Reasoning in Argumentation Frameworks Using Quantified Boolean Formulas

Uwe Egly, Stefan Woltran
Institut für Informationssysteme 184/3
Technische Universität Wien
Favoritenstraße 9–11
A–1040 Wien, Austria
email: [uwe,stefan]@kr.tuwien.ac.at

Abstract. This paper describes a generic approach to implement propositional argumentation frameworks by means of quantified Boolean formulas (QBFs). The motivation to this work is based on the following observations: Firstly, depending on the underlying deductive system and the chosen semantics (i.e., the kind of extension under consideration), reasoning in argumentation frameworks can become computationally involving up to the fourth level of the polynomial hierarchy. This makes the language of QBFs a suitable target formalism since decision problems from the polynomial hierarchy can be efficiently represented in terms of QBF. Secondly, several practicably efficient solvers for QBFs are currently available, and thus can be used as black-box engines in potential implementations of argumentation frameworks. Finally, the definition of suitable QBF modules provides us with a tool box in order to capture a broad range of reasoning tasks associated to formal argumentation.

1. Introduction

In daily life, we use arguments and counter-arguments in discussions in order to "convince" our opponent to our point of view. Argumentation frameworks [7] have been used to formalize the reasoning underlying argumentation. They provide what "convince" means and how arguments may be defeated by counter-arguments.

Reasoning underlying argumentation is a general principle. Many of the well-known non-monotonic reasoning formalisms [21,23] can be faithfully interpreted within argumentation frameworks [3]. Consequently, these frameworks formalize not only the mentioned reasoning underlying argumentation, but can be used to interpret, compare, and implement a wide range of different reasoning principles. Since the main difference between two distinct reasoning principles is the underlying derivability operator, the interpretation of both principles is generic except the definition of this operator. Therefore, argumentation frameworks provide not only a theoretical setting for studying different reasoning mechanisms, but also can be used as a practical underpinning for implementations. However, as shown by Dimopoulos, Nebel, and Toni [5] some combinations of derivability operators and notions of extensions make reasoning in such argumentation frameworks computationally involving as witnessed by hardness results up to the fourth level of the polynomial hierarchy.

In this paper, we propose an implementation of argumentation frameworks which is based on the satisfiability problem of quantified Boolean formulas (QBFs), an extension of classical propositional logic in which formulas may contain quantifications over propositional atoms. The motivation to consider QBFs for argumentation frameworks is as follows:

First, in recent years we observed a parallel and mutually influencing development of QBF solvers on the one hand, and design of applications, on the other hand. This situation is similar to the emerge of the success of satisfiability solvers in the mid nineties, where first impressive results have been achieved by employing SAT solvers in the area of planning [17,18]. Since QBFs are a more expressive language than propositional logic, their range of application is naturally larger than that of SAT (under the reasonable assumption that reductions are computable in polynomial time). In fact, various problems from different areas have been considered as applications for QBF, including conformant planning problems [24], reasoning from inconsistent knowledge bases [1,2], nonmonotonic reasoning [8,13,25], verification [16,20], and theorem proving [9]. Moreover, there has been made a significant progress in the development of QBF solvers in the last few years [19].

Second, the different semantics captured by argumentation frameworks are all uniformly represented in our QBF setting. Our aim is, not at least, to illustrate how basic QBF modules can be used as building blocks for assembling realizations of numerous reasoning tasks in different instantiations of the framework. Notably, the different complexity behavior does not prohibit a uniform implementation method due to the power of QBFs and their solvers. In fact, the high complexity of some variants together with our encodings provide highly complex but structured problems for benchmarking QBF solvers. Currently, such structured problems are barely going beyond the second level of the polynomial hierarchy.

The outline of the paper is as follows. After some formal preliminaries, we start with the description of abstract argumentation frameworks. The terminology is due to the fact that the underlying derivability operator remains *abstract* in the sense that only some necessary criteria have to be satisfied, but the operator is not specified in a concrete way. Then we provide corresponding abstract translation schemes by means of QBFs. Finally, we briefly describe some case studies, which instantiate the generic framework to propositional reasoning principles. In terms of the QBF framework, this is obtained by plugging in a QBF module which concretely describes the derivability operator of the respective formalism. Due to space restrictions, we shall only sketch these concrete realizations, which may serve as a basis for implementation by invoking QBF systems.

2. Formal Preliminaries

2.1. Quantified Boolean Formulas

Quantified Boolean formulas (QBFs) generalize ordinary propositional formulas by the admission of quantifications over propositional variables. In particular, the language of QBFs contains, for any atom p, unary operators of form $\forall p$ and $\exists p$, called *universal* and *existential quantifiers*, respectively. Informally, a QBF of form $\forall p\, \exists q\, \Phi$ means that for all truth assignments of p there is a truth assignment of q such that Φ is true.

An occurrence of a propositional variable p in a QBF Φ is *free* iff it does not appear in the scope of a quantifier $\mathsf{Q}p$ ($\mathsf{Q} \in \{\forall, \exists\}$), otherwise the occurrence of p is *bound*. If Φ contains no free variable occurrences, then Φ is *closed*, otherwise Φ is *open*. Furthermore, we write $\Phi[p/\phi]$ to denote the result of uniformly substituting each free occurrence of the variable p in Φ by a formula ϕ.

By an *interpretation*, I, we understand a set of atoms. Informally, an atom p is true under I iff $p \in I$. In general, the truth value, $\nu_I(\Phi)$, of a QBF Φ under an interpretation I is recursively defined as follows:

1. if $\Phi = \top$, then $\nu_I(\Phi) = 1$;
2. if $\Phi = p$ is an atom, then $\nu_I(\Phi) = 1$ if $p \in I$, and $\nu_I(\Phi) = 0$ otherwise;
3. if $\Phi = \neg\Psi$, then $\nu_I(\Phi) = 1 - \nu_I(\Psi)$;
4. if $\Phi = (\Phi_1 \wedge \Phi_2)$, then $\nu_I(\Phi) = min(\{\nu_I(\Phi_1), \nu_I(\Phi_2)\})$;
5. if $\Phi = \forall p\, \Psi$, then $\nu_I(\Phi) = \nu_I(\Psi[p/\top] \wedge \Psi[p/\bot])$;
6. if $\Phi = \exists p\, \Psi$, then $\nu_I(\Phi) = \nu_I(\Psi[p/\top] \vee \Psi[p/\bot])$.

The truth conditions for $\bot$, $\vee$, $\rightarrow$, and $\leftrightarrow$ follow from the above in the usual way. We say that Φ is *true under* I iff $\nu_I(\Phi) = 1$, otherwise Φ is *false under* I. If $\nu_I(\Phi) = 1$, then I is a *model* of Φ. If Φ has some model, then Φ is said to be *satisfiable*. If Φ is true under any interpretation, then Φ is *valid*. Observe that a closed QBF is either valid or unsatisfiable, because closed QBFs are either true under each interpretation or false under each interpretation. Hence, for closed QBFs, there is no need to refer to particular interpretations. Two sets of QBFs (or ordinary boolean formulas) are *logically equivalent* iff they possess the same models. In the sequel, we use the following abbreviation in the context of QBFs: For a set $P = \{p_1, \ldots, p_n\}$ of propositional variables and a quantifier $\mathsf{Q} \in \{\forall, \exists\}$, we let $\mathsf{Q}P\,\Phi$ stand for the formula $\mathsf{Q}p_1\mathsf{Q}p_2 \cdots \mathsf{Q}p_n\,\Phi$.

In the same way as the satisfiability problem of classical propositional logic is the "prototypical" problem of NP, i.e., being an NP-complete problem, the satisfiability problem of QBFs in *prenex form* possessing $k-1$ quantifier alternations is the "prototypical" problem of the k-th level of the polynomial hierarchy, as expressed by the following well-known result taken from [26].

Proposition 1 *Given a propositional formula ϕ with its atoms partitioned into $i \geq 1$ pairwise distinct sets $P_1, \ldots, P_i$, deciding whether $\exists P_1 \forall P_2 \ldots \mathsf{Q}_i P_i \phi$ is true is Σ_i^p-complete, where $\mathsf{Q}_i = \exists$ if i is odd and $\mathsf{Q}_i = \forall$ if i is even, Dually, deciding whether $\forall P_1 \exists P_2 \ldots \mathsf{Q}'_i P_i \phi$ is true is Π_i^p-complete, where $\mathsf{Q}'_i = \forall$ if i is odd and $\mathsf{Q}_i = \exists$ if i is even.*

This complexity landscape can be extended to arbitrary closed QBFs if the maximal number of quantifier alternations along a path in the QBF's formula tree is taken into account. In turn, an arbitrary QBF can be transformed into an equivalent QBF in prenex form, although this transformation is not deterministic and crucial for the performance of QBF solvers requiring the input formula being in prenex conjunctive normal form (for details, see [10,11]).

Finally, we highlight the used reduction approach. Given a decision problem Π, we aim at finding a translation scheme $\mathcal{T}_\Pi$ into closed QBFs, such that

1. $\mathcal{T}_\Pi(\cdot)$ is faithful, i.e., $\mathcal{T}_\Pi(K)$ is true iff K is a yes-instance of Π;
2. For each instance K, $\mathcal{T}_\Pi(K)$ is computable in polynomial time with respect to the size of K; and

3. determining the truth of the QBFs resulting from $\mathcal{T}_\Pi(\cdot)$ is not computationally harder (by means of Proposition 1) than the computational complexity of Π.

2.2. Abstract Argumentation Frameworks

In this section, we introduce the notions around argumentation frameworks, where we basically follow the definitions in [5]. Abstract argumentation frameworks are defined on top of a deductive system $(\mathcal{L}_\mathcal{A}, \mathcal{R})$, where $\mathcal{L}_\mathcal{A}$ is some formal language over an alphabet $\mathcal{A}$ and $\mathcal{R}$ is a set of inference rules inducing a monotonic derivability relation $\vdash$. For a theory $T \subseteq \mathcal{L}_\mathcal{A}$, we identify, as usual, its deductive closure by

$$Th(T) = \{\alpha \in \mathcal{L}_\mathcal{A} \mid T \vdash \alpha\}.$$

An *abstract (assumption-based) framework* is a triple $(T, A, \overline{(\cdot)})$, where $T, A \subseteq \mathcal{L}_\mathcal{A}$, with A being the set of *assumptions*, and $\overline{(\cdot)}$ is a mapping from A to $\mathcal{L}_\mathcal{A}$. For an $\alpha \in A$, $\overline{\alpha}$ is the *contrary* of α. An *extension* of a framework $(T, A, \overline{(\cdot)})$ is a theory $Th(T \cup S)$ with $S \subseteq A$. If no confusion can arise, an extension is often referred to as S alone. A set $S \subseteq A$ *attacks* an $\alpha \in A$ iff $T \cup S \vdash \overline{\alpha}$, and S attacks an $S' \subseteq A$ iff S attacks an $\alpha \in S'$. Consequently, S attacks *itself* iff there exists an $\alpha \in S$, such that S attacks α.

A set $S \subseteq A$ is *closed* iff $S = A \cap Th(T \cup S)$. Frameworks, in which it is guaranteed that each such S is closed, are called *flat*. Given a framework $(T, A, \overline{(\cdot)})$, a set $S \subseteq A$ is *stable* iff

1. S is closed,
2. S does not attack itself, and
3. S attacks each $\alpha \in A \setminus S$.

A set $S \subseteq A$ is *admissible* iff

1. S is closed,
2. S does not attack itself, and
3. for all closed $S' \subseteq A$, it holds that if S' attacks S, then S attacks S'.

Finally, S is *preferred* if it is admissible and maximal with respect to set inclusion. For a set $S \subseteq A$, which is stable (resp. admissible, preferred), the extension $Th(T \cup S)$ is called stable (resp. admissible, preferred).

A framework $(T, A, \overline{(\cdot)})$ is called *normal*, iff every maximal closed set not attacking itself is stable. Finally, a framework is *simple*, iff, for inconsistent T, there is no admissible extension, and otherwise there exists a least admissible extension $S = A \cap Th(T)$.

Given a framework $(T, A, \overline{(\cdot)})$, the *credulous reasoning problem* is to decide whether a given $\varphi \in \mathcal{L}_\mathcal{A}$ is contained in $Th(T \cup S)$ for *some* extension S. The *skeptical reasoning problem* is to decide whether $\varphi \in \mathcal{L}_\mathcal{A}$ is contained in $Th(T \cup S)$ for *all* extensions S.

The attentive reader might have observed that we did not define what kind of derivability operator is associated with the abstract argumentation framework. This is not an error but a feature. In the next section, we will continue with a translation of abstract argumentation frameworks to QBFs. These translations will again be independent from a concrete derivability operator which will come into the play when we instantiate the framework.

3. Abstract Translation Schemes to QBFs

In this section we discuss the general encodings, leaving the concrete check for the derivability operator $\vdash$ unresolved. Afterwards we shall present some concrete realizations in detail.

Given the propositional language $\mathcal{L}_\mathcal{A}$ underlying an argumentation framework, we assume the language of QBFs, $\mathcal{L}_{\mathsf{QBF}}$, implicitly as defined over a sufficiently large alphabet, consisting of all propositional atoms $\mathcal{A}$ in $\mathcal{L}_\mathcal{A}$ plus a set of additional mutual disjoint *guessing variables* $G = \{g_\alpha \mid \alpha \in \mathcal{L}_\mathcal{A}\}$ which we duplicate whenever needed, i.e., $G' = \{g'_\alpha \mid \alpha \in \mathcal{L}_\mathcal{A}\}$, $G'' = \{g''_\alpha \mid \alpha \in \mathcal{L}_\mathcal{A}\}$, etc. Indeed, we shall use subsets of the guessing variables to guess subsets $S \subseteq \mathcal{L}_\mathcal{A}$. Given an interpretation I and a subset $S \subseteq \mathcal{L}_\mathcal{A}$, we say that *$I$ characterizes S via G* iff it holds that $g_\alpha \in I$ iff $\alpha \in S$.

First, we define an abstract QBF module for encodings of $\vdash$, which is later replaced by concrete instantiations. Then we provide the general encodings for checking closure, the notion of attacking, and for characterizing stable, admissible, and preferred extensions. Afterwards, we briefly discuss simplifications for frameworks which are flat, normal, or simple.

Definition 1 *Let $(\mathcal{L}_\mathcal{A}, \mathcal{R})$ be a deductive system with an induced derivability relation $\vdash$, let $T, A \subseteq \mathcal{L}_\mathcal{A}$, and $\alpha \in \mathcal{L}_\mathcal{A}$. Moreover, let $2^{\mathcal{L}_\mathcal{A}}$ denote the power set of $\mathcal{L}_\mathcal{A}$. Then a function*

$$f^G: \quad 2^{\mathcal{L}_\mathcal{A}} \times 2^{\mathcal{L}_\mathcal{A}} \times \mathcal{L}_\mathcal{A} \;\mapsto\; \mathcal{L}_{\mathsf{QBF}}$$

is called an encoding *for $\vdash$, iff*

1. *$f^G(T, A, \alpha)$ has free variables $G = \{g_a \mid a \in A\}$, and,*
2. *for each interpretation I characterizing S via G, it holds that $f^G(T, A, \alpha)$ is true under I iff $T \cup S \vdash \alpha$.*

As an example, consider some theory T, a set $A = \{\beta, \gamma\}$, and an encoding f^G for $\vdash$ with free variables g_β, g_γ. Now consider, $f^G(T, A, \alpha)$ is true only under the following interpretations (over $\{g_\beta, g_\gamma\}$): $I_1 = \{g_\beta\}$, $I_2 = \{g_\gamma\}$, and $I_3 = \{g_\beta, g_\gamma\}$. Now since f^G is an encoding, we derive from these models that

(i) $T \cup \{\beta\} \vdash \alpha$,
(ii) $T \cup \{\gamma\} \vdash \alpha$, and
(iii) $T \cup \{\beta, \gamma\} \vdash \alpha$

hold, while $T \vdash \alpha$ does not hold since $f^G(T, A, \alpha)$ is not true under $I_4 = \emptyset$.

We are now well prepared to characterize all necessary ingredients for characterizing reasoning in argumentation frameworks via QBFs.

Theorem 1 *Let $F = (T, A, \overline{(\cdot)})$ be a framework over a deductive system inducing $\vdash$, f^G an encoding of $\vdash$ with free variables $G = \{g_a \mid a \in A\}$, and I an interpretation characterizing $S \subseteq A$ via G. Then the following holds.*

1. *S is closed iff I is a model of*

$$\mathit{closed}_F^G := \bigwedge_{a \in A} \big(g_a \leftrightarrow f^G(T, A, a)\big). \qquad (1)$$

2. *S does not attack itself iff I is a model of*

$$noattack_F^G := \bigwedge_{a \in A} \big(g_a \rightarrow \neg f^G(T, A, \overline{a})\big). \tag{2}$$

3. *S is stable iff I is a model of*

$$stable_F^G := closed_F^G \ \wedge noattack_F^G \wedge \bigwedge_{a \in A} \big(\neg g_a \rightarrow f^G(T, A, \overline{a})\big) \tag{3}$$

$$:= closed_F^G \ \wedge \bigwedge_{a \in A} \big(g_a \leftrightarrow \neg f^G(T, A, \overline{a})\big). \tag{4}$$

4. *S is admissible iff I is a model of*

$$adm_F^G := closed_F^G \ \wedge noattack_F^G \ \wedge \forall G' \Big[closed_F^{G'} \wedge \Big(\bigvee_{a \in A} \big(g_a \wedge f^{G'}(T, A, \overline{a})\big)\Big) \rightarrow \Big(\bigvee_{a \in A} \big(g'_a \wedge f^G(T, A, \overline{a})\big)\Big)\Big]. \tag{5}$$

Observe that the third arguments in the functions f^G in $closed_F^G$ and $stable_F^G$ are different, i.e., we have a in (1) but $\overline{a}$ in (4). However, in some cases and in particular for a flat framework, the test for closure (i.e., the conjunct $closed_F^G$) can be removed from (4), resulting in

$$stable_F^G := \bigwedge_{a \in A} \big(g_a \leftrightarrow \neg f^G(T, A, \overline{a})\big). \tag{6}$$

Concerning admissible extensions, by applying Theorem 2 in [4], the encoding can now be considerably simplified for flat frameworks.

Proposition 2 *A set $S \subseteq A$ is admissible for a flat framework $(T, A, \overline{(\cdot)})$, iff S does not attack itself, and for the set $S' = \{\alpha \in A \setminus S \mid S$ does not attack $\alpha\} \cup S$, it holds that S' does not attack S.*

Theorem 2 *Let $F = (T, A, \overline{(\cdot)})$ be a flat framework over a deductive system inducing $\vdash$, f^G an encoding of $\vdash$ with free variables $G = \{g_a \mid a \in A\}$, and I an interpretation characterizing $S \subseteq A$ via G. Then S is admissible iff*

$$adm_F^G := noattack_F^G \wedge \exists G' \Big[\bigwedge_{a \in A} \Big(g'_a \leftrightarrow \big(g_a \vee \neg f^G(T, A, \overline{a})\big)\Big) \wedge \bigwedge_{a \in A} \big(g_a \rightarrow \neg f^{G'}(T, A, \overline{a})\big)\Big]\Big) \tag{7}$$

is true under I.

It remains to discuss the notion of preferred extensions. In order to encode the maximality test, which is employed to characterize preferred sets, we use the following concept.

Definition 2 *Let $G = \{g_a \mid a \in A\}$ and $G' = \{g'_a \mid a \in A\}$ be disjoint indexed sets of atoms with the same cardinality. Define*

$$G < G' := \bigwedge_{a \in A} \Big(g_a \rightarrow g'_a\Big) \wedge \neg \bigwedge_{a \in A} \Big(g'_a \rightarrow g_a\Big).$$

Theorem 3 *Let $F = (T, A, \overline{(\cdot)})$ be a framework over a deductive system inducing $\vdash$, f^G an encoding of $\vdash$ with free variables $G = \{g_a \mid a \in A\}$, and I an interpretation characterizing $S \subseteq A$ via G. Then S is preferred iff*

$$pref^G_F := adm^G_F \wedge \neg \exists G' \Big((G < G') \wedge adm^{G'}_F\Big)$$

is true under I.

Observe that the entire encoding now uses three copies of guessing variables, namely G, G', and also G'' which occurs in $adm^{G'}_F$. Also note that we can choose between two realizations of adm^G_F to be used in $pref^G_F$ depending whether F is flat (Theorem 2) or not (3. in Theorem 1). This leads to a different quantifier structure in $pref^G_F$ mirroring the different generic complexity results for the preferability semantics as reported in Theorem 8 in [5].

To conclude this section, we turn our attention to the basic scheme to encode the reasoning problems. We denote by $stable(F)$ (resp. $adm(F)$, $pref(F)$) the set of stable (resp. admissible, preferred) extensions of a framework F.

Theorem 4 *Let $F = (T, A, \overline{(\cdot)})$ be a framework over a deductive system $(\mathcal{L}_\mathcal{A}, \mathcal{R})$, $\varphi \in \mathcal{L}_\mathcal{A}$, and $\mathcal{E} \in \{stable, adm, pref\}$. Then*

1. *φ is contained in some $E \in \mathcal{E}(F)$ iff $\exists G(\mathcal{E}^G_F \wedge f^G(T, A, \varphi))$ is true;*
2. *φ is contained in all $E \in \mathcal{E}(F)$ iff $\forall G(\mathcal{E}^G_F \rightarrow f^G(T, A, \varphi))$ is true.*

Recall that we have already discussed that, for flat frameworks, these encodings can be simplified accordingly. Moreover, there exist also shortcuts with respect to the encoded reasoning tasks. For instance, since any preferred extension is also admissible and any admissible extension is a subset of preferred extension, deciding whether φ is contained in some preferred extension is the same as deciding whether φ is contained in some admissible extension, which provides an easier encoding (i.e., an encoding with less quantifier alternations). As well, we can apply the (easier) stable encodings in order to deal with preferred extensions in the context of normal frameworks .

Generally speaking, after translating the encodings of the reasoning tasks into prenex normal form, we claim that our encodings are adequate with respect to the generic complexity results of abstract frameworks derived in Theorem 8 in [5], whenever an adequate encoding for the derivability operator $\vdash$ is provided.

4. Examples for Encodings

In this section, we instantiate our abstract translation framework to concrete translations.

4.1. Basic Frameworks

First, we analyze the simple framework as treated, for instance, in [6,2]. In particular, we show that our generic approach coincides with the encodings to propositional logic given by Besnard and Doutre in [2] and thus generalizes their methodology.

We recall the definition of this basic framework.

Definition 3 *A* basic argumentation framework *is a pair* (A, R) *where* A *is a set of* arguments *and* $R \subseteq A \times A$*. If* $(a, b) \in R$ *then we say that* a attacks b*. A set* $S \subseteq A$ *attacks an argument* b *if some* $a \in S$ *attacks* b*.*

The attentive reader might have observed that we used the letter A for denoting the set of assumptions in the abstract framework as well as for denoting the set of arguments in basic frameworks in Definition 3. We will see that this usage is not misleading because the arguments in the basic framework play the role of assumptions in the abstract framework.

How can we represent a basic framework in our general abstract setting? We simply consider the logical system $(A, \emptyset)$, that is, the arguments are our basic vocabulary $\mathcal{L}_{\mathcal{A}}$ (i.e., a set of atomic formulas) and the set of (additional) inference rules is empty. This choice immediately implies that (i) $A \vdash a$ iff $a \in A$ and (ii) $Th(A) = A$. With a slight abuse of notation, we use $f^G(\emptyset, A, a) := g_a$ and, for any set of arguments B, $f^G(\emptyset, A, B) := \bigvee_{b \in B} g_b$ in the encodings. Then a basic argumentation framework (A, R) is simulated by the general framework $(\emptyset, A, \overline{(\cdot)})$ with $\overline{a} = \{b \mid (b, a) \in R\}$. Obviously, the framework is flat, and thus we can avoid the check for closure within our encodings.

We start with stable extensions, where our encoding (6) reduces to

$$\begin{aligned} stable_F^G := \bigwedge_{a \in A} \big(g_a \leftrightarrow \neg f^G(\emptyset, A, \overline{a})\big) \ &:= \bigwedge_{a \in A} \big(g_a \leftrightarrow (\neg \bigvee_{b \in \overline{a}} g_b)\big) \\ := \bigwedge_{a \in A} \big(g_a \leftrightarrow (\bigwedge_{b:(b,a) \in R} \neg g_b)\big). \end{aligned}$$

The latter formula coincides with the encoding from Proposition 5 in [2] by replacing the guessing atoms g_a with the corresponding atoms a, for each $a \in A$.

Admissible extensions for (A, R) are characterized using (7) from Theorem 2. By evaluating $f^G(\emptyset, A, \overline{a})$ and $f^{G'}(\emptyset, A, \overline{a})$, we get

$$\begin{aligned} adm_F^G := \bigwedge_{a \in A} \big(g_a \rightarrow (\neg \bigvee_{b:(b,a) \in R} g_b)\big) \wedge \\ \exists G' \Big[\bigwedge_{a \in A} \Big(g'_a \leftrightarrow \big(g_a \vee \neg \bigvee_{b:(b,a) \in R} g_b\big)\Big) \wedge \bigwedge_{a \in A} \Big(g_a \rightarrow \neg \bigvee_{b:(b,a) \in R} g'_b)\Big)\Big]. \end{aligned}$$

Now we "plug in" the definition of the g'_a's from the first conjunct in the second line to the second conjunct. We then can omit these definitions and the existential quantifiers and get

$$adm_F^G := \bigwedge_{a \in A} (g_a \rightarrow \neg \bigvee_{b:(b,a)\in R} g_b) \wedge \bigwedge_{a \in A} \Big(g_a \rightarrow \neg \bigvee_{b:(b,a)\in R} (g_b \vee \neg \bigvee_{c:(c,b)\in R} g_c)\Big)$$

We rewrite the second conjunct of adm_F^G, viz.

$$\bigwedge_{a \in A} \Big(g_a \rightarrow \neg \bigvee_{b:(b,a)\in R} (g_b \vee \neg \bigvee_{c:(c,b)\in R} g_c)\Big)$$

to

$$\bigwedge_{a \in A} \Big(g_a \rightarrow \neg \bigvee_{b:(b,a)\in R} g_b\Big) \wedge \bigwedge_{a \in A} \Big(g_a \rightarrow \neg \bigvee_{b:(b,a)\in R} \bigwedge_{c:(c,b)\in R} \neg g_c\Big). \qquad (8)$$

The first conjunct of this expression absorbs the first conjunct in adm_F^G, and thus adm_F^G is equivalent to (8), which itself is equivalent (module variable renaming as in the case of stable extensions above) to the encoding presented in [2].

Finally, for the encoding of the preferred extensions, we use a different concept as in [2], where the preferred extensions are characterized via *maximal* models of propositional formulas. Since we have the full power of QBFs, we can characterize these extensions via ordinary models using our encoding schema from above. Maximality is checked on the object level (i.e., within the resulting QBF). In particular, we get the following theorem.

Theorem 5 *Let (A, R) be an argumentation framework, $F = (\emptyset, A, \overline{(\cdot)})$ the corresponding abstract framework, and adm_G^F as reduced above. Moreover, let $S \subseteq A$, $I \subseteq G$, such that $a \in S$ iff $g_a \in I$ for each $a \in A$. Then S is preferred iff*

$$pref_F^G := adm_F^G \wedge \neg\exists G' \Big((G < G') \wedge adm_F^{G'}\Big)$$

is true under I.

Using our generic scheme, we additionally get immediately the encodings for the reasoning problems as discussed in Theorem 4.

4.2. Abductive Framework

We proceed with another simple framework, namely Theorist [22], which has been shown to be captured by abstract frameworks as follows. We use $(T, A, \overline{(\cdot)})$, with T and A being sets of propositional formulas. For each $a \in A$, $\bar{a}$ is just $\neg a$, and $\vdash$ is the classical derivability operator. We first have to encode $\vdash$.

Proposition 3 *For any propositional theories T and A, and for any formula φ, let V be the set of atoms occurring in T, A, or φ, and $G = \{g_a \mid a \in A\}$ be new atoms. Then,*

$$f^G(T, A, \varphi) := \forall V \Big(\big(\bigwedge_{t \in T} t \wedge \bigwedge_{a \in A} (g_a \rightarrow a)\big) \rightarrow \varphi \Big)$$

is an encoding of classical derivability in the sense of Definition 1.

With this instantiation, we can encode all reasoning tasks under consideration. In particular, we can characterize *extensions* in the sense of [22]. Such an extension is defined as $Th(T \cup S)$, where $S \subseteq A$ and S is a maximal subset of A (with respect to set inclusion), such that $T \cup S$ remains consistent. The relation to argumentation frameworks is as follows.

Proposition 4 ([3]) *Given a Theorist abductive framework (T, A), E is an extension of (T, A) iff E is a stable extension of the corresponding argumentation framework.*

Hence, using the concrete realization of the derivability operator from Proposition 3, we immediately obtain an encoding for Theorist-like extensions by plugging f^G into the abstract encodings for stable expansions given in Theorem 1.

4.3. Auto-epistemic Logic

We consider auto-epistemic logic (AEL) [21] in the context of argumentation frameworks. Then AEL has as the underlying language $\mathcal{L}_\mathcal{A}$ a modal logic with the modal operator L, but $\mathcal{R}$ are the classical inference rules. As assumptions we have propositional atoms $L\alpha$ and $\neg L\alpha$. The contrary of $\neg L\alpha$ is α, and the contrary of $L\alpha$ is $\neg L\alpha$.

We instantiate our encodings for stable extensions following this framework. Therefore, consider $F = (T, A, \overline{(\cdot)})$, with T a modal theory, A containing literals $L\alpha$ and $\neg L\alpha$, for each subformula $L\alpha$ in T, and $\overline{(\cdot)}$ as above. Since $\vdash$ is the classical inference operator, we use f_G as defined in Proposition 3. The exact relation between stable extensions of the framework and stable expansions of an auto-epistemic theory (cf. [21]) is as follows, see Theorem 3.11 in [3].

Proposition 5 *A theory E is a stable extension of the framework corresponding to a modal theory T iff E is a consistent stable expansion of T.*

Hence, our abstract encodings (together with the concrete realization for f^G as defined in Proposition 3) capture stable expansions of T. Moreover, one can show that these encodings reduce (after some simplifications) to the ones presented in [13].

However, AEL provides argumentation frameworks which are neither normal, simple, or flat. Thus, none of the previously mentioned shortcuts in the encodings can be applied and we end up, in the worst case, with QBFs possessing up to three quantifier alternations. In fact, this holds in the case of skeptical reasoning under preferred extensions, i.e., deciding whether a given formula φ is contained in all preferred extensions of a given argumentation framework. This problem was shown to be Π_4^p-complete [5] and our encodings match this intrinsic complexity. For illustration, we briefly sketch the structure of quantifier dependencies for the QBFs which encode this particular problem. According to Theorem 4, we have formulas of the form $\forall G(pref_F^G \rightarrow f^G(T, A, \varphi))$. Observe that $pref_F^G$ has negative polarity in this formula. By inspecting the quantifiers in $pref_F^G$ (according to Theorem 3), we get additional quantifiers $\exists G' \forall G''$ the latter from the subformula $adm_F^{G'}$. The final quantifier $\exists V$ stems from the occurrences of the encodings of the classical derivability (which are present in both polarities within $adm_F^{G'}$), Hence, we end up here with quantifier dependencies $\forall G \exists G' \forall G'' \exists V$.

5. Discussion

Due to the lack of space, we just mention briefly some further potentials of our translation framework based on QBFs. As shown in [3], non-monotonic modal logics, especially auto-epistemic logic [21] can be captured via argumentation frameworks. Moreover, by the notions of admissible and preferred extensions, additional semantics for such logics are obtained. All of these semantics in turn can then be computed via QBFs using our generic framework. For some non-monotonic modal logics such as auto-epistemic logic, this is more or less straight forward since the corresponding argumentation framework relies on classical derivability, which we already encoded sufficiently in Proposition 3. With the obtained encodings at hand, we may also provide an answer to a question raised in [5] where the authors ask how the preferred and admissible semantics of auto-epistemic logic relate to the semantics of parsimonious and moderately grounded expansions [12]. Since the latter have been reduced to QBFs in [13], we thus have a uniform axiomatization of all the formal systems in question.

A further application is to encode different tasks for default logic [23], which is also an instantiation of the abstract framework as shown in [3]. In this case, we have to express a different derivability operator, namely classical logic augmented with monotonic rules. Such an encoding has already been used in the literature, see for instance [8].

Finally, we briefly discuss an important question towards the concrete implementation of argumentation frameworks using QBF solvers. As is apparent by the presented encodings, the resulting QBFs are not in any specific normal form. However, most of the available QBF solvers require the input to be in *prenex conjunctive normal form*. Thus, a further transformation is necessary. This transformation is usually performed in two steps, namely *prenexing* and a transformation of the resulting purely propositional matrix into conjunctive normal form. The drawbacks of this transformation are an increase in both formula size and variable number, or, even worse, the formula's structure is disrupted. Moreover, prenexing cannot be carried out deterministically and the chosen normalization strategy crucially influences the runtimes (also depending on the concrete solver used), see e.g., [10]. However, there are a few solvers which are able to handle arbitrary QBFs (e.g., [11,14]) and recent results [15,11] show that non-normal form approaches are highly beneficial on certain instances. Future work thus includes a careful evaluation how QBF solvers of different types behave on encodings from our framework.

References

[1] O. Arieli and M. Denecker. Reducing preferential paraconsistent reasoning to classical entailment. *Journal of Logic and Computation*, 13(4):557–580, 2003.

[2] P. Besnard and S. Doutre. Checking the acceptability of a set of arguments. In *Proceedings of the 10th International Workshop on Non-Monotonic Reasoning (NMR'04)*, pages 59–64, 2004.

[3] A. Bondarenko, P. M. Dung, R. A. Kowalski, and F. Toni. An abstract, argumentation-theoretic approach to default reasoning. *Artificial Intelligence*, 93(1–2):63–101, 1997.

[4] Y. Dimopoulos, B. Nebel, and F. Toni. Preferred arguments are harder to compute than stable extension. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI'99)*, pages 36–43. Morgan Kaufmann Publishers, 1999.

[5] 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.

[6] S. Doutre and J. Mengin. Preferred extensions of argumentation frameworks: Query answering and computation. In *Proceedings of the First International Joint Conference on Automated Reasoning, (IJCAR'01)*, volume 2083 of LNAI, pages 272–288, Springer, 2001.

[7] 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):321–358, 1995.

[8] U. Egly, T. Eiter, H. Tompits, and S. Woltran. Solving advanced reasoning tasks using quantified boolean formulas. In *Proceedings of the 17th National Conference on Artificial Intelligence (AAAI'00)*, pages 417–422. AAAI Press/MIT Press, 2000.

[9] U. Egly, R. Pichler, and S. Woltran. On deciding subsumption problems. *Annals of Mathematics and Artificial Intelligence*, 43(1–4):255–294, 2005.

[10] U. Egly, M. Seidl, H. Tompits, S. Woltran, and M. Zolda. Comparing different prenexing strategies for quantified boolean formulas. In *Proceedings of the 6th International Conference on the Theory and Applications of Satisfiability Testing (SAT'03). Selected Revised Papers*, volume 2919 of *LNCS*, pages 214–228, 2004.

[11] U. Egly, M. Seidl, and S. Woltran. A solver for QBFs in nonprenex form. Accepted for publication in *Proc. ECAI'06*, 2006.

[12] T. Eiter and G. Gottlob. Complexity of reasoning with parsimonious and moderately grounded expansions. *Fundamenta Informaticae*, 17(1-2):31–53, 1992.

[13] T. Eiter, V. Klotz, H. Tompits, and S. Woltran. Modal nonmonotonic logics revisited: efficient encodings for the basic reasoning tasks. In *Proc. TABLEAUX'02*, volume 2381 of *LNCS*, pages 100–114. Springer, 2002.

[14] M. Ghasemzadeh, V. Klotz, and C. Meinel. Embedding memoization to the semantic tree search for deciding QBFs. In *Proc. AI 2004*, volume 3339 of *LNCS*, pages 681–693, 2004.

[15] E. Giunchiglia, M. Narizzano, and A. Tacchella. Quantifier structure in search based procedures for QBFs. In *Proc. DATE 2006*, 2006.

[16] J. Katz, Z. Hanna, and N. Dershowitz. Space-efficient bounded model checking. In *Proc. DATE 2005*, pages 686–687. IEEE Computer Society, 2005.

[17] H. Kautz, D. McAllester, and B. Selman. Encoding plans in propositional logic. In *Proceedings of the Fifth International Conference on Principles of Knowledge Representation and Reasoning (KR'96)*, pages 374–384. Morgan Kaufmann Publishers, 1996.

[18] H. Kautz and B. Selman. Planning as satisfiability. In *Proceedings of the Tenth European Conference on Artificial Intelligence (ECAI'92)*, pages 359–363. John Wiley & Sons, 1992.

[19] D. Le Berre, M. Narizzano, L. Simon, and A. Tacchella. The second QBF solvers comparative evaluation. In *Proceedings of the 7th International Conference on Theory and Applications of Satisfiability Testing (SAT'04), Revised Selected Papers*, volume 3542 of *LNCS*, pages 376–392. Springer, 2005.

[20] A. C. Ling, D. P. Singh, and S. D. Brown. FPGA logic synthesis using quantified boolean satisfiability. In *Proceedings of the 8th International Conference on Theory and Applications of Satisfiability Testing (SAT'05)*, volume 3569 of *LNCS*, pages 444–450. Springer, 2005.

[21] R. C. Moore. Semantical considerations on nonmonotonic logic. *Artificial Intelligence*, 25:75–94, 1985.

[22] D. Poole. A logical framework for default reasoning. *Artificial Intelligence*, 36(1):270–48, 1988.

[23] R. Reiter. A logic for default reasoning. *Artificial Intelligence*, 13(1–2):81–132, 1980.

[24] J. Rintanen. Constructing conditional plans by a theorem prover. *Journal of Artificial Intelligence Research*, 10:323–352, 1999.

[25] H. Tompits and S. Woltran. Towards implementations for advanced equivalence checking in answer-set programming. In *Proceedings of the 21st International Conference on Logic Programming (ICLP'05)*, volume 3668 of *LNCS*, pages 189–203. Springer, 2005.

[26] C. Wrathall. Complete sets and the polynomial-time hierarchy. *Theoretical Computer Science*, 3(1):23–33, 1976.