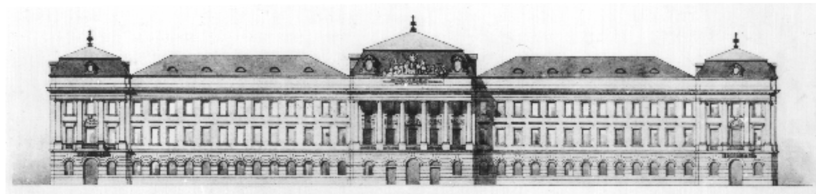
dbai

TECHNICAL REPORT



INSTITUT FÜR INFORMATIONSSYSTEME

ABTEILUNG DATENBANKEN UND ARTIFICIAL INTELLIGENCE

Improved Answer-Set Programming Encodings for Abstract Argumentation

DBAI-TR-2015-92

Sarah A. Gaggl, Norbert Manthey, Alessandro Ronca, Johannes P. Wallner, Stefan Woltran

Institut für Informationssysteme
Abteilung Datenbanken und
Artificial Intelligence
Technische Universität Wien
Favoritenstr. 9
A-1040 Vienna, Austria
Tel: +43-1-58801-18403
Fax: +43-1-58801-18493
sekret@dbai.tuwien.ac.at
www.dbai.tuwien.ac.at

DBAI TECHNICAL REPORT
JULY 2015

TECHNISCHE UNIVERSITÄT WIEN
Vienna University of Technology

DBAI TECHNICAL REPORT

DBAI TECHNICAL REPORT DBAI-TR-2015-92, JULY 2015

Improved Answer-Set Programming Encodings for Abstract Argumentation

Sarah Gaggl [1] **Nobert Manthey** [1] **Alessandro Ronca** [2]
Johannes P. Wallner [3] **Stefan Woltran** [4]

Abstract. The design of efficient solutions for abstract argumentation problems is a crucial step towards advanced argumentation systems. One of the most prominent approaches in the literature is to use Answer-Set Programming (ASP) for this endeavor. In this paper, we present new encodings for three prominent argumentation semantics using the concept of conditional literals in disjunctions as provided by the ASP-system clingo. Our new encodings are not only more succinct than previous versions, but also outperform them on standard benchmarks.

[1]TU Dresden. E-mail: {sarah.gaggl,norbert.manthey}@tu-dresden.de
[2]La Sapienza, Rome. Email: alessandro.a.ronca@gmail.com
[3]University of Helsinki. Email: jwallner@cs.helsinki.fi
[4]TU Wien. E-mail: woltran@dbai.tuwien.ac.at

Acknowledgements: This work has been funded by the Austrian Science Fund (FWF) through projects Y698 and I1102, by the German Research Foundation (DFG) through project HO 1294/11-1, and by Academy of Finland through grants 251170 COIN and 284591.

Contents

1 Introduction

Abstract Argumentation [12, 27] is at the heart of many advanced argumentation systems [2, 6] and is concerned with finding jointly acceptable arguments by taking only their inter-relationships into account. Efficient solvers for abstract argumentation are thus an important development, a fact that is also witnessed by a new competition which takes place in 2015 for the first time [9][1].

To date, several approaches for implementing abstract argumentation exist, many of them following the so-called reduction-based (see [10]) paradigm: hereby, existing efficient software which has originally been developed for other purposes is used. Prominent examples for this approach are (i) the CSP-based system ConArg [3], (ii) SAT-based approaches (e.g. [8, 20]) and (iii) systems which rely on Answer-Set Programming (ASP); see [29] for a comprehensive survey. In fact, ASP [4] is particularly well-suited since ASP systems by default enumerate all solutions of a given program, thus enabling the enumeration of extensions of an abstract argumentation framework in an easy manner. Moreover, disjunctive ASP is capable of expressing problems being even complete for the 2nd level of the polynomial hierarchy. In fact, several semantics for abstract argumentation like preferred, semi-stable [7], or stage [31] are of this high complexity [14, 18].

One particular candidate for an ASP reduction-based system is ASPARTIX [21, 19]. Here, a fixed program for each semantics is provided and the argumentation framework under consideration is just added as an input-database. The program together with the input-database is then handed over to an ASP system of choice in order to calculate the extensions. This makes the ASPARTIX approach easy to adapt and an appealing rapid-prototyping method. The proposed encodings in ASPARTIX for the high-complexity semantics mentioned above come, however, with a certain caveat. This stems from the fact that encodings for such complex programs have to follow a certain saturation pattern, where restricted use of cyclic negation has to be taken care of (we refer to [21] for a detailed discussion). The original encodings followed the definition of the semantics quite closely and thus resulted in quite complex and tricky loop-techniques which are a known feature for ASP experts, but hard to follow for ASP laymen. Moreover, experiments in other domains indicated that such loops also potentially lead to performance bottlenecks.

In this work, we thus aim for new and simpler encodings for the three semantics of preferred, semi-stable, and stage extensions. To this end, we provide some alternative characterizations for these semantics and design our new encodings along these characterizations in such a way that costly loops are avoided. Instead we make use of the ASP language feature of conditional literals in disjunction [28, 23]. Moreover, we perform exhaustive experimental evaluation against the original ASPARTIX-encodings, the ConArg system, and another ASP-variant [19] which makes use of the ASP front-end *metasp* [24], where the required maximization is handled via meta-programming. Our results show that the new ASP encodings not only outperform the previous variants, but also makes ASPARTIX more powerful than ConArg.

The novel encodings together with the benchmark instances are available under `http://dbai.tuwien.ac.at/research/project/argumentation/systempage/#conditional`. An extended version of the paper containing all proofs is available as a technical report [22].

[1] See `http://argumentationcompetition.org` for further information.

2 Background

2.1 Abstract Argumentation

First, we recall the main formal ingredients for argumentation frameworks [12, 1] and survey relevant complexity results (see also [17]).

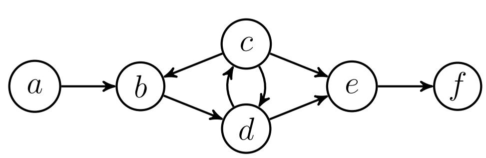
Definition 1. *An* argumentation framework (AF) *is a pair* $F = (A, R)$ *where* A *is a set of arguments and* $R \subseteq A \times A$ *is the attack relation. The pair* $(a, b) \in R$ *means that* a *attacks* b*. An argument* $a \in A$ *is* defended *by a set* $S \subseteq A$ *if, for each* $b \in A$ *such that* $(b, a) \in R$*, there exists a* $c \in S$ *such that* $(c, b) \in R$*. We define the* range of S *(w.r.t.* R*) as* $S^+_R = S \cup \{x \mid \exists y \in S \text{ such that } (y, x) \in R\}$.

Semantics for argumentation frameworks are given via a function σ which assigns to each AF $F = (A, R)$ a set $\sigma(F) \subseteq 2^A$ of extensions. We shall consider here for σ the functions stb, adm, prf, $stage$, and sem which stand for stable, admissible, preferred, stage, and semi-stable semantics respectively.

Definition 2. *Let* $F = (A, R)$ *be an AF. A set* $S \subseteq A$ *is* conflict-free (in F)*, if there are no* $a, b \in S$*, such that* $(a, b) \in R$*.* $cf(F)$ *denotes the collection of conflict-free sets of* F*. For a conflict-free set* $S \in cf(F)$*, it holds that*

- $S \in stb(F)$*, if* $S^+_R = A$*;*
- $S \in adm(F)$*, if each* $s \in S$ *is defended by* S*;*
- $S \in prf(F)$*, if* $S \in adm(F)$ *and there is no* $T \in adm(F)$ *with* $T \supset S$*;*
- $S \in sem(F)$*, if* $S \in adm(F)$ *and there is no* $T \in adm(F)$ *with* $T^+_R \supset S^+_R$*;*
- $S \in stage(F)$*, if there is no* $T \in cf(F)$ *in* F*, such that* $T^+_R \supset S^+_R$*.*

Example 1. *Consider the AF* $F = (A, R)$ *with* $A = \{a, b, c, d, e, f\}$ *and* $R = \{(a, b), (b, d), (c, b), (c, d), (c, e), (d, c), (d, e), (e, f)\}$*, and the graph representation of* F*:*



We have $stb(F) = stage(F) = sem(F) = \{\{a, d, f\}, \{a, c, f\}\}$*. The admissible sets of* F *are* $\emptyset$*,* $\{a\}$*,* $\{c\}$*,* $\{a, c\}$*,* $\{a, d\}$*,* $\{c, f\}$*,* $\{a, c, f\}$*,* $\{a, d, f\}$*, and* $prf(F) = \{\{a, c, f\},\{a, d, f\}\}$*.*

We recall that each AF F possesses at least one preferred, semi-stable, and stage extension, while $stb(F)$ might be empty. However, it is well known that $stb(F) \neq \emptyset$ implies $stb(F) = stage(F) = sem(F)$ as also seen in the above example.

Next, we provide some alternative characterisations for the semantics of our interest. They will serve as the basis of our encodings.

The alternative characterisation for preferred extensions relies on the following idea. An admissible set S is preferred, if each other admissible set E (which is not a subset of S) is in conflict with S.

Proposition 1. *Let $F = (A, R)$ be an AF and $S \subseteq A$ be admissible in F. Then, $S \in prf(F)$ if and only if, for each $E \in adm(F)$ such that $E \not\subseteq S$, $E \cup S \notin cf(F)$.*

Proof. Let $S \in adm(F)$ and assume there exists an admissible (in F) set $E \not\subseteq S$, such that $E \cup S \in cf(F)$. It is well known (see, e.g. [16], Lemma 1) that if two sets E_1, E_2 defend themselves in an AF F, then also $E_1 \cup E_2$ defends itself in F. It follows that $E \cup S \in adm(F)$ and by assumption $S \subset E \cup S$. Thus, $S \notin prf(F)$. For the other direction, let $S \in adm(F)$ but $S \notin prf(F)$. Hence, there exists an $S' \supset S$ such that $S' \in adm(F)$. Clearly, $S' \not\subseteq S$ but $S' = (S \cup S') \in cf(F)$. □

We turn to semi-stable and stage semantics. In order to verify whether a candidate extension S is a stage (resp. semi-stable) extension of an AF F, we check whether for any set S' such that $S' \supset S^+_R$ there is no conflict-free (resp. admissible) set E such that $S' \subseteq E^+_R$. We also show that is sufficient to check this for minimal such sets S'. Observe that the above check is trivially true if S is already stable, mirroring the observation that $stb(F) = stage(F) = sem(F)$ whenever $stb(F) \neq \emptyset$.

Definition 3. *Let $F = (A, R)$ be an AF and $S \subseteq A$. A* cover *of S in F is any $E \subseteq A$ such that $S \subseteq E^+_R$. The set of covers of S in F is denoted by $\Gamma_F(S)$.*

Proposition 2. *Let $F = (A, R)$ be an AF and $S \in cf(F)$ (resp. $S \in adm(F)$). The following propositions are equivalent: (1) S is a stage (resp. semi-stable) extension of F; (2) for each $a \in A \setminus S^+_R$, there is no $E \in \Gamma_F(S^+_R \cup \{a\})$ such that $E \in cf(F)$ (resp. $E \in adm(F)$); (3) for each S' with $S^+_R \subset S' \subseteq A$, there is no $E \in \Gamma_F(S')$, such that $E \in cf(F)$ (resp. $E \in adm(F)$).*

Proof. We give the proof for stage extensions. The result for semi-stable proceeds analogously. (1)⇒(3): Suppose there is an S' with $S^+_R \subset S' \subseteq A$, such that some $E \in \Gamma_F(S')$ is conflict-free in F. By definition, $S^+_R \subset S' \subseteq E^+_R$. Hence, $S \notin stage(F)$. (2)⇒(1): Suppose $S \notin stage(F)$. Thus there exists $T \in cf(F)$ with $S^+_R \subset T^+_R$. Let $a \in T^+_R \setminus S^+_R$. It follows that $T \in \Gamma_F(S \cup \{a\})$. (3)⇒(2) is clear. □

Finally, we turn to the complexity of reasoning in AFs for two major decision problems. For a given AF $F = (A, R)$ and an argument $a \in A$, credulous reasoning under σ denotes the problem of deciding whether there exists an $E \in \sigma(F)$ s.t. $a \in E$. Skeptical Acceptance under σ is the problem of deciding whether for all $E \in \sigma(F)$ it holds that $a \in E$. Credulous reasoning for preferred semantics is NP-complete, while credulous reasoning for semi-stable and stage semantics is Σ^P_2-complete. For preferred, semi-stable, and stage semantics skeptical reasoning is Π^P_2-complete [12, 11, 14, 15, 18].

2.2 Answer-Set Programming

We give an overview of the syntax and semantics of disjunctive logic programs under the answer-sets semantics [25].

We fix a countable set $\mathcal{U}$ of *(domain) elements*, also called *constants*; and suppose a total order $<$ over the domain elements. An *atom* is an expression $p(t_1, \dots, t_n)$, where p is a *predicate* of arity $n \geq 0$ and each t_i is either a variable or an element from $\mathcal{U}$. An atom is *ground* if it is free of variables. $B_{\mathcal{U}}$ denotes the set of all ground atoms over $\mathcal{U}$. A *(disjunctive) rule* r is of the form

$$a_1 \mid \cdots \mid a_n \leftarrow b_1, \dots, b_k,\ not\, b_{k+1}, \dots,\ not\, b_m \tag{1}$$

with $n \geq 0$, $m \geq k \geq 0$, $n + m > 0$, where $a_1, \dots, a_n, b_1, \dots, b_m$ are literals, and "not" stands for *default negation*. The *head* of r is the set $H(r) = \{a_1, \dots, a_n\}$ and the *body* of r is $B(r) = \{b_1, \dots, b_k,\ not\, b_{k+1}, \dots,\ not\, b_m\}$. Furthermore, $B^+(r) = \{b_1, \dots, b_k\}$ and $B^-(r) = \{b_{k+1}, \dots, b_m\}$. A rule r is *normal* if $n \leq 1$ and a *constraint* if $n = 0$. A rule r is *safe* if each variable in r occurs in $B^+(r)$. A rule r is *ground* if no variable occurs in r. A *fact* is a ground rule without disjunction and empty body. An *(input) database* is a set of facts. A program is a finite set of disjunctive rules. For a program π and an input database D, we often write $\pi(D)$ instead of $D \cup \pi$. If each rule in a program is normal (resp. ground), we call the program normal (resp. ground).

For any program π, let U_π be the set of all constants appearing in π. $Gr(\pi)$ is the set of rules $r\sigma$ obtained by applying, to each rule $r \in \pi$, all possible substitutions σ from the variables in r to elements of U_π. An *interpretation* $I \subseteq B_{\mathcal{U}}$ *satisfies* a ground rule r iff $H(r) \cap I \neq \emptyset$ whenever $B^+(r) \subseteq I$ and $B^-(r) \cap I = \emptyset$. I satisfies a ground program π, if each $r \in \pi$ is satisfied by I. A non-ground rule r (resp., a program π) is satisfied by an interpretation I iff I satisfies all groundings of r (resp., $Gr(\pi)$). $I \subseteq B_{\mathcal{U}}$ is an *answer set* of π iff it is a subset-minimal set satisfying the *Gelfond-Lifschitz reduct* $\pi^I = \{H(r) \leftarrow B^+(r) \mid I \cap B^-(r) = \emptyset, r \in Gr(\pi)\}$. For a program π, we denote the set of its answer sets by $\mathcal{AS}(\pi)$.

Modern ASP solvers offer additional language features. Among them we make use of the *conditional literal* [28, 23]. In the head of a disjunctive rule literals may have conditions, e.g. consider the head of rule "$\mathbf{p}(X) : \mathbf{q}(X) \leftarrow$". Intuitively, this represents a head of disjunctions of atoms $\mathbf{p}(a)$ where also $\mathbf{q}(a)$ is true.

2.3 ASP Encodings for AFs

For our novel encodings we utilize basic encodings for AFs, conflict-free sets, and admissible sets from [21]. An AF is represented as a set of facts.

Definition 4. *Let $F = (A, R)$ be an AF. We define $\hat{F} = \{\mathbf{arg}(a) \mid a \in A\} \cup \{\mathbf{att}(a, b) \mid (a, b) \in R\}$.*

In the following definition we first formalize the correspondence between an extension, as subset of arguments, and an answer set of an ASP encoding; then we extend it to the one between sets of extensions and answer sets respectively.

Listing 1 Module π_{cf}

```
in(X) ← arg(X), not out(X).
out(X) ← arg(X), not in(X).
← att(X,Y), in(X), in(Y).
```

Listing 2 Module π_{def}

```
defeated(X) ← in(Y), att(Y,X).
undefended(X) ← att(Y,X), not defeated(Y).
← in(X), undefended(X).
```

Definition 5. *Let $\mathcal{S} \subseteq 2^{\mathcal{U}}$ be a collection of sets of domain elements and let $\mathcal{I} \subseteq 2^{B_{\mathcal{U}}}$ be a collection of sets of ground atoms. We say that $S \in \mathcal{S}$ and $I \in \mathcal{I}$ correspond to each other, in symbols $S \cong I$, iff $S = \{a \mid \mathbf{in}(a) \in I\}$. We say that $\mathcal{S}$ and $\mathcal{I}$ correspond to each other, in symbols $\mathcal{S} \cong \mathcal{I}$, iff (i) for each $S \in \mathcal{S}$, there exists an $I \in \mathcal{I}$, such that $I \cong S$; and (ii) for each $I \in \mathcal{I}$, there exists an $S \in \mathcal{S}$, such that $S \cong I$.*

It will be convenient to use the following notation and result later in Section 3.

Definition 6. *Let $I, J \in 2^{B_{\mathcal{U}}}$ be sets of ground atoms. We say that I and J are equivalent, in symbols $I \equiv J$, iff $\{\mathbf{in}(a) \mid \mathbf{in}(a) \in I\} = \{\mathbf{in}(a) \mid \mathbf{in}(a) \in J\}$.*

Let $I, J \in 2^{B_{\mathcal{U}}}$, and $S \in 2^{\mathcal{U}}$. If $I \equiv J$ and $I \cong S$, then $J \cong S$.

In Listing 1 we see the ASP encoding for conflict-free sets, while Listing 2 shows defense of arguments. The encoding for admissible sets is given by $\pi_{adm} = \pi_{cf} \cup \pi_{def}$. The following has been proven in [21, Proposition 3.2].

Proposition 3. *For any AF $F = (A, R)$, and any $I \in \mathcal{AS}(\pi_{cf}(\hat{F}))$, $\mathcal{P} = \{\{a \mid \mathbf{in}(a) \in I\}, \{a \mid \mathbf{out}(a) \in I\}\}$ is a partition of A.*

Correctness of the encodings π_{cf} and π_{adm} was proven in [21].

Proposition 4. *For any AF F, we have (i) $cf(F) \cong \mathcal{AS}(\pi_{cf}(\hat{F}))$, and (ii) $adm(F) \cong \mathcal{AS}(\pi_{adm}(\hat{F}))$.*

Next, we characterize the encoding π_{range} (Listing 3), which, given a module computing some extension S (via $\mathbf{in}$) of an AF (A, R), returns its range S_R^+ (via $\mathbf{range}$) and also collects the arguments not contained in the range. We indicate via $\mathbf{unstable}$ that S is not stable, i.e. $S_R^+ \subset A$.

Let $F = (A, R)$ be an AF, and π be a program not containing the predicates $\mathbf{range}(\cdot)$, $\mathbf{out\_of\_range}(\cdot)$ and $\mathbf{unstable}$. Let $I \subseteq B_A$ and $S \subseteq A$ s.t. $I \cong S$. Furthermore let $\pi^+ = \pi \cup \pi_{range}$ and

$$
\begin{aligned}
I^+ = I &\cup \{\mathbf{range}(a) \mid a \in S_R^+\} \cup \{\mathbf{out\_of\_range}(a) \mid a \in A \setminus S_R^+\} \\
&\cup \{\mathbf{unstable} \mid S_R^+ \subset A\}.
\end{aligned}
\tag{2}
$$

Listing 3 Module π_{range}

```
range(X) ← in(X).
range(Y) ← in(X), att(X,Y).
out_of_range(X) ← not range(X), arg(X).
unstable ← out_of_range(X), arg(X).
```

Listing 4 Module π_{eq}

```
eq_upto(Y) ← inf(Y), in(Y), inN(Y).
eq_upto(Y) ← inf(Y), out(Y), outN(Y).
eq_upto(Y) ← succ(Z,Y), in(Y), inN(Y), eq_upto(Z).
eq_upto(Y) ← succ(Z,Y), out(Y), outN(Y), eq_upto(Z).
eq ← sup(Y), eq_upto(Y).
```

Then, $I \in \mathcal{AS}(\pi(\hat{F}))$, if and only if $I^+ \in \mathcal{AS}(\pi^+(\hat{F}))$.

The preferred, semi-stable [21] and stage semantics [19] utilize the so-called *saturation technique*. We sketch here the basic ideas. Intuitively, in the saturation technique encoding for preferred semantics we make a first guess for a set of arguments in the framework, and then we verify if this set is admissible (via module π_{adm}). To verify if this set is also subset maximal admissible, a second guess is carried out via a disjunctive rule. If this second guess corresponds to an admissible set that is a proper superset of the first one, then the first one cannot be a preferred extension. Using the saturation technique now ensures that if all second guesses “fail” to be a strictly larger admissible set of the first guess, then there is one answer-set corresponding to this preferred extension. Usage of default negation within the saturation technique for the second guess is restricted, and thus a loop-style encoding is employed that checks if the second guess is admissible and a proper superset of the first guess.

Roughly, a loop construct in ASP checks a certain property for the least element in a set (here we use the predicate $\mathbf{inf}(\cdot)$), and then checks this property “iteratively” for each (immediate) successor (via predicate $\mathbf{succ}(\cdot,\cdot)$). If the property holds for the greatest element ($\mathbf{sup}(\cdot)$), it holds for all elements. In Listing 4 we illustrate loop encodings, where we see a partial ASP encoding used for preferred semantics in [21] that derives **eq** if the first and second guesses are equal, i.e. the predicates corresponding to the guesses via $\mathbf{in}(\cdot)$, resp. $\mathbf{out}(\cdot)$, and $\mathbf{inN}(\cdot)$, resp. $\mathbf{outN}(\cdot)$, are true for the same constants.

Another variant of ASP encodings for preferred, semi-stable and stage semantics is developed by [19]. There so-called meta-asp encodings are used, which allow for minimizing statements w.r.t. subset inclusion directly in the ASP language [24]. For instance, π_{adm} can then be augmented with a minimizing statement on the predicate **out**, to achieve an encoding of preferred semantics.

3 Encodings

Here we present our new encodings for preferred, semi-stable, and stage semantics via the novel characterizations.

Listing 5 Module π_{satprf^2}

```
1 nontrivial ← out(X).
2 witness(X):out(X) ← nontrivial.
3 spoil | witness(Z):att(Z,Y) ← witness(X), att(Y,X).
4 spoil ← witness(X), witness(Y), att(X,Y).
5 spoil ← in(X), witness(Y), att(X,Y).
6 witness(X) ← spoil, arg(X).
7 ← not spoil, nontrivial.
```

3.1 Encoding for Preferred Semantics

The encoding for preferred semantics is given by $\pi_{prf^2} = \pi_{adm} \cup \pi_{satprf^2}$, where π_{satprf^2} is provided in Listing 5. We first give the intuition of the program. A candidate S for being preferred in an AF $F = (A, R)$ is computed by the program π_{adm} via the $\mathbf{in}(\cdot)$ predicate, and is already known admissible. If all arguments in A are contained in S we are done[2]. Otherwise, the remainder of the program π_{satprf^2} (Lines 2 and 7) is used to check whether there exists a set $E \in adm(F)$ such that $E \not\subseteq S$ and not in conflict with S. We start to build E by guessing some argument not contained in S (Line 2) and then in Line 3 we repeatedly add further arguments to E unless the set defends itself (otherwise we eventually derive $\mathbf{spoil}$). Then, we check whether E is conflict-free (Line 4) and E is not in conflict with S (Line 5). If we are able to reach this point without deriving $\mathbf{spoil}$, then the candidate S cannot be an answer-set (Line 7). This is in line with Proposition 1, which states that in this case S is not preferred.

By inspecting Listing 5 we also see important differences w.r.t. the encodings for preferred semantics of [21]. In our new encodings, the "second guess" via predicate $\mathbf{witness}(\cdot)$ is constructed through conditional disjunction instead of simple disjunction. Usage of the former allows to construct the witness set already with defense of arguments in mind. Furthermore loops, such as the one shown in Listing 4 that checks if the second guess is equal to first one or a loop construct that checks if every argument is defended, can be avoided, since these checks are partially incorporated into Line 2 of Listing 5 and into simpler further checks.

Correctness of this new encoding is stated and proved in the following proposition.

Proposition 5. *For any AF F, we have $prf(F) \cong \mathcal{AS}(\pi_{prf^2}(\hat{F}))$.*

Proof. According to Definition 5, we have to prove (i) and (ii). With line numbers we refer here to the ASP encoding shown in Listing 5. We employ the *splitting theorem* [26] in order to get a characterisation of $\mathcal{AS}(\pi_{prf^2}(\hat{F}))$, in which the sub-programs π_{satprf^2} and π_{adm} are considered separately. The splitting set is $C_{prf^2} = \{\mathbf{arg}(\cdot), \mathbf{att}(\cdot,\cdot), \mathbf{in}(\cdot), \mathbf{out}(\cdot), \mathbf{defeated}(\cdot), \mathbf{undefended}(\cdot)\}$, and we obtain

$$\mathcal{AS}(\pi_{prf^2}(\hat{F})) = \bigcup_{J \in \mathcal{AS}(\pi_{adm}(\hat{F}))} \mathcal{AS}(J \cup \pi_{satprf^2}). \tag{3}$$

[2]Note, this is only the case when there are no attacks in F.

Proof (i). We prove that each preferred extension $S \in prf(F)$ has a corresponding answer-set $I \in \mathcal{AS}(\pi_{prf^2}(\hat{F}))$. From Equation (3) we know that $I \in \mathcal{AS}(\pi_{prf^2}(\hat{F}))$ if $I \in \mathcal{AS}(J \cup \pi_{satprf^2})$, for some $J \in \mathcal{AS}(\pi_{adm}(\hat{F}))$. Moreover $S \in prf(F)$ implies $S \in adm(F)$, hence by Proposition 4 there is $J \in \mathcal{AS}(\pi_{adm}(\hat{F}))$ s.t. $J \cong S$. In the following we distinguish between two complementary cases.

In case $R = \emptyset$, the set $S = A$ is the only preferred one, since it is trivially admissible and it cannot be contained in another set of arguments. We show $I = J$ is a subset-minimal model of $(J \cup \pi_{satprf^2})^J$. The subset-minimality is evident. Then, $\mathbf{out}(a) \notin J$ for any $a \in A$ by Proposition 3, hence J satisfies the rule at Line 1. Since $\mathbf{nontrivial} \notin J$, J satisfies the rules at Lines 2, and 7. Every other rule is satisfied because $\mathbf{att}(a, b) \notin \hat{F}$ for any $a, b \in A$.

In case $R \neq \emptyset$ we can build an interpretation I and prove that I is an answer-set by contraposition, i.e. if there is an $L \subset I$ which satisfies $(J \cup \pi_{satprf^2})^I$, then $S \notin prf(F)$. We define $I = J \cup \{\mathbf{spoil}, \mathbf{nontrivial}\} \cup \{\mathbf{witness}(a) \mid a \in A\}$. We have $I \cong S$ since $I \equiv J$. The set I satisfies $(J \cup \pi_{satprf^2})^I$ (got from $Gr(J \cup \pi_{satprf^2})$ by just removing the rule at Line 7), as $J \subseteq I$ and I contains all the heads of the rules in $(J \cup \pi_{satprf^2})^I$. Notice that $R \neq \emptyset$ guarantees that the head of the rule at Line 2 is non-empty.

Now we describe the necessary shape of L, in order to prove the main assertion next. L must contain $\mathbf{nontrivial}$ because of the rule at Line 1. Indeed $\mathbf{out}(c) \in L$ for some $c \in A \setminus S$, since $J \subseteq L$ with $J \cong S$ and $S \in cf(F)$ (since $S \in adm(F)$), which implies the existence of $c \in A \setminus S$ (we cannot have simultaneously $R \neq \emptyset$, $S \in cf(F)$ and $S = A$), which implies $\mathbf{out}(c) \in J$ by Proposition 3. We have $\mathbf{spoil} \notin L$, otherwise also $\{\mathbf{witness}(a) \mid a \in A\}$ would be in L (because of the rule at Line 6), making L equal to I, but they are different by assumption.

Now we show that, given L, it is possible to find a set $U \in adm(F)$ s.t. $U \not\subseteq S$ and $U \cup S \in cf(F)$, which implies $S \notin prf(F)$ by Proposition 1. We define $U = \{a \mid \mathbf{witness}(a) \in L\}$, and we show all the required properties:

$\underline{U \in cf(F)}$, otherwise we would have two arguments a, b attacking each other, meaning $\{\mathbf{witness}(a), \mathbf{witness}(b), \mathbf{att}(a, b)\} \subseteq L$, which implies $B(r) \subseteq L$ and $H(r) \not\subseteq L$ for some rule r in the grounding of the rule at Line 3, since $\mathbf{spoil} \notin L$.

<u>Each $a \in U$ is defended by U</u>, otherwise it would be possible to find two atoms $\mathbf{witness}(a) \in L$ $[a \in U]$[3] and $\mathbf{att}(b, a) \in L$ $[(b, a) \in R]$ for which there is no $\mathbf{witness}(c) \in L$ $[c \in U]$ s.t. $\mathbf{att}(c, b) \in L$ $[(c, b) \in R]$, thus violating the rule at Line 4, since $\mathbf{spoil} \notin L$.

$\underline{U \not\subseteq S}$. Indeed if we assume $U \subseteq S$, then for every $\mathbf{witness}(a) \in L$ we have $a \in S$ (by definition of U), which corresponds to $\mathbf{in}(a) \in J$ ($S \cong J$), implying $\mathbf{out}(a) \notin J$ (by Proposition 3), making it impossible for L to satisfy the rule at Line 2, since $\mathbf{nontrivial} \in L$.

$\underline{\{U \cup S\} \in cf(F)}$. The sets U and S are conflict-free, so we have to show that there cannot be attack relations between the two sets: an argument $a \in S$ cannot attack an argument $b \in U$, otherwise we would have $\{\mathbf{witness}(b), \mathbf{in}(a), \mathbf{att}(a, b)\} \subseteq L$, which implies $B(r) \subseteq L$ and $H(r) \not\subseteq L$ for some rule in the grounding of the rule at Line 5, since $\mathbf{spoil} \notin L$; an argument $b \in U$ cannot attack an argument $a \in S$, otherwise an argument $c \in S$ should attack b by admissibility of

[3] In this proof, the square brackets are used to point out an immediate implication of the statement preceding them. Usually the statement is about the framework F and the implication about an interpretation, or the other way around.

S, thus violating the previous point.

Proof (ii). We prove that each $I \in \mathcal{AS}(\pi_{prf^2}(\hat{F}))$ corresponds to an $S \in prf(F)$. From Equation (3) we see that $I \in \mathcal{AS}(\pi_{prf^2}(\hat{F}))$ only if $I \in \mathcal{AS}(J \cup \pi_{satprf^2})$ for some $J \in \mathcal{AS}(\pi_{adm}(\hat{F}))$. We have $I \equiv J$, because $J \subseteq I$, and I does not have any additional ground atom $\mathbf{in}(a)$, since $\mathbf{in}(\cdot)$ does not appear in the head of any rule of π_{satprf^2}. By Proposition 4 there exists $S \in adm(F)$ s.t. $S \cong J$, hence $S \cong I$ by Section 2.3. We show that S is also preferred in F, by distinguishing between two complementary cases.

<u>$\mathbf{nontrivial} \notin I$</u>: we have $\mathbf{out}(a) \notin I$ for any $a \in A$, otherwise the rule at Line 1 would be violated. By Proposition3 this implies $\mathbf{in}(a) \in I$ for every $a \in A$, and the same is true for J ($J \cong I$), which we know to be admissible. Hence, $S = A$ and $S \in prf(F)$.

<u>$\mathbf{nontrivial} \in I$</u>: we prove that S is preferred by contraposition, i.e. if $S \notin prf(F)$ then I is not a subset-minimal model of $(J \cup \pi_{satprf^2})^I$. We have that I must have a clear shape in order to satisfy $(J \cup \pi_{satprf^2})^I$. In particular $J \subseteq I$. Then $\mathbf{spoil} \in I$ because of the rule at Line 7 hence, $\mathbf{witness}(a) \in I$ for each $\mathbf{arg}(a) \in I$ because of the rule at Line 6. Summing up we have $J \cup \{\mathbf{nontrivial}, \mathbf{spoil}\} \cup \{\mathbf{witness}(a) \mid a \in A\} \subseteq I$. Finally we show that $I \notin \mathcal{AS}(J \cup \pi_{satprf^2})$, since we are able to build an interpretation $L \subset I$ satisfying the reduct $(J \cup \pi_{satprf^2})^I$. We remind that $S \notin prf(F)$ means that there exists $T \in prf(F)$ s.t. $S \subset T$. We use T to build the interpretation $L = J \cup \{\mathbf{nontrivial}\} \cup \{\mathbf{witness}(a) \mid a \in T\}$. We have $L \subset I$, because it does not contain $\mathbf{spoil}$ and $T \subseteq A$. In the following we show that L is a model of the reduct, because it contains J and it satisfies each rule in $Gr(\pi_{satprf^2})$.

L satisfies the rule at Line 2, because there exists $\mathbf{witness}(a) \in L$ s.t. $\mathbf{out}(a) \in L$, for some $a \in T \setminus S$ (the element a exists because T is a proper superset of S). [4]

Since T is admissible, for each $a \in T$ [$\mathbf{witness}(a) \in L$] attacked by $b \in A$ [$\mathbf{att}(b, a) \in \hat{F}$] there exists $c \in T$ [$\mathbf{witness}(c) \in L$] attacking b [$\mathbf{att}(c, b) \in \hat{F}$]. Hence L satisfies the rule at Line 3, even though $\mathbf{spoil} \notin L$.

L does not contain the body of any rule in the grounding of the rule at Line 4, otherwise T would not be conflict free. L does not contain the body of any rule in the grounding of the rule at Line 5, otherwise T would not be conflict free, since $S \subset T$. L does not contain the body of any rule in the grounding of the rule at Line 6, because it does not contain $\mathbf{spoil}$. □ □

3.2 Encodings for Semi-Stable and Stage Semantics

Semi-stable semantics The encoding for semi-stable semantics is given by $\pi_{sem^2} = \pi_{adm} \cup \pi_{range} \cup \pi_{satsem^2}$, with π_{satsem^2} shown in Listing 6. We first give the intuition. A candidate S for being semi-stable is computed by the program $\pi^+_{adm} = \pi_{adm} \cup \pi_{range}$ via the $\mathbf{in}(\cdot)$ predicate and is known admissible. The module π_{range} computes the range and derives $\mathbf{unstable}$ iff the extension is not stable. If S is stable, we are done. Otherwise the remainder of the program π_{satsem^2} is used to check whether an admissible cover E of a superset of the range S^+_R exists. Starting from S^+_R (Line 2), a superset is achieved by adding at least one element out of it (Line 1). Then a cover

[4] If $a \in T \setminus S$, then $a \notin S$, then $\mathbf{in}(a) \notin I$ ($S \cong I$), then $\mathbf{in}(a) \notin J$ ($I \equiv J$), then $\mathbf{out}(a) \in J$ (by Proposition 3), then $\mathbf{out}(a) \in L$ ($J \subset L$). Summing up, if $a \in T \setminus S$, then $\mathbf{out}(a) \in L$, and $\mathbf{witness}(a) \in L$ by definition.

Listing 6 Module π_{satsem^2}

```
1 larger_range(X):out_of_range(X) ← unstable.
2 larger_range(X) ← range(X), unstable.
3 witness(X) | witness(Z):att(Z,X) ← larger_range(X), unstable.
4 spoil ← witness(X), witness(Y), att(X,Y), unstable.
5 spoil | witness(Z):att(Z,Y) ← witness(X), att(Y,X), unstable.
6 witness(X) ← spoil, arg(X), unstable.
7 larger_range(X) ← spoil, arg(X), unstable.
8 ← not spoil, unstable.
```

is found (Line 3), which is admissible (Lines 3 and 4). If we are able to reach this point without deriving **spoil** (that is always a possibility for satisfying the constraints), then the candidate S cannot be an answer-set (Line 3). This is in line with Proposition 2, which states that in this case S is not semi-stable. Here we state the correctness of the encoding, a full proof is given in [22].

Proposition 6. *For any AF $F = (A, R)$, we have $sem(F) \cong \mathcal{AS}(\pi_{sem^2}(\hat{F}))$.*

Stage semantics The encoding for stage semantics is given by $\pi_{stage^2} = \pi_{cf} \cup \pi_{range} \cup \pi_{satsem^2} \setminus \{r_{admcov}\}$, where r_{admcov} is the rule at Line 5 of Listing 6. The only differences w.r.t. the encoding for semi-stable semantics are: (i) it employs π_{cf} instead of π_{adm}, thus the candidate sets are only conflict-free; and (ii) it lacks the rule at Line 5, hence it considers all the conflict-free covers of the candidate set, which is still in line with Proposition 2. A proof sketch for the forthcoming correctness result is given in [22].

Proposition 7. *For any AF $F = (A, R)$, we have $stage(F) \cong \mathcal{AS}(\pi_{stage^2}(\hat{F}))$.*

4 Evaluation

We tested the novel encodings (NEW) extensively and compared them to the original (ORIGINAL) and metasp (META) encodings as well as to the system ConArg [3]. For the novel and original encodings we used *Clingo 4.4* and for the metasp encodings we used *gringo3.0.5/clasp3.1.1* all from the Potassco group[5]. As benchmarks, we considered a collection of frameworks which have been used by different colleagues for testing before consisting of structured and randomly generated AFs, resulting in 4972 frameworks. In particular we used parts of the instances Federico Cerutti provided to us which have been generated towards an increasing number of SCCs [30]. Further benchmarks were used to test the system *dynpartix* and we included the instances provided by the ICCMA 2015 organizers. The full set is available at `http://dbai.tuwien.ac.at/research/project/argumentation/systempage/#conditional`.

For each framework the task is to enumerate all solutions. The computation has been performed on an Intel Xeon E5-2670 running at 2.6 GHz. From the 16 available cores we used only every

[5] `http://potassco.sourceforge.net`

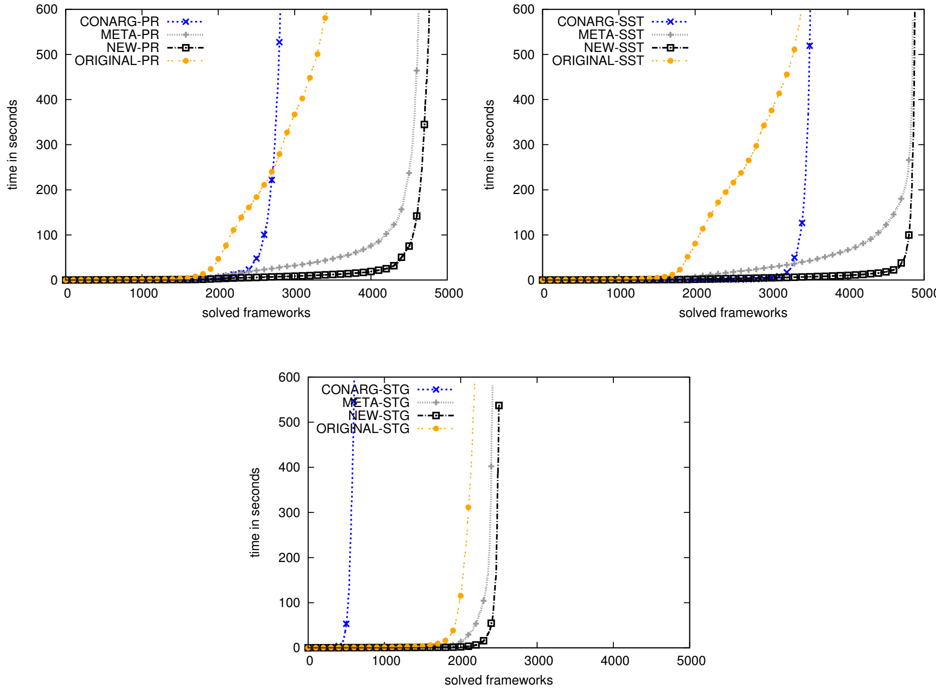


Figure 1 Runtimes for preferred (PR), semi-stable (SST) and stage (STG) semantics.

PR	usc	solved	med	SST	usc	solved	med	STG	usc	solved	med
ConArg	60	2814	43.65	ConArg	50	3509	1.03	ConArg	-	606	600.00
Original	-	3425	180.36	Original	-	3386	211.96	Original	-	2185	600.00
Meta	1	4626	20.83	Meta	2	4830	17.70	Meta	5	2419	600.00
New	101	4765	5.77	New	13	4879	3.30	New	82	2501	384.92

Table 1 Summary of test results.

fourth core to allow a better utilization of the CPU's cache. We applied a 10 minutes timeout, allowing to use at most 6.5 GB of main memory.

It turns out that for each semantics the new encodings significantly outperform the original ones as well as the system ConArg. Furthermore, there is a clear improvement to the metasp encodings, as illustrated in Fig. 1 which shows the cactus plots of the required runtime to solve frameworks (x-axis) with the respective timeout (y-axis) for the three discussed semantics. While for preferred and semi-stable semantics the novel encodings are able to solve more than 4700 instances (out of 4972), one can observe a different trend for stage semantics. There, the new encodings return the best result with 2501 solved instances. Table 1 gives a summary of the test results, where *usc* denotes the unique solver contribution, i.e. the number of AFs which could only be solved by the particular solver, *solved* gives the number of solved instances by the solver, and *med* is the median of the computation time of the solver. Interestingly, ConArg is able to solve 60 (resp. 50) instances for preferred (resp. semi-stable) semantics which are not solvable by the other systems. However, the novel encodings are able to uniquely solve 101 (resp. 82) instances for preferred (resp. stage) semantics. The original encodings have no *unique solver contribution* for all of the considered semantics, thus it is save to replace them with the new encodings. The entries for the median also show that all the novel encodings perform much faster than the other systems, except for semi-stable where ConArg has the lowest median. However, here ConArg is able to solve about 1300 instances less than the novel encodings.

Another interesting observation is that the grounding size of all new encodings is significantly smaller than of both the original and the metasp encodings.

5 Conclusion

In this work, we have developed novel ASP encodings for computationally challenging problems arising in abstract argumentation. Our new encodings for preferred, semi-stable, and stage semantics avoid complicated loop constructs present in previous encodings. In addition to being more succinct, our empirical evaluation showed that a significant performance boost was achieved compared to the earlier ASP encodings, and that our encodings outperform the state-of-the-art system ConArg. From an ASP perspective, our results indicate that loops in saturation encodings (as used in the previous encodings in [21]) are a severe performance bottleneck which should be avoided.

In future work, we plan to compare our results also with the systems CEGARTIX [20] and ArgSemSAT [8]. Furthermore, we also aim for finding better ASP encodings for the ideal [13] and eager semantics [5].

A Proofs

Let $I, J \in 2^{B_\mathcal{U}}$, and $S \in 2^\mathcal{U}$. If $I \equiv J$ and $I \cong S$, then $J \cong S$.

Proof. If $I \equiv J$, then $\{\mathbf{in}(a) \mid \mathbf{in}(a) \in I\} = \{\mathbf{in}(a) \mid \mathbf{in}(a) \in J\}$, then $\{a \mid \mathbf{in}(a) \in I\} = \{a \mid \mathbf{in}(a) \in J\}$. Since $S = \{a \mid \mathbf{in}(a) \in I\}$, therefore $S = \{a \mid \mathbf{in}(a) \in J\}$, meaning $J \cong S$. □

Let $F = (A, R)$ be an AF, and π be a program not containing the predicates $\mathbf{range}(\cdot)$, $\mathbf{out\_of\_range}(\cdot)$ and $\mathbf{unstable}$. Let $I \subseteq B_A$ and $S \subseteq A$ s.t. $I \cong S$. Furthermore let $\pi^+ = \pi \cup \pi_{range}$ and

$$\begin{aligned} I^+ = I &\cup \{\mathbf{range}(a) \mid a \in S_R^+\} \cup \{\mathbf{out\_of\_range}(a) \mid a \in A \setminus S_R^+\} \\ &\cup \{\mathbf{unstable} \mid S_R^+ \subset A\}. \end{aligned} \tag{2}$$

Then, $I \in \mathcal{AS}(\pi(\hat{F}))$, if and only if $I^+ \in \mathcal{AS}(\pi^+(\hat{F}))$.

Proof. We prove the two implications. We make use of the *splitting theorem* [26], with the splitting set containing all the predicates in π.

$$\mathcal{AS}(\pi^+(\hat{F})) = \bigcup_{I \in \mathcal{AS}(\pi(\hat{F}))} \mathcal{AS}(I \cup \pi^+) \tag{4}$$

With line numbers we refer here to the ASP encoding shown in Listing 3.

Only-If Direction. Given $I \in \mathcal{AS}(\pi(\hat{F}))$ and Equation (2), we show $I^+ \in \mathcal{AS}(I \cup \pi^+)$, which implies $I^+ \in \mathcal{AS}(\pi^+(\hat{F}))$ by Equation (4).

We have that I^+ satisfies $(I \cup \pi^+)^{I^+}$. It contains I. It satisfies the rules at Lines 1 and 2, since it contains $\mathbf{range}(a)$ for each $a \in S_R^+$, thus for each $a \in A$ [$\mathbf{arg}(a) \in \hat{F}$][3] s.t. $a \in S$ [$\mathbf{in}(a) \in I$, since $I \cong S$] or $b \in S$ [$\mathbf{in}(b) \in I$, since $I \cong S$] and $(b, a) \in R$ [$\mathbf{att}(b, a) \in \hat{F}$]. It satisfies the rule at Line 3, because it contains $\mathbf{out\_of\_range}(a)$ for each $a \in A \setminus S_R^+$, which means $\mathbf{arg}(a) \in \hat{F}$ and $\mathbf{range}(a) \notin I^+$. It satisfies the rule at Line 4, because it contains $\mathbf{unstable}$ if there is at least one $a \in A \setminus S_R^+$, hence $\mathbf{arg}(a) \in \hat{F}$ and $\mathbf{range}(a) \notin I^+$.

Any subset of I^+ cannot satisfy the reduct. If we remove $\mathbf{range}(a)$ for any $a \in S_R^+$, then one of the rules at Lines 1 or 2 is violated. If we remove $\mathbf{out\_of\_range}(a)$ for any $a \in A \setminus S_R^+$, then we have $\mathbf{range}(a) \notin I^+$ and $\mathbf{arg}(a) \in I^+$, violating the rule at Line 3. If we remove $\mathbf{unstable}$ in the case $S_R^+ \subset A$, we have a situation similar to the previous point.

If Direction. Given $I^+ \in \mathcal{AS}(\pi^+(\hat{F}))$, we show $I \in \mathcal{AS}(\pi(\hat{F}))$ with Equation (2). According to Equation (4), $I^+ \in \mathcal{AS}(\pi^+(\hat{F}))$ implies $I^+ \in \mathcal{AS}(I \cup \pi^+)$ for some $I \in \mathcal{AS}(\pi(\hat{F}))$. The answer-set I cannot contain any atom $\mathbf{unstable}$, $\mathbf{range}(a)$ or $\mathbf{out\_of\_range}(a)$. Indeed the predicates $\mathbf{unstable}$, $\mathbf{range}(\cdot)$ and $\mathbf{out\_of\_range}(\cdot)$ do not appear in $\pi(\hat{F})$, by assumption. We show that I^+, in addition to I, must contain all and only the atoms specified in Equation (2):

- I^+ must contain $\mathbf{range}(a)$ for every $a \in S_R^+$. If $a \in S$, then $\mathbf{in}(a) \in I$ ($I \cong S$), then $\mathbf{in}(a) \in I^+$ ($I \subseteq I^+$), then $\mathbf{range}(a) \in I^+$ by the rule at Line 1. If $b \in S$ and $(b,a) \in R$, then $\mathbf{in}(b) \in I$ ($I \cong S$), then $\mathbf{in}(b) \in I^+$ ($I \subseteq I^+$), and $\mathbf{att}(b,a) \in I^+$. Therefore $\mathbf{range}(a) \in I^+$ by the rule at Line 2.

- I^+ must not contain $\mathbf{range}(a)$ for any $a \in A \setminus S_R^+$. In this case $a \notin S$ [$\mathbf{in}(a) \notin I^+$][3] and for any $b \in S$ it holds $(b,a) \notin R$ [$\mathbf{in}(b) \in I^+ \Rightarrow \mathbf{att}(b,a) \notin I^+$]. Hence I^+ does not contain the body of any grounding of the rule at Line 2, and adding any $\mathbf{range}(a)$ would violate the subset-minimality.

- I^+ must contain $\mathbf{out\_of\_range}(a)$ for every $a \in A \setminus S_R^+$. In this case $\mathbf{range}(a) \notin I^+$ (see above), which implies $\{\mathbf{out\_of\_range}(a) \leftarrow \mathbf{arg}(a)\} \subseteq (\pi^+)^{I^+}$, from the rule at Line 3. Since $\mathbf{arg}(a) \in I^+$ for every $a \in A$, $\mathbf{out\_of\_range}(a)$ must be in I^+.

- I^+ must not contain $\mathbf{out\_of\_range}(a)$ for any $a \in S_R^+$. In this case $\mathbf{range}(a) \in I^+$ (see above), and $\{\mathbf{out\_of\_range}(a) \leftarrow \mathbf{arg}(a)\} \not\subseteq (\pi^+)^{I^+}$ for any $a \in A$. Adding $\mathbf{out\_of\_range}(a)$ would violate the subset-minimality.

- I^+ must contain $\mathbf{unstable}$ if $S_R^+ \subset A$. In this case there is some $a \in A \setminus S_R^+$, which implies $\mathbf{out\_of\_range}(a) \in I^+$ (see above), and we have that the body of a grounding of the rule at Line 4, since $\mathbf{arg}(a) \in I^+$ for every $a \in A$.

- I^+ must not contain $\mathbf{unstable}$ if $S_R^+ = A$. In this case $\mathbf{out\_of\_range}(a) \notin I^+$ for any $a \in A$, hence I^+ does not contain the body of any grounding of the rule at Line 4, thus adding $\mathbf{unstable}$ would violate the subset-minimality of the model. □

□

Proposition 6. *For any AF $F = (A,R)$, we have $sem(F) \cong \mathcal{AS}(\pi_{sem^2}(\hat{F}))$.*

Proof. According to Definition 5, we have to prove (i) and (ii). With line numbers we refer here to the ASP encoding shown in Listing 6. Using the *splitting theorem* we can get a characterisation of $\mathcal{AS}(\pi_{sem^2}(\hat{F}))$, in which the sub-programs π_{satsem^2} and $\pi^+_{adm} = \pi_{adm} \cup \pi_{range}$ are considered separately. The splitting set is $C_{sem^2} = \{\mathbf{arg}(\cdot), \mathbf{att}(\cdot,\cdot), \mathbf{in}(\cdot), \mathbf{out}(\cdot), \mathbf{defeated}(\cdot), \mathbf{undefended}(\cdot), \mathbf{range}(\cdot), \mathbf{out\_of\_range}(\cdot), \mathbf{unstable}\}$.

$$\mathcal{AS}(\pi_{sem^2}(\hat{F})) = \bigcup_{J \in \mathcal{AS}(\pi^+_{adm}(\hat{F}))} \mathcal{AS}(J \cup \pi_{satsem^2}) \tag{5}$$

Proof (i). We prove that each semi-stable extension $S \in sem(F)$ has a corresponding answer-set $I \in \mathcal{AS}(\pi_{sem^2}(\hat{F}))$. We know that S is admissible as well, hence by Proposition 4 there exists K s.t. $K \cong S$ and $K \in \mathcal{AS}(\pi_{adm}(\hat{F}))$. Moreover by Equation (5) $I \in \mathcal{AS}(\pi_{sem^2}(\hat{F}))$ if $I \in \mathcal{AS}(J \cup \pi_{satsem^2})$ for some $J \in \mathcal{AS}(\pi^+_{adm}(\hat{F}))$. In the following we distinguish between two complementary cases.

In the case $S \in stb(F)$, i.e. $S_R^+ = A$, we have by Section 2.3 that $J = K \cup \{\mathbf{range}(a) \mid a \in A\}$ is an answer-set of $\pi_{adm}^+(\hat{F})$. This J satisfies $(J \cup \pi_{satsem^2})^J$, since it does not contain $\mathbf{unstable}$, and it is obviously subset-minimal. So we can take $I = J$, which corresponds to S by Section 2.3, since clearly $J \equiv K$.

In the case $S \notin stb(F)$, i.e. $S_R^+ \subset A$, we have by Section 2.3 that the following J is an answer-set of $\pi_{adm}^+(\hat{F})$:

$$J = K \cup \{\mathbf{unstable}\} \cup \{\mathbf{range}(a) \mid a \in S_R^+\} \cup \{\mathbf{out\_of\_range}(a) \mid a \in A \setminus S_R^+\}$$

We define I as follows:

$$I = J \cup \{\mathbf{spoil}\} \cup \{\mathbf{witness}(a) \mid a \in A\} \cup \{\mathbf{larger\_range}(a) \mid a \in A\}$$

The reduct $(J \cup \pi_{satsem^2})^I$ is obtained by just removing the rule at Line 8, because $\mathbf{spoil} \in I$. The interpretation I satisfies the reduct, since it contains J and the head of every rule in $Gr(\pi_{satsem^2})$. Moreover we have $I \cong S$ by Section 2.3, since clearly $I \equiv K$. Now we show the subset-minimality of I by contrapositive, i.e. if an interpretation $L \subset I$ satisfying the reduct $(J \cup \pi_{satsem^2})^I$ exists, then we are able to build a set S' s.t. $S_R^+ \subset S'$ and it has an admissible cover E, which excludes S from the semi-stable extensions by Proposition 2.

$$S' = \{a \mid \mathbf{larger\_range}(a) \in L\} \quad E = \{a \mid \mathbf{witness}(a) \in L\}$$

We notice that $J \subseteq L$ because J appears in the reduct, in particular $\mathbf{unstable} \in L$; and that $\mathbf{spoil} \notin L$, otherwise L should contain $\mathbf{witness}(a)$ and $\mathbf{larger\_range}(a)$ for every $a \in A$ because of the rule at Lines 6 and 7, thus coinciding with I, but they are different by assumption.

We first show that $S_R^+ \subset S'$. There exists $a \in S' \setminus S_R^+$, which can be rephrased as $\mathbf{larger\_range}(a) \in L$ and $\mathbf{out\_of\_range}(a) \in J$ (equivalently $\in L$), by definition of S' and J. The condition is guaranteed by the rule at Line 1, since $\mathbf{unstable} \in L$. We have $S_R^+ \subseteq S'$, which can be expressed as $\mathbf{range}(a) \in J$ (equivalently $\in L$) for every $\mathbf{larger\_range}(a) \in L$, by definition of J and S'. The condition is guaranteed by the rule at Line 2, since $\mathbf{unstable} \in L$.

At last we show the desired properties of E. The set E is conflict-free, otherwise we would have $a, b \in E$ [$\mathbf{witness}(a)$, $\mathbf{witness}(b) \in L$][3] and $(a, b) \in R$ [$\mathbf{att}(a, b) \in \hat{F}$], thus violating the rule at Line 4, since $\mathbf{spoil} \notin L$ and $\mathbf{unstable} \in L$. The set E defends itself (hence admissible), otherwise we would have $a \in E$ [$\mathbf{witness}(a) \in L$] and $c \in A$ [$\mathbf{arg}(c) \in \hat{F}$] s.t. $(c, a) \in R$ [$\mathbf{att}(c, a) \in \hat{F}$], and there does not exist $b \in E$ [$\mathbf{witness}(b) \in L$] s.t. $(b, c) \in R$ [$\mathbf{att}(b, c) \in \hat{F}$], thus violating the rule at Line 5, since $\mathbf{spoil} \notin L$ and $\mathbf{unstable} \in L$. The set E is a cover of S', i.e. $S' \subseteq E_R^+$, otherwise an argument $a \in S'$ [$\mathbf{larger\_range}(a) \in L$] would not be in E [$\mathbf{witness}(a) \notin L$], nor in E^+ [there is no $\mathbf{witness}(b) \in L$ s.t. $\mathbf{att}(b, a) \in L$], thus violating the rule at Line 3, since $\mathbf{unstable} \in L$.

Proof (ii). We prove that each answer-set $I \in \mathcal{AS}(\pi_{stage^2}(\hat{F}))$ corresponds to a semi-stable extension $S \in sem(F)$. Moreover we have the following preliminary results: by Equation (5) $I \in \mathcal{AS}(\pi_{sem^2}(\hat{F}))$ only if $I \in \mathcal{AS}(J \cup \pi_{satsem^2})$ for some $J \in \mathcal{AS}(\pi_{adm}^+(\hat{F}))$; by Section 2.3 $J \in \mathcal{AS}(\pi_{adm}^+(\hat{F}))$ only if $K \in \mathcal{AS}(\pi_{adm}(\hat{F}))$ with $J = K \cup \{\mathbf{range}(a) \mid a \in$

$S_R^+\} \cup \{\mathbf{out\_of\_range}(a) \mid a \in A \setminus S_R^+\} \cup \{\mathbf{unstable} \mid S_R^+ \subset A\}$; by Proposition 4 $K \in \mathcal{AS}(\pi_{adm}(\hat{F}))$ only if $K \cong S$ and $S \in adm(F)$; finally $S \cong I$, because $I \equiv K$ ($J \subseteq I$ and no additional $\mathbf{in}(a)$ is in I, since $\mathbf{in}(\cdot)$ does not appear in the head of any rule of π_{satsem^2}). In the case $\mathbf{unstable} \notin I$, we simply have $S_R^+ = A$ by definition of J, therefore $S \in stb(F)$, in particular $S \in sem(F)$.

In the complementary case $\mathbf{unstable} \in I$, we have that $\mathbf{spoil} \in I$ by the rule at Line 8 hence $\mathbf{witness}(a) \in I$ and $\mathbf{larger\_range}(a) \in I$ for every $a \in A$, by rules at Lines 6 and 7.

$$I \supseteq J \cup \{\mathbf{unstable}, \mathbf{spoil}\} \cup \{\mathbf{witness}(a) \mid a \in A\} \cup \{\mathbf{larger\_range}(a) \mid a \in A\}$$

The reduct $(J \cup \pi_{satsem^2})^I$ is got by just removing the rule at Line 8, since $\mathbf{spoil} \in I$. Now we prove that S is semi-stable by contrapositive, i.e. if $S \notin sem(F)$ then we can find $L \subset I$ satisfying the reduct. We remind that $S \notin sem(F)$ means that there exists $T \in adm(F)$ s.t. $S_R^+ \subset T_R^+$. We use such a T to build the interpretation L:

$$L = J \cup \{\mathbf{unstable}\} \cup \{\mathbf{witness}(a) \mid a \in T\} \cup \{\mathbf{larger\_range}(a) \mid a \in T_R^+\}$$

It is evident $L \subset I$. We show now that L is a model of $(J \cup \pi_{satsem^2})^I$. The rule at Line 1 is satisfied because there is $a \in T_R^+ \setminus S_R^+$, which means $\mathbf{larger\_range}(a) \in L$ and $\mathbf{out\_of\_range}(a) \in J$ (equivalently $\in L$), by definition of L and J. The rule at Line 2 is satisfied, indeed $S_R^+ \subset T_R^+$ means that $\mathbf{range}(a) \in J$ (equivalently $\in L$) only if $\mathbf{larger\_range}(a) \in L$, by definition of L and J. The rule at Line 3 is satisfied, otherwise we would have some $\mathbf{larger\_range}(a) \in L$ $[a \in T_R^+]$[3] s.t. $\mathbf{witness}(a) \notin L$ $[a \notin T]$ and $\{\mathbf{witness}(b), \mathbf{att}(b,a)\} \not\subseteq L$ $[b \notin T$ or $(b,a) \notin R]$ for any $b \in A$, which goes against the definition of range. The rule at Line 4 is satisfied, otherwise we would have $\{\mathbf{witness}(a), \mathbf{witness}(b), \mathbf{att}(a,b)\} \subseteq L$ $[a, b \in T$ and $(a,b) \in R]$, hence T would not be conflict-free. The rule at Line 5 is satisfied, otherwise we would have $\mathbf{witness}(a) \in L$ $[a \in T]$ and $\mathbf{att}(b,a) \in \hat{F}$ $[(b,a) \in R]$ s.t. no $\mathbf{witness}(c) \in L$ $[c \in T]$ and $\mathbf{att}(c,b) \in \hat{F}$ $[(c,b) \in R]$, hence T would not be admissible. Rules at Lines 6 and 7 are satisfied because $\mathbf{spoil} \notin L$. □ □

Proposition 7. *For any AF $F = (A, R)$, we have $stage(F) \cong \mathcal{AS}(\pi_{stage^2}(\hat{F}))$.*

Proof sketch. The sketch is given by comparing the general structure of this proof against the proof of Proposition 6. Using the *splitting theorem* we can get a characterisation of $\mathcal{AS}(\pi_{stage^2}(\hat{F}))$ in which the sub-programs $\pi_{satstage^2}$ and $\pi_{adm}^+ = \pi_{adm} \cup \pi_{range}$ are considered separately. According to Definition 5 we have to prove (i) and (ii).

Proof sketch (i). Similarly to what is done in Proposition 6, given a stage extension S, we build the corresponding answer-set I on top of J as returned by π_{cf}^+. In the case S is also stable, J has a particular shape as guaranteed by Section 2.3, which allows to conclude that J itself is the sought answer-set. In the case S is not stable, the properties of J as stated in Section 2.3, together with the module $\pi_{satstage^2}$, allow to build an interpretation I satisfying the whole encoding, and to show that if I is not subset-minimal we can find a conflict-free cover of a superset of the range of S, making S not stage by Proposition 1. In this last part, the only difference w.r.t. the proof of Proposition 6 is that an interpretation of the reduct does not need to satisfy the rule at Line 5, which encodes defense of a set.

Proof sketch (ii). Similarly to what is done in Proposition 6, given an answer-set I of $\pi_{stage^2}(\hat{F})$ we find the corresponding stage extension S. In particular $I \in \mathcal{AS}(\pi_{stage^2}(\hat{F}))$ implies the existence of a $K \in \mathcal{AS}(\pi_{cf}(\hat{F}))$ corresponding to S conflict-free. By Proposition 1 we know the form of $J \in \mathcal{AS}(\pi^+_{cf}(\hat{F}))$. In the case $\mathbf{unstable} \notin I$ we have immediately that $S^+_R = A$, thus S stable, in particular stage. In the complementary case, thanks to the known shape of J together with the structure of $\pi_{satstage^2}$, we are able to show that if S is not stage, then I is not an answer-set, in particular is not subset-minimal. The main difference w.r.t. the proof of Proposition 6 is that the set T with $S^+_R \subset T^+_R$ is only conflict-free, indeed it does not need to satisfy the rule at Line 5, which encodes defense of a set. □ □

References

[1] Pietro Baroni, Martin W. A. Caminada, and Massimiliano Giacomin. An Introduction to Argumentation Semantics. *The Knowledge Engineering Review*, 26(4):365–410, 2011.

[2] Philippe Besnard and Anthony Hunter. *Elements of Argumentation*. MIT Press, 2008.

[3] Stefano Bistarelli and Francesco Santini. ConArg: A constraint-based computational framework for argumentation systems. In Taghi M. Khoshgoftaar and Xingquan (Hill) Zhu, editors, *Proceedings of the 23rd IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2011)*, pages 605–612. IEEE Computer Society Press, 2011.

[4] Gerhard Brewka, Thomas Eiter, and Mirosław Truszczyński. Answer set programming at a glance. *Communications of the ACM*, 54(12):92–103, 2011.

[5] Martin W. A. Caminada. Comparing Two Unique Extension Semantics for Formal Argumentation: Ideal and Eager. In *Proceedings of the 19th Belgian-Dutch Conference on Artificial Intelligence, BNAIC 2007*, pages 81–87, 2007.

[6] Martin W. A. Caminada and Leila Amgoud. On the Evaluation of Argumentation Formalisms. *Artificial Intelligence*, 171(5-6):286–310, 2007.

[7] Martin W. A. Caminada, Walter A. Carnielli, and Paul E. Dunne. Semi-Stable Semantics. *Journal of Logic and Computation*, 22(5):1207–1254, 2012.

[8] Federico Cerutti, Massimiliano Giacomin, and Mauro Vallati. ArgSemSAT: solving argumentation problems using SAT. In Simon Parsons, Nir Oren, Chris Reed, and Federico Cerutti, editors, *Proceedings of the 5th International Conference on Computational Models of Argument (COMMA 2014)*, volume 266 of *FAIA*, pages 455–456. IOS Press, 2014.

[9] Federico Cerutti, Nir Oren, Hannes Strass, Matthias Thimm, and Mauro Vallati. A benchmark framework for a computational argumentation competition. In Simon Parsons, Nir Oren, Chris Reed, and Federico Cerutti, editors, *Proceedings of the 5th International Conference on Computational Models of Argument (COMMA 2014)*, volume 266 of *FAIA*, pages 459–460. IOS Press, 2014.

[10] Günther Charwat, Wolfgang Dvorák, Sarah Alice Gaggl, Johannes Peter Wallner, and Stefan Woltran. Methods for solving reasoning problems in abstract argumentation - A survey. *Artificial Intelligence*, 220:28–63, 2015.

[11] Yannis Dimopoulos and Alberto Torres. Graph Theoretical Structures in Logic Programs and Default Theories. *Theoretical Computer Science*, 170(1-2):209–244, 1996.

[12] Phan Minh 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.

[13] Phan Minh Dung, Paolo Mancarella, and Francesca Toni. Computing Ideal Sceptical Argumentation. *Artificial Intelligence*, 171(10-15):642–674, 2007.

[14] Paul E. Dunne and Trevor J. M. Bench-Capon. Coherence in finite argument systems. *Artificial Intelligence*, 141(1/2):187–203, 2002.

[15] Paul E. Dunne and Martin W. A. Caminada. Computational Complexity of Semi-Stable Semantics in Abstract Argumentation Frameworks. In Steffen Hölldobler, Carsten Lutz, and Heinrich Wansing, editors, *Proceedings of the 11th European Conference on Logics in Artificial Intelligence (JELIA 2008)*, volume 5293 of *LNCS*, pages 153–165. Springer, 2008.

[16] Paul E. Dunne, Wolfgang Dvořák, Thomas Linsbichler, and Stefan Woltran. Characteristics of multiple viewpoints in abstract argumentation. In Chitta Baral, Giuseppe De Giacomo, and Thomas Eiter, editors, *Proceedings of the 14th International Conference on Principles of Knowledge Representation and Reasoning (KR 2014)*, pages 72–81. AAAI Press, 2014.

[17] Paul E. Dunne and Michael Wooldridge. Complexity of Abstract Argumentation. In Guillermo Simari and Iyad Rahwan, editors, *Argumentation in Artificial Intelligence*, pages 85–104. Springer US, 2009.

[18] Wolfgang Dvořák and Stefan Woltran. Complexity of semi-stable and stage semantics in argumentation frameworks. *Information Processing Letters*, 110(11):425–430, 2010.

[19] Wolfgang Dvořák, Sarah A. Gaggl, Johannes P. Wallner, and Stefan Woltran. Making use of advances in answer-set programming for abstract argumentation systems. In Hans Tompits, Salvador Abreu, Johannes Oetsch, Jörg Pührer, Dietmar Seipel, Masanobu Umeda, and Armin Wolf, editors, *Proceedings of the 19th International Conference on Applications of Declarative Programming and Knowledge Management (INAP 2011), Revised Selected Papers*, volume 7773 of *LNAI*, pages 114–133. Springer, 2013.

[20] Wolfgang Dvořák, Matti Järvisalo, Johannes P. Wallner, and Stefan Woltran. Complexity-sensitive decision procedures for abstract argumentation. *Artificial Intelligence*, 206:53–78, 2014.

[21] Uwe Egly, Sarah A. Gaggl, and Stefan Woltran. Answer-set programming encodings for argumentation frameworks. *Argument & Computation*, 1(2):147–177, 2010.

[22] Sarah A. Gaggl, Norbert Manthey, Alessandro Ronca, Johannes P. Wallner, and Stefan Woltran. Improved Answer-Set Programming Encodings for Abstract Argumentation. Technical Report DBAI-TR-2015-93, Technische Universität Wien, 2015. `http://www.dbai.tuwien.ac.at/research/report/dbai-tr-2015-93.pdf`.

[23] M. Gebser, R. Kaminski, B. Kaufmann, M. Lindauer, M. Ostrowski, J. Romero, T. Schaub, and S. Thiele. *Potassco User Guide*, second edition edition, 2015.

[24] Martin Gebser, Roland Kaminski, and Torsten Schaub. Complex optimization in answer set programming. *Theory and Practice of Logic Programming*, 11(4-5):821–839, 2011.

[25] Michael Gelfond and Vladimir Lifschitz. Classical negation in logic programs and disjunctive databases. *New Generation Computing*, 9(3/4):365–386, 1991.

[26] Vladimir Lifschitz and Hudson Turner. Splitting a logic program. In Pascal Van Hentenryck, editor, *Proceedings of the 11th International Conference on Logic Programming (ICLP 1994)*, pages 23–37. MIT Press, 1994.

[27] Iyad Rahwan and Guillermo. R. Simari, editors. *Argumentation in Artificial Intelligence*. Springer, 2009.

[28] Tommi Syrjänen. *Logic Programs and Cardinality Constraints: Theory and Practice*. PhD thesis, Aalto University, 2009.

[29] Francesca Toni and Marek Sergot. Argumentation and answer set programming. In Marcello Balduccini and Tran C. Son, editors, *Logic Programming, Knowledge Representation, and Nonmonotonic Reasoning: Essays in Honor of Michael Gelfond*, volume 6565 of *LNCS*, pages 164–180. Springer, 2011.

[30] Mauro Vallati, Federico Cerutti, and Massimiliano Giacomin. Argumentation frameworks features: an initial study. In Torsten Schaub, Gerhard Friedrich, and Barry O'Sullivan, editors, *Proceedings of the 21st European Conference on Artificial Intelligence (ECAI 2014)*, volume 263 of *FAIA*, pages 1117–1118. IOS Press, 2014.

[31] Bart Verheij. Two Approaches to Dialectical Argumentation: Admissible Sets and Argumentation Stages. In John-Jules Ch. Meyer and Linda van der Gaag, editors, *Proceedings of the Eighth Dutch Conference on Artificial Intelligence (NAIC'96)*, pages 357–368, 1996.