Chapter 5
Complexity of Abstract Argumentation

Paul E. Dunne and Michael Wooldridge

1 Introduction

The semantic models discussed in Chapter 2 provide an important element of the formal computational theory of abstract argumentation. Such models offer a variety of interpretations for "collection of acceptable arguments" but are unconcerned with issues relating to their implementation. In other words, the extension-based semantics described earlier distinguish different views of what it *means* for a set, S, of arguments to be acceptable, but do not consider the procedures by which such a set might be *identified*.

This observation motivates the study of natural questions relating to the actual implementation of different semantics, e.g., using semantics s what can be stated regarding methods that: decide if $S \in \mathcal{E}_s(\langle \mathcal{A}, \mathcal{R} \rangle)$ for $S \subseteq \mathcal{A}$; or determine if $x \in S$ for at least one (alternatively every) $S \in \mathcal{E}_s(\langle \mathcal{A}, \mathcal{R} \rangle)$, etc? Such questions raise two separate issues: that of *algorithms* by which upper bounds can be obtained; and that of mechanisms by which lower bounds can be established. Some discussion of the former will be given in Chapter 6; the field of *Computational Complexity Theory* provides a number of approaches by which the latter issue can be addressed: these methods and their application within abstract argument systems are the subject of the current chapter. In the next section we give an overview of some basic notions in complexity theory and continue with a review of some fundamental results on complexity in abstract argument systems in Section 3. In Section 4 we consider analogous results within deductive frameworks, *assumption-based frameworks* and a number of complexity-theoretic properties of value-based frameworks. Section 5

Paul E. Dunne
Dept. of Computer Science, University of Liverpool Liverpool UK, e-mail: ped@csc.liv.ac.uk

Michael Wooldridge
Dept. of Computer Science, University of Liverpool Liverpool UK, e-mail: mjw@csc.liv.ac.uk

I. Rahwan, G. R. Simari (eds.), *Argumentation in Artificial Intelligence,*
DOI 10.1007/978-0-387-98197-0_5,

summarises some recent developments concerning novel semantics. Conclusions and selected open questions are discussed in the final section.

2 Elements of Computational Complexity Theory

In crude terms, computational complexity theory deals with classifying computational problems with respect to the resources needed for their solution, e.g., the time required by the fastest program that will solve the problem. In this section we introduce some of the basic concepts in the field of computational complexity.

2.1 Languages and Decision Problems

We think of "computational problems" in terms of *recognising* objects, (e.g., propositional formulae, argumentation frameworks, etc.), which have some *property* of interest, (e.g., instantiations that make the formula true ($\top$), non-empty subsets of arguments that define preferred extensions). For such problems one has a set of *problem instances*, and the goal is to decide whether or not a given instance should be accepted, i.e., has the property of interest. With this approach, *decision problems* are defined by describing the *form* taken by instances *and* the *question* asked of these instances, i.e., the property we want to check. The subset of instances for which positive answers are given is often referred to as a *language*. For example, the decision problem (language) 3-CNF *Satisfiability* (3-SAT) has,

Instance: Propositional formula, $\varphi(x_1,x_2,\ldots,x_n)$ over the variables $\{x_1,\ldots,x_n\}$ in conjunctive normal form with at most three literals in each clause, i.e., φ is specified by a set of *m clauses*, $\{C_1,C_2,\ldots,C_m\}$, with $C_j = y_{j,1} \vee y_{j,2} \vee y_{j,3}$ where $y_{j,k}$ is a literal from $\{x_1,\ldots,x_n,\neg x_1,\ldots,\neg x_n\}$, so that $\varphi = \wedge_{j=1}^{m} C_j$.
Question: Is there an instantiation, $\alpha = \langle a_1,a_2,\ldots,a_n\rangle \in \langle \bot,\top\rangle^n$ for which setting $x_i := a_i$ results in every clause, C_j having at least one literal given the value $\top$?

Notice that this approach allows us to distinguish ideas of problem *size*. Although this could be captured in terms of the number of bits used to encode an instance, there is often some natural parameter that can be used as an alternative, e.g., the size of a 3-CNF formula is usually measured as the number of propositional variables in its definition (n). In general we use $|x|$ to denote the size of a problem instance x.

2.2 Complexity Classes – P, NP, *co*NP *and* PH

The concept of *complexity class* is used to describe problems whose resource requirements are similar. Given a language, L, the problem of deciding whether $x \in L$

is viewed as having an efficient algorithm if there is a *constant value*, k, and program M, for which

- **if** $x \in L$, then M returns "*accept*" **else** M returns "*reject*".
- M returns its answer after at most $|x|^k$ steps.

The program M is said to provide an algorithm for L with run-time n^k, so leading to the complexity class, P, (of *polynomial time decidable* languages) as

$$\text{P} \;=\; \bigcup_{k=0}^{\infty} \{ L \;:\; \text{There is an algorithm with run-time } n^k \text{ deciding } x \in L. \}$$

Note that we generally regard problems as being *computationally easy* or *tractable* if they are polynomial time decidable, although of course, if k is very large, polynomial time decidability may not in fact imply the existence of a *practicable* algorithm to solve the problem.

It is often the case that the question $x \in ?L$ can be phrased in terms of identifying some auxiliary structure (or *witness*) that x is indeed a member of L, e.g., $\varphi \in 3-\text{SAT}$, is witnessed by any instantiation, α, for which $\varphi(\alpha) = \top$. In general, associated with x one may have a set of *possible witnesses*, $W(x)$ to $x \in L$, any such witness having size at most $|x|^r$, for some constant r. Suppose that L_W is the language of pairs $\langle x, y \rangle$ defined by instances, x of L and witnesses $y \in W(x)$ to $x \in L$. Concentrating on languages, L_W in P, we get the complexity classes NP and coNP –

$$\begin{array}{lll} L \in \text{NP} & \text{if} & (x \in L) \;\Leftrightarrow\; \exists\, y \in W(x) \;:\; \langle x, y \rangle \in L_W \\ L \in \text{coNP} & \text{if} & (x \in L) \;\Leftrightarrow\; \forall\, y \in W(x) \;:\; \langle x, y \rangle \notin L_W \end{array}$$

So, for the *complementary* problem to 3-SAT, called 3-UNSAT, $\varphi \in 3-\text{UNSAT}$ if and only if φ has no satisfying instantiation: $3-\text{SAT} \in \text{NP}$ while $3-\text{UNSAT} \in \text{coNP}$.

Looking at the requirements for $L \in \text{P}$, $L \in \text{NP}$, $L \in \text{coNP}$, we note the following pattern: polynomial time decidable languages are characterised by *unary* predicates – P_L – over instances of L, i.e., tests $x \in L$ equate to evaluating the predicate $P_L(x) \;\equiv\; (x \in L)$; languages in NP and coNP are characterised by polynomial time decidable *binary* predicates – $P_L(x, y)$ over instances and possible witnesses so that NP languages are those expressible as $\exists\, y\; P_L(x, y)$ and coNP expressible as $\forall\, y\; P_L(x, y)$. This view naturally suggests extending to $(k+1)$-ary polynomial time decidable predicates $P_L(x, y_1, y_2, \ldots, y_k)$ and the languages characterised as

$$(x \in L) \;\Leftrightarrow\; Q_k\; y_k\; Q_{k-1}\; y_{k-1}\; \cdots\; Q_2\; y_2\; Q_1\; y_1\; P_L(x, y_1, y_2, \ldots, y_k)$$

where $Q_i \in \{\exists, \forall\}$, $Q_i \neq Q_{i+1}$. When $Q_k \;=\; \exists$ (respectively $\forall$) the corresponding class is denoted by Σ_k^p (respectively, Π_k^p). The collection $\cup_{k=0}^{\infty} \Sigma_k^p$ $(= \cup_{k=0}^{\infty} \Pi_k^p)$ is called the *Polynomial Hierarchy* (PH) .

As examples of languages in Σ_k^p and Π_k^p we have the so-called *quantified satisfiability* problems – QSAT_k^{Σ} and QSAT_k^{Π} – whose instances are 3-CNF formulae defined on k disjoint sets of n propositional variables – $X_1, X_2, \ldots, X_k$ – so that

$$\varphi \in \mathrm{QSAT}_k^{\Sigma} \quad \Leftrightarrow \quad \begin{cases} \exists \alpha_1 \forall \alpha_2 \cdots \exists \alpha_{k-1} \forall \alpha_k \ \varphi(\alpha_1, \alpha_2, \ldots, \alpha_k) = \bot & (k \text{ even}) \\ \exists \alpha_1 \forall \alpha_2 \cdots \forall \alpha_{k-1} \exists \alpha_k \ \varphi(\alpha_1, \alpha_2, \ldots, \alpha_k) = \top & (k \text{ odd}) \end{cases}$$

$$\varphi \in \mathrm{QSAT}_k^{\Pi} \quad \Leftrightarrow \quad \begin{cases} \forall \alpha_1 \exists \alpha_2 \cdots \forall \alpha_{k-1} \exists \alpha_k \ \varphi(\alpha_1, \alpha_2, \ldots, \alpha_k) = \top & (k \text{ even}) \\ \forall \alpha_1 \exists \alpha_2 \cdots \exists \alpha_{k-1} \forall \alpha_k \ \varphi(\alpha_1, \alpha_2, \ldots, \alpha_k) = \bot & (k \text{ odd}) \end{cases}$$

It is immediate from the formal definitions of P, NP, coNP and PH that the corresponding sets (of languages) satisfy

$$\mathrm{P} \subseteq \left\{ \begin{array}{c} \mathrm{NP} = \Sigma_1^p \\ \mathrm{coNP} = \Pi_1^p \end{array} \right\} \subseteq \cdots \subseteq \left\{ \begin{array}{c} \Sigma_k^p \\ \Pi_k^p \end{array} \right\} \subseteq \left\{ \begin{array}{c} \Sigma_{k+1}^p \\ \Pi_{k+1}^p \end{array} \right\} \subseteq \cdots$$

It is conjectured that all of these containments are *strict* and that $\Sigma_k^p \neq \Pi_k^p$ for any $k \geq 1$: these generalize the well-known $\mathrm{P} \neq \mathrm{NP}$ conjecture and, to date, are unproven.

2.3 Hardness, Completeness, and Reducibility

The forms presented in Section 2.2 allow problems to be grouped together via *upper bounds*: expressing membership in L in terms of some polynomial time decidable finite arity predicate P_{k+1} places L (at worst) in one of Σ_k^p or Π_k^p. The relationship of *polynomial time many one reducibility* between languages is one key technique underlying arguments that such upper bounds are "optimal". Suppose, given some complexity class, $\mathcal{C}$, we can show that L has the following property:
For *every* $L' \in \mathcal{C}$ there is a *polynomial time* procedure, τ, that transforms instances x of L' to instances $\tau(x)$ of L in such a way that $x \in L'$ *if and only if* $\tau(x) \in L$. We write $L' \leq_m^p L$ to describe this relationship.

What may be said of L in such cases? Certainly, were $L \in \mathrm{P}$ then we could deduce $\mathcal{C} \subseteq \mathrm{P}$: given an instance x of $L' \in \mathcal{C}$, construct the instance $\tau(x)$ of L (polynomial time) and then use the polynomial time method to decide $\tau(x) \in L$. We can thereby deduce that the complexity of L is *at least as high* as the complexity of *any* language in $\mathcal{C}$. A language L for which every $L' \in \mathcal{C}$ has $L' \leq_m^p L$ is said to be $\mathcal{C}$*–hard* . If, in addition, $L \in \mathcal{C}$ then L is called $\mathcal{C}$*–complete* . Just as P is considered as encapsulating all efficiently decidable languages, so the classes of NP–hard, coNP–hard, Σ_k^p–hard and Π_k^p–hard languages are viewed as progressively more and more demanding in terms of their time requirements. Noting the long-standing conjecture about the relationship between these classes a proof that $L \in \mathrm{P}$ is a *positive* statement that L is *tractable*; a proof that L is $\mathcal{C}$-hard for some $\mathcal{C} \in \mathrm{PH}$ (other than P) provides a strong indication that $L \notin \mathrm{P}$, i.e., that L is *intractable*.[1]

While the condition $L' \leq_m^p L$ for *every* $L' \in \mathcal{C}$ may seem somewhat demanding, noting that the relation $\leq_m^p$ is transitive we can replace "$\forall\, L' \in \mathcal{C}\ L' \leq_m^p L$" by "For

[1] It should be noted that $\mathcal{C} \in \mathrm{PH}$ is, almost without exception a class such as Σ_k^p or Π_k^p for some fixed $k > 0$. There are a number of technical consequences which suggest it is extremely unlikely the class PH itself has complete languages, i.e., L such that $\forall L' \in \mathrm{PH}\ L' \leq_m^p L$.

some $\mathcal{C}$-hard, $L' : L' \leq_m^p L$". For the classes introduced in Section 2.2 we have the following results of Cook [10] and Wrathall [45].

Theorem 5.1.

a. *3-*SAT *is* NP*–complete; 3-*UNSAT *is co*NP*–complete.*
b. QSAT$_k^{\Sigma}$ *is* Σ_k^p*–complete;* QSAT$_k^{\Pi}$ *is* Π_k^p*–complete.*

For the results discussed later in this Chapter with very few exceptions the complexity classifications use reductions from 3-SAT or 3-UNSAT.

2.4 More Advanced Ideas

The topics outlined above provide sufficient background for the majority of complexity analyses on decision problems for AFs. There are, however, a number of developments – in particular in the related frameworks discussed in Section 4.1 – which occur in more recent work on complexity of abstract argumentation: here we briefly introduce the basic notions of *oracle-based* complexity classes and the models proposed in work of Cook and Reckhow [11] in order to capture relative complexity of *proof systems*.

Oracle computations, are defined in terms of the availability of a device (or *oracle*) that at the cost of a single step in the algorithm provides the answer to a given language membership query, e.g., oracle computations using 3-SAT may construct a 3-CNF φ, query the 3-SAT oracle as to whether $\varphi \in 3-\mathrm{SAT}$ with the answer determining subsequent steps taken. This notion, when coupled with oracles for NP-complete languages, gives rise to a range of complexity classes differentiated by the particular restrictions placed on the manner in which such calls are made. One important representative of such classes is the so-called *difference* class, D^p formally defined as those languages, L whose members are the intersection of a language $L_1 \in$ NP with a language $L_2 \in$ coNP: a language $L \in$ D^p can thus be decided by a polynomial time algorithm that is allowed to make at most two calls on an NP oracle (which, by virtue of Thm 5.1(a) can be assumed to be 3-SAT): given an instance x of L, test $x \in L_1$ by forming the appropriate 3-SAT instance, $\tau_1(x)$, and calling the oracle; if the response is positive, then $x \in L_2$ is tested in the same way, forming the instance $\tau_2(x)$ using a second oracle call to verify $\tau_2(x) \not\in 3-\mathrm{SAT}$, i.e. $\tau_2(x) \in 3-\mathrm{UNSAT}$. More generally, the class of languages for which polynomial time algorithms using at most $f(|x|)$ calls on some oracle for a language complete in a class $\mathcal{C}$ is denoted by $\mathrm{P}^{\mathcal{C}[f(|x|)]}$ (so that $\mathrm{D}^p \subseteq \mathrm{P}^{\mathrm{NP}[2]}$). Where no restriction is placed on the oracle invocation the notation $\mathrm{P}^{\mathcal{C}}$ is used. As will be seen from the results reviewed in Sect. 4.1, the algorithmic "base class" can be defined within arbitrary complexity classes, not simply P: this leads to classes such as $\mathrm{NP}^{\mathcal{C}}$, etc.

The formalism from [11] has been adopted in order to relate the efficiency of proof procedures for credulous reasoning in AFs to more widely known proof procedures in propositional logic, e.g., resolution, tableau-based, sequents, etc. The model starts from an abstraction of "proof system" Π for a (coNP) language, L as a procedure which given $x \in L$ admits a formal derivation of this fact: the relative efficiency of two processes Π_1 and Π_2 being viewed as the number of derivation steps each requires.[2] This approach precisely formalises two systems as equivalent whenever derivations in one can be simulated by polynomially longer derivations in the other.

3 Fundamental Complexity Results in Argument Frameworks

Faced with a particular semantics and framework there are a number of questions which one may wish to decide: whether a given collection of arguments satisfies the conditions specified by the semantics; whether a particular argument belongs to at least one or every such set; whether there is *any* (non-empty) collectively acceptable subset, etc.. We shall refer to these subsequently as *Verification* (VER$_s$) ; *Credulous Acceptance* (CA$_s$) ; *Sceptical Acceptance* (SA$_s$) ; *Existence* (EX$_s$); and *Non-emptiness* (NE$_s$). Table 5.1 presents the formal definitions.

Table 5.1 Decision Problems in AFs

Problem	Instance	Question
VER$_s$	$\mathcal{G}(\mathcal{A},\mathcal{R})$; $S \subseteq \mathcal{A}$	Is $S \in \mathcal{E}_s(\mathcal{G})$?
CA$_s$	$\mathcal{G}(\mathcal{A},\mathcal{R})$; $x \in \mathcal{A}$	Is there *any* $S \in \mathcal{E}_s(\mathcal{G})$ for which $x \in S$?
SA$_s$	$\mathcal{G}(\mathcal{A},\mathcal{R})$; $x \in \mathcal{A}$	Is x a member of *every* $T \in \mathcal{E}_s(\mathcal{G})$?
EX$_s$	$\mathcal{G}(\mathcal{A},\mathcal{R})$	Is $\mathcal{E}_s(\mathcal{G})$ *non-empty*?
NE$_s$	$\mathcal{G}(\mathcal{A},\mathcal{R})$	Is there any $S \in \mathcal{E}_s(\mathcal{G})$ for which $S \neq \emptyset$?

Before discussing the intractability results that form the main concern of this chapter, we briefly review the cases for which efficient methods are known.

Theorem 5.2.

a. *For s = GR (the grounded semantics), all of the decision problems in Table 5.1 are in* P. *Furthermore the unique subset S for which $S \in \mathcal{E}_{GR}(\langle \mathcal{A},\mathcal{R} \rangle)$ can be* constructed *in polynomial time.*
b. *Given $\langle \mathcal{A},\mathcal{R} \rangle$ and $S \subseteq \mathcal{A}$ deciding if S is conflict-free, admissible, or stable, i.e., the decision problem* VER$_{ST}(\langle \mathcal{A},\mathcal{R} \rangle, S)$*, are all in* P.

[2] In carrying out such comparisons it is presumed that basic derivations in each system are comparable, e.g., can be implemented in polynomial time.

3.1 Intractability Results in Preferred and Stable Semantics

We now turn to the problems defined in Table 5.1 with respect to the other extension based semantics - Preferred and Stable - introduced in Dung [22]. Our main aim is to outline the constructions of Dimopoulos and Torres [21] and Dunne and Bench-Capon [28] from which the classifications shown in Table 5.2 result.

Table 5.2 Complexity of Decision Problems in Preferred (PR) and Stable Semantics (ST)

A	VER$_{PR}$	coNP–complete	[21]
B	CA$_{PR}$	NP–complete	[21]
C	SA$_{PR}$	Π_2^p–complete	[28]
D	NE$_{PR}$	NP–complete	[21]
E	CA$_{ST}$	NP–complete	[21]
F	SA$_{ST}$	coNP–complete/D^p–complete	See discussion below.
G	EX$_{ST}$	NP–complete	attributed to Chvatal in [33]; also [16, 21, 32].

All of the lower bound results in Table 5.2 are obtained as variations on what we shall refer to as the *standard translation* from 3-CNF formulae to AFs.

Definition 5.1. Given $\varphi(z_1,\ldots,z_n)$ a 3-CNF with clauses $\{C_1,\ldots,C_m\}$ the AF, $\mathcal{G}_\varphi(\mathcal{A}_\varphi,\mathcal{R}_\varphi)$ constituting the *standard translation* from φ has

$$\begin{aligned}\mathcal{A}_\varphi &= \{\varphi\}\cup\{C_1,\ldots,C_m\}\cup\{z_1,\ldots,z_n\}\cup\{\neg z_1,\ldots,\neg z_n\}\\ \mathcal{R}_\varphi &= \{\langle C_j,\varphi\rangle \;:\; 1\le j\le m\} \;\cup\; \{\langle z_i,\neg z_i\rangle,\langle\neg z_i,z_i\rangle \;:\; 1\le i\le n\}\\ &\quad\;\cup\,\{\langle y_i,C_j\rangle \;:\; y_i \text{ is a literal (i.e., } z_i \text{ or } \neg z_i\text{) of the clause } C_j\}\end{aligned}$$

The AF described in Defn 5.1 is, modulo some minor simplifications, identical to that originally used in [21]. This framework provides an extremely versatile mechanism that underpins almost all the complexity analyses of extension based semantics in abstract argumentation frameworks.[3] The basic form of the standard translation suffices to establish (B) and (E) of Table 5.2, whereas (A) and (D) follow from quite simple modifications to it.

For example consider the claim in Table 5.2(B) that CA$_{PR}$ is NP–complete. First note that CA$_{PR}$ $\in$ NP since we may use the set of all admissible subsets of $\mathcal{A}$ containing x as witnesses to $\langle\mathcal{G},x\rangle \in$ CA$_{PR}$. We may then use the standard translation to prove 3-SAT $\le_m^p$ CA$_{PR}$: given φ form the instance $\langle\mathcal{G}_\varphi,\varphi\rangle$ of CA$_{PR}$. If $\varphi\in$ 3-SAT then the literals instantiated to $\top$ by a satisfying assignment indicate a subset S of the arguments $\{z_1,\ldots,z_n,\neg z_1,\ldots,\neg z_n\}$ for which $S\cup\{\varphi\}$ is admissible. On the other hand if $\langle\mathcal{G}_\varphi,\varphi\rangle\in$ CA$_{PR}$ then an admissible set containing φ must include a conflict-free subset of literals arguments that collectively attack all of the clause arguments: instantiating the corresponding literals to $\top$ produces a satisfying assignment of φ.

[3] Exceptions are a specialized case of CA$_{PR}$ and a select number of reductions dealing with value-based argumentation frameworks, cf. [25, Thm. 8(a), Thms. 23–25], and later in this chapter.

Adapting the standard translation by adding a new argument, ψ that is attacked by φ and attacks all of the literal arguments yields an AF, $\mathcal{H}_\varphi$ for which (both) $\mathcal{H}_\varphi \in$ NE$_{PR}$ and $\mathcal{H}_\varphi \in$ EX$_{ST}$ hold if and only if $\varphi \in$ 3-SAT. Table 5.2(A) is an immediate consequence of the former property using the special case of verifying if the empty set is a preferred extension.

The discussion above accounts for all the cases given in Table 5.2 with the exceptions of SA$_{PR}$ and SA$_{ST}$. We first address the apparent ambiguity in the classification of SA$_{ST}$ – Table 5.2(F). A further modification to the framework $\mathcal{H}_\varphi$ by which the new argument ψ now attacks *every* argument in $\mathcal{G}_\varphi$, gives an AF in which ψ belongs to every stable extension if and only if $\varphi \in$ 3-UNSAT so giving a coNP-hardness lower bound.[4] In principle, one appears to have a coNP method via "$\langle\langle\mathcal{A},\mathcal{R}\rangle,x\rangle \in \text{SA}_{ST} \quad \Leftrightarrow \quad \forall\ S \subseteq \mathcal{A}\ \ (\text{VER}_{ST}(\langle\mathcal{A},\mathcal{R}\rangle,S)\ \Rightarrow\ (x \in S)$". There is, however, a possible objection: $\langle\mathcal{G},x\rangle \in$ SA$_{ST}$ even when $\mathcal{G}$ has *no* stable extension whatsoever, i.e., $\mathcal{G} \notin$ EX$_{ST}$.[5] In order to deal with this objection one might require as a precondition of $\langle\mathcal{G},x\rangle \in$ SA$_{ST}$ that $\mathcal{G} \in$ EX$_{ST}$ leading to an easy D^p upper bound: positive instances are characterised as those in $\text{CA}_{ST} \cap \{\langle\langle\mathcal{A},\mathcal{R}\rangle,x\rangle\ :\ \forall\ S \subseteq \mathcal{A}\ (\text{VER}_{ST}(\langle\mathcal{A},\mathcal{R}\rangle,S)\ \Rightarrow\ (x \in S)\}$

The matching D^p–hardness lower bound provides another illustration of the flexibility of the standard translation: instances $\langle\varphi_1,\varphi_2\rangle$ of the canonical D^p–hard problem $\langle$3-SAT,3-UNSAT$\rangle$ being transformed to an instance $\langle\mathcal{K},\psi_2\rangle$ of SA$_{ST}$. The construction is illustrated in Fig. 5.1. We leave the reader to verify that this framework satisfies both EX$_{ST}$ *and* has ψ_2 a member of every stable extension if and only if $\varphi_1 \in$ 3-SAT *and* $\varphi_2 \in$ 3-UNSAT.[6]

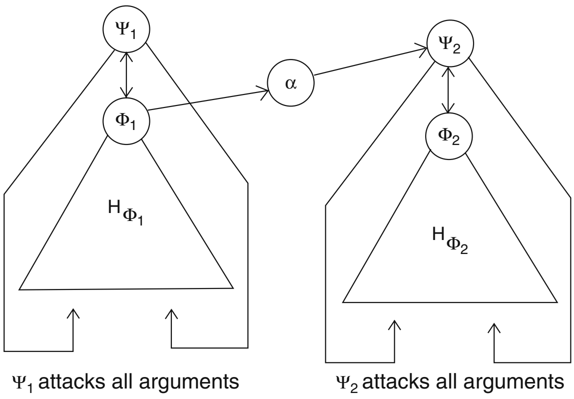

Fig. 5.1 The reduction from $\langle\varphi_1,\varphi_2\rangle \in \langle$3-SAT, 3-UNSAT$\rangle$ to $\langle\mathcal{K},\psi_2\rangle \in$ SA$_{ST}$

[4] In [28] the coNP-hardness result is attributed to Dimopolous and Torres who do not explicitly consider this problem: the commentary of [28, p. 189] observes the lower bound follows from an easy modification to a construction in [21].

[5] Resulting in AFs for which $\langle\langle\mathcal{A},\mathcal{R}\rangle,x\rangle \in$ SA$_{ST}$ and $\langle\langle\mathcal{A},\mathcal{R}\rangle,x\rangle \notin$ CA$_{ST}$ for *every* $x \in \mathcal{A}$, i.e. *every* argument is sceptically accepted but *none* credulously so.

[6] The construction in Fig. 5.1 has not previously appeared in the literature. The issues concerning precise formulations of SA$_{ST}$ appear first to have been raised in [31].

Although the remaining case in Table 5.2 – Π_2^p-completeness of SA_{PR} deals with a, notionally, harder class of languages, again the lower bound follows by adapting the standard translation: in this case to instances $\varphi(y_1,\ldots,y_n,z_1,\ldots,z_n)$ of QSAT_2^{Π}.[7] It is worth noting that the decision problem actually considered via this reduction, in [28], is the so-called *coherence property* of AFs, i.e., whether $\mathcal{G}$ is such that $\mathcal{E}_{PR}(\mathcal{G}) = \mathcal{E}_{ST}(\mathcal{G})$: this problem is shown to be Π_2^p-complete with the classification of SA_{PR} an immediate consequence of the reduction used.

3.2 Dialogue and Relationships to Proof Complexity

The standard translation from 3-CNF (which easily generalises to arbitrary CNF) gives rise to one concrete interpretation of argumentation process in terms of logical proof: in particular, the concept of dialogue based procedures by which two parties attempt to reach agreement on the (credulous) acceptability status of some argument, will be discussed in Chapter 6. If one considers applying such procedures to determining the status of φ (in the standard translation) then a demonstration that φ is *not* admissible corresponds to a formal logical proof that the propositional formula $\neg\varphi$ is a tautology. There are, of course, a number of widely used and well-studied proof mechanisms for propositional logic, the question of interest in terms of complexity in argumentation, is what one can state about the efficiency of dialogue based argumentation processes: that is to say, using the comparative schema proposed in [11], how does the use of dialogue approaches compare to other techniques? This question has been examined with respect to one particular credulous reasoning process: the *two-party immediate response* protocol (TPI) introduced in [44]. Informally, TPI-dialogues involve two protagonists (PRO and CON) debating the acceptability of a given argument x: PRO claiming x to be acceptable and CON adopting the opposite stance. The dialogue is set in the context of an AF where each player takes turns advancing arguments in $\mathcal{A}$: PRO starts by putting forward x. A requirement of the game is that, *whenever possible to do so*, a player must put forward an argument that attacks the most recent argument put forward by their opponent: where this is not possible the player must backtrack to a (specified) earlier point in the discussion or concede. In [29] the number of moves required in this game when played on the standard translation of an *unsatisfiable* CNF is considered. The TPI procedure turns out to be equivalent to a standard propositional proof theory – the so-called CUT-free Gentzen calculus [34] and, as a consequence of [42], there are TPI-disputes requiring exponentially many moves in order to resolve the status of particular arguments.

[7] The reduction originally presented in [28] is not restricted to 3-CNF formulae but describes a general translation from arbitrary propositional formulae over the logical basis $\{\wedge,\vee,\neg\}$.

4 Complexity in Related Abstract Frameworks

As described in several chapters there are a number of abstract treatments of argumentation that build on the basic structures and semantics proposed in Dung [22]. Among such are *assumption-based frameworks* (ABFS) discussed in Chapter 10; the closely related deductive systems considered in Chapter 7; and the value-based argumentation frameworks (VAFS) whose elements have been presented in Chapter 3. Our aim in this section is to review the range of complexity-theoretic results that have been proved within these models. In general we will not give detailed definitions of relevant ideas and refer the reader to the appropriate chapter for these.

4.1 Complexity in Assumption-based Argumentation

Assumption-based frameworks [5], can be interpreted as specific concrete interpretations of abstract argumentation frameworks, i.e. as mechanisms for constructing the structure $\langle \mathcal{A}, \mathcal{R} \rangle$ by generating arguments in $\mathcal{A}$ *and* attacks between these. This approach starts from some *deductive system* – (L,R) in which L is a formal language, e.g., well-formed propositional sentences, and R a set of *inference rules* of the form $\alpha \leftarrow \{\alpha_1,\ldots,\alpha_n\}$ describing the *conclusions* ($\alpha \in L$) that are supported by the *premises* $\{\alpha_1,\ldots,\alpha_n\} \subseteq L$. Such systems have an associated *derivability relation*, $\vdash\ :\ 2^L \rightarrow L$; $\Delta \vdash \alpha$ holds whenever α may be obtained (via R) from Δ. It should be noted that attention is restricted to theories in which the underlying derivability relation is *monotonic*, i.e., $(\Delta \vdash \alpha) \Rightarrow (\Delta' \vdash \alpha)$ for any $\Delta' \supseteq \Delta$.

The key elements added are *assumption sets*, $A \subseteq L$ and the *contrary* function, $\overline{}$ which is a (total) mapping from $\alpha \in A$ to its *contrary* $\overline{\alpha} \in L$. In very simplified terms, (sets of) assumptions define the basis for atomic arguments (in AFS), and the contrary mapping provides the reasons underpinning attacks between arguments. Just as the semantics of “collection of acceptable arguments” in AFS is given by different notions of extension, so too in ABFS the objects of interest are subsets of assumptions defining extensions. In total, viewing an argument as “a statement derivable from some set of assumptions”, leads to the notion of attack between arguments as (the argument) $\Delta \vdash \alpha$ attacks (the argument) $\Delta' \vdash \beta$ if $\Delta \vdash \overline{\gamma}$ for an assumption $\gamma \in \Delta'$. In this way we can take any basic semantics w.r.t. AFS, and define analogues w.r.t. ABFS, e.g., a set of assumptions, Δ, is conflict-free if for every $\alpha \in \Delta$ it is *not* the case that $\Delta \vdash \overline{\alpha}$.

Similarly, one may formulate each of the decision problems of Table 5.1 in ABF settings. There are, however, a number of important distinctions: as a result complexity-theoretic treatments of ABFS use significantly different techniques to those discussed in Section 3. In particular,

a. In AFS both arguments, $\mathcal{A}$, and the attack relation, $\mathcal{R}$, are specified *explicitly*. In ABFS these are *implicit* and dependent on the underlying set of assumptions A *and* the precise deductive theory embodied within (L,R).

b. The deductive system (L,R) is *not* limited to classical propositional logic with the contrary being simply logical negation, e.g., (L,R) and $^{-}$ could be instantiated in terms of a number of non-monotonic logics such as the default logic of [39].

Since the attack relation is defined between assumption sets, in principal one may express any ABF $\langle (L,R), A, ^{-} \rangle$ as an AF, $\langle \mathcal{A}, \mathcal{R} \rangle$: $\mathcal{A} = 2^A$, $\mathcal{R} = \{\langle \Delta, \Delta' \rangle : \Delta \vdash \overline{\gamma}$ for some $\gamma \in \Delta'\}$. There is, however, one complication: in practice not *every* subset of A is of interest, only those that satisfy the technical requirement of being *closed*, i.e., $\Delta \subseteq A$ is closed if and only if $\alpha \in A \setminus \Delta \Rightarrow \neg(\Delta \vdash \alpha)$. While this provides a starting point for algorithms and upper bound constructions, for *lower bounds* such approaches yield little of benefit: $|\langle (L,R), A, ^{-} \rangle|$ is exponentially smaller than the corresponding AF.

A detailed investigation of complexity-theoretic issues within ABFs has been presented in a series of papers by Dimopolous, Nebel and Toni [18, 19, 20]. Using the notation L^{ABF} to distinguish ABF instantiations of decision problems L as presented in Table 5.1 and $L^{\mathrm{ABF,LT}}$ with reference to different formal theories $\mathrm{LT} = (L,R)$, a key element in exact complexity characterisations is the computational complexity of the derivability relation for the underlying logic, i.e., the *derivability problem* (DER) for the formal theory (L,R), has instances $\langle \Delta, \alpha \rangle$ – $\Delta \subseteq L$, $\alpha \in L$ – accepted if and only if $\Delta \vdash \alpha$. For example, in standard propositional logic the derivability problem is coNP–complete: $\Delta \vdash \alpha$ if and only if the formula $\alpha \vee \bigvee_{\varphi \in \Delta} \neg\varphi$ is a tautology.

Combining the notion of “oracle complexity classes” as described in Section 2.4 using oracles for $\mathrm{DER}(\Delta, \alpha)$ provides a generic approach to obtaining *upper bounds* on the complexity of decision problems within ABFs. For example, consider the decision problem $\mathrm{VER}_{ST}^{\mathrm{ABF,LT}}$ of verifying that a given set of assumptions defines a stable extension within $\langle (L,R), A, ^{-} \rangle$, where the derivability problem for LT is in some class $\mathcal{C}$. In order to decide if Δ is accepted:

1. Check that Δ is *closed*.
2. Check that Δ is conflict-free, i.e., $\forall\, \alpha \in \Delta\ \neg(\Delta \vdash \overline{\alpha})$.
3. Check that Δ attacks every assumption $\alpha \notin \Delta$, i.e., $\Delta \vdash \overline{\alpha}$ for each $\alpha \in A \setminus \Delta$.

All of these stages can be carried out using $|A|$ calls to an oracle for DER: (1) tests $(\Delta, \alpha) \notin \mathrm{DER}$ for $\alpha \in A \setminus \Delta$; (2) involves a further $|\Delta|$ calls; and (3) a final set of $|A \setminus \Delta|$ calls. In consequence, $\mathrm{VER}_{ST}^{\mathrm{ABF,LT}} \in \mathrm{P}^{\mathcal{C}}$.

Concentrating on upper bounds for Table 5.1 within the most general settings[8] the upper bounds obtained in [20] are stated in Table 5.3.

It may be noted that with the exception of upper bounds on stability related problems, those relating to preferred and admissible sets of assumptions are rather higher than might be expected having allowed for the additional overhead associated with calls to the DER oracle, e.g., $\mathrm{CA}_{\mathrm{ADM}} \in \mathrm{NP}$ whereas $\mathrm{CA}_{\mathrm{ADM}}^{\mathrm{ABF}} \in \mathrm{coNP}^{\mathrm{NP}^{\mathcal{C}}}$ rather than

[8] That is to say, no specific properties of the underyling frameworks are assumed, e.g., the property “flatness” described in [5].

Table 5.3 Upper bounds for main decision problems in ABFs

Problem	Semantics	Instance (ABF)	ABF bound (DER $\in \mathcal{C}$)	Instance (AF)	AF bound
VER_s	Admissible	$\langle (L,R), A, {}^{-}\rangle$, $\Delta \subseteq A$	$\text{coNP}^{\mathcal{C}}$	$\langle \mathcal{A}, \mathcal{R}\rangle$, $S \subseteq \mathcal{A}$	P
VER_s	Preferred	$\langle (L,R), A, {}^{-}\rangle$, $\Delta \subseteq A$	$\text{coNP}^{\text{NP}^{\mathcal{C}}}$	$\langle \mathcal{A}, \mathcal{R}\rangle$, $S \subseteq \mathcal{A}$	coNP
VER_s	Stable	$\langle (L,R), A, {}^{-}\rangle$, $\Delta \subseteq A$	$\text{P}^{\mathcal{C}}$	$\langle \mathcal{A}, \mathcal{R}\rangle$, $S \subseteq \mathcal{A}$	P
CA_s	Admissible	$\langle (L,R), A, {}^{-}\rangle$, $\varphi \in L$	$\text{NP}^{\text{NP}^{\mathcal{C}}}$	$\langle \mathcal{A}, \mathcal{R}\rangle$, $x \in \mathcal{A}$	NP
CA_s	Preferred	$\langle (L,R), A, {}^{-}\rangle$, $\varphi \in L$	$\text{NP}^{\text{NP}^{\mathcal{C}}}$	$\langle \mathcal{A}, \mathcal{R}\rangle$, $x \in \mathcal{A}$	NP
CA_s	Stable	$\langle (L,R), A, {}^{-}\rangle$, $\varphi \in L$	$\text{NP}^{\mathcal{C}}$	$\langle \mathcal{A}, \mathcal{R}\rangle$, $x \in \mathcal{A}$	NP
SA_s	Preferred	$\langle (L,R), A, {}^{-}\rangle$, $\varphi \in L$	$\text{coNP}^{\text{NP}^{\text{NP}^{\mathcal{C}}}}$	$\langle \mathcal{A}, \mathcal{R}\rangle$, $x \in \mathcal{A}$	Π_2^p
SA_s	Stable	$\langle (L,R), A, {}^{-}\rangle$, $\varphi \in L$	$\text{NP}^{\mathcal{C}}$	$\langle \mathcal{A}, \mathcal{R}\rangle$, $x \in \mathcal{A}$	coNP/D^p

$\text{NP}^{\mathcal{C}}$. That the reasoning problems exhibit "higher than expected" complexity in ABFs is not on account of the (additional) closure checking stage, despite the fact this does not feature in corresponding AF algorithms.[9] The increased complexity arises from the nature of the attack relation, e.g. deciding $\langle \mathcal{G}, x\rangle \in \text{CA}_{PR}$ involves: guess $S \subseteq \mathcal{A}$, confirm that $x \in S$ and S is conflict-free; check S attacks each y that attacks S. Suppose, however, we consider the analogous version for $\text{CA}_{PR}^{\text{ABF,LT}}$: guess $\Delta \subseteq \mathcal{A}$; confirm that Δ is closed, $\Delta \vdash \varphi$, and Δ does not attack itself; finally check that any *closed assumption set* attacking Δ is itself attacked by Δ. This final stage requires tests involving Δ and *all* other *sets* of assumptions, rather than (as is effectively the case in AFs and suffices for stability) checking a property of Δ in relation to *single* assumptions.

We conclude this overview of complexity in ABFs by noting that for the credulous and sceptical reasoning variants, the classifications of Table 5.3 turn out to be optimal for a wide range of instantiations of $\langle (L,R), A, {}^{-}\rangle$ modelling non-classical logics such as DL [39], AEL [38], etc. The typical approach to lower bound proofs, e.g., as illustrated in the specific examples of DL and AEL, uses bounds on the complexity of DER: the cases DL and AEL being coNP–complete. While for DL, one can show $\text{CA}_{PR}^{\text{ABF,DL}} \in \text{NP}^{\text{NP}} = \Sigma_2^p$, no reduction in the generic upper bound is possible for AEL: $\text{CA}_{PR}^{\text{ABF,AEL}} \in \text{NP}^{\text{NP}^{\text{NP}}}$, i.e Σ_3^p. The Σ_2^p (resp. Σ_3^p) hardness reductions use instances of QSAT_2^{Σ} (resp. QSAT_3^{Σ}) to define ABFs instantiated as DL (resp. AEL) systems: detailed constructions may be found in [20].

4.2 Complexity in Value-based Argumentation Frameworks

We recall, from Chapter 3, that *value-based argumentation frameworks* (VAFs) augment the basic $\langle \mathcal{A}, \mathcal{R}\rangle$ abstraction of Dung's AFs by introducing a finite set of *values*, $\mathcal{V}$, and a mapping $\eta : \mathcal{A} \to \mathcal{V}$ describing the abstract value, $\eta(x)$ endorsed by $x \in \mathcal{A}$, so a VAF is described via a four tuple, $\mathcal{G}^{(\mathcal{V})} = \langle \mathcal{A}, \mathcal{R}, \mathcal{V}, \eta\rangle$. In VAFs the underlying structures are the completely abstract frameworks of [22]: whereas ABFs provide

[9] In fact this stage is redundant in a number of ABF models, e.g DL.

a basis for argument *construction* and attacks between arguments, the motivation behind VAFs is to offer an *explanatory* mechanism accounting for choices between distinct justifiable collections, S and T, which are not collectively acceptable, i.e., S and T may be admissible under Dung's semantics, however, $S \cup T$ fails to be. Such occurrences raise the question of the supporting reasons as to which of S or T is adopted: as developed in Chapter 3, VAFs rationalize these choices in terms of *value orderings* on $\mathcal{V}$. Any commitment to a preference of $v_i \in \mathcal{V}$ over $v_j \in \mathcal{V}$ (written $v_i \succ v_j$) induces a simplification of $\langle \mathcal{A}, \mathcal{R}, \mathcal{V}, \eta \rangle$ whereby every attack $\langle x, y \rangle \in \mathcal{R}$ for which $\eta(x) = v_j$ and $\eta(y) = v_i$ can be removed. Under the restrictions discussed in Chapter 3, applying this refinement of $\mathcal{R}$, any total ordering, α of $\mathcal{V}$, will result in an *acyclic* framework, $\mathcal{G}_\alpha^{(\mathcal{V})}$: as has been noted elsewhere such a framework will have $\mathcal{E}_{GR}(\mathcal{G}_\alpha^{(\mathcal{V})}) = \mathcal{E}_{PR}(\mathcal{G}_\alpha^{(\mathcal{V})}) = \mathcal{E}_{ST}(\mathcal{G}_\alpha^{(\mathcal{V})})$. Value orderings thus motivate the two principal decision problems that have been reviewed in algorithmic and complexity studies of VAFs: *Subjective Acceptance* (SBA) and *Objective Acceptance* (OBA) . Both take as an instance a VAF $\mathcal{G}^{(\mathcal{V})}$ and argument $x \in \mathcal{A}$.

$$
\begin{aligned}
\langle \mathcal{G}^{(\mathcal{V})}, x \rangle \in \text{SBA} &\Leftrightarrow \exists\, \alpha \text{ a total ordering of } \mathcal{V} : \text{CA}_{PR}(\mathcal{G}_\alpha^{(\mathcal{V})}, x) \\
\langle \mathcal{G}^{(\mathcal{V})}, x \rangle \in \text{OBA} &\Leftrightarrow \forall\, \alpha \text{ total orderings of } \mathcal{V} : \text{CA}_{PR}(\mathcal{G}_\alpha^{(\mathcal{V})}, x)
\end{aligned}
$$

The acyclic form of $\mathcal{G}_\alpha^{(\mathcal{V})}$ gives $\text{CA}_{PR}(\mathcal{G}_\alpha^{(\mathcal{V})}, x) \in \text{P}$ hence SBA$\in$NP and OBA$\in$coNP.

Both bounds turn out to be exact, as shown in [30, 3]: these again use variants of the standard translation from Defn 5.1. This may appear surprising given that although the standard translation is well-suited to relating *subsets* (of arguments) to instantiations of propositional variables, it is less clear how it could be applied to deal with relating *orderings* of values to such instantiations. The device used in [30] associates a "neutral" value with the formula and clause arguments in the VAF defined from $\varphi(Z_n)$ and replaces the mutually attacking pairs $\{z_i, \neg z_i\}$ with a cycle of four arguments $p_i \rightarrow q_i \rightarrow r_i \rightarrow s_i \rightarrow p_i$. Two arguments ($p_i$ and r_i) are assigned the value pos_i to promote "$z_i = \top$ in a satisfying assignment of φ" while the others (q_i and s_i) are given the value neg_i in order to promote "$z_i = \bot$ in a satisfying assignment of φ". Although their definition and these classifications suggest that SBA (resp. OBA) are closely related to CA_{PR} (resp. SA_{PR} for *coherent* AFs) , recent work, discussed in Section 5 highlights several differences between the nature of decision problems in VAFs and, what appear to be analogous problems in AFs.

5 Recent Developments

The computational complexity of the standard semantics (preferred, stable, grounded) in AFs settings is, in the most general case, now well understood: exact complexity bounds having been established for each of the canonical decision problems given in Table 5.1 with respect to these semantics. There has, however, continued to be extensive development of this aspect of the formal theory of argumentation, driven by a number of reasons. Among these – and forming the topics reviewed in this section – one has: the various proposals for novel extension-based semantics, some of which have been discussed in Chapter 2, e.g., Ideal semantics [23, 24], Semi-stable semantics [6, 7], Prudent semantics [12]. A second consideration concerns the extent to which intractability issues may be alleviated by constructing efficient algorithmic approaches applicable to AFs which are restricted in some way, e.g., by analogy with the known tractable case of *acyclic* topologies.

5.1 Novel extension-based semantics and their complexity

In this section we outline recent treatments of complexity in three of the developments of Dung's standard AF semantics: prudent, ideal, and semi-stable semantics.

We recall that the rationale underlying *prudent semantics* stems from the potential problematic side effects that might eventuate by regarding as collectively acceptable, arguments $\{x,y\}$ for which x "indirectly attacks" y. An indirect attack by x on y is present in $\langle \mathcal{A}, \mathcal{R} \rangle$ if "there exists a finite sequence $x_0, \ldots, x_{2n+1}$ such that (1) $x = x_0$ and $y = x_{2n+1}$ and (2) for each $0 \leq i < 2n$, $\langle x_i, x_{i+1} \rangle \in \mathcal{R}$" [22, p. 332]. It should be noted that this formulation, which we have quoted *verbatim*, presents some ambiguity, which is significant from complexity-theoretic and semantic perspectives: it fails to distinguish "indirect attacks" in which no argument is repeated, i.e., "simple paths"; from those in which *arguments* but not attacks may be repeated; from those in which attacks may be repeated, e.g. the cases in Fig. 5.2.

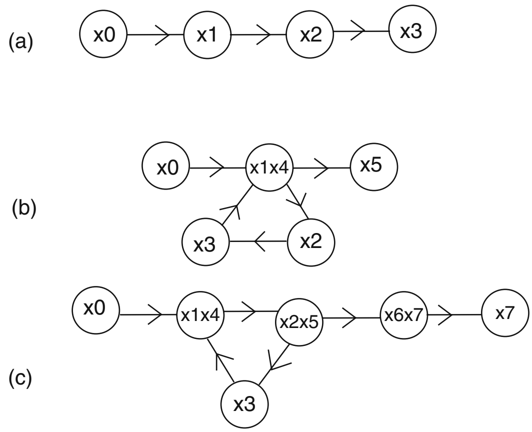

Fig. 5.2 Three possible forms of "indirect" attack – (a) Simple; (b) $x1 = x4$; (c) $\langle x1, x2 \rangle = \langle x4, x5 \rangle$

The concept of conflict-free set from [22], is replaced under the prudent semantics by that of *prudently conflict-free set*, i.e., one in which there is no indirect attack between any two members. There are evident interpretative issues with cases (b) and

(c) in Fig. 5.2, however, the most natural interpretation (where an indirect attack is a simple path) has one significant computational drawback.

Fact 1 *Given $\langle \mathcal{A}, \mathcal{R} \rangle$ and $S \subseteq \mathcal{A}$ deciding if S is prudently conflict-free is co*NP*–complete even if S contains only two arguments.*

Proof. Immediate from the result of Lapaugh and Papadimitriou [35] which shows deciding the existence of a simple *even* length path between two specified arguments in a directed graph to be NP–complete. The extension to *odd* length simple path is trivial, so the lemma follows by observing that a prudently conflict-free set is one in which no simple odd length path is present between two arguments.

Noting the definitions of admissible set, preferred and stable extensions from [22] and the fact that conflict-freeness is an integral part of these, the result of Fact 1 immediately allows us to deduce that under the prudent semantics (so that "conflict-free set" becomes "prudently conflict-free") the respective verification problems are all coNP–complete. Complexity of credulous and sceptical acceptance under the prudent semantics has yet to be studied in depth. The intractability status of key decision problems in this semantics is predicated on the interpretation of "indirect attack" given by (a), i.e., as a simple path. These do not hold if repeated *attacks* – Fig 5.2(c) – are used: here polynomial time methods are available. The status of allowing repeated *arguments* – Fig 5.2(b) – is, to the authors' knowledge still open.

The *ideal semantics* were originally proposed with respect to ABFs, but have a natural formulation in AFs: $S \subseteq \mathcal{A}$ is an ideal *set* within $\langle \mathcal{A}, \mathcal{R} \rangle$ if S is both admissible and a subset of *every* set in $\mathcal{E}_{PR}(\mathcal{A}, \mathcal{R})$; S is an ideal *extension* if it a maximal ideal set. Detailed studies of the complexity of ideal semantics in AFs are presented in [26, 27]. The treatment of complexity issues presented in these papers exploit a number of more advanced techniques, however, the hardness proofs continue to be built on the standard translation of CNF formulae to AFs. In terms of the canonical problems in Table 5.1, the verification problem (for ideal sets) is shown to be coNP-complete, placing this decision problem at the same level of complexity as the verification problem for preferred extensions. Arguably the most radical technique exploited – although widely applied in a number of earlier complexity-theoretic analyses – is the use of *randomized* reductions coupled with structural complexity results from [8, 9], as opposed to standard many-one reducibility ($\leq_m^p$) which features in all of the results discussed earlier.[10] Combining these elements, the verification problem (for ideal extensions) and credulous acceptance problems are shown to be complete for the (conjectured to be) subclass of P^{NP} in which oracle queries are *non-adaptive* (denoted $\mathrm{P}_{||}^{\mathrm{NP}}$). We note that the upper bounds from [26, 27] do not use randomized elements. This complexity class also arises in the known *lower bounds* for both credulous and sceptical acceptance in semi-stable semantics presented in [31]. These lower bounds again apply the structural characterizations of [8] (using $\leq_m^p$ reducibility, i.e., not randomized). Upper bounds, however, are Σ_2^p and Π_2^p, i.e., exact classifications of reasoning problems in semi-stable semantics is open.

[10] Relevant background is outlined in [27] and described in full in [26]. The actual "randomized" element is not explicit but arises from results of [43] for the satisfiability variant used.

5.2 Properties of restricted frameworks

The complexity lower bounds discussed above describe *worst-case* scenarios, i.e., the fact that, for example CA$_{PR}$ is NP–hard, does not imply that every algorithm on every instance will entail unrealistic computational overheads. As will be seen in Chapter 6, if the AF is acyclic, then all of the canonical decision problems of Table 5.1 have polynomial time solutions. In consequence a natural question to consider is whether other graph-theoretic restrictions also result in frameworks with efficient decision processes. Examining this question leads to two classes of results: positive outcomes of the form "decision problem L has a polynomial time algorithm in AFs satisfying some property P"; and negative classifications of the form "decision problem L in frameworks satisfying property P are no easier than the general case". Results of the first type extend (beyond acyclic frameworks) the range of AFs for which tractable solutions exist. Recent work has added to the class of such frameworks: *symmetric* AFs – those for which $\langle x,y\rangle \in \mathcal{R} \Leftrightarrow \langle y,x\rangle \in \mathcal{R}$ – in work of Coste-Marquis *et al.* [13]; *bipartite* AFs (those for which $\mathcal{A}$ may be partitioned into two conflict-free sets) [25]. Using the notion of "treewidth decomposition", see e.g., [4] a select number of problems whose instances are single AFs such as EX$_{ST}$, NE$_{PR}$ admit linear time algorithms given a treewidth decomposition of width k as part of the instance: the construction of these algorithms rely on a deep result (Courcelle's Theorem [14, 15]) demonstrating how efficient algorithms for testing graph-theoretic properties may be obtained given an appropriate logical description of the property (the so-called Monadic Second Order Logic) and a bounded treewidth decomposition of the graph. For more details on this approach and its application in AF settings we refer the reader to [2] and [25].

There are, however, a number of natural properties that fail to yield any reduction in complexity. Typically the approach adopted in proving such results is to demonstrate that frameworks with the property of interest are general enough to effectively "simulate" any framework, e.g., if S is admissible in $\langle\mathcal{A},\mathcal{R}\rangle$ then $S\cup T$ is admissible in $\langle\mathcal{A}\cup\mathcal{B},\mathcal{R}'\rangle$ where the latter AF has a particular property. Using such methods, [25] shows that no reduction in complexity arises in: k-partite AFs ($k \geq 3$); *planar* systems; and those in which no argument attacks or is attacked by more than two other arguments. This remains the case when all restrictions hold simultaneously.

We conclude this overview of recent work by returning to the issue of complexity in VAFs. In contrast to the class of positive cases that have been identified with AFs, the situation with VAFs turns outs out to be far more negative. The most extreme indication of this status is the following result of [25].

Fact 2

a. SBA *is* NP*–complete and* OBA *is co*NP*–complete even if the underlying graph is a* binary tree *and every $v \in \mathcal{V}$ is associated with* at most *three arguments.*
b. *For every $\varepsilon > 0$* SBA *is* NP*–complete and* OBA *is co*NP*–complete even if the underlying graph is a* binary tree *and* $|\mathcal{V}| \leq |\mathcal{A}|^{\varepsilon}$.

Both constructions use reductions from variants of 3-SAT/3-UNSAT, however, these differ in a number of ways from the standard translation (which is clearly is not a binary tree). The situation highlighted by results such as Fact 2 provides further indications that the nature of SBA/OBA in VAFs, while superficially similar to, is in fact radically different from that of CA/SA in AFs.

6 Conclusions and Further Research

In this final section we outline some areas of research which offer a variety of challenging directions through which the algorithmic and complexity foundations of abstract argumentation may be further advanced. We stress that our aim is to focus on general areas rather than particular open questions as such: the reader who has followed the earlier exposition will have noted that a number of specific open issues have already been raised in the text.

6.1 Average case properties

As discussed in Section 5.2, the lower bounds on problem complexity are worst-case, so leaving open the possibility that feasible algorithms may be available in suitable contexts. In addition to the use of restrictions on the form of instances one other approach that has been widely considered in the theory of algorithms is the study of *average-case complexity*. Underpinning this approach one considers a probability distribution, μ, on instances of a decision problem – often, but not invariably so, μ is the uniform distribution whereby each instance is equally likely, proceeding to define the average-case run time of an algorithm P on instances of size n of L as $\sum_{x\in I(n)} \mu(x)\rho(P,x)$ where $\rho(P,x)$ is the run-time of P on instance x. Formal definitions of average-case complexity classes may be found in [36]. To date surprisingly little work has been carried out concerning the application of average-case methods to decision problems in AFs either in terms of algorithmic development or in considering the limitations of such approaches. It remains open to what extent techniques such as those applied to other intractable problems, e.g., [1] for the NP–complete Hamiltonian cycle problem, or [46] for CNF satisfiability could be replicated in AF settings. Of some relevance to such approaches are so-called "phase-transition" effects, which received much attention in the mid-late 1990s as potential indicators of factors separating tractable and intractable classes of problem instances, e.g., the studies of random CNF-SAT from [37, 40]. Analytic studies of such effects appears to indicate connections between suitable witnessing structures, e.g., satisfying assignment, being present "almost certainly" and the performance of algorithms to identify such structures. Of some interest in the context of AF semantics are the results of [41, 17] which give conditions ensuring that a random AF "almost certainly" has a stable extension. There has as yet, however, been no detailed study of the implications of these results for fast on average methods for identifying or enumerating stable extensions. In the same way that the analyses of [41, 17] relate to the existence of stable extensions in AFs, it would be of some interest to examine to consider existence properties of other solution structures in random AFs and algorithmic consequences.

6.2 Approaches to dynamic updates

An important feature of the argumentation forms discussed so far is that, in practice, these are not *static* systems: typically an AF, $\langle \mathcal{A}, \mathcal{R} \rangle$, represents only a "snapshot" of the environment, and, as further facts, information and opinions emerge the form of the initial view may change significantly in order to accommodate these. For example, additional arguments may have to be considered so changing $\mathcal{A}$; existing attacks may cease to apply and new attacks (arising from changes to $\mathcal{A}$) come into force. It is clear that accounting for such dynamic aspects raises a number of issues in terms of assessing the acceptability status of individual arguments. As with the study of average-case properties, the treatment of algorithms and complexity issues relating to determining argument status in dynamically changing environments has been somewhat neglected. Thus, given $\langle \mathcal{A}, \mathcal{R} \rangle$ and $S \subseteq \mathcal{A}$ for which $S \in \mathcal{E}_s(\langle \mathcal{A}, \mathcal{R} \rangle)$ according to some semantics s, natural decision questions are: does $x \in S$ continue to be credulously accepted (w.r.t. to semantics s) in the AF $\langle \mathcal{B}, \mathcal{S} \rangle$ where $\mathcal{B}$ results by removing some arguments from $\mathcal{A}$ and replacing these; similarly $\mathcal{T}$ modifies the attack relation $\mathcal{R}$.

Summary

Complexity issues provide an important foundational element of the formal computational theory of abstract argumentation. Our review of the preceding pages is intended to give a flavour of the class of questions of interest and an appreciation of the techniques that have been brought to bear in addressing these. While some notable progress has been achieved since the appearance of [22] – particularly in understanding of decision properties of the standard semantics and the canonical problems of Table 5.1, nevertheless a significant number of areas and potential analytic tools originating from complexity-theoretic studies, remain unexplored.

References

1. D. Angluin and L. Valiant. Fast probabilistic algorithms for hamiltonian circuits and matchings. *Jnl. of Comp. and System Sci.*, 18:82–93, 1979.
2. S. Arnborg, J. Lagergren, and D. Seese. Easy problems for tree-decomposable graphs. *Jnl. of Algorithms*, 12:308–340, 1991.

3. T. J. M. Bench-Capon, S. Doutre, and P. E. Dunne. Audiences in argumentation frameworks. *Artificial Intelligence*, 171:42–71, 2007.
4. H. L. Bodlaender. A partial *k*-arboretum of graphs with bounded treewidth. *Theoretical Computer Science*, 209:1–45, 1998.
5. A. Bondarenko, P. Dung, R. Kowalski, and F. Toni. An abstract, argumentation-theoretic approach to default reasoning. *Artificial Intelligence*, 93:63–101, 1997.
6. M. Caminada. Semi-stable semantics. In P. E. Dunne and T. J. M. Bench-Capon, editors, *Proc. 1st Int. Conf. on Computational Models of Argument*, volume 144 of *FAIA*, pages 121–130. IOS Press, 2006.
7. M. Caminada. An algorithm for computing semi-stable semantics. In *Proc. of ECSQARU 2007, 9th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, pages 222–234, Hammamet, Tunisia, 2007.
8. R. Chang and J. Kadin. On computing Boolean connectives of characteristic functions. *Math. Syst. Theory*, 28:173–198, 1995.
9. R. Chang, J. Kadin, and P. Rohatgi. On unique satisfiability and the threshold behavior of randomised reductions. *Jnl. of Comp. and Syst. Sci.*, pages 359–373, 1995.
10. S. A. Cook. The complexity of theorem-proving procedures. In *STOC '71: Proc. of the 3rd Annual ACM Symposium on Theory of Computing*, pages 151–158, New York, NY, USA, 1971. ACM.
11. S. A. Cook and R. A. Reckhow. The relative complexity of propositional proof systems. *Journal of Symbolic Logic*, 44(1):36–50, 1979.
12. S. Coste-Marquis, C. Devred, and P. Marquis. Prudent semantics for argumentation frameworks. In *Proc.* 17^{th} *IEEE Intnl.Conf. on Tools with AI (ICTAI 2005)*, pages 568–572. IEEE Computer Society, 2005.
13. S. Coste-Marquis, C. Devred, and P. Marquis. Symmetric argumentation frameworks. In L. Godo, editor, *Proc.* 8^{th} *European Conf. on Symbolic and Quantitative Approaches to Reasoning With Uncertainty (ECSQARU)*, volume 3571 of *LNAI*, pages 317–328. Springer-Verlag, 2005.
14. B. Courcelle. The monadic second-order logic of graphs. I. recognizable sets of finite graphs. *Information and Computation*, 85(1):12–75, 1990.
15. B. Courcelle. The monadic second-order logic of graphs III: tree-decompositions, minor and complexity issues. *Informatique Théorique et Applications*, 26:257–286, 1992.
16. N. Creignou. The class of problems that are linearly equivalent to satisfiability or a uniform method for proving np-completeness. *Theoretical Computer Science*, 145(1-2):111–145, 1995.
17. W. F. de la Vega. Kernels in random graphs. *Discrete Math.*, 82(2):213–217, 1990.
18. Y. Dimopoulos, B. Nebel, and F. Toni. Preferred arguments are harder to compute than stable extensions. In D. Thomas, editor, *Proc. of the 16th International Joint Conference on Artificial Intelligence (IJCAI-99-Vol1)*, pages 36–43, San Francisco, 1999. Morgan Kaufmann Publishers.
19. Y. Dimopoulos, B. Nebel, and F. Toni. Finding admissible and preferred arguments can be very hard. In A. G. Cohn, F. Giunchiglia, and B. Selman, editors, *KR2000: Principles of Knowledge Representation and Reasoning*, pages 53–61, San Francisco, 2000. Morgan Kaufmann.
20. Y. Dimopoulos, B. Nebel, and F. Toni. On the computational complexity of assumption-based argumentation for default reasoning. *Artificial Intelligence*, 141:55–78, 2002.
21. Y. Dimopoulos and A. Torres. Graph theoretical structures in logic programs and default theories. *Theoretical Computer Science*, 170:209–244, 1996.
22. P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming, and *N*-person games. *Artificial Intelligence*, 77:321–357, 1995.
23. P. M. Dung, P. Mancarella, and F. Toni. A dialectical procedure for sceptical assumption-based argumentation. In P. E. Dunne and T. J. M. Bench-Capon, editors, *Proc. 1st Int. Conf. on Computational Models of Argument*, volume 144 of *FAIA*, pages 145–156. IOS Press, 2006.
24. P. M. Dung, P. Mancarella, and F. Toni. Computing ideal sceptical argumentation. *Artificial Intelligence*, 171:642–674, 2007.

25. P. E. Dunne. Computational properties of argument systems satisfying graph-theoretic constraints. *Artificial Intelligence*, 171:701–729, 2007.
26. P. E. Dunne. The computational complexity of ideal semantics. Technical Report ULCS-08-015, Dept. of Comp. Sci., Univ. of Liverpool, August 2008.
27. P. E. Dunne. The computational complexity of ideal semantics I: abstract argumentation frameworks. In *Proc. 2nd Int. Conf. on Computational Models of Argument*, volume 172 of *FAIA*, pages 147–158. IOS Press, 2008.
28. P. E. Dunne and T. J. M. Bench-Capon. Coherence in finite argument systems. *Artificial Intelligence*, 141:187–203, 2002.
29. P. E. Dunne and T. J. M. Bench-Capon. Two party immediate response disputes: properties and efficiency. *Artificial Intelligence*, 149:221–250, 2003.
30. P. E. Dunne and T. J. M. Bench-Capon. Complexity in value-based argument systems. In *Proc. 9th JELIA*, volume 3229 of *LNAI*, pages 360–371. Springer-Verlag, 2004.
31. P. E. Dunne and M. Caminada. Computational complexity of semi-stable semantics in abstract argumentation frameworks. In *Proc. 11th JELIA*, volume 5293 of *LNAI*, pages 153–165. Springer-Verlag, 2008.
32. A. Fraenkel. Planar kernel and grundy with $d \leq 3$, $d_{out} \leq 2$, $d_{in} \leq 2$ are NP–complete. *Discrete Appl. Math.*, 3(4):257–262, 1981.
33. M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of* NP-*Completeness*. W. H. Freeman: New York, 1979.
34. G. Gentzen. Investigations into logical deductions, 1935. In M. E. Szabo, editor, *The Collected Papers of Gerhard Gentzen*, pages 68–131. North-Holland Publishing Co., Amsterdam, 1969.
35. A. S. Lapaugh and C. H. Papadimitriou. The even path problem for graphs and digraphs. *Networks*, 14(4):597–614, 1984.
36. L. Levin. Average case complete problems. *SIAM J. Comput.*, 15:285–286, 1986.
37. D. Mitchell, B. Selman, and H. Levesque. Hard and easy distributions of sat problems. In *Proc. AAAI-92*, pages 459–465. AAAI/MIT Press, 1992.
38. R. C. Moore. Semantical considerations on nonmonotonic logic. *Artificial Intelligence*, 25:75–94, 1985.
39. R. Reiter. A logic for default reasoning. *Artificial Intelligence*, 13:81–132, 1980.
40. B. Selman, H. Levesque, and D. Mitchell. A new method for solving hard satisfiability problems. In *Proc. 10th National Conf. on Art. Intellig.*, pages 440–446, 1992.
41. I. Tomescu. Almost all digraphs have a kernel. *Discrete Math.*, 84(2):181–192, 1990.
42. A. Urquhart. The complexity of Gentzen systems for propositional logic. *Theoretical Computer Science*, 66(1):87–97, 1989.
43. L. G. Valiant and V. V. Vazirani. NP is as easy as detecting unique solutions. *Theoretical Computer Science*, 47:85–93, 1986.
44. G. Vreeswijk and H. Prakken. Credulous and sceptical argument games for preferred semantics. In *Proc. of JELIA'2000, The 7th European Workshop on Logic for Artificial Intelligence.*, pages 224–238, Berlin, 2000. Springer LNAI 1919, Springer Verlag.
45. C. Wrathall. Complete sets and the polynomial-time hierarchy. *Theoretical Computer Science*, 3:23–33, 1976.
46. L. Wu and C. Tang. Solving the satisfiability problem by using randomized approach. *Inf. Proc. Letters*, 41:187–190, 1992.