On the Computational Complexity of Naive-based Semantics for Abstract Dialectical Frameworks

Sarah Alice Gaggl
Computational Logic Group
TU Dresden, Germany

Sebastian Rudolph
Computational Logic Group
TU Dresden, Germany

Hannes Strass
Computer Science Institute
Leipzig University, Germany

Abstract

Abstract dialectical frameworks (ADFs) are a powerful generalization of Dung's abstract argumentation frameworks. ADFs allow to model argumentation scenarios such that ADF semantics then provide interpretations of the scenarios. Among the considerable number of ADF semantics, the naive-based ones are built upon the fundamental concept of conflict-freeness. Intuitively, a three-valued interpretation of an ADF's statements is conflict-free iff all true statements can possibly be accepted, and all false statements cannot possibly be accepted.

In this paper, we perform an exhaustive analysis of the computational complexity of naive-based semantics. The results are quite interesting, for some of them involve little-known classes of the so-called Boolean hierarchy (another hierarchy in between classes of the polynomial hierarchy). Furthermore in credulous and sceptical entailment, the complexity can be different depending on whether we check for truth or falsity of a specific statement.

1 Introduction

Over the last decade, argumentation theory emerged as one of the major fields in artificial intelligence and non-monotonic reasoning. There, abstract argumentation frameworks (AFs) as introduced by Dung [1995] became a key formalism with applications to a variety of non-monotonic reasoning problems such as logic programming, inconsistency handling, legal reasoning and many others [Rahwan *et al.*, 2009].

The basic Dung-style framework only consists of a set of *abstract arguments* and a binary relation between them, denoted as *attacks*. The evaluation of such an AF is then based on model-theoretic semantics, which allow to select sets of arguments that can "stand together". The need to represent more complex relations between the abstract entities led to a wide range of extensions, which allow to handle preferences and values on arguments [Amgoud and Cayrol, 2002; Bench-Capon, 2003], weights [Dunne *et al.*, 2011], probabilities [Li *et al.*, 2011] and introduce a positive relation between arguments, so-called *supports* [Amgoud *et al.*, 2008]. Recently, abstract dialectical frameworks (ADFs) have been introduced [Brewka and Woltran, 2010; Brewka *et al.*, 2013] as a powerful generalization of Dung's framework. ADFs allow for more general interactions between statements, for example support, joint attack, joint support and mixed combinations. Furthermore, ADFs can also handle preferences, on both the statements and the links [Brewka and Woltran, 2010; Brewka *et al.*, 2013] as well as probabilities [Polberg and Doder, 2014]. Moreover, ADFs can not only be seen as an extension of Dung's AFs but also as a target language for compilation from more concrete and application-based languages (e.g. Carneades [Brewka and Gordon, 2010]), and thus, serve as "argumentation middleware" [Brewka *et al.*, 2013].

An ADF consists of a set of *statements*, a set of *links* between the statements and for each statement an *acceptance condition*, a Boolean formula over the parents of the statement. The acceptance of a statement thus only depends on the status of its parents. As for AFs, there are many semantics for ADFs which allow to decide on the status of the statements. One special kind of semantics are the stage [Strass, 2013; Gaggl and Strass, 2014] and the recently introduced nai_2 semantics [Gaggl and Strass, 2014]. Both are generalisations of the respective AF semantics (stage and *cf2*) [Verheij, 1996; Baroni *et al.*, 2005] and based on three-valued conflict-free interpretations. Semantics based on conflict-freeness are also referred to as *naive-based* semantics in the literature, since they form further refinements of the notion *naive*, referring to information-maximal conflict-free interpretations. For AFs, naive-based semantics are capable to handle cycles in a more uniform way than admissible-based semantics (see [Gaggl and Dvořák, 2014] for an extensive study).

Typical reasoning problems for ADFs are model verification, sceptical and credulous reasoning and existence of a non-trivial interpretation. Analysing the computational complexity of these reasoning problems is a crucial topic for theoretical and practical reasons. First, complexity results often serve as an indicator for how difficult and how expressive a reasoning task can be. Second, knowing about the complexity of a reasoning problem is essential for the development of adequate algorithms and systems. A comprehensive complexity analysis of the ADF semantics defined by Brewka *et al.* [2013] has been given by Strass and Wallner [2014]. However, the naive-based semantics are a more recent development and their complexity has not received any attention yet.

In this paper, we address these open problems and perform an exhaustive study of the computational complexity of naive-based semantics for abstract dialectical frameworks. More precisely, we analyse all reasoning tasks mentioned earlier (model verification, non-trivial existence, and credulous and sceptical reasoning) for the conflict-free, naive, stage and nai_2 semantics. The results show that these tasks are (sometimes considerably) more difficult than their counterparts in AFs. While for the standard Dung semantics (admissible, preferred, complete, stable), their ADF generalisations are mildly more complex (one level up in the polynomial hierarchy [Strass and Wallner, 2014]), for the naive-based semantics, the differences can be far more significant. For example, deciding whether an argument is true in every naive extension can be done in logarithmic space for AFs,while it is hard at least for the second level of the polynomial hierarchy in the case of ADFs. The complexity becomes even higher (completeness for the third level) if we want to check whether a statement is *false* in every naive interpretation of an ADF. In general, different complexities for entailment of truth and entailment of falsity seems to be quite uncommon in logic-based formalisms. We can trace the reason for this difference in naive-based semantics for ADFs back to the definition of a conflict-free interpretation, which basically requires different strengths of justification depending on which truth value is assigned to a statement: If a statement s is assigned truth value true, then this must be justified by statement s being possibly acceptable, that is, there must be an assignment to the remaining statements such that the acceptance condition of s is fulfilled. On the other hand, if a statement is assigned truth value false, then this must be justified by statement *s not* being possibly acceptable, that is, a satisfying assignment of the acceptance condition must not exist. Quite possibly even more interesting (and the hardest proof of all our results) is the complexity of deciding existence of non-trivial conflict-free interpretations. We show that the problem is complete for the second level of the Boolean hierarchy [Wechsung, 1985]. The Boolean hierarchy consists of classes that are composed of Boolean combinations of problems from NP and complements thereof. A somewhat better-known example is the class DP, a logical "and" of one NP- and one coNP-problem.

The remainder of the paper is structured as follows. We introduce the necessary background on ADFs and complexity theory in Section 2. The main part of the analysis is performed in Section 3, where we grouped our results according to the four decision problems verification (Section 3.1), existence of a non-trivial interpretation (Section 3.2), and entailment (Section 3.3). We conclude the paper in Section 4.

2 Background

For functions $f : A \to B$ and $g : C \to D$, we denote the *update of* f *with* g by $f \circ g : A \cup C \to B \cup D$ with $x \mapsto g(x)$ if $x \in C$, and $x \mapsto f(x)$ otherwise. So even if $x \in A \cap C$ and $f(x)$ is defined, we have $(f \circ g)(x) = g(x)$. For a function $f : A \to B$ and $b \in B$ we denote $f^{-1}(b) = \{a \in A \mid f(a) = b\}$. For $A' \subseteq A$ the function $f|_{A'} : A' \to B$ is the restriction of f's domain to A'.

We will make use of many standard concepts of classical propositional logic in this paper, including the usual notions of formulas, interpretations and models and satisfiability. Our analysis will be based on three-valued interpretations, mappings $v : S \to \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$ that assign one of the truth values true ($\mathbf{t}$), false ($\mathbf{f}$) or unknown ($\mathbf{u}$) to each statement. The three truth values are partially ordered by $\leq_i$ according to their information content: we have $\mathbf{u} <_i \mathbf{t}$ and $\mathbf{u} <_i \mathbf{f}$ and no other pair in $<_i$, which intuitively means that the classical truth values contain strictly more information than the truth value unknown. The information ordering $\leq_i$ extends in a straightforward way to valuations v_1, v_2 over S in that $v_1 \leq_i v_2$ if and only if $v_1(s) \leq_i v_2(s)$ for all $s \in S$.

Given a three-valued interpretation v and a formula φ, the partial evaluation of φ with v takes the two-valued part of v and replaces the evaluated variables by their truth values.

Definition 1. Let φ be a propositional formula over vocabulary S and for an $M \subseteq S$ let $v : M \to \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$ be a three-valued interpretation. The *partial valuation of* φ *by* v is the formula $\varphi^v = \varphi[s/\mathbf{t} : v(s) = \mathbf{t}][s/\mathbf{f} : v(s) = \mathbf{f}]$.

2.1 Abstract dialectical frameworks (ADFs)

An abstract dialectical framework (ADF) is a directed graph whose nodes represent statements or positions which can be accepted or not. The links represent dependencies: the status of a node s only depends on the status of its parents $par(s)$, that is, the nodes with a direct link to s. Each node s has an associated acceptance condition C_s specifying the exact conditions under which s is accepted. C_s is a function assigning to each subset of $par(s)$ one of the truth values $\mathbf{t}$, $\mathbf{f}$. Intuitively, if for some $R \subseteq par(s)$ we have $C_s(R) = \mathbf{t}$, then s will be accepted provided the nodes in R are accepted and those in $par(s) \setminus R$ are not accepted.

Definition 2. An *abstract dialectical framework* is a tuple $D = (S, L, C)$ where

- S is a set of statements (positions, nodes),
- $L \subseteq S \times S$ is a set of links,
- $C = \{C_s\}_{s \in S}$ is a collection of total functions $C_s : 2^{par(s)} \to \{\mathbf{t}, \mathbf{f}\}$, one for each statement s. The function C_s is called *acceptance condition of* s.

It is often convenient to represent acceptance conditions C_s as propositional formulas φ_s over $par(s)$. (We will do so in this paper, and furthermore restrict ourselves to finite ADFs.) Then, clearly, for $M \subseteq par(s)$ we have $C_s(M) = \mathbf{t}$ iff $M \models \varphi_s$. It might be the case that a link $(r, s) \in L$ in an ADF bears no actual significance. Formally, r *is redundant in* φ_s iff for every two-valued interpretation $v : par(s) \to \{\mathbf{t}, \mathbf{f}\}$, the formulas $\varphi_s^{v \circ \{r \mapsto \mathbf{t}\}}$ and $\varphi_s^{v \circ \{r \mapsto \mathbf{f}\}}$ are equivalent. That is, if (r, s) is redundant then $v(r)$ has no influence on the truth value of φ_s^v whatsoever. Several semantics for ADFs can be defined by using three-valued interpretations v to partially evaluate acceptance formulas φ_s [Brewka *et al.*, 2013; Gaggl and Strass, 2014]. We use the following three:

Definition 3. Let $D = (S, L, C)$ be an ADF. A three-valued interpretation $v : S \to \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$ is

- *conflict-free*, i.e. $v \in cfi(D)$, iff for all $s \in S$ we have:
  - $v(s) = \mathbf{t}$ implies that φ_s^v is satisfiable,
  - $v(s) = \mathbf{f}$ implies that φ_s^v is unsatisfiable;

- *naive*, i.e. $v \in nai(D)$, iff v is $\leq_i$-maximal conflict-free;
- *stage*, i.e. $v \in stg(D)$, iff the set $v_{\mathbf{u}} = v^{-1}(\mathbf{u})$ is $\subseteq$-minimal with respect to v being conflict-free.

The following definitions are from [Gaggl and Strass, 2014].

Definition 4. Let $D = (S, L, C)$ be an ADF and $p, s \in S$. We say that s *depends on* p if there is a path from p to s in L but no path from s to p in L. Now let $M \subseteq S$. A statement $s \in S$ is *independent modulo* M iff for each $p \in S$, if s depends on p then $p \in M$. A set $M \subseteq S$ is *independent* iff there is no $s \in M$ that depends on a $p \in S \setminus M$. Lastly, define $ind_D(M) = \{s \in S \mid s \text{ is independent modulo } M \text{ in } D\}$.

Note that dependence here implicitly speaks about strongly connected components (SCCs). Given an independent subset M of statements of an ADF, ignoring all other statements again yields an ADF.

Definition 5. Let $D = (S, L, C)$ be an ADF and $M \subseteq S$ be an independent set. The ADF D *restricted to* M is given by $D|_M = (M, L \cap (M \times M), \{\varphi_s\}_{s \in M})$.

Note that $D|_M$ really is an ADF since its acceptance formulas by presumption do not mention statements not in M.

Definition 6. Let $D = (S, L, C)$ be an ADF, $M \subseteq S$ and $v : M \to \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$. The ADF D *reduced with* v *on* M is given by $[\![D]\!]_M^v = (S, [\![L]\!]_M^v, \{[\![\varphi_s]\!]_M^v\}_{s \in S})$ with

$$[\![\varphi_s]\!]_M^v = \begin{cases} \mathbf{t} & \text{if } s \in M \text{ and } v(s) = \mathbf{t} \\ \mathbf{f} & \text{if } s \in M \text{ and } v(s) = \mathbf{f} \\ \neg s & \text{if } s \in M \text{ and } v(s) = \mathbf{u} \\ clean(\varphi_s^v) & \text{otherwise, where} \end{cases}$$

$$clean(\varphi_s^v) = \varphi_s^v[r/\mathbf{t} : r \text{ is redundant in } \varphi_s^v].$$

$$[\![L]\!]_M^v = \{(r, s) \mid r \in S \text{ and } r \text{ occurs in } [\![\varphi_s]\!]_M^v\}$$

That is, $clean(\varphi_s^v)$ removes redundant parents of s from φ_s^v.

Definition 7. Let $D = (S, L, C)$ be an ADF. The set of nai_2 interpretations of D is recursively defined as follows:

$$nai_2(D) = nai_2(ind_D(\emptyset), D) \text{ where for } M \subseteq S\text{:}$$
$$nai_2(M, D) = nai(D) \text{ in case } M = S\text{, and otherwise:}$$
$$nai_2(M, D) = \bigcup_{w \in nai(D|_M)} nai_2\big(ind_{[\![D]\!]_M^w}(M), [\![D]\!]_M^w\big)$$

2.2 Complexity theory

Assume some fixed finite vocabulary Σ with $|\Sigma| > 1$. A language $L \subseteq \Sigma^*$ is in P iff it can be recognised by a deterministic Turing machine in polynomial time. Complexity class NP contains all problems L that have a polytime-computable witness relation; that is, $L \in \mathsf{NP}$ iff there are $W_L \in \mathsf{P}$ and $k \in \mathbb{N}$ such that: $x \in L$ iff there is a y such that $(x, y) \in W_L$ and $|y| \leq |x|^k$. For any class $\mathcal{C}$ of languages, its complement class is $\mathsf{co}\mathcal{C} = \{\overline{L} \mid L \in \mathcal{C}\}$. For example, the class coNP contains all languages L whose complement $\overline{L} = \Sigma^* \setminus L$ is in NP. These two classes give rise to the polynomial hierarchy, that can be defined (using oracle Turing machines) as follows: $\Delta_0^P = \Sigma_0^P = \Pi_0^P = \mathsf{P}$, and for $i \geq 0$, $\Delta_{i+1}^P = \mathsf{P}^{\Sigma_i^P}$, $\Sigma_{i+1}^P = \mathsf{NP}^{\Sigma_i^P}$, $\Pi_{i+1}^P = \mathsf{coNP}^{\Sigma_i^P}$. For any complexity class $\mathcal{C}$, a Turing machine with access to a $\mathcal{C}$-oracle can be understood as having a constant-time decision subroutine for problems in $\mathcal{C}$. For each level i of the polynomial hierarchy, the classes Σ_i^P and Π_i^P have canonical complete problems. For Σ_i^P it is as follows: Given a quantified Boolean formula (QBF) $\Phi = \exists P_1 \forall P_2 \exists P_3 \ldots Q_i P_i \psi$, determine whether Φ is true, where $Q_i \in \{\forall, \exists\}$ depending on whether i is even or odd. For Π_i^P the canonical complete problem is similar, but starts with universal quantification.

While these classes are fairly standard, NP and coNP also give rise to the so-called *Boolean hierarchy*. It is rather little-known and defined as follows [Wechsung, 1985]. Firstly, for given complexity classes $\mathcal{C}_1$ and $\mathcal{C}_2$ define the new classes $\mathcal{C}_1 \wedge \mathcal{C}_2 = \{L_1 \cap L_2 \mid L_1 \in \mathcal{C}_1, L_2 \in \mathcal{C}_2\}$ and $\mathcal{C}_1 \vee \mathcal{C}_2 = \{L_1 \cup L_2 \mid L_1 \in \mathcal{C}_1, L_2 \in \mathcal{C}_2\}$. Next, set $\mathsf{C}_0^{\mathsf{BH}} = \mathsf{D}_0^{\mathsf{BH}} = \mathsf{P}$ and for $i \geq 0$ define $\mathsf{C}_{i+1}^{\mathsf{BH}} = \mathsf{coNP} \wedge \mathsf{D}_i^{\mathsf{BH}}$ and $\mathsf{D}_{i+1}^{\mathsf{BH}} = \mathsf{NP} \vee \mathsf{C}_i^{\mathsf{BH}}$.[1] (Intuitively, $\mathsf{C}_i^{\mathsf{BH}}$ is for "conjunction" and $\mathsf{D}_i^{\mathsf{BH}}$ is for "disjunction".) For example, $\mathsf{D}_1^{\mathsf{BH}} = \mathsf{NP}$ and $\mathsf{C}_1^{\mathsf{BH}} = \mathsf{coNP}$, while $\mathsf{D}_2^{\mathsf{BH}} = \mathsf{NP} \vee \mathsf{C}_1^{\mathsf{BH}} = \mathsf{NP} \vee \mathsf{coNP}$ and $\mathsf{C}_2^{\mathsf{BH}} = \mathsf{coNP} \wedge \mathsf{D}_1^{\mathsf{BH}} = \mathsf{coNP} \wedge \mathsf{NP}$. The class $\mathsf{C}_2^{\mathsf{BH}}$ was independently discovered and called DP by Papadimitriou and Yannakakis [1982]. Its complement $\mathsf{coDP} = \mathsf{D}_2^{\mathsf{BH}}$ contains all languages L for which there are $L_1 \in \mathsf{NP}$ and $L_2 \in \mathsf{coNP}$ with $L = L_1 \cup L_2$. The Boolean hierarchy and the polynomial hierarchy are closely interrelated: Chang and Kadin [1996] showed that the polynomial hierarchy collapses (to the third level) if the Boolean hierarchy collapses.

3 Complexity Results

We will consider the following decision problems for any semantics $\sigma \in \{cfi, nai, stg, nai_2\}$.

- Ver_σ: Given an ADF D over S and an interpretation $v : S \to \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$, is $v \in \sigma(D)$?
- Exists_σ: Given an ADF D over S, does there exist a non-trivial interpretation $v \in \sigma(D)$, that is, one with $v(S) \neq \{\mathbf{u}\}$?
- $\mathsf{Cred}_\sigma^{\mathbf{t}}$ / $\mathsf{Cred}_\sigma^{\mathbf{f}}$: Given an ADF D over S and an $s \in S$, does there exist an interpretation $v \in \sigma(D)$ with $v(s) = \mathbf{t}$ / $v(s) = \mathbf{f}$?
- $\mathsf{Scep}_\sigma^{\mathbf{t}}$ / $\mathsf{Scep}_\sigma^{\mathbf{f}}$: Given an ADF D over S and an $s \in S$, is $v(s) = \mathbf{t}$ / $v(s) = \mathbf{f}$ for all $v \in \sigma(D)$?

In several reductions of this paper, we consider quantified Boolean formulas over vocabularies $P \uplus Q$ with their matrix ψ in either DNF (a disjunction of monomials) or CNF (a conjunction of clauses). They will be used to provide hardness results through reducing checking whether the QBF evaluates to true to some relevant problem at hand. Sometimes, we cannot use ψ as is, but have to replace atoms from part of its vocabulary, say P, by new literals from a distinct copy of P, the atoms $P' = \{p' \mid p \in P\}$. We will then denote by ψ' the formula obtained from ψ by replacing all positive occurrences of an atom $p \in P$ by the literal $\neg p'$ for the respective $p' \in P'$. For example, for

[1] This is the Boolean hierarchy between $\Delta_1^P = \mathsf{P}$ and Δ_2^P; there is a Boolean hierarchy between Δ_i^P and Δ_{i+1}^P for all $i \geq 1$ [Chang and Kadin, 1996] (using Σ_i^P and Π_i^P instead of NP and coNP).

the DNF $\psi = (p_1 \wedge q_1 \wedge \neg p_2) \vee (\neg q_2 \wedge \neg p_1 \wedge p_3)$ we get $\psi' = (\neg p_1' \wedge q_1 \wedge \neg p_2) \vee (\neg q_2 \wedge \neg p_1 \wedge \neg p_3')$. The following property of this replacement will be important for us.

Proposition 1. *Let ψ be a DNF over P. For every interpretation $v : P \to \{\mathbf{t}, \mathbf{f}\}$, there exists a $w : P \cup P' \to \{\mathbf{f}, \mathbf{u}\}$ such that ψ^v is a tautology if and only if ψ'^w is a tautology.*

A similar result holds for satisfiability if ψ is in CNF. We are now ready to present the main results of this paper, tight complexity bounds for all semantics among conflict-free, naive, stage and nai_2 for all decision problems introduced above. The results are grouped together in subsections according to the decision problems.

3.1 Interpretation verification

We start out with verifying if a given interpretation is conflict-free. Roughly, this is done using one satisfiability check and one unsatisfiability check, and the completeness result tells us that we most likely cannot do any better.

Proposition 2. Ver_{cfi} *is* DP*-complete.*

Proof. in DP: Let D be an ADF over S and $v : S \to \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$ be an interpretation. To verify that v is conflict-free for D, we have to verify that (1) for all $s \in S$ with $v(s) = \mathbf{t}$, the formula φ_s^v is satisfiable, and (2) for all $s \in S$ with $v(s) = \mathbf{f}$, the formula φ_s^v is unsatisfiable. This can be done in DP by verifying that $\bigwedge_{s \in S, v(s)=\mathbf{t}} \varphi_s^v$ is satisfiable and $\bigvee_{s \in S, v(s)=\mathbf{f}} \varphi_s^v$ is unsatisfiable. Clearly these formulas can be computed in polynomial time.

DP-hard: Let ϕ and ψ be propositional formulas over disjoint vocabularies P_1 and P_2, respectively. We construct an ADF over statements $S = P_1 \cup P_2 \cup \{x, y\}$ and an interpretation $v : S \to \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$ such that v is conflict-free for D if and only if ϕ is satisfiable and ψ is unsatisfiable. Set $\varphi_p = \neg p$ for all $p \in P_1 \cup P_2$, furthermore set $\varphi_x = \phi$ and $\varphi_y = \psi$. Finally, define v such that $p \mapsto \mathbf{u}$ for $p \in P_1 \cup P_2$, and $x \mapsto \mathbf{t}$ and $y \mapsto \mathbf{f}$. □

To showcase the reductions used in the proofs of our results, we present one particular reduction that is used to show the Π_2^P-hardness of most interpretation verification problems.

Reduction 1. Let $\Phi = \exists P \forall Q \psi$ be a QBF with ψ in DNF. Define an ADF D_Φ over $S = P \cup P' \cup Q \cup \{y, z\}$ with:

$$\varphi_p = \neg p \wedge (\neg y \vee z) \qquad \text{for } p \in P$$
$$\varphi_{p'} = \neg p' \wedge (\neg y \vee z) \qquad \text{for } p \in P$$
$$\varphi_q = \neg q \qquad \text{for } q \in Q$$
$$\varphi_y = \neg y \vee \bigwedge_{p \in P} (p \vee p')$$
$$\varphi_z = \neg z \wedge \neg \psi'$$

Here ψ' is ψ where all positive occurrences of p are replaced by $\neg p'$. Finally, define the interpretation $v : S \to \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$ such that $v(y) = \mathbf{t}$ and all other statements are mapped to $\mathbf{u}$.

Intuitively, p and p' serve to guess a valuation for P where setting $p \in S$ to false encodes setting $p \in P$ to false, and setting $p' \in S$ to false encodes setting $p \in P$ to true. All $p, p' \in S$ cannot be set to true, and only be set to false if z is false and y is true; in turn, z can only be set to false if $\neg\psi'$ is unsatisfiable; statement y can only be set to $\mathbf{t}$ or $\mathbf{u}$. Setting y to true in a conflict-free interpretation v guarantees that for each $p \in P$ at most one of p is false or p' is false in v, but never both. These ideas are reused and (sometimes significantly) elaborated upon in later results.

Recall that an interpretation $v : S \to \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$ is naive iff it is conflict-free and $\leq_i$-maximal with respect to being conflict-free. Thus, to verify that a given interpretation v is *not* naive, we first check (using an NP oracle) whether v is conflict-free. If v is not conflict-free, we are done; otherwise, we can guess an interpretation v' with $v <_i v'$ and verify in DP (using the NP-oracle again) that v' is conflict-free. Once more, this is the best we can do.

Theorem 3. Ver_{nai} *is* Π_2^P*-complete.*

For verifying stage interpretations, membership works in the same way as for naive. For hardness, a close look at Reduction 1 reveals that it also works for stage semantics.

Theorem 4. Ver_{stg} *is* Π_2^P*-complete.*

The same hardness reduction (Reduction 1) even works for the nai_2 semantics. It is somewhat harder to show containment in Π_2^P via a reduction to Ver_{nai}: intuitively, this is done by parallelising the single (independent) verifications of nai interpretations in all SCCs of a given ADF D.

Theorem 5. Ver_{nai_2} *is* Π_2^P*-complete.*

Proof. in Π_2^P: We show a reduction to Ver_{nai}. Let D be an ADF over S and the interpretation $v : S \to \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$ be given. We recursively compute the unique decomposition of D with respect to v. In the following we denote the independent sets for each recursive call by M_i for $0 \leq i < n$, that is, $M_0 = ind_D(\emptyset)$ and $M_{i+1} = ind_{[\![D]\!]_{M_i}^{v_i}}(M_i)$ with $v_i = v|_{M_i}$. In each recursive call we make a new, distinct copy D_i of the ADF $D|_{M_i}$ and the respective restricted interpretation $v_i = v|_{M_i}$, that is, for $0 \leq i < n$ define an ADF $D_i = (S_i, L_i, C_i)$ with statements $S_i = \{s_i \mid s \in M_i\}$, links $L_i = \{(s_i, t_i) \mid s_i, t_i \in S_i, (s, t) \in L\}$, acceptance formulas $\varphi_{s_i} = \varphi_s[s/s_i : s \in M_i]$ and furthermore an interpretation $w_i : S_i \to \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$ with $w_i(s_i) = v_i(s)$ for all $s \in M_i$. Let $M_k = S$ be the independent set for the last recursive call. (Clearly $k < n$.) Now we have that $v \in nai_2(D)$ if and only if $w' \in nai(D')$, where

$$w' = \bigcup_{i=0}^{k} w_i \quad \text{and} \quad D' = \bigcup_{i=0}^{k} D_i$$

The computation of D' can be done in at most n steps (for $|S| = n$) with at most $\frac{n(n+1)}{2}$ statements in D'. □

3.2 Existence of non-trivial interpretations

Deciding whether a given ADF D has at least one non-trivial conflict-free interpretation turns out to be complete for the less well-known complexity class coDP. Intuitively, a coDP-problem allows us to choose whether we "want" to solve an NP- or a coNP-problem, but we have to solve at least one of them correctly. Showing coDP-hardness for Exists_{cfi} is comparably easy. The canonical coDP-complete problem is the following, SAT-OR-UNSAT: Given two propositional formulas ϕ and ψ, is ϕ satisfiable or ψ unsatisfiable? Note that the "or" is to be understood logically, that is, it suffices to answer at least one of the questions correctly. The reduction from SAT-OR-UNSAT to Exists_{cfi} now works as follows: given two propositional formulas ϕ and ψ over vocabularies P_1 and P_2, we construct an ADF D over $S = P_1 \cup P_2 \cup \{y, z\}$ with $\varphi_p = \neg p$ for $p \in P_1 \cup P_2$, $\varphi_y = \neg y \vee \phi$, and $\varphi_z = \neg z \wedge \psi$. It is easy to see that D has a non-trivial conflict-free interpretation v with $v(y) = \mathbf{t}$ iff ϕ is satisfiable, and that D has a non-trivial conflict-free interpretation v with $v(z) = \mathbf{f}$ iff ψ is unsatisfiable. In combination, D has a non-trivial conflict-free interpretation iff ϕ is satisfiable or ψ is unsatisfiable.

Showing membership of Exists_{cfi} for coDP is quite tricky. The first useful observation is that there are essentially only two distinct types of non-trivial conflict-free interpretations:

(1) those $v : S \to \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$ with $v^{-1}(\mathbf{t}) \neq \emptyset$, that is, where some statement is mapped to true;

(2) those with $v(S) \subseteq \{\mathbf{u}, \mathbf{f}\}$ and $v^{-1}(\mathbf{f}) \neq \emptyset$, that is, all statements are mapped to undefined or false and at least one is mapped to false.

The proof works by showing that existence of non-trivial conflict-free interpretations of type (1) can be decided in NP, and that the existence of those of type (2) can be decided in coNP. In combination, membership for coDP follows.

Showing the first part is straightforward: to decide whether some statement $s \in S$ can be set to true without violating conflict-freeness, we construct the propositional formula $\bigvee_{s \in S} \varphi_s^{\{s \mapsto \mathbf{t}\}}$ and check if it is satisfiable. If for some $s \in S$ the formula $\varphi_s^{\{s \mapsto \mathbf{t}\}}$ is satisfiable, then $v : S \to \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$ with $v(s) = \mathbf{t}$ and $v(s') = \mathbf{u}$ for $s' \in S \setminus \{s\}$ is conflict-free. Otherwise, no $s \in S$ can be set to true in a conflict-free way.

Showing the second part about interpretations $v : S \to \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$ with $v(S) \subseteq \{\mathbf{u}, \mathbf{f}\}$ (we call them $\mathbf{uf}$-interpretations) constitutes the main portion of the proof. Roughly, conflict-free $\mathbf{uf}$-interpretations are closed under the greatest lower bound operator $\sqcup_i$ associated to the information ordering $\leq_i$ on interpretations. That is, whenever v_1 and v_2 are $\mathbf{uf}$-interpretations that are conflict-free for D, then the interpretation $v_1 \sqcup_i v_2$ is a $\mathbf{uf}$-interpretation that is conflict-free for D as well. Since both $v_1 \leq_i v_1 \sqcup_i v_2$ and $v_2 \leq_i v_1 \sqcup_i v_2$ by definition, there is a unique $\leq_i$-greatest conflict-free $\mathbf{uf}$-interpretation $v_{\max} : S \to \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$ of D. Our task is to decide whether $v_{\max}$ is non-trivial. We first show how to do this by computing $v_{\max}$ in polynomial time using an NP oracle. The procedure works constructively and begins with the interpretation $v_0 : S \to \{\mathbf{f}\}$, that is, mapping all statements to false. The computation now stepwise $(j = 0, 1, \ldots, n-1)$ reassigns $v_{j+1}(s) = \mathbf{u}$ for $s \in v_j^{-1}(\mathbf{f})$ whenever it is the case that assigning $v_j(s) = \mathbf{f}$ is actually not justified because $\varphi_s^{v_j}$ is satisfiable (otherwise, it keeps $v_{j+1}(s) = v_j(s) = \mathbf{f}$). To answer the satisfiability queries, the procedure can use the NP-oracle. The proof finally shows how to combine all the oracle queries into one satisfiability check. This is done by encoding the whole computation into a propositional formula ϕ_{cfi} of polynomial size such that the formula is satisfiable if and only if there is a possible computation that starts with $v_0(S) = \{\mathbf{f}\}$ and ends in the trivial $v_n(S) = \{\mathbf{u}\}$. Since such a computation would show that $v_{\max}$ is trivial, there is a non-trivial conflict-free $\mathbf{uf}$-interpretation of D if and only if the formula ϕ_{cfi} is unsatisfiable. This then shows containment in coNP for checking whether there is a non-trivial conflict-free interpretation of type (2), and thus concludes the coDP-containment proof.

Theorem 6. *Exists_{cfi} is coDP-complete.*

Fortunately, this amount of work "pays off" in that deciding the existence of non-trivial conflict-free interpretations also decides the existence of naive, stage and nai_2 interpretations. The first technical result towards establishing that is the following lemma. It shows how every conflict-free interpretation v gives rise to a naive (or stage) interpretation v' that is "above v" with respect to some ordering. In case of naive, the ordering is clearly the information ordering $\leq_i$. In case of stage, the ordering is given by comparing the statements that are assigned the truth value $\mathbf{u}$ by the two interpretations.

Lemma 7. *Let D be an ADF over S. For every interpretation $v : S \to \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$ that is conflict-free for D, there exists:*

1. *a naive interpretation $v' : S \to \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$ with $v \leq_i v'$;*
2. *a stage interpretation $v'' : S \to \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$ with $v''_{\mathbf{u}} \subseteq v_{\mathbf{u}}$.*

The lemma can be used to show that not only do the non-trivial existence problems coincide, but also credulous entailment for conflict-free and naive semantics are equivalent. Intuitively, if an ADF D has a conflict-free interpretation v with, say, $v(s) = \mathbf{t}$, then the lemma above guarantees the existence of a naive w with $v \leq_i w$ and thus $w(s) = v(s) = \mathbf{t}$.

Proposition 8. *The following decision problems coincide:*

1. Exists_{cfi}, Exists_{nai}, Exists_{stg}, Exists_{nai_2};
2. $\mathsf{Cred}^{\mathbf{t}}_{cfi}$ *and* $\mathsf{Cred}^{\mathbf{t}}_{nai}$;
3. $\mathsf{Cred}^{\mathbf{f}}_{cfi}$ *and* $\mathsf{Cred}^{\mathbf{f}}_{nai}$.

3.3 Entailment

While verification is a quite basic reasoning task, and non-trivial interpretation existence is mostly used to figure out if a given knowledge base is sensible at all, the entailment problem is most likely to be repeatedly used in practice. Recalling that ADFs are intended for modelling argumentation scenarios, entailment queries then allow to answer questions about these scenarios, such as, "Is it the case that there is one possible interpretation of this scenario where statement a is true?" For the conflict-free semantics, this problem is, while infeasible in a conservative sense, still relatively easy.

Theorem 9. *$\mathsf{Cred}^{\mathbf{t}}_{cfi}$ is NP-complete.*

Astonishingly, for similar questions of the form, "Is it the case that there is one possible (conflict-free) interpretation of this scenario where statement a is *false*?", giving an answer becomes harder! This asymmetry is quite remarkable, and has its cause in the asymmetry of the definition of a conflict-free interpretation: $v : S \to \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$ is conflict-free iff for each $s \in S$ with $v(s) = \mathbf{t}$ the formula φ_s^v is satisfiable, and for each $s \in S$ with $v(s) = \mathbf{f}$ the formula φ_s^v is unsatisfiable. So in one case, there is a satisfiability check, in the other there is an unsatisfiability check. To decide credulous entailment, we clearly have to guess an interpretation $v : S \to \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$ with a desired property (such as $v(s) = \mathbf{t}$ or $v(s) = \mathbf{f}$). And while the witnesses for verifying $v(s) = \mathbf{t}$ can be guessed alongside v, such is not possible when having to verify $v(s) = \mathbf{f}$. Formally, the hardness part of the result below is proved via a reduction from the problem of deciding whether a quantified Boolean formula $\exists P \forall Q \psi$ is true.

Theorem 10. *$\mathsf{Cred}^{\mathbf{f}}_{cfi}$ is Σ_2^P-complete.*

There is a straightforward way to show that a statement $s \in S$ is *not* sceptically entailed as true by an ADF D over S: guess an interpretation $v : S \to \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$ with $v(s) \neq \mathbf{t}$ and show that v is naive. Since Ver_{nai} is in Π_2^P, this straightforward approach yields containment of $\mathsf{Scep}^{\mathbf{t}}_{nai}$ in Π_3^P. Fortunately, there is an easier way: we guess an interpretation $v : S \to \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$ with $v(s) = \mathbf{u}$, and verify (using the NP oracle) that v is conflict-free for D, while the augmented interpretation $v \circ \{s \mapsto \mathbf{t}\}$ is *not* conflict-free for D. Intuitively, this identifies statement $s \in S$ as a "troublemaker", as the one reason that violates conflict-freeness in all interpretations with at least as much information as v. Since among these interpretations at least one must be naive, we have our desired counterexample for sceptical entailment. This yields containment in Π_2^P; as it turns out, that is the best possible bound.

Theorem 11. *$\mathsf{Scep}^{\mathbf{t}}_{nai}$ is Π_2^P-complete.*

The straightforward approach to decide sceptical entailment of truth clearly also works for sceptical entailment of falsity. In this case, however, it turns out that there is no shortcut. For the (quite technical) proof of the result, we adapt and combine proof techniques from [Strass and Wallner, 2014, Theorem 20] and Theorem 3.

Theorem 12. *$\mathsf{Scep}^{\mathbf{f}}_{nai}$ is Π_3^P-complete.*

For naive semantics, we have seen (1) an asymmetry in deciding (credulous/sceptical) truth and falsity; and (2) a steady rise in complexity from credulous truth up to sceptical falsity. For stage semantics, surprisingly, these differences vanish: All four decision problems are (more or less) equally hard, namely in the third level of the polynomial hierarchy. For the first problem, this is shown by considering QBFs $\exists P \forall Q \exists R \psi$.

Theorem 13. *$\mathsf{Cred}^{\mathbf{t}}_{stg}$ is Σ_3^P-complete.*

For hardness of deciding credulous falsity, we can use a simple extension of the hardness construction used above: basically, the construction relies on a statement y that can be set to $\mathbf{u}$ if the given QBF $\exists P \forall Q \exists R \psi$ is true, and must be set to $\mathbf{f}$ otherwise (due to the inherent $\subseteq$-minimisation of $v_{\mathbf{u}}$ in stage semantics). The actual reduction now works over a statement z with acceptance formula $\varphi_z = y$; consequently, z can be set to true iff y can be set to $\mathbf{u}$. In the extended construction below, we now add another statement a with acceptance formula $\varphi_a = \neg z$. Both statements will always be assigned opposite truth values from $\{\mathbf{t}, \mathbf{f}\}$, thus proving the next result.

Proposition 14. *$\mathsf{Cred}^{\mathbf{f}}_{stg}$ is Σ_3^P-complete.*

	cfi	*nai*	*stg*	nai_2
Ver_σ	DP-c	Π_2^P-c	Π_2^P-c	Π_2^P-c
Exists_σ	coDP-c	coDP-c	coDP-c	coDP-c
$\mathsf{Cred}^{\mathbf{t}}_\sigma$	NP-c	NP-c	Σ_3^P-c	Σ_3^P-c
$\mathsf{Cred}^{\mathbf{f}}_\sigma$	Σ_2^P-c	Σ_2^P-c	Σ_3^P-c	Σ_3^P-c
$\mathsf{Scep}^{\mathbf{t}}_\sigma$	trivial	Π_2^P-c	Π_3^P-c	Π_3^P-c
$\mathsf{Scep}^{\mathbf{f}}_\sigma$	trivial	Π_3^P-c	Π_3^P-c	Π_3^P-c

Table 1: Complexity results for naive-based semantics of abstract dialectical frameworks; $\mathcal{C}$-c stands for $\mathcal{C}$-complete.

To show that a statement s is *not* sceptically entailed as false in an ADF D, we guess an interpretation $v : S \to \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$ with $v(s) \neq \mathbf{f}$ and verify in Π_2^P that v is stage. This approach is optimal, as completeness shows.

Theorem 15. *$\mathsf{Scep}^{\mathbf{f}}_{stg}$ is Π_3^P-complete.*

In the step from $\mathsf{Scep}^{\mathbf{f}}_{stg}$ to $\mathsf{Scep}^{\mathbf{t}}_{stg}$ we can use the same construction extension as in the step from $\mathsf{Cred}^{\mathbf{t}}_{stg}$ to $\mathsf{Cred}^{\mathbf{f}}_{stg}$.

Proposition 16. *$\mathsf{Scep}^{\mathbf{t}}_{stg}$ is Π_3^P-complete.*

For the nai_2 semantics, we can directly use that the relevant entailment decision problems (or their complements, respectively) are polynomially interreducible.

Proposition 17. *The following problems can be polynomially reduced to each other:*

- $\mathsf{Cred}^{\mathbf{t}}_{nai_2}$ *and* $\mathsf{Cred}^{\mathbf{f}}_{nai_2}$,
- $\mathsf{Scep}^{\mathbf{t}}_{nai_2}$ *and* $\mathsf{Scep}^{\mathbf{f}}_{nai_2}$,
- co-$\mathsf{Scep}^{\mathbf{f}}_{nai_2}$ *and* $\mathsf{Cred}^{\mathbf{t}}_{nai_2}$.

Together with the observation that the hardness reduction of Theorem 12 works for the nai_2 semantics as well, the proposition leads to the following results.

Theorem 18. *$\mathsf{Cred}^{\mathbf{t}}_{nai_2}$ and $\mathsf{Cred}^{\mathbf{f}}_{nai_2}$ are Σ_3^P-complete. $\mathsf{Scep}^{\mathbf{t}}_{nai_2}$ and $\mathsf{Scep}^{\mathbf{f}}_{nai_2}$ are Π_3^P-complete.*

4 Discussion

We presented numerous novel results on the computational complexity of naive-based semantics for abstract dialectical frameworks. An overview is above in Table 1. The main lesson learned is that naive-based semantics for ADFs are – computationally speaking – not at all "naive".

Our analysis paves the way for implementing naive-based ADF semantics, for example by adding adequate ASP encodings for the verification and existence problem to the DIAMOND system [Ellmauthaler and Strass, 2014]. For the sceptical and credulous entailment problems in the third level of the polynomial hierarchy, encodings based on QBFs seem possible [Diller *et al.*, 2014]. In future work, we also intend to identify computationally more amenable fragments; the subclass of *bipolar* ADFs is a promising candidate. Furthermore, the recently introduced stg_2 semantics [Gaggl and Strass, 2014] is as yet unanalysed in terms of complexity.

References

[Amgoud and Cayrol, 2002] Leila Amgoud and Claudette Cayrol. A reasoning model based on the production of acceptable arguments. *Annals of Mathematics and Artificial Intelligence*, 34(1-3):197–215, 2002.

[Amgoud *et al.*, 2008] Leila Amgoud, Claudette Cayrol, Marie-Christine Lagasquie-Schiex, and Pierre Livet. On bipolarity in argumentation frameworks. *International Journal of Intelligent Systems*, 23:1–32, 2008.

[Baroni *et al.*, 2005] Pietro Baroni, Massimiliano Giacomin, and Giovanni Guida. SCC-recursiveness: A general schema for argumentation semantics. *Artificial Intelligence*, 168(1–2):162–210, 2005.

[Bench-Capon, 2003] Trevor J. M. Bench-Capon. Persuasion in practical argument using value-based argumentation frameworks. *Journal of Logic and Computation*, 13(3):429–448, 2003.

[Brewka and Gordon, 2010] Gerhard Brewka and Thomas F. Gordon. Carneades and abstract dialectical frameworks: A reconstruction. In Pietro Baroni, Federico Cerutti, Massimiliano Giacomin, and Guillermo Ricardo Simari, editors, *Proceedings of the Third International Conference on Computational Models of Argument (COMMA)*, volume 216 of *FAIA*, pages 3–12. IOS Press, 2010.

[Brewka and Woltran, 2010] Gerhard Brewka and Stefan Woltran. Abstract Dialectical Frameworks. In Fangzhen Lin, Ulrike Sattler, and Miroslaw Truszczynski, editors, *Proceedings of the Twelfth International Conference on Principles of Knowledge Representation and Reasoning (KR)*, pages 102–111. AAAI Press, 2010.

[Brewka *et al.*, 2013] Gerhard Brewka, Hannes Strass, Stefan Ellmauthaler, Johannes P. Wallner, and Stefan Woltran. Abstract Dialectical Frameworks Revisited. In Francesca Rossi, editor, *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence (IJCAI)*, pages 803–809. IJCAI/AAAI, 2013.

[Chang and Kadin, 1996] Richard Chang and Jim Kadin. The Boolean hierarchy and the polynomial hierarchy: A closer connection. *SIAM Journal on Computing*, 25(2):340–354, 1996.

[Diller *et al.*, 2014] Martin Diller, Johannes P. Wallner, and Stefan Woltran. Reasoning in abstract dialectical frameworks using quantified boolean formulas. In Parsons et al. [2014], pages 241–252.

[Dung, 1995] Phan M. Dung. On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-Person Games. *Artificial Intelligence*, 77(2):321–358, 1995.

[Dunne *et al.*, 2011] Paul E. Dunne, Anthony Hunter, Peter McBurney, Simon Parsons, and Michael Wooldridge. Weighted argument systems: Basic definitions, algorithms, and complexity results. *Artificial Intelligence*, 175(2):457–486, 2011.

[Ellmauthaler and Strass, 2014] Stefan Ellmauthaler and Hannes Strass. The DIAMOND system for computing with abstract dialectical frameworks. In Parsons et al. [2014], pages 233–240.

[Gaggl and Dvořák, 2014] Sarah A. Gaggl and Wolfgang Dvořák. Stage semantics and the SCC-recursive schema for argumentation semantics. *Journal of Logic and Computation*, 2014.

[Gaggl and Strass, 2014] Sarah A. Gaggl and Hannes Strass. Decomposing abstract dialectical frameworks. In Parsons et al. [2014], pages 281–292.

[Li *et al.*, 2011] Hengfei Li, Nir Oren, and Timothy J. Norman. Probabilistic argumentation frameworks. In Sanjay Modgil, Nir Oren, and Francesca Toni, editors, *Revised Selected Papers of the First International Workshop on Theory and Applications of Formal Argumentation (TAFA)*, volume 7132 of *LNCS*, pages 1–16. Springer, 2011.

[Papadimitriou and Yannakakis, 1982] Christos H. Papadimitriou and Mihalis Yannakakis. The complexity of facets (and some facets of complexity). In Harry R. Lewis, Barbara B. Simons, Walter A. Burkhard, and Lawrence H. Landweber, editors, *Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing (STOC)*, pages 255–260. ACM, 1982.

[Parsons *et al.*, 2014] Simon Parsons, Nir Oren, Chris Reed, and Federico Cerutti, editors. *Proceedings of the Fifth International Conference on Computational Models of Argument (COMMA)*, volume 266 of *FAIA*. IOS Press, 2014.

[Polberg and Doder, 2014] Sylwia Polberg and Dragan Doder. Probabilistic abstract dialectical frameworks. In Eduardo Fermé and João Leite, editors, *Proceedings of the Fourteenth European Conference on Logics in Artificial Intelligence (JELIA)*, volume 8761 of *LNCS*, pages 591–599. Springer, 2014.

[Rahwan *et al.*, 2009] Iyad Rahwan, Guillermo R. Simari, and Johan van Benthem. *Argumentation in artificial intelligence*, volume 47. Springer, 2009.

[Strass and Wallner, 2014] Hannes Strass and Johannes P. Wallner. Analyzing the Computational Complexity of Abstract Dialectical Frameworks via Approximation Fixpoint Theory. In Chitta Baral, Giuseppe De Giacomo, and Thomas Eiter, editors, *Proceedings of the Fourteenth International Conference on Principles of Knowledge Representation and Reasoning (KR)*, pages 101–110. AAAI Press, 2014.

[Strass, 2013] Hannes Strass. Approximating operators and semantics for abstract dialectical frameworks. *Artificial Intelligence*, 205:39–70, 2013.

[Verheij, 1996] Bart Verheij. Two approaches to dialectical argumentation: Admissible sets and argumentation stages. In J.-J. Ch. Meyer and L. C. van der Gaag, editors, *Proceedings of the Eighth Dutch Conference on Artificial Intelligence (NAIC)*, pages 357–368, 1996.

[Wechsung, 1985] Gerd Wechsung. On the Boolean closure of NP. In Lothar Budach, editor, *Proceedings of the International Conference on Fundamentals of Computation Theory (FCT)*, volume 199 of *LNCS*, pages 485–493. Springer, 1985.