The Complexity of the Warranted Formula Problem in Propositional Argumentation

Robin Hirsch*
r.hirsch@cs.ucl.ac.uk

Nikos Gorogiannis*
n.gkorogiannis@cs.ucl.ac.uk

6th October 2009

Abstract

The notion of *warrant* or *justification* is one of the central concepts in formal models of argumentation. The dialectical definition of warrant is expressed in terms of recursive defeat: an argument is warranted if each of its counter-arguments is itself defeated by a warranted counter-argument. However, few complexity results exist on checking whether an argument is warranted in the context of deductive models of argumentation, i.e., models where an argument is a deduction of a claim from a set of premises using some logic. We investigate the computational complexity of checking whether a claim is warranted in propositional argumentation under two natural definitions of warrant and show that it is PSPACE-complete in both cases.

1 Introduction

Argumentation, as a subject of research within Artificial Intelligence, is concerned with the study of *arguments* and their interrelationships. For example, an argument may attack another argument thus providing reasons for rejecting the latter. Arguments can be considered as opaque objects to whose internal structure we have no recourse (in which case a set of arguments along with an *attack* relation is an *abstract argumentation framework* [Dun95]), or they can be considered as structures built on top of a deductive system, where the claim of the argument has to be entailed by its premises [BDKT97, BH00, BH01, PWA03, GS04].

Given a set of arguments as a context and an attack relation over that set, the question of whether a particular argument or claim is somehow *acceptable* or *warranted* by prevailing on opposing arguments, is one of the central topics of interest in argumentation. Since a multitude of theories of argumentation exists, several definitions of acceptability of arguments are to be found in the literature. Apart from the theoretical properties that each potential definition

*Department of Computer Science, University College London, Gower Street, London, WC1E 6BT, UK

of acceptability may have, the algorithmic issues pertaining are of interest, especially with a view to applications of argumentation.

Several complexity results exist regarding the notion of acceptability in abstract argumentation frameworks [DBC02, DBC04, Dun06, Dun07], generally placing the complexity of checking various notions of acceptability within the first two levels of the polynomial hierarchy. Also, the complexity of acceptability has been studied in the context of *assumption-based frameworks* [BDKT97], in [DNT99, DNT00, DNT02]. We discuss these results and their relation to deductive and propositional argumentation in Section 6. However, few complexity results exist in relation to the theory of propositional argumentation as set out in [BH00, BH01, PWA03].

The contribution of this paper, therefore, is to study the computational complexity of the *warranted formula problem* which asks whether a claim is warranted in a propositional, deductive argumentation system (see [BH00, BH01, PWA03]). Our main result is that this decision problem, under its usual definition as well as under a useful variation of that, is PSPACE-complete.

The outline is as follows. In Section 2 we introduce the supporting notions and definitions. The simpler decision problem of whether there exists an argument supporting a claim is shown to be Σ_2^p-complete in Section 3. Then, in Sections 4 and 5 we show that the decision problem of checking whether a claim is warranted, under the usual definition and a modified one respectively, is in both cases PSPACE-complete. Finally, we discuss the results and conclude in Section 6.

2 Preliminaries

We will use a propositional language with a countable set of propositional letters. If S is a set of propositional (quantifier-free) formulae and ψ is a propositional formula, we write $S \vdash \psi$ if ψ can be proved from S according to some sound and complete proof system for propositional logic. We will denote by $\bigwedge S$ the conjunction of all formulae in a finite set S of propositional formulae. Also, we use $|\phi|$ for the length of a formula ϕ; when S is a set of formulae, we use $\|S\|$ to denote the sum of the lengths of the formulae in S.

We will also make use of *quantified boolean formulae* and we give the required definitions here. A quantified boolean formula (QBF) is defined recursively as

$$\theta ::= p \mid \neg\theta \mid (\theta_1 \vee \theta_2) \mid \exists p\theta$$

where p is an arbitrary proposition taken from some countable set of propositions. We use standard abbreviations, $\wedge, \rightarrow, \leftrightarrow, \forall$ etc. A *valuation* is a map from the set of propositions to $\{\top, \bot\}$. If p is a proposition, we write $v \equiv_p w$ if v, w are valuations and for all propositions $q \neq p$ we have $v(q) = w(q)$. A quantified boolean formula is *closed* if all propositions occur within the scope of a quantifier. We evaluate the truth of a quantified boolean formula θ with respect to a valuation v, by defining a truth predicate "$\models$", as follows.

$$\begin{aligned} v \models p &\iff v(p) = \top \\ v \models \neg\theta &\iff v \not\models \theta \\ v \models (\theta_1 \wedge \theta_2) &\iff v \models \theta_1 \text{ and } v \models \theta_2 \\ v \models \exists p\theta &\iff w \models \theta, \text{ for some } w \equiv_p v \end{aligned}$$

A quantified boolean formula θ is satisfiable (respectively, valid) iff for some (all) valuation(s) v, we have $v \models \theta$. A closed quantified boolean formula is satisfiable iff it is valid and if this is the case we say that the formula is *true*.

DEFINITION 1. *An instance of the* Quantified Boolean Formula Problem *(***QBF***) is a closed formula*

$$\exists p_0 \forall p_1 \exists p_2 \ldots \exists p_{n-1} \forall p_n \phi$$

for some odd $n \geq 0$ and propositions $p_0, p_1, \ldots, p_n$, where ϕ is an ordinary propositional formula. The yes-instances of **QBF** *are the true formulae of this form.*

It is known that **QBF** is PSPACE-complete (see for example [GJ79, theorem 7.10]. **QBF** instances are not normally required to have strictly alternating quantifiers, nor an even number of quantifiers, nor are the instances normally assumed to be closed, but these restrictions do not affect the complexity as we can always add dummy quantifiers without changing the validity of a formula .

We now review the required definitions on argumentation. The following definition, of a propositional argument, is based on [BH00, BH01, PWA03].

DEFINITION 2. *Let ϕ be a propositional formula and let S be a set of propositional formulae. The pair (S, ϕ) is an* argument for ϕ *(written $A(S, \phi)$) if*

- *S is consistent,*
- *$S \vdash \phi$, and*
- *S is minimal (i.e., no proper subset of S proves ϕ).*

The set S is called the support for the argument, and ϕ its claim.

An instance of the *argument problem* (**ARG**) is a pair (S, ϕ), where a pair (S, ϕ) is a yes-instance iff $A(S, \phi)$. An instance of the *argument existence problem* ($\exists$**ARG**) is a pair (ϕ, Δ), where Δ is a set of formulae. The pair (ϕ, Δ) is a yes-instance of $\exists$**ARG** if there is $S \subseteq \Delta$ and $A(S, \phi)$; it is a no-instance otherwise.

There are different ways of formalising what we mean by a counter-argument. Counter-arguments may contradict the claim of an argument, one of the formulae in the support, or the conjunction of some or all the formulae in the support. This last type of counter-argument is the most general and it is the basis of the definition we adopt here. Since the main topic of this article is complexity, it is reasonable to assume that the knowledgebase Δ is finite, and this allows us to simplify the notion of a counter-argument using the $\bigwedge$ notation. Let (S, ϕ) be an argument (note that S can now be assumed finite). A counter-argument to (S, ϕ) is an argument of the form $(T, \neg \bigwedge S)$. In [BH01] this is called a *canonical undercut*. From now on, we will use the terms counter-argument and undercut interchangeably.

Thus, if there exists an argument (S, ϕ) such that $S \subseteq \Delta$, then S acts as a kind of support for the claim ϕ. We have also a notion of attack, i.e., that an argument $(T, \neg \bigwedge S)$ with $T \subseteq \Delta$, is a canonical undercut for the argument (S, ϕ). A natural extension, built on top of these notions is the notion of *warrant* or *justification* [Pol87, Nut94, BH01, GS04]. In the argumentation literature, warrant status is usually defined in terms of dialectical, or argument, trees. However, in this paper we will give an equivalent recursive definition that will simplify our proofs. Informally, a warranted formula must be supported by an argument and, recursively, if there are any counter-arguments then they are not warranted. Such a recursive definition of whether a claim ϕ is warranted given a set of beliefs Δ is given below.

One problem with such a definition of warrant is that it ostensibly allows infinite chains of arguments and counter-arguments, which unless addressed, leave the status of a formula undetermined. In our definition below, fairly standard in the literature and based on [BH00, BH01], we insist that in a chain of arguments and counter-arguments each argument must involve at least one "new" formula not used in the previous part of the chain.

DEFINITION 3. *Let ϕ be a propositional formula and let U, Δ be sets of formulae such that $U \subseteq \Delta$. We say that "*ϕ is warranted over (Δ, U)*" and we write $W(\phi, \Delta, U)$ iff there is a subset $S \subseteq \Delta$ such that*

- *$S \not\subseteq U$,*
- *$A(S, \phi)$ and*
- *$\neg W(\neg \bigwedge S, \Delta, U \cup S)$.*

The set U can be thought of as the set of 'already used' formulae. An instance (ϕ, Δ) of the *warranted formula problem* (**WFP**) consists of a propositional formula ϕ and a finite set of propositional formulae Δ. It is a yes-instance if ϕ is warranted over $(\Delta, \emptyset)$ and it is a no-instance otherwise. The formula ϕ is called the *claim* and Δ is called the *knowledge base*. It is the complexity of this decision problem, firstly, that we wish to determine, and we do this in Section 4.

The following lemma establishes that no inconsistent formula can be warranted.

LEMMA 4. *If $\phi \vdash \bot$ then $\neg W(\phi, \Delta, U)$, for any Δ, U.*

Proof. Working backwards, if $W(\phi, \Delta, U)$ then there is $S \subseteq \Delta$ with $A(S, \phi)$ so, from definition 2, S is consistent, and by soundness of $\vdash$, ϕ is also consistent. $\square$

Alternative definitions of warranted formulae are possible, particularly when modifying the termination conditions. Here, we present a second definition of warrant, where an undercut is allowed if it is based *entirely* on new formulae.

DEFINITION 5. *An instance* (ϕ, Δ) *of the* alternative warranted formula problem *(which we write as* $\mathbf{WFP}_2$*) consists of a claim* ϕ *and a knowledge base* Δ*, as before.*

(ϕ, Δ) *is a yes-instance of* $\mathbf{WFP}_2$ *if* $W_2(\phi, \Delta)$*, which is defined to hold iff*

$$\exists S \subseteq \Delta \ \left(A(S, \phi) \wedge \neg W_2\left(\neg \bigwedge S, \Delta \setminus S\right)\right)$$

Observe that an argument (S, ϕ) can only be undercut by an argument with a support contained in $\Delta \setminus S$, according to this definition, i.e. a set consisting only of 'unused' formulae. One advantage of this definition is that all tautologies are warranted, since the empty set is permitted as support for an argument for ϕ and there are no possible undercuts for the empty set by this definition (whereas in definition 3, tautologies are not warranted). The complexity of the alternative warranted formula problem is discussed in Section 5.

We recall the required elements of complexity theory. The classes P and NP are as usual, the class of decision problems solvable in polynomial time and non-deterministic polynomial time, respectively. We will denote by **PSAT** the decision problem of ascertaining whether a propositional formula is satisfiable, which is well-known to be NP-complete. PSPACE is the class of decision problems solvable in polynomial space on a deterministic Turing machine. An oracle for a class C can be thought of as a sub-routine that answers a query in C in constant time. We will use oracles to define certain classes in the Polynomial Hierarchy [SM73]. If a *deterministic* Turing machine with access to a C-oracle solves a decision problem in polynomial time then we say that the decision problem is in P^{C}, and similarly if a *non-deterministic* Turing machine with access to a C-oracle solves a decision problem in polynomial time then we say that the decision problem is in NP^{C}. Define the class $\Sigma_2^p = \mathsf{NP}^{\mathsf{NP}}$. The canonical Σ_2^p-complete problem is $\exists\forall$ [Sto76]. An instance of $\exists\forall$ is a quantified boolean formula of the form

$$\exists p_0 \exists p_1 \ldots \exists p_{k-1} \forall q_0 \ldots \forall q_{m-1} \phi$$

where ϕ is a quantifier free, propositional formula. Such a formula is a yes-instance iff it is true.

A PSPACE-complete decision problem that we will employ in our proofs is a variant of the *generalised geography problem.* Normally this decision problem is defined in terms of a winning strategy for the second player in a certain geography game, but here we give an equivalent recursive definition.

DEFINITION 6 (Undirected Edge Geography Problem (**UEGP**))**.** *Let* V *be a set (of vertices). Define the set of* undirected edges $\varepsilon(V)$ *of* V *by*

$$\varepsilon(V) = \{\{v, w\} : v \neq w \in V\}$$

An instance (v, E) *of* **UEGP** *over the finite set* V *consists of a vertex* $v \in V$ *and a subset* E *of* $\varepsilon(V)$*. Such an instance is a yes-instance (and we write* $(v, E) \in \mathbf{UEGP}$*) iff*

$$\forall w \in V[\{v, w\} \in E \rightarrow (w, E \setminus \{\{v, w\}\}) \notin \mathbf{UEGP}]$$

(clearly, since E is finite, this recursive definition is well-founded).

UEGP is known to be PSPACE-complete [FSU93].

To help our inductive proofs, it is convenient to modify this problem by including an extra parameter F representing forbidden edges. An instance (v, E, F) of **UEGP'** over V consists of a vertex $v \in V$ and two subsets $E, F \subseteq \varepsilon(V)$.

$$(v, E, F) \in \textbf{UEGP'} \iff \forall w[\{v, w\} \in E \backslash F \to (w, E, F \cup \{\{v, w\}\}) \notin \textbf{UEGP'}] \tag{1}$$

Evidently, for any sets of edges E, F,

$$(v, E \setminus F) \in \textbf{UEGP} \iff (v, E, F) \in \textbf{UEGP'} \tag{2}$$

3 The Argument Existence Problem

In this section we examine the complexity of the argument existence problem. This result has previously appeared in [PWA03].

LEMMA 7. *The complexity of the argument problem (**ARG**) is in P^{NP}. The argument existence problem ($\exists$**ARG**) is Σ_2^p-complete.*

Proof. Algorithm 1, Argument, decides whether a pair (S, ϕ) is an argument, i.e., whether $A(S, \phi)$ is true. It is a deterministic, linear time algorithm which calls the subroutine Consistent at most $2 + |S|$ times, which computes whether its formula argument is consistent. Propositional consistency is known to be NP-complete, therefore each call takes non-deterministic polynomial time in $\|S\| + |\phi|$. Hence the complexity of Argument is within P^{NP}.

Algorithm 1 Argument(S, ϕ)

```
if not Consistent(⋀ S) then
    return false
end if
if Consistent(⋀ S ∧ ¬φ) then  /* S ⊬ φ */
    return false
end if
for all s ∈ S do
    if not Consistent (⋀(S \ {s}) ∧ ¬φ) then  /* S is not minimal */
        return false
    end if
end for
return true
```

Building on Algorithm 1, Algorithm 2 decides whether a pair (ϕ, Δ) is a yes-instance of $\exists$**ARG**. This algorithm makes a non-deterministic choice of a subset S of Δ and then calls Argument once. As we have seen, running Argument

Algorithm 2 $\exists\mathsf{Argument}(\phi, \Delta)$

Choose $S \subseteq \Delta$
return $\mathsf{Argument}(S, \phi)$

is deterministic and involves a linear number of calls to $\mathsf{Consistent}$. Hence the complexity of $\exists$**ARG** is in $\mathsf{NP}^{\mathsf{NP}} = \Sigma_2^p$.

It remains to prove that $\exists$**ARG** is Σ_2^p-hard. We prove this by reducing $\exists\forall$ to $\exists$**ARG**. As previously mentioned, an instance of $\exists\forall$ is a quantified boolean formula of the form

$$\exists p_0 \exists p_1 \ldots \exists p_{k-1} \forall q_0 \ldots \forall q_{m-1} \phi \tag{3}$$

where ϕ is a quantifier free, propositional formula. We write p for an arbitrary proposition in $\{p_i : i < k\}$ and q for an arbitrary proposition from $\{q_i : i < m\}$. We can assume, by adding additional existential quantifiers if necessary, that all propositions are bound by quantifiers. Moreover, we can also assume that $\{p_0, \ldots, p_{k-1}\} \cap \{q_0, \ldots, q_{m-1}\} = \emptyset$ (just delete $\exists p_i$ if $p_i = q_j$ for some $j < m$). Such an instance is a yes-instance if it is true. The reduction maps such an instance to the instance (ϕ, Δ) of $\exists$**ARG** where

$$\Delta = \{p, \neg p : p \text{ occurs in } \phi\}.$$

We check that the reduction is correct. If (3) is a yes-instance then it is true. That means that there is a valuation v such that $v \models \forall q_0 \ldots \forall q_{m-1}\phi$. It follows, from the completeness of $\vdash$, that ϕ can be proved from $S = \{p : v(p) = \top\} \cup \{\neg p : v(p) = \bot\}$, so S is a consistent set of formulae proving ϕ. The set S might not be minimal, but it must contain a minimal set proving ϕ. Hence (ϕ, Δ) is a yes-instance of $\exists$**ARG**. Conversely, if (ϕ, Δ) is a yes-instance of $\exists$**ARG** then there is $S \subseteq \Delta$ such that $A(S, \phi)$. Now, for any valuation v where $v(p) = \top \iff p \in S$ we know, by the consistency of S, that $v \models \bigwedge S$. By the soundness of $\vdash$ and since $S \vdash \phi$ it follows that $v \models \phi$, for all such valuations. Since $\{q_0, \ldots, q_{m-1}\}$ is disjoint from $\{p_0, \ldots, p_{k-1}\}$, it follows that $v \models \forall q_0 \ldots \forall q_{m-1}\phi$. Hence (3) is a yes-instance of $\exists\forall$. $\square$

This establishes the complexity of $\exists$**ARG**. We turn to the complexity of the warranted formula problem next.

4 The Warranted Formula Problem

First we establish that **WFP** is in PSPACE. This result is related to [BHW08], where Besnard et al use Quantified Boolean Formulae to express several decision problems related to argument trees within a propositional argumentation framework. Although they do not explicitly deal with warrant, a polynomial-time reduction from **WFP** to **QBF** validity is possible in principle, and the algorithms in that paper could perhaps be used to achieve this. Here, we provide a direct algorithm that works within polynomial space.

LEMMA 8. WFP *can be solved in* PSPACE*.*

Proof. Consider Algorithm 3. We prove that this exponential-time deterministic algorithm solves **WFP** and, if implemented correctly, uses polynomial space, by induction over $|\Delta \setminus U|$. We will calculate a bound on the space usage, in terms of $x = |\phi| + \|\Delta\|$.

Algorithm 3 WFP(ϕ, Δ, U)

```
for all S ⊆ Δ do
    if S ⊈ U and A(S, φ) and not WFP(¬⋀S, Δ, U ∪ S) then
        return true
    end if
end for
return false
```

First note, by lemma 7, that for any $S \subseteq \Delta$ we can check $A(S, \phi)$ in polynomial space, since $\mathsf{P}^{\mathsf{NP}} \subseteq \mathsf{PSPACE}$. Let q be a polynomial such that $A(S, \phi)$ can be solved using $q(x)$ space and let $q^+(x) = x + q(x)$.

Claim: The space needed to run algorithm 3 on input (ϕ, Δ, U) is at most $|\Delta| + |\Delta \setminus U| \times q^+(x)$.

The claim is proved by induction over $|\Delta \setminus U|$. The algorithm starts by allocating $|\Delta|$ space to keep a record of which set $S \subseteq \Delta$ has been selected for the current iteration of the loop. This is needed to decide whether to terminate on completing the current iteration and, if not, which set S should be selected for the next iteration. For the base case, $\Delta = U$, each time it enters the loop, the algorithm checks $S \not\subseteq U$ in space $|\Delta|$ which fails immediately (because $S \subseteq \Delta \to S \subseteq U$), and this space for checking $S \subseteq U$ can be released and re-used in the next iteration. The space needed is therefore at most $|\Delta|$.

Now let $k > 0$ and suppose we have proved the claim for all cases where $|\Delta \setminus U| < k$. Suppose we run the algorithm with parameters (ϕ, Δ, U) where $|\Delta \setminus U| = k$. As before, the algorithm first allocates $|\Delta|$ space for keeping a record of the current set S in each iteration of the loop. On entering the loop, the algorithm first checks $S \not\subseteq U$ and $A(S, \phi)$, using space $q(x)$ at most. If it passes these tests it then calls $\mathbf{WFP}(\neg \bigwedge S, \Delta, S \cup U)$. By our induction hypothesis, the space needed for this recursive call is at most $|\Delta| + |\Delta \setminus (S \cup U)| \times q^+(x) \leq |\Delta| + (|\Delta \setminus U| - 1) \times q^+(x)$ (since $S \cup U$ properly contains U in this case). The total space needed is thus

$$
\begin{aligned}
& |\Delta| + q(x) + \big(|\Delta| + (|\Delta \setminus U| - 1) \times q^+(x)\big) \\
\leq \quad & q^+(x) + \big(|\Delta| + (|\Delta \setminus U| - 1) \times q^+(x)\big) \\
\leq \quad & |\Delta| + |\Delta \setminus U| \times q^+(x)
\end{aligned}
$$

as required. This proves that algorithm 3 runs in polynomial space.

We prove that the algorithm is correct by induction over $|\Delta \setminus U|$. Consider the base case, $\Delta = U$. In this case there is no subset S of Δ that is not a subset of U, hence ϕ is not warranted over (Δ, U). For the same reason, the algorithm fails the $S \not\subseteq U$ test each time it enters the loop, and eventually returns **false**, correctly. For the inductive step, the formula ϕ is warranted over (Δ, U) iff there is $S \subseteq \Delta$ such that $S \not\subseteq U$, $A(S, \phi)$ and $\neg W(\neg \bigwedge S, \Delta, U \cup S)$, by definition 3. By the inductive hypothesis, when $S \not\subseteq U$, we have $W(\neg \bigwedge S, \Delta, U \cup S)$ iff the algorithm returns TRUE on input $(\neg \bigwedge S, \Delta, U \cup S)$. Hence $W(\phi, \Delta, U)$ iff there is $S \subseteq \Delta$, $S \not\subseteq U$, $A(S, \phi)$ and $\neg W(\neg \bigwedge S, \Delta, U \cup S)$ iff there is $S \subseteq \Delta$, $S \not\subseteq U$, $A(S, \phi)$ and the algorithm returns FALSE on input $(\neg \bigwedge S, \Delta, U \cup S)$ iff the algorithm returns TRUE on input (ϕ, Δ, U). □

Now we have to prove that **WFP** is PSPACE-hard.

DEFINITION 9 (The Reduction)**.** *Let (v_0, E) be an instance of UEGP over V. We will reuse the vertices in V as propositions. For a set of edges $E \subseteq \varepsilon(V)$, let*

$$\widehat{E} = \{\neg v \vee \neg w, v, w : \{v, w\} \in E\}.$$

We will prove (Theorem 13) that the mapping

$$(v_0, E) \mapsto (v_0, \{v_0\} \cup \widehat{E})$$

is a polynomial time reduction of **UEGP** to **WFP**. The next two lemmas are very easy and we omit the proofs.

LEMMA 10. *The minimal inconsistent subsets of $\widehat{E}$ are*

$$\{\{v, w, \neg v \vee \neg w\} : \{v, w\} \in E\}.$$

LEMMA 11. *Let $S \subseteq \widehat{E}$ and $v \in V$. Then $A(S, \neg v)$ holds if and only if there is $w \in V$ such that $\{v, w\} \in E$ and $S = \{\neg v \vee \neg w, w\}$.*

LEMMA 12. *Let $E, F \subseteq \varepsilon(V)$. Then,*

$$W(\neg w, \widehat{E}, \widehat{F} \cup \{v\}) \iff W(v \vee \neg w, \widehat{E}, \widehat{F} \cup \{v\}).$$

Proof. $W(v \vee \neg w, \widehat{E}, \widehat{F} \cup \{v\})$ holds iff there is a support $S \subseteq \widehat{E}$, such that $A(S, v \vee \neg w)$, $S \not\subseteq \widehat{F} \cup \{v\}$ and $\neg W(\neg \bigwedge S, \widehat{E}, \widehat{F} \cup \{v\} \cup S)$. By lemmas 10 and 11, the only possible supports for the claim $v \vee \neg w$ are $\{v\}$ and sets of the form $\{\neg w \vee \neg u, u\}$, where $\{w, u\} \in \varepsilon(V)$. But $\{v\} \subseteq \widehat{F} \cup \{v\}$, so $\{v\}$ is not allowed as a support for the claim $v \vee \neg w$, and the support $S = \{\neg w \vee \neg u, u\}$ is only allowed if $S \subseteq \widehat{E}$ but $S \not\subseteq \widehat{F} \cup \{v\}$, i.e., if $\{w, u\} \in E \setminus F$. Note that $u \neq v$ because $\{v, \neg w \vee \neg v\}$ is not a minimal subset that entails $v \vee \neg w$. Hence $W(v \vee \neg w, \widehat{E}, \widehat{F} \cup \{v\})$ holds iff

$$\exists u \ \left[\{w, u\} \in E \setminus F \wedge \neg W\left(\neg \bigwedge \{\neg w \vee \neg u, u\}, \widehat{E}, \widehat{F} \cup \{v\} \cup \{\neg w \vee \neg u, u\}\right)\right].$$

But similarly, the only possible supports for the claim $\neg w$ are of the form $\{\neg w \vee \neg u, u\}$ where $\{u, w\} \in E \setminus F$. Thus,

$$\begin{aligned}
&W(v \vee \neg w, \widehat{E}, \widehat{F} \cup \{v\}) \iff \\
&\iff \exists u \Big[\{w, u\} \in E \setminus F \wedge \\
&\qquad\qquad \neg W \left(\neg \bigwedge \{\neg w \vee \neg u, u\}, \widehat{E}, \widehat{F} \cup \{v\} \cup \{\neg w \vee \neg u, u\} \right) \Big] \\
&\iff W(\neg w, \widehat{E}, \widehat{F} \cup \{v\})
\end{aligned}$$

□

Putting everything together, we obtain Theorem 13.

THEOREM 13. *Let $E, F \subseteq \varepsilon(V)$. Then*

$$(v, E, F) \in \textbf{UEGP'} \iff \neg W(\neg v, \widehat{E}, \widehat{F} \cup \{v\})$$

Proof. The proof is by induction over $|E \setminus F|$. The base case, when $E = F$ and thus $|E \setminus F| = 0$, is obvious. The inductive step is proved as follows.

$$\begin{aligned}
&(v, E, F) \in \textbf{UEGP'} \iff \\
&\iff \forall w \left[\{v, w\} \in E \setminus F \to (w, E, F \cup \{\{v, w\}\}) \notin \textbf{UEGP'}\right] && \text{(using (1))} \\
&\iff \forall w \left[\{v, w\} \in E \setminus F \to W\left(\neg w, \widehat{E}, (F \cup \widehat{\{\{v, w\}\}}) \cup \{w\}\right)\right] && \text{(ind. hyp.)} \\
&\iff \neg \exists w \left[\{v, w\} \in E \setminus F \wedge \neg W\left(\neg w, \widehat{E}, (F \cup \widehat{\{\{v, w\}\}})\right)\right] \\
&&& (w \in \widehat{\{\{v, w\}\}},\ \forall \equiv \neg\exists\neg) \\
&\iff \neg \exists w \left[\{v, w\} \in E \setminus F \wedge \neg W\left(v \vee \neg w, \widehat{E}, (F \cup \widehat{\{\{v, w\}\}})\right)\right] \\
&&& (\text{lem. } 12,\ v \in \widehat{\{\{v, w\}\}}) \\
&\iff \neg \exists w \left[\{v, w\} \in E \setminus F \wedge \neg W\left(\neg \bigwedge \{\neg v \vee \neg w, w\}, \widehat{E}, (F \cup \widehat{\{\{v, w\}\}})\right)\right] \\
&&& \text{(prop. log.)} \\
&\iff \neg \exists S \left[A(S, \neg v),\ S \subseteq \widehat{E},\ S \not\subseteq \widehat{F} \cup \{v\},\ \neg W\left(\neg \bigwedge S, \widehat{E}, \widehat{F} \cup \{v\} \cup S\right)\right] \\
&&& \text{(lem. 11)} \\
&\iff \neg W\left(\neg v, \widehat{E}, \widehat{F} \cup \{v\}\right) && \text{(def. 3)}
\end{aligned}$$

□

COROLLARY 14. *The mapping $(v_0, E) \mapsto (v_0, \{v_0\} \cup \widehat{E})$ is a polynomial time reduction of* **UEGP** *to* **WFP**.

Proof. Suppose v_0 is not incident with any edge from E. Then (v_0, E) is clearly a yes-instance of **UEGP**. There is exactly one argument, namely $(\{v_0\}, v_0)$, for v_0 contained in $\{v_0\} \cup \widehat{E}$ and since we are assuming that v_0 is not incident with any edges of E we see by lemma 11 that there are no arguments for $\neg v_0$ in $\{v_0\} \cup \widehat{E}$, hence $W(v_0, \{v_0\} \cup \widehat{E}, \emptyset)$, so the mapping sends a yes-instance to a yes-instance, in this case.

Now suppose v_0 is incident with at least one edge of E. Then $\{v_0\} \cup \widehat{E} = \widehat{E}$. There is exactly one argument for v_0 in $\{v_0\} \cup \widehat{E}$, namely $\{v_0\}$. So,

$$
\begin{aligned}
(v_0, \{v_0\} \cup \widehat{E}) \in \mathbf{WFP} &\iff \neg W(\neg v_0, \{v_0\} \cup \widehat{E}, \{v_0\}) && \text{(def. 3)} \\
&\iff \neg W(\neg v_0, \widehat{E}, \{v_0\}) && (v_0 \in \widehat{E}) \\
&\iff (v_0, E, \emptyset) \in \textbf{UEGP'} && \text{(thm. 13)} \\
&\iff (v_0, E) \in \mathbf{UEGP} && (2)
\end{aligned}
$$

□

COROLLARY 15. WFP *is* PSPACE*-complete.*

5 A Variation of the Warranted Formula Problem

We now turn our attention to the variation of the warranted formula problem given in Definition 5. This version of the problem is typically more stringent since in it, it is required that in a dialogue between opponents, one may only use completely new formulae to propose an undercut. At the same time, this version has the advantage that all tautologies are warranted, which is not the case for Definition 3. It is easy to check that algorithm 4 solves $\mathbf{WFP}_2$ and runs in polynomial space (as in the proof of lemma 8), so $\mathbf{WFP}_2 \in$ NPSPACE = PSPACE.

Algorithm 4 A PSPACE algorithm for $\mathbf{WFP}_2(\phi, \Delta)$

---

for all $S \subseteq \Delta$ **do**
  if $A(S, \phi)$ and $(S = \emptyset$ or $\neg\mathbf{WFP}_2(\neg \bigwedge S, \Delta \setminus S))$ **then**
    return true
  end if
end for
return false

In the rest of this section we present a reduction of the PSPACE-complete problem **QBF** to $\mathbf{WFP}_2$, thus proving that this new problem is also PSPACE-complete. The reduction from the Undirected Edge Geography Problem to **WFP** that we used in the previous section is not a correct reduction from UEGP to $\mathbf{WFP}_2$, because a move in the geography game might follow an edge to a previously visited node n, but the corresponding set of formulas will not be permitted as a counter-argument, because the proposition n has already been used in a previous argument. The authors spent some time attempting to modify this reduction in order to prove the PSPACE-hardness of $\mathbf{WFP}_2$, but we did not succeed.

Instead, we reduce **QBF** to $\mathbf{WFP}_2$. Before we present the reduction itself, we observe a few facts that will simplify our exposition. Given an instance of **QBF**

$$\exists p_0 \forall p_1 \exists p_2 \ldots \exists p_{n-1} \forall p_n \phi \tag{4}$$

where n is odd, we can see that in linear time, using the De Morgan Laws and eliminating double negations, we can replace ϕ by an equivalent formula ϕ_1 in *negation normal form* where the only binary connectives are $\vee$ and $\wedge$ and negation only occurs immediately in front of propositions and ϕ_1 uses only the propositions $p_0, \ldots, p_n$. Henceforth we will assume that ϕ is in negation normal form and only uses these propositions.

We will introduce now the propositions and formulae that support our proposed reduction.

For $i \leq n$, let Π^i be the following set of propositions

$$\Pi^i = \left\{ (+P)^i_j, (-P)^i_j : j \leq i \right\}.$$

Π^i is a language with propositions corresponding to the first $i+1$ propositions of the instance (4). These are meant to capture the positive and negative appearences of a proposition, but without actually employing negations, so that inconsistencies are carefully controlled. The claim and knowledge base will use propositions in

$$\{Q_i, R_i : i \leq n\} \cup \bigcup_{i \leq n} \Pi^i.$$

For $i \leq n$ let

$$\mathsf{val}^i = \bigwedge_{j \leq i} \left((+P)^i_j \vee (-P)^i_j \right).$$

Asserting val^i forces the selection of a set of propositions that correspond to a partial valuation, up to the proposition p_i. For $i < n$ let

$$\mathsf{ext}^i = \left((+P)^{i+1}_{i+1} \vee (-P)^{i+1}_{i+1} \right) \wedge \bigwedge_{j \leq i} \left[\left((+P)^i_j \wedge (-P)^{i+1}_j \right) \vee \left((-P)^i_j \wedge (+P)^{i+1}_j \right) \right].$$

ext^i asserts that there are two partial valuations, one at level i and one at level $i+1$, giving opposite truth values for the first i propositions.

Finally, we will use the following formulae for $i \leq n$.

$$\begin{array}{rcl}
\lambda_i & = & (\neg \mathsf{val}^i \vee (\neg Q_i \wedge \neg R_i)) \\
\rho_i & = & \left\{ \begin{array}{ll} (R_i \wedge (\neg \mathsf{ext}^i \vee \neg Q_{i+1})) & i < n \\ R_n & i = n \end{array} \right.
\end{array}$$

We will say a bit more about the purpose of these formulae just before the proof of Lemma 21.

DEFINITION 16. *Consider an instance (4) of* **QBF**. *We define an instance*

$$(Q_0 \wedge \mathsf{val}^0,\ \Delta)$$

of $\mathbf{WFP}_2$. *The claim is* $(Q_0 \wedge \mathsf{val}^0) = (Q_0 \wedge ((+P)^0_0 \vee (-P)^0_0))$ *and the knowledge base is*

$$\Delta = \bigcup_{i \leq n} \Pi^i \cup \{Q_i, \lambda_i, \rho_i : i \leq n\} \cup \{\neg\phi'\}$$

where ϕ' *is obtained from* ϕ *by replacing each positive occurrence of* p_j *by* $(+P)^n_j$ *and each occurrence of* $\neg p_j$ *by* $(-P)^n_j$, *for* $j \leq n$ *(so occurrences of all propositions in* ϕ' *are positive).*

We introduce some supporting definitions here. For $i \leq n$ let $\Delta^i = \Pi^i \cup \{Q_i, \lambda_i, \rho_i\}$ (so $\Delta = \bigcup_{i \leq n} \Delta^i \cup \{\neg\phi'\}$). Let $\overline{s} = \langle s_j : j \leq n\rangle \in \{+,-\}^{n+1}$ be any sequence of $+$'s and $-$'s of length $n+1$. Let $i \leq n$ and let $\overline{s} \in \{+,-\}^{n+1}$. We write $\overline{s}^i$ for the element of $\{+,-\}^{n+1}$ obtained from $\overline{s}$ by changing the sign of s_i. We may write $\overline{s}^{\,\pm i}$ to denote either $\overline{s}$ or $\overline{s}^i$. For $k \leq n+1$ write $\overline{s}{\restriction}_k$ for the sequence $\langle s_j : j \leq k\rangle$.

Let $v_{\overline{s}}$ be the valuation defined on $\{p_0, \ldots, p_n\}$ by $v(p_j) = \top \iff s_j = +$. Define the sets of propositions

$$\begin{aligned} X_i(\overline{s}) &= \left\{(s_j P)^i_j : j \leq i\right\} \subseteq \Pi^i \\ \overline{X_i}(\overline{s}) &= \left\{(-s_j P)^i_j : j \leq i\right\} = \Pi^i \setminus X_i(\overline{s}) \end{aligned}$$

where $-s_j$ is the 'opposite sign' to s_j. Observe that $X_i(\overline{s})$ is determined by $\overline{s}{\restriction}_i$. We now claim that

$$v_{\overline{s}}(\phi) = \top \iff X_n(\overline{s}) \models \phi' \tag{5}$$

for any $\overline{s} \in \{+,-\}^{n+1}$. To prove this claim, first recall that ϕ is in negation normal form. If ϕ is a literal then there is $j \leq n$ and either $\phi = p_j$ or $\phi = \neg p_j$. In the first case $v_{\overline{s}}(p_j) = \top \iff s_j = + \iff (s_j P)^n_j \in X_n(\overline{s})$ and in the second case $v_{\overline{s}}(\neg p_j) = \top \iff s_j = - \iff (s_j P)^n_j \in X_n(\overline{s})$. Either way, (5) holds. If ϕ is a disjunction, say $\phi = \bigvee_k \phi_k$, then inductively (5) holds for each ϕ_k, so $v_{\overline{s}}(\bigvee_k \phi_k) = \top \iff \exists k\ v_{\overline{s}}(\phi_k) = \top \iff_{(IH)} \exists k\ X_n(\overline{s}) \models \phi_k \iff X_n(\overline{s}) \models \phi$. The case where ϕ is a conjunction is similar (replace $\exists$ by $\forall$). This proves (5).

LEMMA 17. *Let* $i \leq n$, $T \subseteq \Delta$.

$$T \vdash \mathsf{val}^i \iff \exists \overline{s} \in \{+,-\}^{n+1},\ T \supseteq X_i(\overline{s})$$

and for $i < n$

$$T \vdash \mathsf{ext}^i \iff \exists \overline{s} \in \{+,-\}^{n+1},\ T \supseteq X_i(\overline{s}) \cup \overline{X_{i+1}}(\overline{s})$$

Proof. Clearly, $X_i(\overline{s}) \vdash \mathsf{val}^i$, since each of the i conjuncts of val^i is proved by $X_i(\overline{s})$. Conversely, let $T \vdash \mathsf{val}^i$. For each $j \leq i$ we have $T \vdash ((+P)^i_j \vee (-P)^i_j)$. The only formula including a positive occurrence of $(+P)^i_j$ in Δ is the formula

$(+P)^i_j$ itself, and similarly for $(-P)^i_j$. Hence, for each $j \leq i$, either $(+P)^i_j$ or $(-P)^i_j$ belongs to T. Let $\overline{s} \in \{+,-\}^{n+1}$ be an arbitrary sequence, subject to $s_j = + \iff (+P)^i_j \in T$. Then $T \supseteq X_i(\overline{s})$, as required.

Similarly, the right to left implication in the second part is clear. For the left to right implication, suppose $T \vdash \mathsf{ext}^i$. For each $j \leq i$ we have $T \vdash ((+P)^i_j \wedge (-P)^{i+1}_j) \vee ((-P)^i_j \wedge (+P)^{i+1}_j)$ and $T \vdash (+P)^{i+1}_{i+1} \vee (-P)^{i+1}_{i+1}$. The propositions of ext^i only occur positively in formulae in Δ as themselves. Let $\overline{s} \in \{+,-\}^{n+1}$ be arbitrary subject to $s_j = + \iff (+P)^i_j, (-P)^{i+1}_j \in T$, for $j \leq i$, and $s_{i+1} = + \iff (-P)^{i+1}_{i+1} \in T$. As before we get $T \supseteq X_i(\overline{s}) \cup \overline{X_{i+1}}(\overline{s})$. □

LEMMA 18. *The minimal inconsistent subsets of* $\Delta \setminus \{\neg\phi'\}$ *are*

$$X_i(\overline{s}) \cup \{\lambda_i, Q_i\}, \quad X_i(\overline{s}) \cup \{\lambda_i, \rho_i\}$$

for $i \leq n$, $\overline{s} \in \{+,-\}^{n+1}$ *and*

$$\{\rho_i, Q_{i+1}\} \cup X_i(\overline{s}) \cup \overline{X_{i+1}}(\overline{s}), \quad \{\rho_i, Q_{i+1}\} \cup X_i(\overline{s}) \cup \overline{X_{i+1}}(\overline{s}^{i+1})$$

for $i < n$, $\overline{s} \in \{+,-\}^{n+1}$.

If $\neg\phi' \in S$ *and* S *is minimal inconsistent then* $S \subseteq \Pi^n \cup \{\neg\phi'\}$. *A set* $X_n(\overline{s}) \cup \{\neg\phi'\}$ *is inconsistent iff* $v_{\overline{s}} \models \phi$.

Proof. Every subset of $\bigcup_{i \leq n} \Pi^i \cup \{Q_i : i \leq n\}$ is clearly consistent (consider the valuation that makes all propositions true) so any inconsistent subset of $\Delta \setminus \{\neg\phi'\}$ includes λ_i or ρ_i for some $i \leq n$. Suppose $\lambda_i \in S$ and S is minimal inconsistent. Then $S \setminus \{\lambda_i\}$ is consistent and $S \setminus \{\lambda_i\} \vdash \neg\lambda_i \equiv \mathsf{val}^i \wedge (Q_i \vee R_i)$. By lemma 17, since $S \setminus \{\lambda_i\} \vdash \mathsf{val}^i$ we have $S \supseteq X_i(\overline{s})$ for some $\overline{s}$. And since $S \setminus \{\lambda_i\} \vdash (Q_i \vee R_i)$ either $Q_i \in S$ or $\rho_i \in S$ (note that the only positive occurrence of Q_i or R_i is in the formulae Q_i, ρ_i respectively). Thus the minimal inconsistent subsets of $\Delta \setminus \{\neg\phi'\}$ that include λ_i are $X_i(\overline{s}) \cup \{\lambda_i, Q_i\}$ and $X_i(\overline{s}) \cup \{\lambda_i, \rho_i\}$ for $\overline{s} \in \{+,-\}^{n+1}$. Any other minimal inconsistent subset of $\Delta \setminus \{\neg\phi'\}$ must exclude λ_j (all $j \leq n$) but include ρ_i for some $i \leq n$. So suppose $\lambda_j \notin S$ (all j) and $\rho_i \in S$ (some i), $S \setminus \{\rho_i\}$ is consistent and $S \setminus \{\rho_i\} \vdash \neg\rho_i$. Supposing $i = n$, we get that $S \setminus \{\rho_i\} \vdash \neg R_n$ but the only negative occurrence of R_n in any formula in Δ is in λ_n and we are assuming that $\lambda_n \notin S$, so $S \setminus \{\rho_i\} \not\vdash \neg R_n$ and therefore it cannot be that $i = n$. Assume that $i < n$, then $S \setminus \{\rho_i\} \vdash \neg\rho_i \equiv (\neg R_i \vee (\mathsf{ext}^i \wedge Q_{i+1}))$. As before, it must be that $S \setminus \{\rho_i\} \not\vdash \neg R_i$, hence $S \setminus \{\rho_i\} \vdash \mathsf{ext}^i \wedge Q_{i+1}$. By lemma 17, $S \supseteq X_i(\overline{s}) \cup \overline{X_{i+1}}(\overline{s})$, for some $\overline{s}$ and since the only positive occurrence of Q_{i+1} in any formula in Δ is as itself, we also have $Q_{i+1} \in S$. Hence the other minimal inconsistent subsets of $\Delta \setminus \{\neg\phi'\}$ are $\{\rho_i, Q_{i+1}\} \cup X_i(\overline{s}) \cup \overline{X_{i+1}}(\overline{s})$.

If S is minimal inconsistent and $\neg\phi' \in S$ then $S \setminus \{\neg\phi'\}$ is consistent and $S \setminus \{\neg\phi'\} \vdash \phi'$. The propositions in ϕ' appear only positively, and the only positive occurrences of these propositions in Δ occur as themselves and they are found in Π^n. It follows by minimality that $S \subseteq \Pi^n$.

Let $\overline{s} \in \{+,-\}^{n+1}$. Then $X_n(\overline{s}) \cup \{\neg\phi'\}$ is inconsistent iff $X_n(\overline{s}) \vdash \phi'$ iff $v_{\overline{s}} \models \phi$, by (5). □

LEMMA 19. *Let S, T be disjoint, consistent subsets of Δ and suppose that $A(T, \neg \bigwedge S)$. There must be a minimal inconsistent subset $X \subseteq \Delta$ such that $X \cap S, X \setminus S \neq \emptyset$ and $T = X \setminus S$.*

LEMMA 20. *Suppose that there is exactly one set $S \subseteq \Delta$ such that $A(S, \phi)$. Then the following are equivalent: (i) $W_2(\phi, \Delta)$, (ii) $W_2(\bigwedge S, \Delta)$, and (iii) $\neg W_2(\neg \bigwedge S, \Delta \setminus S)$.*

Proof. First we prove the following claim: given the assumptions in the lemma, if for some $T \subseteq \Delta$ it is the case that $A(T, \bigwedge S)$, then $T = S$. To prove the claim, suppose $A(T, \bigwedge S)$. Then, $T \vdash \phi$ will be true, and therefore, there is a minimal set $T' \subseteq T$ such that $T' \vdash \phi$, meaning that $A(T', \phi)$ is true. But then, by the assumption in the lemma, $T' = S$ and, by the requirement of minimality of T in $A(T, \bigwedge S)$, $T = T' = S$. This proves the claim.

Now we prove the lemma.

$$
\begin{aligned}
& W_2(\phi, \Delta) \\
& \iff \exists T \subseteq \Delta,\ A(T, \phi) \wedge \neg W_2(\neg \bigwedge T, \Delta \setminus T) && \text{(by def. 5)} \\
& \iff \neg W_2(\neg \bigwedge S, \Delta \setminus S) && \text{(by uniqueness of } S\text{)} \\
& \iff W_2(\bigwedge S, \Delta) && \text{(by def. 5 and claim)}
\end{aligned}
$$

□

LEMMA 21. *Let $i < n$, $\overline{s} \in \{+, -\}^{n+1}$.*

$$
W_2\left(\bigwedge X_i(\overline{s}) \wedge Q_i, \Delta \setminus \bigcup_{j<i} \Delta^j\right)
\iff \left(\begin{array}{c}
\neg W_2(\bigwedge X_{i+1}(\overline{s}) \wedge Q_{i+1}, \Delta \setminus \bigcup_{j \leq i} \Delta^j) \\
\wedge \\
\neg W_2(\bigwedge X_{i+1}(\overline{s}^{i+1}) \wedge Q_{i+1}, \Delta \setminus \bigcup_{j \leq i} \Delta^j)
\end{array}\right)
$$

Before we give the formal proof, it might be helpful to give a rough description of the roles of the formulae λ_i and ρ_i, as well as that of the propositions Q_i, R_i. The intention is that if an argument with support $X_i(\overline{s})$ is made, there are two undercutting arguments and their supports are $X_{i+1}(\overline{s})$ and $X_{i+1}(\overline{s}^{i+1})$. We could have fixed this by including in our knowledge base the formula α_i that asserts that $X_i(\overline{s}) \cup X_{i+1}(\overline{s}) \cup \{\alpha_{i+1}\}$ is inconsistent. Now, if $X_i(\overline{s}) \cup \{\alpha_i\}$ is played there are two undercutting arguments, $X_{i+1}(\overline{s}^{\pm i+1}) \cup \{\alpha_{i+1}\}$. If these two arguments were the only undercutting arguments, then the sequences $\overline{s}$ would properly reflect partial valuations used to evaluate (4). The problem is that $\overline{X_i}(\overline{s}) \cup \overline{X_{i+1}}(\overline{s}^{\pm i+1})$ also undercuts $X_i(\overline{s}) \cup \{\alpha_i\}$ and cannot itself be undercut. In order to avoid these kinds of undercuts, we devise the knowledge base so that every formula in Π^i has already been used before an undercutting argument $X_{i+1}(\overline{s}^{\pm i+1})$ is made. The proposition Q_i is included in the support

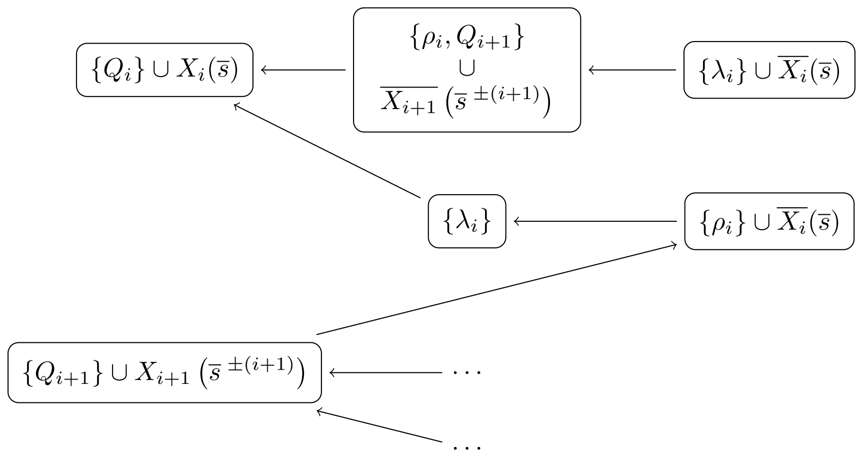

Figure 1: Attacks and counter-attacks on $\{Q_i\} \cup X_i(\overline{s})$ within $\Delta \setminus \bigcup_{j<i} \Delta^j$.

of an argument at level i only when Π^i is not yet exhausted. If the formula ρ_i is included in the support of an argument then the whole of Π^i will be exhausted.

See figure 1. Each node represents an argument, although the claim of the argument has been suppressed and only the support is visible. An arc between two arguments denotes undercutting. When $i = 0$ there are two sequences of $+$'s and $-$'s of length one, hence two possible nodes at the root of the graph, namely $\left\{Q_0, +P_0^0\right\}$ and $\left\{Q_0, -P_0^0\right\}$, and these are the only supports of the claim of the reduction.

The set $\{\lambda_i\}$ undercuts $\{Q_i\} \cup X_i(\overline{s})$. The set $\{\rho_i\} \cup \overline{X_i}(\overline{s})$ undercuts $\{\lambda_i\}$, but in the process "exhausts" all propositions at level i: now, $(+P)^i_j, (-P)^i_j$ for $j \leq i$ have all been used up. The set $\{\rho_i\} \cup \overline{X_i}(\overline{s})$ can now be undercut by $\{Q_{i+1}\} \cup X_{i+1}(\overline{s}^{\pm i+1})$ and by no other sets.

Figure 1 shows that $\{Q_i\} \cup X_i(\overline{s})$ can be undercut in another way, by $\{\rho_i, Q_{i+1}\} \cup \overline{X_{i+1}}(\overline{s}^{\pm i+1})$, but this undercut will not be warranted because it is itself undercut by $\{\lambda_i\} \cup \overline{X_i}(\overline{s})$ which cannot be undercut. Now we give the formal proof of the lemma.

Proof. Since each proposition in $X_i(\overline{s}) \cup \{Q_i\}$ only occurs positively as itself in Δ it follows, for $S \subseteq \Delta$, that $A(S, \bigwedge X_i(\overline{s}) \wedge Q_i) \iff S = X_i(\overline{s}) \cup \{Q_i\}$. By lemma 20,

$$
\begin{aligned}
&W_2\Big(\bigwedge X_i(\overline{s}) \wedge Q_i, \Delta \setminus \bigcup_{j<i} \Delta^j\Big)\\
&\iff \neg W_2\Big(\neg\Big(\bigwedge X_i(\overline{s}) \wedge Q_i\Big), \Delta \setminus \Big(\bigcup_{j<i} \Delta^j \cup X_i(\overline{s}) \cup \{Q_i\}\Big)\Big)\\
&\iff \neg W_2\Big(\neg\Big(\bigwedge X_i(\overline{s}) \wedge Q_i\Big), \Big(\Delta \setminus \bigcup_{j\leq i} \Delta^j\Big) \cup \{\lambda_i, \rho_i\} \cup \overline{X_i}(\overline{s})\Big) \qquad (6)
\end{aligned}
$$

using $\Delta^i = X_i(\overline{s}) \cup \overline{X_i}(\overline{s}) \cup \{Q_i, \lambda_i, \rho_i\}$. We consider whether $\neg W_2(\neg(\bigwedge X_i(\overline{s}) \wedge Q_i), (\Delta \setminus \bigcup_{j\leq i} \Delta^j) \cup \{\lambda_i, \rho_i\} \cup \overline{X_i}(\overline{s}))$ holds. Let $S \subseteq (\Delta \setminus \bigcup_{j\leq i} \Delta^j) \cup \{\lambda_i, \rho_i\} \cup \overline{X_i}(\overline{s})$. By lemmas 18, 19, $A(S, \neg(\bigwedge X_i(\overline{s}) \wedge Q_i))$ holds if and only if

$$S = \{\rho_i, Q_{i+1}\} \cup \overline{X}_{i+1}(\overline{s}),\ S = \{\rho_i, Q_{i+1}\} \cup \overline{X}_{i+1}(\overline{s}^{i+1}) \text{ or } S = \{\lambda_i\}$$

(the set $\{\lambda_i, \rho_i\}$ also entails the given claim but it is not minimal). The first two of these three supports for $\neg(\bigwedge X_i(\overline{s}) \wedge Q_i)$ are not warranted though, because both of them are undercut by $\{\lambda_i\} \cup \overline{X_i}(\overline{s})$ and there is no undercut on this latter set contained within $\Delta \setminus (\bigcup_{j\leq i} \Delta^j \cup \{\lambda_i, \rho_i, Q_{i+1}\})$. Thus

$$
\begin{aligned}
&\neg W_2\Big(\neg\Big(\bigwedge X_i(\overline{s}) \wedge Q_i\Big), \Big(\Delta \setminus \bigcup_{j\leq i} \Delta^j\Big) \cup \{\lambda_i, \rho_i\} \cup \overline{X_i}(\overline{s})\Big)\\
&\iff W_2\Big(\neg\lambda_i, \Big(\Delta \setminus \bigcup_{j\leq i} \Delta^j\Big) \cup \{\rho_i\} \cup \overline{X_i}(\overline{s})\Big) \qquad (7)
\end{aligned}
$$

By lemmas 18, 19 again, the one and only support for $\neg\lambda_i$ in $(\Delta \setminus \bigcup_{j\leq i} \Delta^j) \cup \{\rho_i\} \cup \overline{X_i}(\overline{s})$ is $\{\rho_i\} \cup \overline{X_i}(\overline{s})$. By lemma 20 again,

$$
\begin{aligned}
&W_2\Big(\neg\lambda_i, \Big(\Delta \setminus \bigcup_{j\leq i} \Delta^j\Big) \cup \{\rho_i\} \cup \overline{X_i}(\overline{s})\Big)\\
&\iff \neg W_2\Big(\neg\Big(\rho_i \wedge \bigwedge \overline{X_i}(\overline{s})\Big), \Delta \setminus \bigcup_{j\leq i} \Delta^j\Big) \qquad (8)
\end{aligned}
$$

Continuing, the only supports for $\neg\left(\rho_i \wedge \bigwedge \overline{X_i}(\overline{s})\right)$ within $\Delta \setminus \bigcup_{j\leq i} \Delta^j$ are

$$\{Q_{i+1}\} \cup X_{i+1}(\overline{s}), \quad \{Q_{i+1}\} \cup X_{i+1}(\overline{s}^{i+1})$$

Hence

$$
\begin{aligned}
&\neg W_2\Big(\neg\Big(\rho_i \wedge \bigwedge \overline{X_i}(\overline{s})\Big), \Delta \setminus \bigcup_{j\leq i} \Delta^j\Big)\\
&\iff \begin{cases}
W_2\left(\neg\left(\bigwedge X_{i+1}(\overline{s}) \wedge Q_{i+1}\right), \Delta \setminus (\bigcup_{j\leq i} \Delta^j \cup X_{i+1}(\overline{s}) \cup \{Q_{i+1}\})\right)\\
\text{and}\\
W_2\Big(\neg\Big(\bigwedge X_{i+1}(\overline{s}^{i+1}) \wedge Q_{i+1}\Big),\\
\qquad \Delta \setminus (\bigcup_{j\leq i} \Delta^j \cup X_{i+1}(\overline{s}^{i+1} \cup \{Q_{i+1}\}))\Big)
\end{cases} \qquad (9)
\end{aligned}
$$

Putting all this together,

$$
\begin{aligned}
& W_2\Big(\bigwedge X_i(\overline{s}) \wedge Q_i, \Delta \setminus \bigcup_{j<i} \Delta^j\Big) \\
\iff & \neg W_2\Big(\neg\Big(\bigwedge X_i(\overline{s}) \wedge Q_i\Big), \Big(\Delta \setminus \bigcup_{j\le i} \Delta^j\Big) \cup \{\lambda_i, \rho_i\} \cup \overline{X_i}(\overline{s})\Big) && \text{(by 6)} \\
\iff & W_2\Big(\neg\lambda_i, \Big(\Delta \setminus \bigcup_{j\le i} \Delta^j\Big) \cup \{\rho_i\} \cup \overline{X_i}(\overline{s})\Big) && \text{(by 7)} \\
\iff & \neg W_2\Big(\neg\left(\rho_i \wedge \bigwedge \overline{X_i}(\overline{s})\right), \Delta \setminus \bigcup_{j\le i} \Delta^j\Big) && \text{(by 8)} \\
\iff & \begin{cases}
W_2\Big(\neg\left(\bigwedge X_{i+1}(\overline{s}) \wedge Q_{i+1}\right), \Delta \setminus \left(\bigcup_{j\le i} \Delta^j \cup X_{i+1}(\overline{s}) \cup \{Q_{i+1}\}\right)\Big) \\
\text{and} \\
W_2\Big(\neg\Big(\bigwedge X_{i+1}(\overline{s}^{i+1}) \wedge Q_{i+1}\Big), \\
\qquad \Delta \setminus \left(\bigcup_{j\le i} \Delta^j \cup X_{i+1}(\overline{s}^{i+1}) \cup \{Q_{i+1}\}\right)\Big)
\end{cases} && \text{(by 9)} \\
\iff & \begin{cases}
W_2\Big(\neg\Big(\bigwedge X_{i+1}(\overline{s}) \wedge Q_{i+1}\Big), \\
\qquad \Big(\Delta \setminus \bigcup_{j\le i+1} \Delta^j\Big) \cup \overline{X_{i+1}}(\overline{s}) \cup \{\rho_{i+1}, \lambda_{i+1}\}\Big) \\
\text{and} \\
W_2\Big(\neg\Big(\bigwedge X_{i+1}(\overline{s}^{i+1}) \wedge Q_{i+1}\Big), \\
\qquad \Big(\Delta \setminus \bigcup_{j\le i+1} \Delta^j\Big) \cup \overline{X_{i+1}}(\overline{s}^{i+1}) \cup \{\rho_{i+1}, \lambda_{i+1}\}\Big)
\end{cases} \\
\iff & \begin{cases}
\neg W_2\Big(\bigwedge X_{i+1}(\overline{s}) \wedge Q_{i+1}, \Delta \setminus \bigcup_{j<i+1} \Delta^j\Big) \\
\text{and} \\
\neg W_2\Big(\bigwedge X_{i+1}(\overline{s}^{i+1}) \wedge Q_{i+1}, \Delta \setminus \bigcup_{j<i+1} \Delta^j\Big)
\end{cases} && \text{(by (6))}
\end{aligned}
$$

□

LEMMA 22. *Let $i \le n$, $\overline{s} \in \{+,-\}^{n+1}$. If $n-i$ is even then*

$$
\begin{aligned}
& W_2\left(\bigwedge \left(X_i(\overline{s}) \cup \{Q_i\}\right), \left(\Delta \setminus \bigcup_{j\le i} \Delta^j\right) \cup X_i(\overline{s}) \cup \{Q_i\}\right) \\
& \iff v_{\overline{s}} \not\models \exists p_{i+1} \forall p_{i+2} \ldots \forall p_n \phi
\end{aligned}
$$

and if $n-i$ is odd,

$$W_2\left(\bigwedge\left(X_i(\overline{s})\cup\{Q_i\}\right),\left(\Delta\setminus\bigcup_{j\leq i}\Delta^j\right)\cup X_i(\overline{s})\cup\{Q_i\}\right)$$
$$\iff v_{\overline{s}}\models\forall p_{i+1}\exists p_{i+2}\ldots\forall p_n\phi$$

Proof. The proof is by induction over $n-i$. For the base case, let $i=n$, so $n-i$ is even. We have to prove that $W_2(\bigwedge(X_n(\overline{s})\cup\{Q_n\}),(\Delta\setminus\bigcup_{j\leq n}\Delta^j)\cup X_n(\overline{s})\cup\{Q_n\})$ holds iff $v_{\overline{s}}\not\models\phi$. Note that $\Delta\setminus\bigcup_{j\leq n}\Delta^j=\{\neg\phi'\}$. The formula $\bigwedge X_n(\overline{s})\wedge Q_n$ is supported by $X_n(\overline{s})\cup\{Q_n\}$ and by no other set within $X_n(\overline{s})\cup\{Q_n,\neg\phi'\}$. The only possible set supporting $\neg\left(\bigwedge X_n(\overline{s})\wedge Q_n\right)$ contained in $\{\neg\phi'\}$ is of course $\{\neg\phi'\}$, and $\{\neg\phi'\}$ proves $\neg\left(\bigwedge X_n(\overline{s})\wedge Q_n\right)$ if and only if $X_n(\overline{s})\models\phi'$ iff $v_{\overline{s}}(\phi)=\top$, by the last part of lemma 18 and equation (5). Thus $W_2(\bigwedge(X_n(\overline{s})\cup\{Q_n\}),X_n(\overline{s})\cup\{Q_n,\neg\phi'\})$ holds iff $v_{\overline{s}}\not\models\phi$, proving the base case.

Now let $n-i>0$ and suppose the lemma holds for all smaller values of $n-i$. By lemma 21, $W_2(\bigwedge X_i(\overline{s})\wedge Q_i,\Delta\setminus\bigcup_{j<i}\Delta^j)$ holds iff

$$\begin{aligned}&\neg W_2\left(\bigwedge X_{i+1}(\overline{s})\wedge Q_{i+1},\Delta\setminus\bigcup_{j\leq i}\Delta^j\right)\wedge\\&\neg W_2\left(\bigwedge X_{i+1}(\overline{s}^{i+1})\wedge Q_{i+1},\Delta\setminus\bigcup_{j\leq i}\Delta^j\right).\end{aligned}\tag{10}$$

If $n-i$ is odd, this is equivalent to $v_{\overline{s}}\models\exists p_{i+2}\forall p_{i+3}\ldots\forall p_n\phi$ and $v_{\overline{s}^{i+1}}\models\exists p_{i+2}\forall p_{i+3}\ldots\forall p_n\phi$ by the inductive hypothesis ($n-(i+1)$ is even) which is equivalent to $v_{\overline{s}}\models\forall p_{i+1}\exists p_{i+2}\forall p_{i+3}\ldots\forall p_n\phi$, as required. If $n-i$ is even, (10) is equivalent to $v_{\overline{s}}\not\models\exists p_{i+2}\forall p_{i+3}\ldots\forall p_n\phi$ and $v_{\overline{s}^{i+1}}\not\models\exists p_{i+2}\ldots\forall p_n\phi$, which is equivalent to $v_{\overline{s}}\not\models\exists p_{i+1}\forall p_{i+2}\ldots\forall p_n\phi$, as required. □

THEOREM 23. *Definition 16 is a polynomial time reduction of* **QBF** *to* **WFP**$_2$.

Proof. The reduction maps the instance (4) of **QBF** to the instance $(Q_0\wedge\mathsf{val}^0,\Delta)$ of **WFP**$_2$. There are two supports for the claim viz $\left\{Q_0,(+P)_0^0\right\}$ and $\left\{Q_0,(-P)_0^0\right\}$. Hence the claim is warranted over Δ iff either $W_2(Q_0\wedge(+P)_0^0,\Delta)$ or $W_2(Q_0\wedge(-P)_0^0,\Delta)$, i.e., iff there is $\overline{s}\in\{+,-\}^{n+1}$ such that $W_2(Q_0\wedge\bigwedge X_0(\overline{s}),\Delta)$. By lemma 22, recalling that n is odd, this is equivalent to $v_{\overline{s}}\models\forall p_1\exists p_2\ldots\forall p_n\phi$ or $v_{\overline{s}^0}\models\forall p_1\ldots\forall p_n\phi$ (here $\overline{s}$ is arbitrary) which is equivalent to $v_{\overline{s}}\models\exists p_0(\forall p_1\ldots\forall p_n\phi)$. Hence the reduction is correct. □

6 Discussion and Conclusions

We have shown that deciding whether a claim is warranted in the frameworks of [BH00, BH01, PWA03] is PSPACE-complete. We discuss this result within the context of existing complexity results for other frameworks.

Dung's seminal paper [Dun95] considers abstract Argumentation Frameworks (AFs), where arguments are seen as abstract entities and, using a notion of attack that is essentially a binary relation on the set of arguments, several definitions of acceptability are introduced. Given a finite set of arguments, the complexity of ascertaining whether an argument belongs to the set of acceptable arguments according to a specific notion of acceptability, is a decidable decision problem. Results concerning the complexity of some definitions of acceptability, as well as further refinements thereof in the literature, have been produced [DBC02, DBC04, Dun06, Dun07] and the complexity lies generally in the first two levels of the polynomial hierarchy — apparently lower than the PSPACE result of the current paper. But these results do not carry over to deductive argumentation. A naive translation from the warranted argument problem of definition 3 to an abstract Argumentation Framework would potentially produce an exponential number of arguments (in general, one can expect $|\Delta| \cdot 2^{|\Delta|}$ arguments). If one thus constructs an abstract argumentation framework that contains as nodes the arguments of a deductive system of argumentation, then the graph could be of exponential size, rendering the complexity results expressed as functions of the graph-size unhelpful. Indeed, this process resembles the exponential jump in complexity observed when going from the usual representations of graphs as inputs to decision problems (e.g., as incidence matrices) to succinct representations (e.g., circuits). Some work on directly linking propositional argumentation with abstract argumentation frameworks exists [WDP06], but has not addressed the complexity of warrant.

A parallel line of research on argumentation based on deductive systems concerns Assumption-Based Frameworks (ABFs) by Bondarenko et al [BDKT97], where an underlying object language and associated logic provide the deductive processes on which the validity of an argument depends. In this framework an argument is effectively a conflict-free set of assumptions, *without a specified claim*. The corresponding decision problems in ABFs have been studied separately, and results exist [DNT99, DNT00, DNT02] on the complexity of various semantics, generally situated in the first four levels of the polynomial hierarchy. Once again, the complexity results do not carry over to propositional argumentation. The reasons are twofold: first, the definition of argument diverges, making a correspondence difficult; second, the definitions of acceptability in ABFs also diverge from the definition of warrant in propositional argumentation.

Further work includes the study of other deductive argumentation frameworks, as yet unexplored from a complexity perspective. A prime candidate for such study is *defeasible logic programming* (DeLP) [SL92, GS04] which also employs the notion of warrant based on dialectical trees. Two key differences between the definition of warrant in DeLP and those we have considered here are (i) the formulae of DeLP are restricted to generalised horn clause formulae with modus-ponens as the only inference rule, and (ii) when undercutting an argument there is no requirement that the formulae in the undercut are partly or entirely unused formulae, instead there is a weaker requirement that the support of the undercut is not contained in the support of a previously played argument. In principle, this means that a chain of arguments, undercuts and recursive undercuts could have exponential length (where as in both definition 3 and 5, such a chain could have at most linear length). Indeed, consider the knowledge with the following strict rules: $p_1, \ldots, p_n \rightarrow q$, $p_1, \ldots, p_n \rightarrow \neg q$, $a_i \rightarrow p_i$ and $b_i \rightarrow p_i$ for $i \leq n$ and with the following defeasible rules (presumptions, in the terminology of DeLP): a_i and b_i for $i \leq n$.[1] The tree corresponding to the query $W_{\mathrm{DeLP}}(q, \Delta)$ would have branches of length 2^n. Currently, we only know that the complexity of the warranted argument problem for DeLP is between PSPACE (through a reduction from the geography problem) and EXPSPACE and we plan to investigate this problem further.

References

[BDKT97] A. Bondarenko, P.M. Dung, R.A. Kowalski, and F. Toni. An Abstract, Argumentation-Theoretic Approach to Default Reasoning. *Artificial Intelligence*, 93:63–101, 1997.

[BH00] Ph. Besnard and A. Hunter. Towards a Logic-Based Theory of Argumentation. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence, AAAI'2000*, pages 411–416. MIT Press, 2000.

[BH01] Ph. Besnard and A. Hunter. A logic-based theory of deductive arguments. *Artificial Intelligence*, 128(1-2):203–235, 2001.

[BHW08] Ph. Besnard, A. Hunter, and S. Woltran. Encoding Deductive Argumentation in Quantified Boolean Formulae. Technical report, Technical University of Vienna, 2008.

[DBC02] P.E. Dunne and T.J.M. Bench-Capon. Coherence in finite argument systems. *Artificial Intelligence*, 141(1):187–203, 2002.

[DBC04] P.E. Dunne and T.J.M. Bench-Capon. Complexity in Value-Based Argument Systems. In *Logics in Artificial Intelligence, 9th European Conference, JELIA 2004*, volume 3229 of *Lecture Notes in Computer Science*, pages 360–371. Springer, 2004.

[DNT99] Y. Dimopoulos, B. Nebel, and F. Toni. Preferred Arguments are Harder to Compute than Stable Extension. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, IJCAI 99*, pages 36–43. Morgan Kaufmann, 1999.

[DNT00] Y. Dimopoulos, B. Nebel, and F. Toni. Finding Admissible and Preferred Arguments Can be Very Hard. In *Principles of Knowledge Representation and Reasoning, KR2000*, pages 53–61, 2000.

[1] In DeLP, strict rules can be thought of as part of the entailment relation, and defeasible rules as the formulae from which arguments are built.

[DNT02] Y. Dimopoulos, B. Nebel, and F. Toni. On the computational complexity of assumption-based argumentation for default reasoning. *Artificial Intelligence*, 141(1/2):57–78, 2002.

[Dun95] P.M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321–357, 1995.

[Dun06] P.E. Dunne. Complexity Properties of Restricted Abstract Argument Systems. In *Computational Models of Argument: Proceedings of COMMA 2006*, volume 144 of *Frontiers in Artificial Intelligence and Applications*, pages 85–96. IOS Press, 2006.

[Dun07] P.E. Dunne. Computational properties of argument systems satisfying graph-theoretic constraints. *Artificial Intelligence*, 171(10-15):701–729, 2007.

[FSU93] A.S. Fraenkel, E.R. Scheinerman, and D. Ullman. Undirected edge geography. *Theoretical Computer Science*, 112:371–381, 1993.

[GJ79] M Garey and D Johnson. *Computers and Intractability — a guide to the theory of* **NP***-completeness.* Freeman and Co., 1979.

[GS04] A.J. García and G.R. Simari. Defeasible Logic Programming: An Argumentative Approach. *Theory and Practice of Logic Programming*, 4(1-2):95–138, 2004.

[Nut94] D. Nute. Defeasible Logic. In D. Gabbay, C.J. Hogger, and J.A. Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming*, pages 355–395. Oxford University Press, 1994.

[Pol87] J.L. Pollock. Defeasible Reasoning. *Cognitive Science*, 11(4):481–518, 1987.

[PWA03] S. Parsons, M. Wooldridge, and L. Amgoud. Properties and Complexity of Some Formal Inter-agent Dialogues. *Journal of Logic and Computation*, 13(3):347–376, 2003.

[SL92] G.R. Simari and R.P. Loui. A Mathematical Treatment of Defeasible Reasoning and its Implementation. *Artificial Intelligence*, 53(2-3):125–157, 1992.

[SM73] L.J. Stockmeyer and A.R. Meyer. Word Problems Requiring Exponential Time: Preliminary Report. In *Fifth Annual ACM Symposium on Theory of Computing*, pages 1–9. ACM, 1973.

[Sto76] L.J. Stockmeyer. The polynomial-time hierarchy. *Theoretical Computer Science*, 3(1):1–22, 1976.

[WDP06] M.J. Wooldridge, P.E. Dunne and S. Parsons. On the Complexity of Linking Deductive and Abstract Argument Systems. In *Proceedings of The Twenty-First National Conference on Artificial Intelligence (AAAI-06)*, pages 299-304. AAAI Press, July 2006.