1. $\mathcal{R}_1$ *is not* **pbts***.*

Although $\mathcal{R}_1$ is not **pbts** and $tw(\mathbf{Ch}_\infty(\mathcal{D}_*, \mathcal{R}_1)) = \infty$, it is still possible to find a finite universal model for the knowledge base $(\mathcal{D}_*, \mathcal{R}_1)$. Toward this end, let us consider the closely related rule set $\mathcal{R}'_1 = \{\rho'_1, \rho'_2\}$ such that (i) the rule $\rho'_1 = r(x, y) \wedge r(y, z) \rightarrow r(x, z)$ and (ii) the rule $\rho'_2 = r(x, y) \rightarrow r(x, x) \wedge r(y, y)$. Observe that if we close $\mathcal{D}_*$ under applications of $\mathcal{R}'_1$, then we obtain the instance $\mathcal{I} = \{r(a, a), r(a, b), r(b, b)\}$, which can be mapped into $\mathbf{Ch}_2(\mathcal{D}_*, \mathcal{R}_1) \subseteq \mathbf{Ch}_\infty(\mathcal{D}_*, \mathcal{R}_1)$ (depicted in Figure 1) via the identity homomorphism h.

It is obvious that $\mathcal{I}$ is finite, but even more, $\mathcal{I}$ serves as a universal model for the KB $(\mathcal{D}_*, \mathcal{R}_1)$. Modelhood is straightforward to establish by considering the satisfaction of each rule from $\mathcal{R}_1$ on $\mathcal{I}$. Universality is a consequence of the fact that $\mathcal{I}$ homomorphically maps into $\mathbf{Ch}_\infty(\mathcal{D}_*, \mathcal{R}_1)$, which is a universal model of $(\mathcal{D}_*, \mathcal{R}_1)$ (meaning $\mathbf{Ch}_\infty(\mathcal{D}_*, \mathcal{R}_1)$ homomorphically maps into any model of $(\mathcal{D}_*, \mathcal{R}_1)$), and thus, $\mathcal{I}$ homomorphically maps into any model of $(\mathcal{D}_*, \mathcal{R}_1)$.

As it so happens, closing any database $\mathcal{D}$ under $\mathcal{R}'_1$ yields a finite universal model for $(\mathcal{D}, \mathcal{R}_1)$ in general, implying that $\mathcal{R}_1$ is **fes**. To provide some intuition regarding this point, first let $\mathcal{I}^*$ be an instance obtained by closing an arbitrary database $\mathcal{D}$ under $\mathcal{R}'_1$. We will first explain why $\mathcal{I}^*$ is finite, then explain why it is a model of $\mathcal{R}_1$, and last, explain why $\mathcal{I}^*$ is a universal model of $\mathcal{R}_1$. First, since $\mathcal{R}'_1$ never introduces new terms, it is easy to see that the resulting instance $\mathcal{I}^*$ will be finite (as witnessed in the above example). Second, $\mathcal{I}^*$ will be a model of $(\mathcal{D}, \mathcal{R}_1)$ for the following reasons: as with $\mathcal{R}_1$, the rule set $\mathcal{R}'_1$ introduces transitive edges as well as loops at initial vertices of binary edges, meaning that any instance that satisfies $\mathcal{R}'_1$ will satisfy $\rho_2, \rho_3 \in \mathcal{R}_1$. Additionally, whereas $\mathcal{R}_1$ extends the terminal vertex of a binary edge with a fresh binary edge to a fresh null (via the ρ_1 rule), $\mathcal{R}'_1$ introduces *loops* at the terminal vertex of a binary edge (via the ρ'_2 rule). The introduction of such loops (e.g. in $\mathcal{I}^*$) ensures the satisfaction of $head(\rho_1)$, and therefore, of ρ_1. Thus, $\mathcal{I}^* \models \mathcal{R}_1$. Last, similar to the example with $\mathcal{I}$ above, one can show that $\mathcal{I}^*$ can be mapped into $\mathbf{Ch}_\infty(\mathcal{D}, \mathcal{R}_1)$ by means of the identity homomorphism (mapping all constants in $\mathcal{D}$ to themselves), establishing the universality of $\mathcal{I}^*$. We therefore have the following:

Lemma 2. $\mathcal{R}_1$ *is* **fes***.*

4 Greediness

We now discuss a property of derivations referred to as *greediness*. In essence, a derivation is greedy when the frontier of any applied rule consists solely of constants from a given KB and/or nulls introduced by a *single* previous rule application. Such derivations were defined by Thomazo et al. [14] and were used to identify the (query-decidable) class of *greedy bounded treewidth sets* (**gbts**), that is, the class of rule sets that produce only *greedy derivations* (defined below) when applied to a database.

In this section, we also identify a new query-decidable class of rule sets, referred to as *weakly greedy bounded treewidth sets* (**wgbts**). The **wgbts** class serves as a weakened version of **gbts**, and contains rule sets that admit at least one greedy derivation of any derivable instance. It is straightforward to confirm that **wgbts** generalizes **gbts** since if a rule set is **gbts**, then every derivation of a derivable instance is greedy, implying that every derivable instance has *some* greedy derivation. Yet, what is non-trivial to show is that **wgbts** *properly subsumes* **gbts**. We prove this fact by means of a proof-theoretic argument and counter-example. First, we show under what conditions we can permute rule applications in a given derivation (see Lemma 3 below), and second, we provide a rule set which has non-greedy derivations (meaning, the rule set is not **gbts**), but where every derivation can be transformed into a greedy derivation by means of rule permutations and replacements.

Let us formally define greedy derivations, followed by examples to demonstrate the concept of (non-)greediness, and after, we will define **gbts** and **wgbts** on the basis thereof.

Definition 2 (Greedy Derivation [14]). *We define an $\mathcal{R}$-derivation*

$$\delta = \mathcal{I}_0, (\rho_1, h_1, \mathcal{I}_1), \ldots, (\rho_n, h_n, \mathcal{I}_n)$$

to be greedy *iff for each i such that $0 < i \leq n$, there exists a $j < i$ such that $h_i(fr(\rho_i)) \subseteq \mathrm{Nul}(\overline{h}_j(head(\rho_j))) \cup \mathrm{Con}(\mathcal{I}_0, \mathcal{R}) \cup \mathrm{Nul}(\mathcal{I}_0)$.*

To give examples of non-greedy and greedy derivations, let us define the database $\mathcal{D}_\dagger := \{p(a), q(b)\}$ and the rule set $\mathcal{R}_2 := \{\rho_1, \rho_2, \rho_3, \rho_4\}$, with

$\rho_1 = p(x) \to \exists yz.q(x, y, z)$
$\rho_2 = r(x) \to \exists yz.s(x, y, z)$
$\rho_3 = p(x) \wedge r(y) \to \exists zwuv.q(x, z, w) \wedge s(y, u, v)$
$\rho_4 = q(x, y, z) \wedge s(w, u, v) \to \exists o.t(x, y, w, u, o)$

An example of a non-greedy derivation is the following:

$$\delta_1 = \mathcal{D}_\dagger, (\rho_1, h_1, \mathcal{I}_1), (\rho_1, h_2, \mathcal{I}_2), (\rho_2, h_3, \mathcal{I}_3), (\rho_4, h_4, \mathcal{I}_4),$$

where $h_1(x) = h_2(x) = a$, $h_3(x) = b$, $h_4(x) = a$, $h_4(y) = y_0$, $h_4(z) = z_0$, $h_4(w) = b$, $h_4(u) = y_2$, and $h_4(v) = z_2$, and all instances participating in δ_1 are as follows:

$\mathcal{D}_\dagger = \{p(a), r(b)\}$ $\mathcal{I}_3 = \mathcal{I}_2 \cup \{s(b, y_2, z_2)\}$

$\mathcal{I}_1 = \mathcal{D}_\dagger \cup \{q(a, y_0, z_0)\}$ $\mathcal{I}_4 = \mathcal{I}_3 \cup \{t(a, y_0, b, y_2, o)\}$

$\mathcal{I}_2 = \mathcal{I}_1 \cup \{q(a, y_1, z_1)\}$

The above derivation is not greedy because

$$h_4(fr(\rho_4)) = \{a, y_0, b, y_2\} = \{y_0\} \cup \{y_2\} \cup \{a, b\} \cup \emptyset \subseteq \mathrm{Nul}(\overline{h}_1(head(\rho_1))) \cup \mathrm{Nul}(\overline{h}_3(head(\rho_2))) \cup \mathrm{Con}(\mathcal{D}_\dagger, \mathcal{R}_2) \cup \mathrm{Nul}(\mathcal{D}_\dagger)$$

That is to say, the frontier of the last rule application (i.e. the application of ρ_4) contains nulls introduced by *two* previous rule applications (as opposed to containing nulls from just a single previous rule application), namely, the first application of ρ_1 and the application of ρ_2. In contrast, the following is an example of a greedy derivation

$$\delta_2 = \mathcal{D}_\dagger, (\rho_3, h'_1, \mathcal{I}'_1), (\rho_1, h'_2, \mathcal{I}'_2), (\rho_4, h'_3, \mathcal{I}'_3),$$

where $h'_1(x) = a$, $h'_1(y) = b$, $h'_2(x) = a$, $h'_3(x) = a$, $h'_3(y) = y_0$, $h'_3(z) = z_0$, $h'_3(w) = b$, $h'_3(u) = y_2$, $h'_3(v) = z_2$, and all instances participating in δ_2 are as follows:

$$\begin{aligned} \mathcal{D}_\dagger &= \{p(a), r(b)\} \\ \mathcal{I}'_1 &= \mathcal{D}_\dagger \cup \{q(a, y_0, z_0), s(b, y_2, z_2)\} \\ \mathcal{I}'_2 &= \mathcal{I}'_1 \cup \{q(a, y_1, z_1)\} \\ \mathcal{I}'_3 &= \mathcal{I}'_2 \cup \{t(a, y_0, b, y_2, o)\} \end{aligned}$$

One can confirm the greediness of δ_2 by observing that the frontier of every rule application contains nothing but constants and/or nulls introduced by a single previous rule application.

Definition 3 ((Weakly) Greedy Bounded-Treewidth Set). *Let $\mathcal{R}$ be a rule set. $\mathcal{R}$ is a* greedy bounded-treewidth set (**gbts**) *iff for any database $\mathcal{D}$, if $\mathcal{D} \xrightarrow{\delta}_{\mathcal{R}} \mathcal{I}$, then δ is greedy. $\mathcal{R}$ is a* weakly greedy bounded-treewidth set (**wgbts**) *iff for any database $\mathcal{D}$, if $\mathcal{D} \xrightarrow{\delta}_{\mathcal{R}} \mathcal{I}$, then there exists some greedy $\mathcal{R}$-derivation δ' such that $\mathcal{D} \xrightarrow{\delta'}_{\mathcal{R}} \mathcal{I}$.*

Remark 1. Observe that **gbts** and **wgbts** are characterized on the basis of derivations *from a given database*, that is, derivations of the form $\mathcal{I}_0, (\rho_1, h_1, \mathcal{I}_1), \dots, (\rho_n, h_n, \mathcal{I}_n)$ where $\mathcal{I}_0 = \mathcal{D}$ is a database. In such a case, a derivation of the above form is greedy *iff* for each i with $0 < i \leq n$, there exists a $j < i$ such that $h_i(fr(\rho_i)) \subseteq \mathrm{Nul}(\overline{h}_j(head(\rho_j))) \cup \mathrm{Con}(\mathcal{D}, \mathcal{R})$. The reason being, when $\mathcal{I}_0 = \mathcal{D}$ is a database, then $\mathrm{Con}(\mathcal{I}_0, \mathcal{R}) \cup \mathrm{Nul}(\mathcal{I}_0) = \mathrm{Con}(\mathcal{D}, \mathcal{R}) \cup \mathrm{Nul}(\mathcal{D}) = \mathrm{Con}(\mathcal{D}, \mathcal{R})$ since $\mathrm{Nul}(\mathcal{D}) = \emptyset$ as databases only contain constants (and not nulls) by definition.

As noted above, it is straightforward to show that **wgbts** subsumes **gbts**.

Proposition 2. *Every* **gbts** *ruleset is* **wgbts**.

Still, establishing that **wgbts** strictly subsumes **gbts**, i.e. there are rule sets within **wgbts** that are outside **gbts**, requires more effort. As it so happens, the rule set $\mathcal{R}_2$ (defined above) serves as such a rule set, admitting non-greedy $\mathcal{R}_2$-derivations, but where it can be shown that every instance derivable using the rule set admits a greedy $\mathcal{R}_2$-derivation. As a case in point, observe that the $\mathcal{R}_2$-derivations δ_1 and δ_2 both derive the same instance $\mathcal{I}_4 = \mathcal{I}'_3$, however, δ_1 is a non-greedy $\mathcal{R}_2$-derivation of the instance and δ_2 is a greedy $\mathcal{R}_2$-derivation of the instance. Clearly, the existence of the non-greedy $\mathcal{R}_2$-derivation δ_1 witnesses that $\mathcal{R}_2$ is not **gbts**. To establish that $\mathcal{R}_2$ nevertheless falls within the **wgbts** class, we show that every non-greedy $\mathcal{R}_2$-derivation can be transformed into a greedy $\mathcal{R}_2$-derivation by means of two operations: (i) rule permutations and (ii) rule replacements.

Regarding rule permutations, we consider under what conditions we may swap consecutive applications of rules in a derivation to yield a new derivation of the same instance. For example, in the $\mathcal{R}_2$-derivation δ_1 above, we may swap the consecutive applications of ρ_1 and ρ_2 to obtain the following derivation:

$$\delta'_1 = \mathcal{D}_\dagger, (\rho_1, h_1, \mathcal{I}_1), (\rho_2, h_3, \mathcal{I}_1 \cup (\mathcal{I}_3 \setminus \mathcal{I}_2)), \\ (\rho_1, h_2, \mathcal{I}_3), (\rho_4, h_4, \mathcal{I}_4).$$

$\mathcal{I}_1 \cup (\mathcal{I}_3 \setminus \mathcal{I}_2) = \{p(a), r(b), q(a, y_0, z_0), s(b, y_2, z_2)\}$ is derived by applying ρ_2 and the application of ρ_1 reclaims the instance $\mathcal{I}_3$. Therefore, the same instance $\mathcal{I}_4$ remains the conclusion. Although one can confirm that δ'_1 is indeed an $\mathcal{R}_2$-derivation, thus serving as a successful example of a rule permutation (meaning, the rule permutation yields another $\mathcal{R}_2$-derivation), the following question still remains: for a rule set $\mathcal{R}$, under what conditions will permuting rules within a given $\mathcal{R}$-derivation always yield another $\mathcal{R}$-derivation?

We pose an answer to this question, formulated as the *permutation lemma* below, which states that an application of a rule ρ may be permuted before an application of a rule ρ' so long as the former rule does not depend on the latter.[6] Furthermore, it should be noted that such rule permutations preserve the greediness of derivations. In the context of the above example, ρ_2 may be permuted before ρ_1 in δ_1 because the former does not depend on the latter.

Lemma 3 (Permutation Lemma). *Let $\mathcal{R}$ be a rule set with $\mathcal{I}_0$ an instance. Suppose we have a (greedy) $\mathcal{R}$-derivation of the following form:*

$$\mathcal{I}_0, \ldots, (\rho_i, h_i, \mathcal{I}_i), (\rho_{i+1}, h_{i+1}, \mathcal{I}_{i+1}), \ldots, (\rho_n, h_n, \mathcal{I}_n)$$

If ρ_{i+1} does not depend on ρ_i, then the following is a (greedy) $\mathcal{R}$-derivation as well:

$$\mathcal{I}_0, \ldots, (\rho_{i+1}, h_{i+1}, \mathcal{I}'_i), (\rho_i, h_i, \mathcal{I}_{i+1}), \ldots, (\rho_n, h_n, \mathcal{I}_n)$$

where $\mathcal{I}'_i = \mathcal{I}_{i-1} \cup (\mathcal{I}_{i+1} \setminus \mathcal{I}_i)$.

[6] Rule dependence is defined in Section 2 and is based on the work of Baget [1].

As a consequence of the above lemma, rules may always be permuted in a given $\mathcal{R}$-derivation so that its structure mirrors the graph of rule dependencies $G(\mathcal{R})$ (defined in Section 2). In other words, given a rule set $\mathcal{R}$ and an $\mathcal{R}$-derivation δ, we may permute all applications of rules that serve as sources in $G(\mathcal{R})$ (which do not depend on any rules in $\mathcal{R}$) to the beginning of δ, followed by all rule applications that depend only on sources, and so forth, with any applications of rules serving as sinks in $G(\mathcal{R})$ concluding the derivation. For example, in the graph of rule dependencies of $\mathcal{R}_2$, the rules ρ_1, ρ_2, and ρ_3 serve as source nodes (since they do not depend on any rules in $\mathcal{R}_2$) and the rule ρ_4 is a sink node depending on each of the aforementioned three rules, i.e. $G(\mathcal{R}_2) = (V, E)$ with $V = \{\rho_1, \rho_2, \rho_3, \rho_4\}$ and $E = \{(\rho_i, \rho_4) \mid 1 \leq i \leq 3\}$. Hence, in any given $\mathcal{R}_2$-derivation δ, any application of ρ_1, ρ_2, or ρ_3 can be permuted backward (toward the beginning of δ) and any application of ρ_4 can be permuted forward (toward the end of δ).

Beyond the use of rule permutations, we also transform $\mathcal{R}_2$-derivations by making use of rule replacements. In particular, observe that $head(\rho_3)$ and $body(\rho_3)$ correspond to conjunctions of $head(\rho_1)$ and $head(\rho_2)$, and $body(\rho_1)$ and $body(\rho_2)$, respectively. Thus, we can replace the first application of ρ_1 and the succeeding application of ρ_2 in δ_1' above by a single application of ρ_3, thus yielding the following $\mathcal{R}_2$-derivation:

$$\delta_1'' = \mathcal{D}_\dagger, (\rho_3, h, \mathcal{I}_1 \cup (\mathcal{I}_3 \setminus \mathcal{I}_2)), (\rho_1, h_2, \mathcal{I}_3), (\rho_4, h_4, \mathcal{I}_4),$$

where $h(x) = a$ and $h(y) = b$. Interestingly, if one inspects the above $\mathcal{R}_2$-derivation, they will find that it is identical to the greedy $\mathcal{R}_2$-derivation δ_2 defined earlier in the section, and so, we have shown how to take a non-greedy $\mathcal{R}_2$-derivation (viz. δ_1) and transform in into a greedy $\mathcal{R}_2$-derivation (viz. δ_2) by means of rule permutations and replacements. In the same fashion, one can prove in general that any non-greedy $\mathcal{R}_2$-derivation can be transformed into a greedy $\mathcal{R}_2$-derivation, thus giving rise to the following theorem, and demonstrating that $\mathcal{R}_2$ is indeed **wgbts**. For the interested reader, a rigorous proof can be found in the appendix.

Theorem 2. *$\mathcal{R}_2$ is* **wgbts***, but not* **gbts***, and thus,* **wgbts** *properly subsumes* **gbts**.

5 Derivation Graphs

We now discuss *derivation graphs* – a concept introduced by Baget et al. [5] and used to establish that certain classes of rule sets (e.g. weakly frontier guarded rule sets [6]) are **fts**. A derivation graph has the structure of a directed acyclic graph and encodes *how* atoms are derived throughout the course of an $\mathcal{R}$-derivation. By applying so-called *reduction operations*, a derivation graph may (under certain conditions) be transformed into a treelike graph that serves as a tree decomposition of an $\mathcal{R}$-derivable instance.

Below, we define derivation graphs and discuss how such graphs are transformed into tree decompositions by means of reduction operations. To increase comprehensibility, we provide an example of a derivation graph (shown in Figure 2) and give an example of applying each reduction operation (shown in Figure 3). After, we identify two (query-decidable) classes of rule sets on the basis of derivation graphs, namely, *cycle-free derivation graph sets* (**cdgs**) and *weakly cycle-free derivation graph sets* (**wcdgs**). Despite their prima facie distinctness, the **cdgs** and **wcdgs** classes coincide with **gbts** and **wgbts** classes, respectively, thus showing how the latter classes can be characterized in terms of derivation graphs. Let us now formally define derivation graphs, and after, we will demonstrate the concept by means of an example.

Definition 4 (Derivation Graph). *Let $\mathcal{D}$ be a database, $\mathcal{R}$ be a rule set, $C = \mathrm{Con}(\mathcal{D}, \mathcal{R})$, and δ be the $\mathcal{R}$-derivation $\mathcal{D}, (\rho_1, h_1, \mathcal{I}_1), \ldots, (\rho_n, h_n, \mathcal{I}_n)$. The* derivation graph of δ *is the tuple $G_\delta := (\mathrm{V}, \mathrm{E}, \mathrm{At}, \mathrm{L})$, where $\mathrm{V} := \{X_0, \ldots, X_n\}$ is a finite set of* nodes, $\mathrm{E} \subseteq \mathrm{V} \times \mathrm{V}$ *is a set of* arcs, *and the functions $\mathrm{At} : V \to \mathcal{I}_n$ and $\mathrm{L} : E \to 2^{\mathrm{Ter}(\mathcal{I}_n)}$ decorate nodes and arcs, respectively, such that:*

1. $\mathrm{At}(X_0) := \mathcal{D}$ *and* $\mathrm{At}(X_i) = \mathcal{I}_i \setminus \mathcal{I}_{i-1}$ *with* $\mathrm{Ter}(X_i) := \mathrm{Ter}(\mathrm{At}(X_i)) \cup C$*;*
2. $(X_i, X_j) \in \mathrm{E}$ *iff there is a $p(\mathbf{t}) \in \mathrm{At}(X_i)$ and a frontier atom $p(\mathbf{t}')$ in ρ_j such that $h_j(p(\mathbf{t}')) = p(\mathbf{t})$. We then set $\mathrm{L}(X_i, X_j) = \big(h_j\big(\mathrm{Ter}(p(\mathbf{t}')) \cap fr(\rho_j)\big)\big) \setminus C$.*

We refer to the node X_0 as the initial node*, and we define the set of* non-constant terms *associated with a node to be $\overline{C}(X) = \mathrm{Ter}(X) \setminus C$.*

To provide an example of a derivation graph, we let $\mathcal{D}_\ddagger = \{p(a, b)\}$ and $\mathcal{R}_3 = \{\rho_1, \rho_2, \rho_3, \rho_4\}$ where

$$
\begin{aligned}
\rho_1 &= p(x, y) \to \exists z.q(y, z)\\
\rho_2 &= q(x, y) \to \exists z.(r(x, y) \wedge r(y, z))\\
\rho_3 &= r(x, y) \wedge q(z, x) \to s(x, y)\\
\rho_4 &= r(x, y) \wedge s(z, w) \to t(y, w)
\end{aligned}
$$

Let us consider the following derivation:

$$\delta = \mathcal{D}_\ddagger, (\rho_1, h_1, \mathcal{I}_1), (\rho_2, h_2, \mathcal{I}_2), (\rho_3, h_3, \mathcal{I}_3), (\rho_4, h_4, \mathcal{I}_4)$$

where $h_1(x) = a$, $h_1(y) = b$, $h_2(x) = b$, $h_2(y) = z_0$, $h_3(x) = z_0$, $h_3(y) = z_1$, $h_3(z) = b$, $h_4(x) = b$, $h_4(y) = h_4(z) = z_0$, $h_4(w) = z_1$, and the instances participating in δ are as follows:

$$
\begin{aligned}
\mathcal{D}_\ddagger &= \{p(a, b)\} & \mathcal{I}_3 &= \mathcal{I}_2 \cup \{s(z_0, z_1)\}\\
\mathcal{I}_1 &= \mathcal{D}_\ddagger \cup \{q(b, z_0)\} & \mathcal{I}_4 &= \mathcal{I}_3 \cup \{t(z_0, z_1)\}\\
\mathcal{I}_2 &= \mathcal{I}_1 \cup \{r(b, z_0), r(z_0, z_1)\}
\end{aligned}
$$

The derivation graph $G_\delta = (\mathrm{V}, \mathrm{E}, \mathrm{At}, \mathrm{L})$ corresponding to δ is shown in Figure 2 and contains fives nodes; in particular, $\mathrm{V} = \{X_0, X_1, X_2, X_3, X_4\}$. Each node $X_i \in \mathrm{V}$ is associated with a set $\mathrm{At}(X_i)$ of atoms depicted in the associated circle (e.g. $\mathrm{At}(X_2) = \{r(a, z_0), r(z_0, z_1)\}$), and each arc $(X_i, X_j) \in \mathrm{E}$ is represented as a directed arrow with $\mathrm{L}(X_i, X_j)$ shown as the associated set of terms (e.g. $\mathrm{L}(X_3, X_4) = \{z_1\}$). For each node $X_i \in \mathrm{V}$, the set $\mathrm{Ter}(X_i)$ of terms associated with the node is equal to $\mathrm{Ter}(\mathrm{At}(X_i)) \cup \{a, b\}$ (e.g. $\mathrm{Ter}(X_3) = \{z_0, z_1, a, b\}$) since $C = \mathrm{Con}(\mathcal{D}_\ddagger, \mathcal{R}_3) = \{a, b\}$.

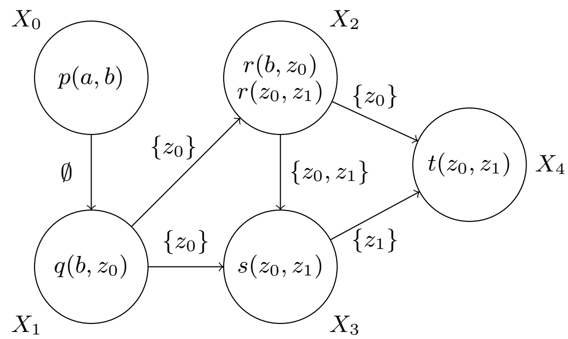

Fig. 2. The derivation graph G_δ.

As can be witnessed via the above example, derivation graphs satisfy a set of properties akin to those characterizing tree decompositions [5, Proposition 12].

Lemma 4 (Decomposition Properties). *Let $\mathcal{D}$ be a database, $\mathcal{R}$ be a rule set, and $C = \mathrm{Con}(\mathcal{D}, \mathcal{R})$. If $\mathcal{D} \xrightarrow{\delta}_{\mathcal{R}} \mathcal{I}$, then G_δ satisfies the following properties:*

1. $\bigcup_{X_n \in \mathrm{V}} \mathrm{Ter}(X_n) = \mathrm{Ter}(\mathcal{I})$;
2. *For each $p(\mathbf{t}) \in \mathcal{I}$, there is an $X_n \in \mathrm{V}$ such that $p(\mathbf{t}) \in \mathrm{At}(X_n)$;*
3. *For each term $x \in \overline{C}(\mathcal{I})$, the subgraph of G_δ induced by the nodes X_n such that $x \in \overline{C}(X_n)$ is connected;*
4. *For each $X_n \in \mathrm{V}$ the size of $\mathrm{Ter}(X_n)$ is bounded by an integer that only depends on the size of $(\mathcal{D}, \mathcal{R})$, viz.* $\max\{|\mathrm{Ter}(\mathcal{D})|, |\mathrm{Ter}(head(\rho_i))|_{\rho_i \in \mathcal{R}}\} + |C|$.

Let us now introduce our set of *reduction operations*. As remarked above, in certain circumstances such operations can be used to transform derivation graphs into tree decompositions of an instance.

We make use of three reduction operations, namely, (i) *arc removal*, denoted $(\mathsf{ar})^{[i,j]}$, (ii) *term removal*, denoted $(\mathsf{tr})^{[i,j,k,t]}$, and (iii) *cycle removal*, denoted $(\mathsf{cr})^{[i,j,k,\ell]}$. The first two reduction operations were already proposed by Baget et al. [5].[7] However, we have introduced cycle removal as a new operation as it will assist us in characterizing **gbts** and **wgbts** in terms of derivation graphs.

[7] Baget et al. [5] presented (tr) and (ar) as a single operation referred to as *redundant arc removal*.

Definition 5 (Reduction Operations). *Let $\mathcal{D}$ be a database, $\mathcal{R}$ be a rule set, $\mathcal{D} \xrightarrow{\delta}_{\mathcal{R}} \mathcal{I}_n$, and G_δ be the derivation graph of δ. We define the set* RO *of* reduction operations *as:*

$$\{(\mathsf{ar})^{[i,j]}, (\mathsf{tr})^{[i,j,k,t]}, (\mathsf{cr})^{[i,j,k,\ell]} \mid i,j,k,\ell \leq n,\ t \in \mathrm{Ter}(\mathcal{I}_n)\}$$

which are specified below, and let $(\mathsf{r})\Sigma(G_\delta)$ denote the output of applying the operation (r) to the (potentially reduced) derivation graph $\Sigma(G_\delta) = (\mathrm{V}, \mathrm{E}, \mathrm{At}, \mathrm{L})$, where $\Sigma \in \mathsf{RO}^*$ is a reduction sequence*, that is, Σ is a (potentially empty) sequence of reduction operations.*

1. *Arc Removal $(\mathsf{ar})^{[i,j]}$: Whenever $(X_i, X_j) \in \mathrm{E}$ and $\mathrm{L}(X_i, X_j) = \emptyset$, then $(\mathsf{ar})^{[i,j]}\Sigma(G_\delta) := (\mathrm{V}, \mathrm{E}', \mathrm{At}, \mathrm{L}')$ where $\mathrm{E}' := \mathrm{E} \setminus \{(X_i, X_j)\}$ and $\mathrm{L}' = \mathrm{L} \restriction \mathrm{E}'$.*
2. *Term Removal $(\mathsf{tr})^{[i,j,k,t]}$: If $(X_i, X_k), (X_j, X_k) \in \mathrm{E}$ with $X_i \neq X_j$ and $t \in \mathrm{L}(X_i, X_k) \cap \mathrm{L}(X_j, X_k)$, then $(\mathsf{tr})^{[i,j,k,t]}\Sigma(G_\delta) := (\mathrm{V}, \mathrm{E}, \mathrm{At}, \mathrm{L}')$ where L' is obtained from L by removing t from $\mathrm{L}(X_j, X_k)$.*
3. *Cycle Removal $(\mathsf{cr})^{[i,j,k,\ell]}$: If $(X_i, X_k), (X_j, X_k) \in \mathrm{E}$ and there exists a node $X_\ell \in \mathrm{V}$ with $\ell < k$ such that*

$$\mathrm{L}(X_i, X_k) \cup \mathrm{L}(X_j, X_k) \subseteq \mathrm{Ter}(X_\ell)$$

then, $(\mathsf{cr})^{[i,j,k,\ell]}\Sigma(G_\delta) := (\mathrm{V}, \mathrm{E}', \mathrm{At}, \mathrm{L}')$ where

$$\mathrm{E}' := \big(\mathrm{E} \setminus \{(X_i, X_k), (X_j, X_k)\}\big) \cup \{(X_\ell, X_k)\}$$

and L' is obtained from $\mathrm{L} \restriction \mathrm{E}'$ by setting $L(X_\ell, X_k)$ to $\mathrm{L}(X_i, X_k) \cup \mathrm{L}(X_j, X_k)$.

Last, we say that a reduction sequence $\Sigma \in \mathsf{RO}^*$ is a complete reduction sequence *relative to a derivation graph G_δ iff $\Sigma(G_\delta)$ is cycle-free.*

Remark 2. When there is no danger of confusion, we will take the liberty to write (tr), (ar), and (cr) without the superscript parameters. For instance, given a derivation graph G_δ, the (reduced) derivation graph $(\mathsf{cr})(\mathsf{tr})(G_\delta)$ is obtained by applying an instance of (tr) followed by an instance of (cr) to G_δ. When applying a reduction operation we always explain *how* it is applied, so the exact operation is known.

We now describe the functionality of each reduction operation and illustrate each by means of an example. We will apply each to transform the derivation graph G_δ (shown in Figure 2) into a tree decomposition of $\mathcal{I}_4$ (which was defined above). The (tr) operation deletes a term t within the intersection of the sets labeling two converging arcs. For example, we may apply (tr) to the derivation graph G_δ from Figure 2, deleting the term z_0 from the label of the arc (X_1, X_3), and yielding the reduced derivation graph $(\mathsf{tr})(G_\delta)$, which is shown first in Figure 3. We may then apply (ar) to $(\mathsf{tr})(G_\delta)$, deleting the arc (X_1, X_3), which is labeled with the empty set, to obtain the reduced derivation graph $(\mathsf{ar})(\mathsf{tr})(G_\delta)$ shown middle in Figure 3.

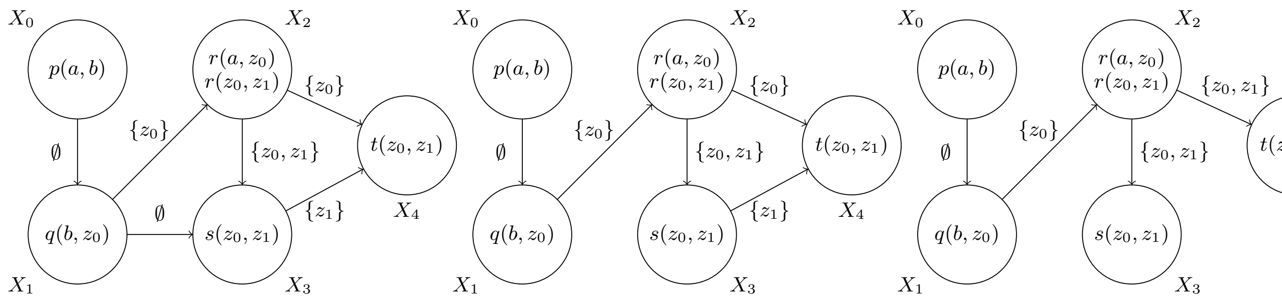

Fig. 3. Read from left-to-right: the three reduced derivation graphs are $(\mathsf{tr})(G_\delta)$, $(\mathsf{ar})(\mathsf{tr})(G_\delta)$, and $(\mathsf{cr})(\mathsf{ar})(\mathsf{tr})(G_\delta)$.

The (cr) operation is more complex and works by considering two converging arcs (X_i, X_k) and (X_j, X_k) in a (reduced) derivation graph. If there exists a node X_ℓ whose index ℓ is less than the index k of the child node X_k and $\mathrm{L}(X_i, X_k) \cup \mathrm{L}(X_j, X_k) \subseteq \mathrm{Ter}(X_\ell)$, then the converging arcs (X_i, X_k) and (X_j, X_k) may be deleted and the arc (X_ℓ, X_k) introduced and labeled with $\mathrm{L}(X_i, X_k) \cup \mathrm{L}(X_j, X_k)$. As an example, the reduced derivation graph $(\mathsf{cr})(\mathsf{ar})(\mathsf{tr})(G_\delta)$ (shown third in Figure 3) is obtained from $(\mathsf{ar})(\mathsf{tr})(G_\delta)$ (shown bottom-left in Figure 3) by applying (cr) in the following manner to the convergent arcs (X_2, X_4) and (X_3, X_4): since for X_2 (whose index 2 is less than the index 4 of X_4) $\mathrm{L}(X_2, X_4) \cup \mathrm{L}(X_3, X_4) \subseteq \mathrm{Ter}(X_2)$, we may delete the arcs (X_2, X_4) and (X_3, X_4) and introduce the arc (X_2, X_4) labeled with $\mathrm{L}(X_2, X_4) \cup \mathrm{L}(X_3, X_4) = \{z_0\} \cup \{z_1\} = \{z_0, z_1\}$. Observe that the reduced derivation graph $(\mathsf{cr})(\mathsf{ar})(\mathsf{tr})(G_\delta)$ is free of cycles, witnessing that $\Sigma = (\mathsf{cr})(\mathsf{ar})(\mathsf{tr})$ is a complete reduction sequence relative to G_δ. Moreover, if we replace each node by the set of its terms and disregard the labels on arcs, then $\Sigma(G_\delta)$ can be read as a tree decomposition of $\mathcal{I}_4$. In fact, one can show that every reduced derivation graph satisfies the decomposition properties mentioned in Lemma 4 above.

Lemma 5. *Let $\mathcal{D}$ be a database and $\mathcal{R}$ be a rule set. If $\mathcal{D} \xrightarrow{\mathcal{R}}_\delta \mathcal{I}$, then for any reduction sequence Σ, $\Sigma(G_\delta) = (\mathrm{V}, \mathrm{E}, \mathrm{At}, \mathrm{L})$ satisfies the decomposition properties 1-4 in Lemma 4.*

As illustrated above, derivation graphs can be used to derive tree decompositions of $\mathcal{R}$-derivable instances. By the fourth decomposition property (see Lemma 4 above), the width of such a tree decomposition is bounded by a constant that depends only on the given knowledge base. Thus, if a rule set $\mathcal{R}$ always yields derivation graphs that are reducible to *cycle-free* graphs – meaning that (un)directed cycles do not occur within the graph – then all $\mathcal{R}$-derivable instances have tree decompositions that are uniformly bounded by a constant. This establishes that the rule set $\mathcal{R}$ falls within the **pbts** class, and therefore, **fts** class, confirming that query entailment is decidable with $\mathcal{R}$. We define two classes of rule sets by means of reducible derivation graphs:

Definition 6 ((Weakly) Cycle-free Derivation Graph Set). *Let $\mathcal{R}$ be a rule set. $\mathcal{R}$ is a* cycle-free derivation graph set (**cdgs**) *iff for any database $\mathcal{D}$, if $\mathcal{D} \xrightarrow{\delta}_{\mathcal{R}} \mathcal{I}$, then G_δ can be reduced to a cycle-free graph via the reduction operations. $\mathcal{R}$ is a* weakly cycle-free derivation graph set (**wcdgs**) *iff for any database $\mathcal{D}$, if $\mathcal{D} \xrightarrow{\delta}_{\mathcal{R}} \mathcal{I}$, then there exists a derivation δ' such that $\mathcal{D} \xrightarrow{\delta'}_{\mathcal{R}} \mathcal{I}$ and $G_{\delta'}$ can be reduced to a cycle-free graph via the reduction operations.*

It is straightforward to confirm that **wcdgs** subsumes **cdgs**, and that both classes are subsumed by **pbts**.

Proposition 3. *Let $\mathcal{R}$ be a rule set.*

1. *If $\mathcal{R}$ is* **cdgs***, then $\mathcal{R}$ is* **wcdgs***;*
2. *If $\mathcal{R}$ is* **wcdgs***, then $\mathcal{R}$ is* **pbts**.

Furthermore, as mentioned above, **gbts** and **wgbts** coincide with **cdgs** and **wcdgs**, respectively. By making use of the (cr) operation, one can show that the derivation graph of any greedy derivation is reducible to a cycle-free graph, thus establishing that **gbts** $\subseteq$ **cdgs** and **wgbts** $\subseteq$ **wcdgs**. To show the converse (i.e. that **cdgs** $\subseteq$ **gbts** and **wcdgs** $\subseteq$ **wgbts**) however, requires more work. In essence, one shows that for every (non-source) node X_i in a cycle-free (reduced) derivation graph there exists another node X_j such that $j < i$ and the frontier of the atoms in $\mathrm{At}(X_i)$ only consist of constants and/or nulls introduced by the atoms in $\mathrm{At}(X_j)$. This property is preserved under *reverse* applications of the reduction operations, and thus, one can show that if a derivation graph is reducible to a cycle-free graph, then the above property holds for the original derivation graph, implying that the derivation graph encodes a greedy derivation. Based on such arguments, one can prove the following:

Theorem 3. *Let $\mathcal{R}$ be a rule set.*

1. *$\mathcal{R}$ is* **gbts** *iff $\mathcal{R}$ is* **cdgs***;*
2. *$\mathcal{R}$ is* **wgbts** *iff $\mathcal{R}$ is* **wcdgs**.

An interesting consequence of the above theorem concerns the redundancy of (ar) and (tr) in the presence of (cr). In particular, since we know that (i) if a derivation graph can be reduced to a cycle-free graph, then the derivation graph encodes a greedy derivation, and (ii) the derivation graph of any greedy derivation can be reduced to an cycle-free graph by means of applying the (cr) operation only, it follows that if a derivation graph can be reduced to a cycle-free graph, then it can be reduced by only applying the (cr) operation. We refer to this phenomenon as *reduction-admissibility*, which is defined below.

Definition 7 (Reduction-admissible). *We say that a reduction operation* (r) *is* reduction-admissible *iff for any rule set $\mathcal{R}$ and $\mathcal{R}$-derivation δ, if G_δ is reducible to a cycle-free graph with* (r)*, then G_δ is reducible to a cycle-free graph without* (r)*.*

Corollary 1. *Let $\mathcal{R}$ be a rule set.*

1. *The reduction operations* (tr) *and* (ar) *are reduction-admissible;*
2. *The* **wcdgs** *class properly contains the* **cdgs** *class;*
3. *If $\mathcal{R}$ is* **cdgs***,* **gbts***,* **wcdgs***, or* **wgbts***, then BCQ entailment is decidable.*

6 Conclusion

tim: TODO

References

1. Baget, J.F.: Improving the forward chaining algorithm for conceptual graphs rules. In: Proceedings of the Ninth International Conference on Principles of Knowledge Representation and Reasoning. p. 407–414. KR'04, AAAI Press (2004)
2. Baget, J.F., Leclère, M., Mugnier, M.L., Salvat, E.: Extending decidable cases for rules with existential variables. In: Proceedings of the 21st International Jont Conference on Artifical Intelligence. p. 677–682. IJCAI'09, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2009)
3. Baget, J.F., Leclère, M., Mugnier, M.L., Salvat, E.: Extending decidable cases for rules with existential variables. In: Proceedings of the 21st International Joint Conference on Artificial Intelligence. p. 677–682. IJCAI'09, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2009)
4. Baget, J.F., Mugnier, M.L.: Extensions of simple conceptual graphs: the complexity of rules and constraints. Journal of Artificial Intelligence Research **16**, 425–465 (2002)
5. Baget, J.F., Leclère, M., Mugnier, M.L., Salvat, E.: On rules with existential variables: Walking the decidability line. Artificial Intelligence **175**(9), 1620–1654 (2011). https://doi.org/https://doi.org/10.1016/j.artint.2011.03.002, https://www.sciencedirect.com/science/article/pii/S0004370211000397
6. Calì, A., Gottlob, G., Kifer, M.: Taming the infinite chase: Query answering under expressive relational constraints. Journal of Artificial Intelligence Research **48**, 115–174 (2013). https://doi.org/10.1613/jair.3873, https://doi.org/10.1613/jair.3873
7. Chandra, A.K., Lewis, H.R., Makowsky, J.A.: Embedded implicational dependencies and their inference problem. In: Proceedings of the 13th Annual ACM Symposium on Theory of Computing (STOC'81). pp. 342–354. ACM (1981). https://doi.org/10.1145/800076.802488, https://doi.org/10.1145/800076.802488
8. Deutsch, A., Nash, A., Remmel, J.B.: The chase revisited. In: Lenzerini, M., Lembo, D. (eds.) Proceedings of the 27th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS'08). pp. 149–158. ACM (2008). https://doi.org/10.1145/1376916.1376938, https://doi.org/10.1145/1376916.1376938
9. Fagin, R., Kolaitis, P.G., Miller, R.J., Popa, L.: Data exchange: semantics and query answering. Theoretical Computer Science **336**(1), 89–124 (2005). https://doi.org/https://doi.org/10.1016/j.tcs.2004.10.033, https://www.sciencedirect.com/science/article/pii/S030439750400725X, database Theory
10. Feller, T., Lyon, T.S., Ostropolski-Nalewaja, P., Rudolph, S.: Finite-Cliquewidth Sets of Existential Rules: Toward a General Criterion for Decidable yet Highly Expressive Querying. In: Geerts, F., Vandevoort, B. (eds.) 26th International Conference on Database Theory (ICDT 2023). Leibniz International Proceedings in Informatics (LIPIcs), vol. 255, pp. 18:1–18:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany (2023). https://doi.org/10.4230/LIPIcs.ICDT.2023.18, https://drops.dagstuhl.de/opus/volltexte/2023/17760
11. Johnson, D.S., Klug, A.: Testing containment of conjunctive queries under functional and inclusion dependencies. In: Proceedings of the 1st ACM SIGACT-SIGMOD Symposium on Principles of Database Systems. p. 164–169. PODS '82, Association for Computing Machinery, New York, NY, USA (1982). https://doi.org/10.1145/588111.588138, https://doi.org/10.1145/588111.588138

12. Maier, D., Mendelzon, A.O., Sagiv, Y.: Testing implications of data dependencies. ACM Trans. Database Syst. **4**(4), 455–469 (dec 1979). https://doi.org/10.1145/320107.320115, https://doi.org/10.1145/320107.320115
13. Thomas, R.: The tree-width compactness theorem for hypergraphs (1988), https://thomas.math.gatech.edu/PAP/twcpt.pdf, unpublished
14. Thomazo, M., Baget, J.F., Mugnier, M.L., Rudolph, S.: A generic querying algorithm for greedy sets of existential rules. In: Brewka, G., Eiter, T., McIlraith, S.A. (eds.) Proceedings of the 13th International Conference on Principles of Knowledge Representation and Reasoning (KR'12). AAAI (2012), http://www.aaai.org/ocs/index.php/KR/KR12/paper/view/4542

A Proofs for Section 3

Proposition 1 *Let $\mathcal{R}$ be a rule set.*

1. *If $\mathcal{R}$ is* **fes***, then $\mathcal{R}$ is* **fts***;*
2. *if $\mathcal{R}$ is* **pbts***, then $\mathcal{R}$ is* **fts***.*

Proof. We argue both claims in turn:

1. If $\mathcal{R}$ is **fes**, then for any database $\mathcal{D}$, $(\mathcal{D}, \mathcal{R})$ has finite universal model $\mathcal{I}^*$ of size n. Hence, the width of each tree decomposition of $\mathcal{I}^*$ is at most n, showing that $(\mathcal{D}, \mathcal{R})$ has a universal model $\mathcal{I}^*$ such that $tw(\mathcal{I}^*) \leq n$.
2. If $\mathcal{R}$ is **pbts**, then we know that for every database $\mathcal{D}$, there exists an $n \in \mathbb{N}$ such that for every $k \in \mathbb{N}$, $tw(\mathbf{Ch}_k(\mathcal{D}, \mathcal{R})) \leq n$. Let $\mathcal{D}$ be an arbitrary database. As $\mathbf{Ch}_k(\mathcal{D}, \mathcal{R})$ is finite for every $k \in \mathbb{N}$ and monotonically increases (relative to the subset relation) as k increases, we have that for every finite subset of $\mathbf{Ch}_\infty(\mathcal{D}, \mathcal{R})$, the treewidth of that subset is bounded by n. Thus, by the the treewidth compactness theorem [13], $tw(\mathbf{Ch}_\infty(\mathcal{D}, \mathcal{R})) \leq n$. Since $\mathbf{Ch}_\infty(\mathcal{D}, \mathcal{R})$ is a universal model of $(\mathcal{D}, \mathcal{R})$, it follows that $(\mathcal{D}, \mathcal{R})$ has a universal model of finite treewidth. Last, since $\mathcal{D}$ was assumed arbitrary, we have that $\mathcal{R}$ is **fts**.

Lemma 2 *$\mathcal{R}_1$ is* **fes***.*

Proof. Let $\mathcal{D}$ be a database, and let $\mathcal{I}^*$ be the instance obtained by closing $\mathcal{D}$ under the rule set $\mathcal{R}'_1 = \{\rho'_1, \rho'_2\}$ where $\rho'_1 = r(x,y) \land r(y,z) \to r(x,z)$ and $\rho'_2 = r(x,y) \to r(x,x) \land r(y,y)$. We know that $\mathcal{I}^*$ will be finite since neither of the above rules introduce new terms and only add edges to $\mathcal{D}$, of which only finitely many can be added. We now argue first that $\mathcal{I}^*$ is a model of $\mathcal{R}_1$, and second, that $\mathcal{I}^*$ is a universal model of $\mathcal{R}_1$.

To establish modelhood, first let us define h to be the identity map, mapping each constant in $\mathcal{D}$ to itself. We will argue that each rule in $\mathcal{R}_1$ is satisfied on $\mathcal{I}^*$. For the rule ρ_1, suppose that $r(h(x), h(y)) \in \mathcal{I}^*$. Then, since $\mathcal{I}^*$ is closed under applications of ρ'_2, we know that $r(h(y), h(y)) \in \mathcal{I}^*$, thus showing that $\mathcal{I}^* \models \exists z r(h(y), z)$, and confirming that ρ_1 is indeed satisfied on $\mathcal{I}^*$. The satisfaction of the rules ρ_2 and ρ_3 follow immediately from the satisfaction of ρ'_1 and ρ'_2, respectively.

To establish that $\mathcal{I}^*$ is a universal model of $(\mathcal{D}, \mathcal{R}_1)$, we show that $\mathcal{I}^*$ can be homomorphically mapped into $\mathbf{Ch}_\infty(\mathcal{D}, \mathcal{R}_1)$ via the homomorphism h (defined above). First, we show that (a) for any $p(a_1, \ldots, a_n) \in \mathcal{I}^*$ with $p \neq r$, $p(h(a_1), \ldots, h(a_n)) \in \mathbf{Ch}_\infty(\mathcal{D}, \mathcal{R}_1)$, and second, we show that (b) if $r(a, b) \in \mathcal{I}^*$, then $r(h(a), h(b)) \in \mathbf{Ch}_\infty(\mathcal{D}, \mathcal{R}_1)$. As each constant of $\mathcal{I}^*$ is mapped to itself by the definition of h, we have that h is a homomorphism from $\mathcal{I}^*$ to $\mathbf{Ch}_\infty(\mathcal{D}, \mathcal{R}_1)$. It is immediate that (a) holds since if $p \neq r$, then $p(a_1, \ldots, a_n) \in \mathcal{D}$, implying that $p(h(a_1), \ldots, h(a_n)) \in \mathbf{Ch}_\infty(\mathcal{D}, \mathcal{R}_1)$ since $\mathcal{D} \subseteq \mathbf{Ch}_\infty(\mathcal{D}, \mathcal{R}_1)$ and h is the identity function on the constants of $\mathcal{D}$.

To show (b), first recall that $\mathcal{I}^*$ is the instance obtained by closing $\mathcal{D}$ under $\mathcal{R}'_1$. It follows that $\mathcal{I}^*$ homomorphically maps into $\mathbf{Ch}_\infty(\mathcal{D}, \mathcal{R}'_1)$. Thus, if we can show that $\mathbf{Ch}_\infty(\mathcal{D}, \mathcal{R}'_1)$ homomorphically maps into $\mathbf{Ch}_\infty(\mathcal{D}, \mathcal{R}_1)$, then we have established that $\mathcal{I}^*$ is a universal model of $(\mathcal{D}, \mathcal{R}_1)$. To show this, we argue by induction on k that if $r(t, t') \in \mathbf{Ch}_k(\mathcal{D}, \mathcal{R}'_1)$, then $r(h(t), h(t')) \in \mathbf{Ch}_\infty(\mathcal{D}, \mathcal{R}_1)$, which implies the desired result.

Base case. Suppose that $k = 0$, i.e. $r(t, t') \in \mathcal{D}$, meaning that $r(t, t')$ is of the form $r(a, b)$ since databases only contain ground atoms. Then, since $\mathcal{D} \subseteq \mathbf{Ch}_\infty(\mathcal{D}, \mathcal{R}_1)$, we have that $r(h(a), h(b)) \in \mathbf{Ch}_\infty(\mathcal{D}, \mathcal{R}_1)$ as $h(a) = a$ and $h(b) = b$.

Inductive step. For the inductive hypothesis, let us suppose that $r(t, t') \in \mathbf{Ch}_{k+1}(\mathcal{D}, \mathcal{R}'_1)$, which, since none of the rules in $\mathcal{R}'_1$ introduce nulls, means that $r(t, t')$ is of the form $r(a, b)$. If $r(a, b) \in \mathbf{Ch}_k(\mathcal{D}, \mathcal{R}'_1)$, then by IH, $r(h(a), h(b)) \in \mathbf{Ch}_\infty(\mathcal{D}, \mathcal{R}_1)$, and we are done; therefore, we assume that $r(a, b) \notin \mathbf{Ch}_k(\mathcal{D}, \mathcal{R}'_1)$. We have two cases to consider: either (i) $r(a, b)$ was introduced to $\mathbf{Ch}_{k+1}(\mathcal{D}, \mathcal{R}'_1)$ by means of ρ'_1 via a homomorphism h_1 or (ii) by means of ρ'_2 via a homomorphism h_2.

In case (i), we know that $r(a, h_1(y)), r(h_1(y), b) \in \mathbf{Ch}_k(\mathcal{D}, \mathcal{R}'_1)$, implying that

$$r(h(a), h(h_1(y))), r(h(h_1(y)), h(b)) \in \mathbf{Ch}_\infty(\mathcal{D}, \mathcal{R}_1)$$

by IH. Let n be the smallest natural number such that

$$r(h(a), h(h_1(y))), r(h(h_1(y)), h(b)) \in \mathbf{Ch}_n(\mathcal{D}, \mathcal{R}_1).$$

Observe that

$$r(h(h_1(a)), h(h_1(y))) = r(h(a), h(h_1(y))) \text{ and}$$
$$r(h(h_1(y)), h(h_1(b))) = r(h(h_1(y)), h(b))$$

since h_1 fixes constants by definition. We know that ρ_2 will be applied at step $n+1$ of the chase via a homormorphism h_3 with $h_3(x) = a$, $h_3(y) = h(h_1(y))$, and $h_3(z) = b$, entailing $r(h(a), h(b)) = r(h_3(a), h_3(b)) \in \mathbf{Ch}_{n+1}(\mathcal{D}, \mathcal{R}_1)$, with $r(h(a), h(b)) = r(h_3(a), h_3(b))$ because $h(a) = a = h_3(a)$ and $h(b) = b = h_3(b)$.

For case (ii), since $r(a, b)$ was introduced by means of ρ'_2 to $\mathbf{Ch}_{k+1}(\mathcal{D}, \mathcal{R}'_1)$ via a homomorphism h_2, we may assume that $r(a, b)$ is of the form $r(c, c)$ as ρ'_2 introduces loops. Therefore, either there is an atom $r(c, h_2(y)) \in \mathbf{Ch}_k(\mathcal{D}, \mathcal{R}'_1)$ or an atom $r(h_2(x), c) \in \mathbf{Ch}_k(\mathcal{D}, \mathcal{R}'_1)$. By IH, either $r(h(h_2(c)), h(h_2(y))) = r(h(c), h(h_2(y))) \in \mathbf{Ch}_\infty(\mathcal{D}, \mathcal{R}_1)$ or $r(h(h_2(x)), h(c)) = r(h(h_2(x)), h(h_1(c))) \in$

$\mathbf{Ch}_\infty(\mathcal{D}, \mathcal{R}_1)$. Let n be the smallest natural number such that $r(h(c), h(h_2(y))) \in \mathbf{Ch}_n(\mathcal{D}, \mathcal{R}_1)$ or $r(h(h_2(x)), h(c)) \in \mathbf{Ch}_n(\mathcal{D}, \mathcal{R}_1)$. In the former case, ρ_3 will be applied at step $n+1$ via a homomorphism h_3 with $h_3(x) = c$ and $h_3(y) = h(h_2(y))$, ensuring that $r(h(c), h(c)) = r(c, c) = r(h_3(c), h_3(c)) \in \mathbf{Ch}_{n+1}(\mathcal{D}, \mathcal{R}_1) \subseteq \mathbf{Ch}_\infty(\mathcal{D}, \mathcal{R}_1)$. In the latter case, ρ_1 will be applied at step $n+1$ via a homomorphism h_4 with $h_4(x) = h(h_2(x))$ and $h_4(y) = c$, ensuring that $r(h_4(c), z) \in \mathbf{Ch}_{n+1}(\mathcal{D}, \mathcal{R}_1)$ with z a fresh null. Similar to the former case, at step $n+2$ the rule ρ_3 will be applied via a homomorphism h_5 with $h_5(x) = c$ and $h_5(y) = z$, implying $r(h(c), h(c)) = r(c, c) = r(h_5(c), h_5(c)) \in \mathbf{Ch}_{n+2}(\mathcal{D}, \mathcal{R}_1) \subseteq \mathbf{Ch}_\infty(\mathcal{D}, \mathcal{R}_1)$.

B Proofs for Section 4

Proposition 2 *Let $\mathcal{R}$ be a rule set. If $\mathcal{R}$ is* **gbts**, *then $\mathcal{R}$ is* **wgbts**.

Proof. Let $\mathcal{D}$ be a database and $\mathcal{R}$ is be a **gbts** rule set. If $\mathcal{D} \xrightarrow{\delta}_{\mathcal{R}} \mathcal{I}$, then δ is greedy as $\mathcal{R}$ is **gbts**. Hence, there exists a greedy $\mathcal{R}$-derivation (viz. δ) of $\mathcal{I}$ from $\mathcal{D}$, showing that $\mathcal{R}$ is **wgbts** as well.

Lemma 3 (Permutation Lemma) *Let $\mathcal{R}$ be a rule set with $\mathcal{I}_0$ an instance. Suppose we have a (greedy) $\mathcal{R}$-derivation of the following form:*

$$\delta = \mathcal{I}_0, \ldots, (\rho_i, h_i, \mathcal{I}_i), (\rho_{i+1}, h_{i+1}, \mathcal{I}_{i+1}), \ldots, (\rho_n, h_n, \mathcal{I}_n)$$

If ρ_{i+1} does not depend on ρ_i, then the following is a (greedy) $\mathcal{R}$-derivation as well:

$$\delta' := \mathcal{I}_0, \ldots, (\rho_{i+1}, h_{i+1}, \mathcal{I}'_i), (\rho_i, h_i, \mathcal{I}_{i+1}), \ldots, (\rho_n, h_n, \mathcal{I}_n)$$

where $\mathcal{I}'_i = \mathcal{I}_{i-1} \cup (\mathcal{I}_{i+1} \setminus \mathcal{I}_i)$.

Proof. By assumption, ρ_{i+1} does not depend on ρ_i, implying $h_{i+1}(\mathit{body}(\rho_{i+1})) \subseteq \mathcal{I}_{i-1}$. Hence, we may apply ρ_{i+1} with h_{i+1} directly to $\mathcal{I}_{i-1}$ yielding the instance $\mathcal{I}'_i = \mathcal{I}_{i-1} \cup (\mathcal{I}_{i+1} \setminus \mathcal{I}_i)$. Since $h_i(\mathit{body}(\rho_i)) \subseteq \mathcal{I}_{i-1} \subseteq \mathcal{I}'_i$, we may apply ρ_i directly after ρ_{i+1} yielding the instance $\mathcal{I}_{i+1}$. Moreover, if δ is greedy, then (i) $h_{i+1}(\mathit{fr}(\rho_{i+1})) \subseteq \mathrm{Nul}(\overline{h}_j(\mathit{head}(\rho_j))) \cup \mathrm{Con}(\mathcal{I}_0, \mathcal{R}) \cup \mathrm{Nul}(\mathcal{I}_0)$ for some $j < i+1$, and (ii) $h_i(\mathit{fr}(\rho_i)) \subseteq \mathrm{Nul}(\overline{h}_k(\mathit{head}(\rho_k))) \cup \mathrm{Con}(\mathcal{I}_0, \mathcal{R}) \cup \mathrm{Nul}(\mathcal{I}_0)$ for some $k < i$. As ρ_{i+1} does not depend on ρ_i, it must be the case that $j \neq i$, and so, we have that δ' will be greedy as well since (i) and (ii) will hold for $j, k < i$ in δ'.

Theorem 2 *$\mathcal{R}_2$ is* **wgbts**, *but not* **gbts**, *and thus,* **wgbts** *properly subsumes* **gbts**.

Proof. We know that **wgbts** subsumes **gbts** by Lemma 2, however, to show that **wgbts** properly subsumes **gbts**, we prove that $\mathcal{R}_2$ is **wgbts**, but not **gbts**. Therefore, let $\mathcal{D}$ be an arbitrary database and $\mathcal{I}$ be an instance such that there exists an $\mathcal{R}_2$-derivation δ_0 of $\mathcal{I}$ from $\mathcal{D}_\dagger$. We show by induction on the length of δ_0 that a greedy $\mathcal{R}_2$-derivation of $\mathcal{I}$ from $\mathcal{D}$ can always be found.

Base case. Any $\mathcal{R}_2$-derivation of an instance $\mathcal{I}$ from $\mathcal{D}$ of length $n = 0$ or $n = 1$ is trivially greedy by Definition 2.

Inductive step. Suppose our derivation δ_0 is of length $n+1$, that is

$$\delta_0 = \mathcal{D}, (\rho_1, h_1, \mathcal{I}_1), \ldots, (\rho_n, h_n, \mathcal{I}_n), (\rho_{n+1}, h_{n+1}, \mathcal{I}_{n+1})$$

By IH, we have that a greedy $\mathcal{R}_2$-derivation δ_1 of $\mathcal{I}_n$ exists; hence, let $\delta_2 = \delta_1, (\rho_{n+1}, h_{n+1}, \mathcal{I}_{n+1})$ and observe that δ_2 is a valid $\mathcal{R}_2$-derivation as we already know by the structure of δ_0 above that ρ_{n+1} is triggered in $\mathcal{I}_n$ with the homomorphism h_{n+1}. If the last rule ρ_{n+1} applied in δ_2 is ρ_1, ρ_2, or ρ_3, then since no such rule depends on any rule in $\mathcal{R}_2$, it must be the case $h_{n+1}(head(\rho_{n+1})) \subseteq \mathcal{D}$, showing that δ_2 is greedy. Therefore, let us assume that the last rule ρ_{n+1} applied is ρ_4. Recall that $body(\rho_4) = \{q(x,y,z), s(w,u,v)\}$, and observe that if ρ_4 is applied, then $h_{n+1}(q(x,y,z)), h_{n+1}(s(w,u,v)) \in \mathcal{I}_n$. We make a case distinction depending on the how $h_{n+1}(q(x,y,z))$ and $h_{n+1}(s(w,u,v))$ entered into the derivation δ_1 below:

1. Suppose that $h_{n+1}(q(x,y,z)), h_{n+1}(s(w,u,v)) \in \mathcal{D}$. Then, δ_2 is greedy since

$$\begin{aligned} h_{n+1}(fr(\rho_{n+1})) &\subseteq \mathrm{Con}(\mathcal{D}) \\ &\subseteq \mathrm{Nul}(\overline{h}_j(head(\rho_j))) \cup \mathrm{Con}(\mathcal{D}, \mathcal{R}) \end{aligned}$$

   for any $j < n+1$.
2. Suppose that $h_{n+1}(q(x,y,z)) \in \mathcal{D}$ and $h_{n+1}(s(w,u,v))$ was introduced by an application of ρ_2 or ρ_3 at $j < n+1$ (i.e. $\rho_j \in \{\rho_2, \rho_3\}$). Then, δ_2 is greedy since

$$\begin{aligned} h_{n+1}(fr(\rho_{n+1})) &\subseteq \mathrm{Nul}(\overline{h}_j(head(\rho_j))) \cup \mathrm{Con}(\mathcal{D}) \\ &\subseteq \mathrm{Nul}(\overline{h}_j(head(\rho_j))) \cup \mathrm{Con}(\mathcal{D}, \mathcal{R}). \end{aligned}$$

3. Suppose that $h_{n+1}(q(x,y,z))$ was introduced by an application of ρ_1 or ρ_3 at $j < n+1$ (i.e. $\rho_j \in \{\rho_1, \rho_3\}$) and $h_{n+1}(s(w,u,v)) \in \mathcal{D}$. Then, δ_2 is greedy since

$$\begin{aligned} h_{n+1}(fr(\rho_{n+1})) &\subseteq \mathrm{Nul}(\overline{h}_j(head(\rho_j))) \cup \mathrm{Con}(\mathcal{D}) \\ &\subseteq \mathrm{Nul}(\overline{h}_j(head(\rho_j))) \cup \mathrm{Con}(\mathcal{D}, \mathcal{R}). \end{aligned}$$

4. Suppose that $h_{n+1}(q(x,y,z))$ and $h_{n+1}(s(w,u,v))$ were introduced by a single application of ρ_3 at $j < n+1$ (i.e. $\rho_j = \rho_3$). Then, δ_2 is greedy since

$$\begin{aligned} h_{n+1}(fr(\rho_{n+1})) &\subseteq \mathrm{Nul}(\overline{h}_j(head(\rho_3))) \cup \mathrm{Con}(\mathcal{D}) \\ &\subseteq \mathrm{Nul}(\overline{h}_j(head(\rho_j))) \cup \mathrm{Con}(\mathcal{D}, \mathcal{R}). \end{aligned}$$

5. Suppose that $h_{n+1}(q(x,y,z))$ was introduced by an application of $\rho_j \in \{\rho_1, \rho_3\}$ and $h_{n+1}(s(w,u,v))$ was introduced by an application of $\rho_k \in \{\rho_2, \rho_3\}$ with $j, k < n+1$. We assume that if ρ_3 introduced both $h_{n+1}(q(x,y,z))$ and $h_{n+1}(s(w,u,v))$, then both applications of ρ_3 are distinct, and we assume w.l.o.g. that $j < k$. Since ρ_k only depends on the database $\mathcal{D}$, we may repeatedly apply the permutation lemma (Lemma 3) to δ_2, permuting the

application of ρ_k earlier in the derivation until we reach the application of ρ_j, yielding:

$$\delta_3 = \mathcal{D}, \dots, (\rho_j, h_j, \mathcal{I}_j), (\rho_k, h_k, \mathcal{I}'_{j+1}), \dots, (\rho_{n+1}, h_{n+1}, \mathcal{I}_{n+1})$$

where $\mathcal{I}'_{j+1} = \mathcal{I}_j \cup (\mathcal{I}_k \setminus \mathcal{I}_{k-1})$. By the permutation lemma, we know that the portion of δ'_3 up to and including $(\rho_n, h_n, \mathcal{I}_n)$ is greedy. We have four cases to consider, and in each case, we show how to transform δ_3 into a greedy derivation δ'_3 of the same conclusion.

(a) If $\rho_j = \rho_1$ and $\rho_k = \rho_2$, then replace $(\rho_j, h_j, \mathcal{I}_j), (\rho_k, h_k, \mathcal{I}'_{j+1})$ in δ_3 with $(\rho_3, h', \mathcal{I}'_{j+1})$ where $h'(p(x)) = h_j(\mathit{body}(\rho_j))$, $h'(r(y)) = h_j(\mathit{body}(\rho_k))$, and $\overline{h'}(\mathit{head}(\rho_3)) = \overline{h}_j(q(x,y,z)) \wedge \overline{h}_k(s(x,y,z))$. This gives the derivation:

$$\delta'_3 = \mathcal{D}, \dots, (\rho_3, h', \mathcal{I}'_{j+1}), \dots, (\rho_{n+1}, h_{n+1}, \mathcal{I}_{n+1})$$

One can confirm that δ'_3 is indeed a valid derivation as $h'(\mathit{body}(\rho_3)) \in \mathcal{D}$, showing that ρ_3 may be applied where it is. Also,

$$\mathcal{I}'_{j+1} = \mathcal{I}_{j-1} \cup \{\overline{h}_j(q(x,y,z)), \overline{h}_k(s(x,y,z))\} = \mathcal{I}_{j-1} \cup \{\overline{h'}(\mathit{head}(\rho_3))\},$$

showing that $\mathcal{I}'_{j+1}$ is indeed derived by applying ρ_3, and for any application of a rule ρ_m with $j < m$ (i.e. for any application of a rule occurring after the application of ρ_3 displayed in δ'_3 above) if it previously depended on ρ_j or ρ_k, it will now depend on the above application of ρ_3, which introduces the same atoms as ρ_j and ρ_k. This also shows that the portion of δ'_3 up to and including $(\rho_n, h_n, \mathcal{I}_n)$ is greedy. Last, it follows that $\rho_{n+1} = \rho_4$ now depends on the above application of ρ_3, showing that

$$h_{n+1}(\mathit{fr}(\rho_{n+1})) \subseteq \mathrm{Nul}(\overline{h'}(\mathit{head}(\rho_3))) \cup \mathrm{Con}(\mathcal{D}, \mathcal{R}),$$

and hence, δ'_3 is greedy.

(b) If $\rho_j = \rho_1$ and $\rho_k = \rho_3$, then replace $(\rho_j, h_j, \mathcal{I}_j)$ in δ_3 with $(\rho_3, h', \mathcal{I}'_j)$ where $h'(p(x)) = h_j(p(x))$, $h'(r(y)) = h_k(r(x))$, $\overline{h'}(\mathit{head}(\rho_3)) = \overline{h}_j(q(x,y,z)) \wedge \overline{h}_k(s(x,y,z))$, and

$$\mathcal{I}'_j = \mathcal{I}_{j-1} \cup \{\overline{h}_j(q(x,y,z)), \overline{h}_k(s(x,y,z))\} = \mathcal{I}'_{j+1}.$$

Thus, we have the derivation:

$$\delta'_3 = \mathcal{D}, \dots, (\rho_3, h', \mathcal{I}'_j), (\rho_k, h_k, \mathcal{I}'_{j+1}), \dots, (\rho_{n+1}, h_{n+1}, \mathcal{I}_{n+1})$$

It is straightforward to confirm that δ'_3 is indeed a valid derivation, and furthermore, for any rule ρ_m with $j < m < n+1$, if it depended on ρ_j, it will now depend on the above application of ρ_3, showing that for any such m we have

$$h_m(\mathit{fr}(\rho_m)) \subseteq \mathrm{Nul}(\overline{h}_j(\mathit{head}(\rho_1))) \cup \mathrm{Con}(\mathcal{D}, \mathcal{R}) \subseteq \mathrm{Nul}(\overline{h'}(\mathit{head}(\rho_3))) \cup \mathrm{Con}(\mathcal{D}, \mathcal{R}).$$

Moreover, $\rho_{n+1} = \rho_4$ can be seen to depend on the application of ρ_3 displayed in δ_3' above, that is to say

$$h_{n+1}(fr(\rho_{n+1})) \subseteq \mathrm{Nul}(\overline{h'}(head(\rho_3))) \cup \mathrm{Con}(\mathcal{D}, \mathcal{R}).$$

Hence, it follows that δ_3' is greedy.

(c) If $\rho_j = \rho_3$ and $\rho_k = \rho_2$, then replace $(\rho_k, h_k, \mathcal{I}_k)$ in δ_3 with $(\rho_3, h', \mathcal{I}_{j+1}')$ where $h'(p(x)) = h_j(p(x))$, $h'(r(y)) = h_k(r(x))$, $\overline{h'}(head(\rho_3)) = \overline{h}_j(q(x,y,z)) \wedge \overline{h}_k(s(x,y,z))$, and

$$\mathcal{I}_{j+1}' = \mathcal{I}_j \cup \{\overline{h}_j(q(x,y,z)), \overline{h}_k(s(x,y,z))\}.$$

Thus, we have the derivation:

$$\delta_3' = \mathcal{D}, \ldots, (\rho_j, h_j, \mathcal{I}_j), (\rho_3, h', \mathcal{I}_{j+1}'), \ldots, (\rho_{n+1}, h_{n+1}, \mathcal{I}_{n+1})$$

It is straightforward to confirm that δ_3' is indeed a valid derivation, and furthermore, for any rule ρ_m with $k < m < n+1$, if it depended on ρ_k, it will now depend on the above application of ρ_3, showing that for any such m we have

$$h_m(fr(\rho_m)) \subseteq \mathrm{Nul}(\overline{h}_k(head(\rho_2))) \cup \mathrm{Con}(\mathcal{D}, \mathcal{R}) \subseteq \mathrm{Nul}(\overline{h'}(head(\rho_3))) \cup \mathrm{Con}(\mathcal{D}, \mathcal{R}).$$

Additionally, $\rho_{n+1} = \rho_4$ can be seen to depend on the application of ρ_3 displayed in δ_3' above, that is to say

$$h_{n+1}(fr(\rho_{n+1})) \subseteq \mathrm{Nul}(\overline{h'}(head(\rho_3))) \cup \mathrm{Con}(\mathcal{D}, \mathcal{R}).$$

Hence, it follows that δ_3' is greedy.

(d) If $\rho_j = \rho_3$ and $\rho_k = \rho_3$, then add $(\rho_3, h', \mathcal{I}_{j+1}')$ after $(\rho_j, h_j, \mathcal{I}_j), (\rho_k, h_k, \mathcal{I}_{j+1}')$ in δ_3 where $h'(p(x)) = h_j(p(x))$, $h'(r(y)) = h_k(r(y))$, and $\overline{h'}(head(\rho_3)) = \overline{h}_j(q(x,y,z)) \wedge \overline{h}_k(s(x,y,z))$. Thus, we have the derivation:

$$\delta_3' = \mathcal{D}, \ldots, (\rho_j, h_j, \mathcal{I}_j), (\rho_k, h_k, \mathcal{I}_{j+1}'), (\rho_3, h', \mathcal{I}_{j+1}'), \ldots, (\rho_{n+1}, h_{n+1}, \mathcal{I}_{n+1})$$

It is straightforward to confirm that δ_3' is indeed a valid derivation. Also, observe

$$h'(fr(\rho_3)) \subseteq \mathrm{Con}(\mathcal{D}, \mathcal{R}) \subseteq \mathrm{Nul}(\overline{h}_l(head(\rho_l))) \cup \mathrm{Con}(\mathcal{D}, \mathcal{R}).$$

for any $l \leq k$. Moreover, for every rule ρ_m with $k < m < n+1$, if

$$h_m(fr(\rho_m)) \subseteq \mathrm{Nul}(\overline{h}_{m'}(head(\rho_{m'}))) \cup \mathrm{Con}(\mathcal{D}, \mathcal{R}).$$

held in δ_3 with $m' < m$, then it will continue to hold in δ_3'. Last, $\rho_{n+1} = \rho_4$ can be seen to depend on the application of ρ_3 displayed in δ_3' above, that is to say

$$h_{n+1}(fr(\rho_{n+1})) \subseteq \mathrm{Nul}(\overline{h'}(head(\rho_3))) \cup \mathrm{Con}(\mathcal{D}, \mathcal{R}).$$

Hence, it follows that δ_3' is greedy, and concludes our proof that $\mathcal{R}_2$ is a **wgbts**, but is not a **gbts**.

C Proofs for Section 5

Lemma 6. *Let $\mathcal{D}$ be a database and $\mathcal{R}$ a rule set. If $\mathcal{D} \xrightarrow{\mathcal{R}}_{\delta} \mathcal{I}$ with $\Sigma(G_\delta) = (\mathrm{V}, \mathrm{E}, \mathrm{At}, \mathrm{L})$ a derivation graph and Σ a reduction sequence, then $\Sigma(G_\delta)$ has the following properties:*

1. *for each non-initial node $X_n \in \mathrm{V}$, there exists a $\rho \in \mathcal{R}$ with $\rho = \varphi(\mathbf{x}, \mathbf{y}) \to \exists \mathbf{z} \psi(\mathbf{y}, \mathbf{z})$ and a homomorphism $\overline{h}$ such that $\mathrm{At}(X_n) = \overline{h}(\psi(\mathbf{y}, \mathbf{z}))$;*
2. *if $(X_n, X_m) \in \mathrm{E}$, then $n < m$.*

Proof. Both claims follow from the definition of a derivation graph along with the fact that the reduction operations only affect arcs and labels.

Definition 8 (x-Generative, Source Node). *Let $\mathcal{D}$ be a database, $\mathcal{R}$ be a rule set, $\mathcal{D} \xrightarrow{\delta}_{\mathcal{R}} \mathcal{I}$, and Σ be a reduction sequence applicable to $G_\delta = (\mathrm{V}, \mathrm{E}, \mathrm{At}, \mathrm{L})$. We define a node in $\Sigma(G_\delta) = (\mathrm{V}', \mathrm{E}', \mathrm{At}', \mathrm{L}')$ to be x-*generative *with $x \in \overline{C}(\mathcal{I})$ iff for every node $X_k \in \mathrm{V}'$, if $x \in \overline{C}(X_k)$, then $n \leq k$. We define a node $X \in \mathrm{V}'$ to be a* source node *iff no node $Y \in \mathrm{V}'$ exists such that $(Y, X) \in \mathrm{E}'$, and we define X to be* non-source node *otherwise.*

Lemma 7. *Let $\mathcal{D}$ be a database, $\mathcal{R}$ be a rule set, $\mathcal{D} \xrightarrow{\delta}_{\mathcal{R}} \mathcal{I}$, and Σ be a reduction sequence with $\Sigma(G_\delta) = (\mathrm{V}, \mathrm{E}, \mathrm{At}, \mathrm{L})$. For any nodes $X_i, X_j \in \mathrm{V}$, if $x \in \overline{C}(X_i) \cap \overline{C}(X_j)$, $(X_i, X_j) \in \mathrm{E}$, and $x \notin \mathrm{L}(X_i, X_j)$, then there exists a node $X_m \in \mathrm{V}$ such that $x \in \overline{C}(X_m)$, $(X_m, X_j) \in \mathrm{E}$, and $x \in \mathrm{L}(X_m, X_j)$.*

Lemma 8. *Let $\mathcal{D}$ be a database and $\mathcal{R}$ be a rule set. If $\mathcal{D} \xrightarrow{\delta}_{\mathcal{R}} \mathcal{I}$, then for any reduction sequence Σ, $\Sigma(G_\delta) = (\mathrm{V}, \mathrm{E}, \mathrm{At}, \mathrm{L})$ satisfies the following two conditions:*

1. *if $x \in \overline{C}(\mathcal{I})$, then there exists a unique x-generative node $X \in \mathrm{V}$;*
2. *if X_n is the x-generative node in $\Sigma(G_\delta)$, then for every $X_k \in \mathrm{V}$ such that $x \in \overline{C}(X_k)$, there is a directed path from X_n to X_k in $\Sigma(G_\delta)$ such that for every node X_ℓ along the path, $\ell \leq k$ and $x \in \overline{C}(X_\ell)$.*

Proof. Statement 1 is evident as there must be a first rule application in δ that introduces the null x. Let $\Sigma(G_\delta) = (\mathrm{V}, \mathrm{E}, \mathrm{At}, \mathrm{L})$. We argue statement 2 by induction on the lexicographic ordering of pairs $(|\delta|, |\Sigma|)$, where $|\delta|$ is the length of the derivation and $|\Sigma|$ is the length of the reduction sequence. Suppose X_n is the x-generative node in $\Sigma(G_\delta)$ and let $X_k \in \mathrm{V}$ such that $x \in \overline{C}(X_k)$. We aim to show that a directed path exists from X_n to X_k such that for every node X_ℓ along the path, $\ell \leq k$ and $x \in \overline{C}(X_\ell)$.

Base case. If $|\delta| = 0$, meaning $\delta = \mathcal{D}$, then the result trivially follows. If $|\Sigma| = 0$, then $\Sigma(G_\delta) = G_\delta$ with $G_\delta = (\mathrm{V}, \mathrm{E}, \mathrm{At}, \mathrm{L})$. If $X_k = X_n$, then the claim trivially holds. However, if $X_k \neq X_n$, then let us consider the derivation $\mathcal{D} \xrightarrow{\delta'}_{\mathcal{R}} \mathcal{I}_{k-1}, (\rho, h, \mathcal{I}_k)$, where the application of ρ produces the node X_k. Since $x \in \overline{C}(X_n)$ and $x \in \overline{C}(X_k)$, we know there exists a node X_m such that $x \in \overline{C}(X_m)$, $(X_m, X_k) \in \mathrm{E}$, and $x \in \mathrm{L}(X_m, X_k)$. By IH, there is a directed path from X_n to X_m such that for every X_ℓ along the path $\ell \leq m$ and $x \in \overline{C}(X_\ell)$.

Therefore, since $(X_m, X_k) \in \mathrm{E}$, we know that such a directed path from X_n to X_k of the required shape exists as well.

Inductive step. Let $(\mathsf{r}) \in \{(\mathsf{tr}), (\mathsf{ar}), (\mathsf{cr})\}$ with $\Sigma = (\mathsf{r})\Sigma'$. Let $\Sigma'(G_\delta) = (\mathrm{V}', \mathrm{E}', \mathrm{At}', \mathrm{L}')$. We consider the cases where (r) is either (ar) or (cr) as the case when (r) is (tr) is trivial as all paths are preserved after the reduction operation is applied.

(ar). Suppose that an arc $(X_i, X_j) \in \mathrm{E}'$ exists such that $\mathrm{L}'(X_i, X_j) = \emptyset$, which is removed by applying (ar) to $\Sigma'(G_\delta)$. By IH, we know that a directed path from X_n to X_k exists in $\Sigma'(G_\delta)$ such that for every node X_ℓ along the path $\ell \leq k$ and $x \in \overline{C}(X_\ell)$. Let us suppose that (X_i, X_j) occurs along this path, since otherwise, the result trivially follows. Then, we know that $x \in \overline{C}(X_i)$ and $x \in \overline{C}(X_j)$. Since $\mathrm{L}'(X_i, X_j) = \emptyset$, we know $x \notin \mathrm{L}'(X_i, X_j)$, and therefore, by Lemma 7, some $X_m \neq X_i$ exists such that $X_m \in \mathrm{V}'$, $x \in \overline{C}(X_m)$, $(X_m, X_j) \in \mathrm{E}'$, and $x \in \mathrm{L}'(X_m, X_j)$. By IH, there exists a directed path from X_n to X_m such that for every node X_ℓ along the path $\ell \leq m$ and $x \in \overline{C}(X_\ell)$. After (ar) is applied, this path will still be present, and so, a path of the desired shape will exist from X_n to X_k.

(cr). Suppose that $(X_i, X_m), (X_j, X_m) \in \mathrm{E}'$, and there exists a node X_ℓ such that $\ell < m$ and $\mathrm{L}'(X_i, X_m) \cup \mathrm{L}'(X_j, X_m) \subseteq \mathrm{Ter}(X_\ell)$. After applying (cr), we suppose that $(X_i, X_m), (X_j, X_m)$ are removed from the set of arcs and (X_ℓ, X_m) is added such that $\mathrm{L}(X_\ell, X_m) = \mathrm{L}'(X_i, X_m) \cup \mathrm{L}'(X_j, X_m)$. By IH, a directed path from X_n to X_k exists in $\Sigma'(G_\delta)$ such that for every node X_u along the path $u \leq k$ and $x \in \overline{C}(X_u)$. We assume w.l.o.g. that (X_i, X_m) occurs along this path, since the other cases are trivial or similar. If $x \in \mathrm{L}'(X_i, X_m)$, then $x \in \overline{C}(X_\ell)$ by assumption, implying that a directed path exists from X_n to X_ℓ of the required form. Hence, after applying (cr), a directed path of the required form will exist consisting of the path from X_n to X_ℓ, the arc (X_ℓ, X_m), and the path from X_m to X_k. However, if $x \notin \mathrm{L}'(X_i, X_m)$, then as in the (ar) case above, there exists some $X_v \neq X_i$ such that $X_v \in \mathrm{V}'$, $x \in \overline{C}(X_v)$, $(X_v, X_m) \in \mathrm{E}'$, and $x \in \mathrm{L}'(X_v, X_m)$ (by Lemma 7). By an argument similar to the (ar) case, we find that a directed path of the required form exists from X_n to X_k in $\Sigma(G_\delta)$.

Lemma 5 *Let $\mathcal{D}$ be a database and $\mathcal{R}$ be a rule set. If $\mathcal{D} \xrightarrow{\delta}_{\mathcal{R}} \mathcal{I}$, then for any reduction sequence Σ, $\Sigma(G_\delta) = (\mathrm{V}, \mathrm{E}, \mathrm{At}, \mathrm{L})$ satisfies the decomposition properties 1-4 in Lemma 4, i.e. the following four conditions:*

1. $\bigcup_{X_n \in \mathrm{V}} \mathrm{Ter}(X_n) = \mathrm{Ter}(\mathcal{I})$;
2. *For each $p(\mathbf{t}) \in \mathcal{I}$, there is an $X_n \in \mathrm{V}$ such that $p(\mathbf{t}) \in \mathrm{At}(X_n)$;*
3. *For each term $x \in \overline{C}(\mathcal{I})$, the subgraph of $\Sigma(G_\delta)$ induced by the nodes X_n such that $x \in \overline{C}(X_n)$ is connected;*
4. *For each $X_n \in \mathrm{V}$ the size of $\mathrm{Ter}(X_n)$ is bounded by an integer that only depends on the size of $(\mathcal{D}, \mathcal{R})$, viz. $max\{|\mathrm{Ter}(\mathcal{D})|, |\mathrm{Ter}(head(\rho_i))|_{\rho_i \in \mathcal{R}}\} + |C|$.*

Proof. It is straightforward to confirm properties 1, 2, and 4. Property 3 follows from Lemma 8.

Proposition 3 *Let $\mathcal{R}$ be a rule set.*

1. *If $\mathcal{R}$ is* **cdgs**, *then $\mathcal{R}$ is* **wcdgs**;
2. *If $\mathcal{R}$ is* **wcdgs**, *then $\mathcal{R}$ is* **pbts**.

Proof. We prove each claim in turn:

1. Suppose that $\mathcal{R}$ is **cdgs** and let $\mathcal{D}$ be an arbitrary database. Then, if $\mathcal{D} \xrightarrow{\mathcal{R}}_{\delta} \mathcal{I}$, it follows that a derivation $\delta' = \delta$ exists such that $\mathcal{D} \xrightarrow{\mathcal{R}}_{\delta'} \mathcal{I}$ and $G_{\delta'}$ can be reduced to a cycle-free graph (since $\mathcal{R}$ is **cdgs**). Hence, $\mathcal{R}$ is **wcdgs**.
2. Suppose that $\mathcal{R}$ is **wcdgs**, $\mathcal{D}$ is a database, let $C = \mathrm{Con}(\mathcal{D}, \mathcal{R})$, and let $n = max\{|\mathrm{Ter}(\mathcal{D})|, |\mathrm{Ter}(head(\rho_i))|_{\rho_i \in \mathcal{R}}\} + |C|$, and assume that $\mathcal{D} \xrightarrow{\delta}_{\mathcal{R}} \mathcal{I}$. Our aim is to show that $tw(\mathcal{I}) \leq n$ in order to show that $\mathcal{R}$ is **pbts**. Since $\mathcal{R}$ is **wcdgs**, we know there exists an $\mathcal{R}$-derivation δ' and a complete reduction sequence Σ such that $\Sigma(G_{\delta'}) = (\mathrm{V}', \mathrm{E}', \mathrm{At}', \mathrm{L}')$ is a cycle-free graph. Let us define a tree decomposition $T = (V, E)$ of $\mathcal{I}$ by making use of $\Sigma(G_{\delta'})$, where $X \in V$ *iff* there exists a node $X' \in \mathrm{V}'$ such that $X = \mathrm{Ter}(X')$. We then define $(X, Y) \in E''$ *iff* there exists an arc $(X', Y') \in \mathrm{E}'$ such that $X = \mathrm{Ter}(X')$ and $Y = \mathrm{Ter}(Y')$. In general, $T' = (V, E'')$ will be a finite forest, so if place each tree of T' in a line and connect the root of the first tree to the root of the second, the root of the second tree to the root of the third, etc., then this yields a tree decomposition $T = (V, E)$ (where E extends E'' with the edges just mentioned). By Lemma 5, T is indeed a tree decomposition, and furthermore, $w(T) \leq n$. Thus, $tw(\mathcal{I}) \leq w(T) \leq n$, establishing the claim.

Definition 9. *Let $\Sigma(G_\delta) = (\mathrm{V}, \mathrm{E}, \mathrm{At}, \mathrm{L})$ be a derivation graph with Σ a reduction sequence and $X_n \in \mathrm{V}$. Moreover, let $\mathcal{D}$ be a database, $\mathcal{R}$ be a rule set, and $C = \mathrm{Con}(\mathcal{D}, \mathcal{R})$. We define the* frontier *$fr(X_n)$ of a node $X_n \in \mathrm{V}$ relative to $(\mathcal{D}, \mathcal{R})$ accordingly:*

$$fr(X_n) = \begin{cases} \emptyset & \text{if } X_n \text{ is a source node;} \\ \overline{h}_i(\mathbf{y}_i) \setminus C & \text{otherwise.} \end{cases}$$

where $\mathrm{At}(X_n) = \overline{h}_i(\psi_i(\mathbf{y}_i, \mathbf{z}_i))$.

Lemma 9. *Let $\mathcal{D}$ be a database, $\mathcal{R}$ be a rule set, $C = \mathrm{Con}(\mathcal{D}, \mathcal{R})$, and assume that $\mathcal{D} \xrightarrow{\mathcal{R}}_{\delta} \mathcal{I}$. Then, for Σ a reduction sequence, the derivation graph $\Sigma(G_\delta) = (\mathrm{V}, \mathrm{E}, \mathrm{At}, \mathrm{L})$ satsifies the following properties:*

1. *for each $X_{n_0} \in \mathrm{V}$ with parent nodes $X_{n_1}, \ldots, X_{n_k} \in \mathrm{V}$,*

$$fr(X_{n_0}) = \bigcup_{i \in \{1,\ldots,k\}} \mathrm{L}(X_{n_i}, X_{n_0});$$

2. *for each $(X_m, X_n) \in \mathrm{E}$, $\mathrm{L}(X_m, X_n) \subseteq \mathrm{Ter}(X_m)$;*
3. *for each $X_{n_0} \in \mathrm{V}$ with parent nodes $X_{n_1}, \ldots, X_{n_k} \in \mathrm{V}$,*

$$\bigcup_{i \in \{1,\ldots,k\}} \mathrm{L}(X_{n_i}, X_{n_0}) \subseteq \bigcup_{i \in \{1,\ldots,k\}} \mathrm{Ter}(X_{n_i}).$$

Proof. Since 3 follows from 2, we only prove 1 and 2. We prove each claim in turn by induction on the length of the reduction sequence Σ.

1. *Base case.* Suppose that $\Sigma = \varepsilon$, so that $\Sigma(G_\delta) = \varepsilon(G_\delta) = G_\delta$. Observe that for any $X_n \in \mathrm{V}$ with (a non-empty set of) parent nodes $X_{n_1}, \ldots, X_{n_k} \in \mathrm{V}$, $fr(X_n) = \overline{h}(\mathbf{y}) \setminus C$ for $\psi(\mathbf{y}, \mathbf{z}) = head(\rho)$ for some $\rho \in \mathcal{R}$. Moreover, by definition, it follows that

$$fr(X_n) = \bigcup_{i \in \{1,\ldots,k\}} \mathrm{L}(X_{n_i}, X_n).$$

Inductive step. We assume for IH that the property holds for $\Sigma(G_\delta)$ and show that the property holds for $(\mathsf{r})\Sigma(G_\delta) = (\mathrm{V}', \mathrm{E}', \mathrm{At}', \mathrm{L}')$ with $(\mathsf{r}) \in \{(\mathsf{tr}), (\mathsf{ar}), (\mathsf{cr})\}$. We make a case distinction based on the last reduction operation (r) applied.

(ar). Let $(X_{n_1}, X_{n_0}) \in \mathrm{E}$ such that $\mathrm{L}(X_{n_1}, X_{n_0}) = \emptyset$. Assume that (ar) was applied, so that $(X_{n_1}, X_{n_0}) \notin \mathrm{E}'$. For any node $X_m \neq X_{n_0}$ in $(\mathsf{r})\Sigma(G_\delta)$ property 2 holds by IH, and for the node X_{n_0} with parent nodes $X_{m_1}, \ldots, X_{m_k}$ in $(\mathsf{ar})\Sigma(G_\delta)$ we have

$$\begin{aligned} fr(X_{n_0}) &= \bigcup_{i \in \{1,\ldots,k\}} \mathrm{L}(X_{m_i}, X_{n_0}) \cup \mathrm{L}(X_{n_1}, X_{n_0}) \\ &= \bigcup_{i \in \{1,\ldots,k\}} \mathrm{L}'(X_{m_i}, X_{n_0}) \cup \emptyset \\ &= \bigcup_{i \in \{1,\ldots,k\}} \mathrm{L}'(X_{m_i}, X_{n_0}) \end{aligned}$$

where the first equality follows by IH, and the second by the definition of L' along with the fact that $\mathrm{L}(X_{n_1}, X_{n_0}) = \emptyset$.

(tr). Let $(X_{n_1}, X_{n_0}), (X_{n_2}, X_{n_0}) \in \mathrm{E}$ with $t \in \mathrm{L}(X_{n_1}, X_{n_0}) \cap \mathrm{L}(X_{n_2}, X_{n_0})$. Suppose we apply (tr), so that $\mathrm{L}'(X_{n_2}, X_{n_0}) = \mathrm{L}(X_{n_2}, X_{n_0}) \setminus \{t\}$. For any node $X_m \neq X_{n_0}$ in $(\mathsf{tr})\Sigma(G_\delta)$ the result holds by IH, and for the node X_{n_0} with parent nodes $X_{m_1}, \ldots, X_{m_k}$ we have

$$fr(X_{n_0}) = \bigcup_{i \in \{1,\ldots,k\}} \mathrm{L}(X_{m_i}, X_{n_0}) = \bigcup_{i \in \{1,\ldots,k\}} \mathrm{L}'(X_{m_i}, X_{n_0})$$

as $t \in \mathrm{L}'(X_{n_1}, X_{n_0})$.

(cr). Let $(X_{n_1}, X_{n_0}), (X_{n_2}, X_{n_0}) \in \mathrm{E}$ with a node $X_m \in \mathrm{V}$ such that $m < n_0$ and $\mathrm{L}(X_{n_1}, X_{n_0}) \cup \mathrm{L}(X_{n_2}, X_{n_0}) \subseteq \mathrm{Ter}(X_m)$. Assume that (cr) was applied, so that $(X_m, X_{n_0}) \in \mathrm{E}'$ with $\mathrm{L}'(X_m, X_{n_0}) = \mathrm{L}(X_{n_1}, X_{n_0}) \cup \mathrm{L}(X_{n_2}, X_{n_0})$. For any node $X_k \neq X_{n_0}$ in $(\mathsf{cr})\Sigma(G_\delta)$, property 2 holds by IH, so let us consider the node X_{n_0}, which has parents $X_{m_1}, \ldots, X_{m_k}$, X_{n_1}, and X_{n_2} in $\Sigma(G_\delta)$ and parents $X_{m_1}, \ldots, X_{m_k}$, and X_m in $(\mathsf{cr})\Sigma(G_\delta)$. By IH, we have the first equality below, and the second follows from the definition of L', giving the desired result:

$$fr(X_{n_0}) = \bigcup_{i\in\{1,\dots,k\}} \mathrm{L}(X_{m_i}, X_{n_0}) \cup \mathrm{L}(X_{n_1}, X_{n_0}) \cup \mathrm{L}(X_{n_2}, X_{n_0})$$
$$= \bigcup_{i\in\{1,\dots,k\}} \mathrm{L}'(X_{m_i}, X_{n_0}) \cup \mathrm{L}'(X_m, X_{n_0}).$$

2. *Base case.* Suppose that $\Sigma = \varepsilon$, so that $\Sigma(G_\delta) = \varepsilon(G_\delta) = G_\delta$. The result immediately follows from the definition of an derivation graph.

   Inductive step. We assume for IH that the property holds for $\Sigma(G_\delta)$ and show that the property holds for $(\mathsf{r})\Sigma(G_\delta) = (\mathrm{V}', \mathrm{E}', \mathrm{At}', \mathrm{L}')$ with $(\mathsf{r}) \in \{(\mathsf{tr}), (\mathsf{ar}), (\mathsf{cr})\}$.

   (tr). Let $(X_{n_1}, X_{n_0}), (X_{n_2}, X_{n_0}) \in \mathrm{E}$ with $t \in \mathrm{L}(X_{n_1}, X_{n_0}) \cap \mathrm{L}(X_{n_2}, X_{n_0})$. Suppose we apply (tr), so that $\mathrm{L}'(X_{n_2}, X_{n_0}) = \mathrm{L}(X_{n_2}, X_{n_0}) \setminus \{t\}$. For any arc $(X_{m_1}, X_{m_0}) \neq (X_{n_2}, X_{n_0})$ in $(\mathsf{tr})\Sigma(G_\delta)$, the result holds by IH, so let us focus on $(X_{n_2}, X_{n_0}) \in \mathrm{E}'$. Observe that $\mathrm{L}'(X_{n_2}, X_{n_0}) \subseteq \mathrm{L}(X_{n_2}, X_{n_0}) \subseteq \mathrm{Ter}(X_{n_2})$.

   (ar). Let $(X_{n_1}, X_{n_0}) \in \mathrm{E}$ such that $\mathrm{L}(X_{n_1}, X_{n_0}) = \emptyset$. Assume that (ar) was applied, so that $(X_{n_1}, X_{n_0}) \notin \mathrm{E}'$. For any $(X_{m_1}, X_{m_0}) \in \mathrm{E}'$, $\mathrm{L}'(X_{m_1}, X_{m_0}) = \mathrm{L}(X_{m_1}, X_{m_0}) \subseteq \mathrm{Ter}(X_{m_1})$ by the definition of L' and IH.

   (cr). Let $(X_{n_1}, X_{n_0}), (X_{n_1}, X_{n_0}) \in \mathrm{E}$ with a node $X_m \in \mathrm{V}$ such that $m < n_0$ and $\mathrm{L}(X_{n_1}, X_{n_0}) \cup \mathrm{L}(X_{n_1}, X_{n_0}) \subseteq \mathrm{Ter}(X_m)$. Assume that (cr) was applied, so that $(X_m, X_{n_0}) \in \mathrm{E}'$ with $\mathrm{L}'(X_m, X_{n_0}) = \mathrm{L}(X_{n_1}, X_{n_0}) \cup \mathrm{L}(X_{n_1}, X_{n_0})$. For any arc $(X_{k_1}, X_{k_2}) \neq (X_m, X_{n_0})$ in $(\mathsf{cr})\Sigma(G_\delta)$, the result holds by IH, so let us focus on $(X_m, X_{n_0}) \in \mathrm{E}$. We have $\mathrm{L}'(X_m, X_{n_0}) = \mathrm{L}(X_{n_1}, X_{n_0}) \cup \mathrm{L}(X_{n_1}, X_{n_0}) \subseteq \mathrm{Ter}(X_m)$ by the definition of L' and the condition required to apply (cr). This concludes the proof of the case.

Definition 10 (Sub-reduction Sequence). *Let $\Sigma = (\mathsf{r}_1) \cdots (\mathsf{r}_n)$ be a reduction sequence. We define a* sub-reduction sequence *Σ' of Σ to be a reduction sequence of the form $(\mathsf{r}_1) \cdots (\mathsf{r}_i)$ with $0 \leq i \leq n$, which is the empty reduction sequence ε when $n = 0$. If Σ' is a sub-reduction sequence of Σ, then we write $\Sigma' \sqsubseteq \Sigma$, and we note that we take Σ' to be the same instances of the reduction operations occurring within the reduction sequence Σ.*

Lemma 10. *Let $\mathcal{D}$ be a database, $\mathcal{R}$ be a rule set, and assume that $\mathcal{D} \xrightarrow{\mathcal{R}}_\delta \mathcal{I}$. Moreover, assume that $\Sigma(G_\delta) = (\mathrm{V}, \mathrm{E}, \mathrm{At}, \mathrm{L})$ is a cycle-free derivation graph with Σ a complete reduction sequence. For each $\Sigma' \sqsubseteq \Sigma$, the derivation graph $\Sigma'(G_\delta) = (\mathrm{V}', \mathrm{E}', \mathrm{At}', \mathrm{L}')$ satisfies the following: For each non-source node $X_n \in \mathrm{V}'$, there exists a node $X_m \in \mathrm{V}'$ such that $m < n$ and $fr(X_n) \subseteq \mathrm{Ter}(X_m)$.*

Proof. We first show (1) that the claim holds for $\Sigma(G_\delta)$, and then (2) show that if the claim holds for $\Sigma'(G_\delta)$ with $\Sigma' = (\mathsf{r})\Sigma''$ and $(\mathsf{r}) \in \{(\mathsf{tr}), (\mathsf{ar}), (\mathsf{cr})\}$, then it holds for $\Sigma''(G_\delta)$.

(1) Let $X_n \in \mathrm{V}$ be a non-source node of $\Sigma(G_\delta)$ with parent nodes X_{n_i} for $i \in \{1, \ldots, k\}$. By Lemma 9, we know that

$$fr(X_n) = \bigcup_{i \in \{1,\ldots,k\}} \mathrm{L}(X_{n_i}, X_{\mathbf{n}}) \subseteq \bigcup_{i \in \{1,\ldots,k\}} \mathrm{Ter}(X_{n_i}).$$

Since Σ is a complete reduction sequence and $\Sigma(G_\delta)$ is cycle-free, we know that $\Sigma(G_\delta)$ is a forest, implying that each non-source node has a single parent node. Hence, X_n has a single parent node X_m, implying that $fr(X_n) \subseteq \mathrm{Ter}(X_m)$, thus confirming the desired result as $m < n$ by Lemma 6.

(2) Let $\Sigma'(G_\delta) = (\mathrm{V}', \mathrm{E}', \mathrm{At}', \mathrm{L}')$, $\Sigma''(G_\delta) = (\mathrm{V}'', \mathrm{E}'', \mathrm{At}'', \mathrm{L}'')$, and suppose that for every non-source node $X_n \in \mathrm{V}'$, there exists a node $X_m \in \mathrm{V}'$ such that $m < n$ and $fr(X_n) \subseteq \mathrm{Ter}(X_m)$. We show the claim by a case distinction on if (tr), (ar), or (cr) was applied last in Σ'.

(tr). Observe that if (tr) was applied last in Σ', then the only difference between $\Sigma'(G_\delta)$ and $\Sigma''(G_\delta)$ is that for some arc $(X_{k_1}, X_{k_0}) \in \mathrm{E}' \cap \mathrm{E}''$, $\mathrm{L}''(X_{k_1}, X_{k_0}) = \mathrm{L}'(X_{k_1}, X_{k_0}) \cup \{t\}$, for some term t. Hence, for an arbitrary non-source node $X_n \in \mathrm{V}''$, $X_n \in \mathrm{V}'$ since $\mathrm{V}'' = \mathrm{V}'$, implying that there exists a node $X_m \in \mathrm{V}' = \mathrm{V}''$ such that $m < n$ and $fr(X_n) \subseteq \mathrm{Ter}(X_m)$, completing the proof of the case.

(ar). If (ar) was applied last in Σ', then the only difference between $\Sigma'(G_\delta)$ and $\Sigma''(G_\delta)$ is that for some arc (X_{k_1}, X_{k_0}), $\mathrm{E}'' = \mathrm{E}' \cup \{(X_{k_1}, X_{k_0})\}$, where $\mathrm{L}''(X_{k_1}, X_{k_0}) = \emptyset$. For any non-source node $X_n \in \mathrm{V}''$ such that X_n is a non-source node in V', the result immediately holds. However, it could be the case that even though X_{k_0} is a non-source node in V'', X_{k_0} is a source node in $\Sigma'(G_\delta)$ as $(X_{k_1}, X_{k_0}) \in \mathrm{E}''$. In this case, by Lemma 9 and the fact that X_{k_1} is the only parent of $X_{k_0} \in \mathrm{V}''$, we know that $fr(X_{k_0}) \subseteq \mathrm{L}''(X_{k_1}, X_{k_0}) = \emptyset$, implying that $fr(X_{k_0}) = \emptyset$. As X_{k_1} is a parent of X_{k_0} in $\Sigma''(G_\delta)$, we know that $k_1 < k_0$ by Lemma 6, and trivially $fr(X_{k_0}) \subseteq \mathrm{Ter}(X_{k_1})$, proving the case.

(cr). If (cr) is applied last in Σ', then the only difference between $\Sigma''(G_\delta)$ and $\Sigma'(G_\delta)$ is that there exist arcs $(X_{n_1}, X_{n_0}), (X_{n_2}, X_{n_0}) \in \mathrm{E}''$ and $\mathrm{E}' = (\mathrm{E}'' \setminus \{(X_{n_1}, X_{n_0}), (X_{n_2}, X_{n_0})\}) \cup \{(X_m, X_{n_0})\}$ as there exists a node $X_m \in \mathrm{V}''$ such that $m < n_0$ and $\mathrm{L}''(X_{n_1}, X_{n_0}) \cup \mathrm{L}''(X_{n_2}, X_{n_0}) = \mathrm{Ter}(X_m)$. Hence, for an arbitrary non-source node $X_k \in \mathrm{V}''$, $X_k \in \mathrm{V}'$ as $\mathrm{V}'' = \mathrm{V}'$, implying the existence of a node $X_{k'}$ such that $k' < k$ and $fr(X_k) \subseteq \mathrm{Ter}(X_{k'})$, thus completing the proof.

Lemma 11. *Let $\mathcal{R}$ be a rule set. Then,*

1. *If $\mathcal{R}$ is* **gbts**, *then $\mathcal{R}$ is* **cdgs**;
2. *if $\mathcal{R}$ is* **wgbts**, *then $\mathcal{R}$ is* **wcdgs**.

Proof. We argue claim 1 since the proof of claim 2 is similar. Let $\mathcal{D}$ be a database, $\mathcal{R}$ be **gbts**, and assume $\mathcal{D} \xrightarrow{\delta}_{\mathcal{R}} \mathcal{I}$. Since $\mathcal{R}$ is **gbts**, we know that the $\mathcal{R}$-derivation

$$\delta = \mathcal{D}, (\rho_1, h_1, \mathcal{I}_1), \ldots, (\rho_n, h_n, \mathcal{I}_n)$$

is greedy. Therefore, for each i such that $0 < i < n$, there exists a $j < i$ such that $h_i(fr(\rho_i)) \subseteq \mathrm{Nul}(\overline{h}_j(head(\rho_j))) \cup \mathrm{Con}(\mathcal{D}, \mathcal{R})$. Let us now show that $\mathcal{R}$ is **cdgs** by arguing that $G_\delta = (\mathrm{V}, \mathrm{E}, \mathrm{At}, \mathrm{L})$ is reducible to a cycle-free graph.

Let us suppose that there exist arcs $(X_{n_1}, X_{n_0}), (X_{n_2}, X_{n_0}) \in \mathrm{E}$. By our assumption that δ is greedy, we know that there exists a node $X_m \in \mathrm{V}$ such that $m < n_0$ and $fr(X_{n_0}) \subseteq \mathrm{Ter}(X_m)$. By Lemma 9, it follows that $\mathrm{L}(X_{n_1}, X_{n_0}) \cup \mathrm{L}(X_{n_2}, X_{n_0}) \subseteq \mathrm{Ter}(X_m)$, meaning we can apply (cr) to G_δ. Observe that $(\mathsf{cr})(G_\delta)$ has one less "convergence point" as (X_{n_1}, X_{n_0}) and (X_{n_2}, X_{n_0}) have been replaced by the single arc (X_m, X_{n_0}). By repeating this process, all such convergence points will be removed, yielding a reduced, cycle-free derivation graph. Hence, $\mathcal{R}$ is **cdgs**.

Lemma 12. *Let $\mathcal{R}$ be a rule set. Then,*

1. *If $\mathcal{R}$ is* **cdgs**, *then $\mathcal{R}$ is* **gbts**;
2. *if $\mathcal{R}$ is* **wcdgs**, *then $\mathcal{R}$ is* **wgbts**.

Proof. We prove claim 2 since claim 1 is shown in a similar fashion. Let $\mathcal{D}$ be a database, $\mathcal{R}$ be **wcdgs**, $C = \mathrm{Con}(\mathcal{D}, \mathcal{R})$, and assume $\mathcal{D} \xrightarrow{\delta}_{\mathcal{R}} \mathcal{I}$. Since $\mathcal{R}$ is **wcdgs**, we know there exists an $\mathcal{R}$-derivation

$$\delta' = \mathcal{D}, (\rho_1, h_1, \mathcal{I}_1), \ldots, (\rho_k, h_k, \mathcal{I}_k)$$

such that $G_{\delta'} = (\mathrm{V}, \mathrm{E}, \mathrm{At}, \mathrm{L})$ is reducible to a cycle-free graph. That is to say, there exists a (complete) reduction sequence Σ such that $\Sigma(G_{\delta'})$ is cycle-free. By Lemma 10, we know that for every $\Sigma' \sqsubseteq \Sigma$, $\Sigma'(G_{\delta'}) = (\mathrm{V}', \mathrm{E}', \mathrm{At}', \mathrm{L}')$ satisfies the following property: For each non-source node $X_n \in \mathrm{V}'$, there exists a node $X_m \in \mathrm{V}'$ such that $m < n$ and $fr(X_n) \subseteq \mathrm{Ter}(X_m)$. In particular, this property holds for $\Sigma' = \varepsilon$, i.e. for $G_{\delta'}$. Since $h_k(fr(\rho_k)) \subseteq C$ when $X_k \in \mathrm{V}$ is a source node, and due to the fact that for each non-source node $X_n \in \mathrm{V}$, $fr(X_n) = \overline{h}_n(fr(\rho_n)) \setminus C = h_n(fr(\rho_n)) \setminus C$, we have that

$$h_n(fr(\rho_n)) \subseteq fr(X_n) \cup C \subseteq \mathrm{Ter}(X_m) = \mathrm{Nul}(\overline{h}_m(head(\rho_m)) \cup \mathrm{Con}(\mathcal{D}, \mathcal{R}),$$

for each $0 < n \leq k$ and some $m < n$, where the last equality above follows from the definition of $\mathrm{Ter}(X_m)$. Therefore, δ' is greedy, showing that $\mathcal{R}$ is **wgbts**.

Theorem 3 *Let $\mathcal{R}$ be a rule set.*

1. *$\mathcal{R}$ is* **gbts** *iff $\mathcal{R}$ is* **cdgs**;
2. *$\mathcal{R}$ is* **wgbts** *iff $\mathcal{R}$ is* **wcdgs**.

Proof. Follows from Lemma 11 and Lemma 12.

Corollary 1 *Let $\mathcal{R}$ be a rule set.*

1. *The reduction operations* (tr) *and* (ar) *are reduction-admissible;*
2. *The* **wcdgs** *class properly contains the* **cdgs** *class;*
3. *If $\mathcal{R}$ is* **cdgs**, **gbts**, **wcdgs**, *or* **wgbts**, *then BCQ entailment is decidable.*

Proof. We prove each claim in turn below:

1. Let δ be an arbitrary $\mathcal{R}$-derivation and assume that G_δ can be reduced to a cycle-free graph. By the proof of Lemma 12 above, δ is a greedy $\mathcal{R}$-derivation. Thus, by the proof of Lemma 11, δ is reducible to a cycle-free graph using only the (cr) operation.
2. By Lemma 2, we know that **wgbts** properly contains **gbts**. Therefore, by Theorem 3 above, **wcdgs** properly contains **cdgs**.
3. Follows from the fact that BCQ entailment is decidable for **fts** and every class of rule sets mentioned is a subset of **fts**.