Online Supplement to Harnessing Incremental Answer Set Solving for Reasoning in Assumption-Based Argumentation

TUOMO LEHTONEN

University of Helsinki, Finland (e-mail: tuomo.lehtonen@helsinki.fi)

JOHANNES P. WALLNER

Graz University of Technology, Austria
(e-mail: wallner@ist.tugraz.at)

MATTI JÄRVISALO

University of Helsinki, Finland (e-mail: matti.jarvisalo@helsinki.fi)

Abstract

This is an online supplement to the article "Harnessing Incremental Answer Set Solving for Reasoning in Assumption-Based Argumentation" published in Theory and Practice of Logic Programming (Lehtonen et al. 2021). This supplement contains further details on algorithmic variants and empirical results not detailed in full in the main article.

1 Algorithms

We provide details on the algorithmic variants not included in the main article.

Algorithm 1 enumerates preferred assumption sets, as a variant of Algorithm 1 in the main article.

Al	gorithm	1 Assum	ption set	enumerati	ion und	er pref	errec	l semantics
----	---------	----------------	-----------	-----------	---------	---------	-------	-------------

Require: ABA framework $F = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{})$

Ensure: return all preferred assumption sets of F

1: $\pi := ABA(F) \cup \pi_{com}$

- 2: while π is satisfiable do
- 3: Let *I* be the found answer set
- 4: $\pi := \pi \cup \{constr(out(I))\}$
- 5: while $\pi \cup in(I)$ is satisfiable do
- 6: Let *I* be the found answer set
- 7: $\pi := \pi \cup \{constr(out(I))\}$
- 8: $E := E \cup \{I\}$

9: **return** *E*

Algorithm 2 details an algorithm for <-admissible assumption set enumeration, as a variant of Algorithm 2 of the main article for deciding credulous acceptance under the same semantics.

Algorithm 2 Assumption set enumeration under <-admissible semantics

Require: ABA⁺ framework $F = (\mathscr{L}, \mathscr{R}, \mathscr{A}, \bar{}, \leq)$ **Ensure:** return all <-admissible assumption sets of F1: $\pi_{cand} := ABA^+(F) \cup \pi_{cf} \cup \pi^+_{undefeated}$ 2: $\pi_{check} := ABA^+(F) \cup \pi^+_{defended} \cup \pi^+_{suspect-defeat}$ 3: **while** π_{cand} is satisfiable **do** 4: Let I be the found answer set 5: **if** $\pi_{check} \cup$ **undefeated** $(I) \cup$ **in**(I) is unsatisfiable **then** $E := E \cup \{I\}$ 6: $\pi_{cand} := \pi_{cand} \cup \{constr(\mathbf{out}(I) \cup \mathbf{in}(I))\}$ 7: **return** E

Algorithm 3 is a variant of Algorithm 3 of the main article which computes a <-complete assumption set (recall that this problem is non-trivial; <-complete assumption sets need not exist for a given framework; if none exist, the algorithm reports "none exist"). Enumerating all <-complete assumptions sets can be achieved via Algorithm 4.

```
Algorithm 3 Finding a <-complete assumption set
```

Require: ABA⁺ framework $F = (\mathscr{L}, \mathscr{R}, \mathscr{A}, \bar{}, \leq)$

Ensure: return a <-complete assumption set of *F* if one exists, Unsatisfiable otherwise

1: $\pi_{cand} := ABA^+(F) \cup \pi_{cf} \cup \pi^+_{undefeated} \cup \pi^+_{prune}$

- 2: $\pi_{checkl} := \text{ABA}^+(F) \cup \pi^+_{defended} \cup \pi^+_{suspect-defeat}$
- 3: $\pi_{check2} := ABA^+(F) \cup \pi^+_{com} \cup \pi^+_{suspect-defeat}$
- 4: while π_{cand} is satisfiable do

5: Let *I* be the found answer set; flag := true

6: **if** $\pi_{checkI} \cup$ **undefeated** $(I) \cup$ **in**(I) unsat. **then**

```
7: for each u \in \mathscr{A} such that undefeated(a) \in I do
```

```
8: if \pi_{check2} \cup undefeated(I) \cup {target(u)} \cup in(I) is
```

```
unsatisfiable then flag := false; break
```

9: **if** flag = true **then return** *I*

```
10: \pi_{cand} := \pi_{cand} \cup \{constr(out(I) \cup in(I))\}
```

```
11: return none exist
```

2

Algorithm 4 Assumption set enumeration under <-complete semantics **Require:** ABA⁺ framework $F = (\mathscr{L}, \mathscr{R}, \mathscr{A}, \bar{}, \leq)$ **Ensure:** return all <-complete assumption sets of *F* 1: $\pi_{cand} := ABA^+(F) \cup \pi_{cf} \cup \pi^+_{undefeated} \cup \pi^+_{prune}$ 2: $\pi_{check1} := \text{ABA}^+(F) \cup \pi^+_{defended} \cup \pi^+_{suspect-defeat}$ 3: $\pi_{check2} := ABA^+(F) \cup \pi^+_{com} \cup \pi^+_{suspect-defeat}$ 4: while π_{cand} is satisfiable **do** Let *I* be the found answer set; *flag* := *true* 5: 6: if $\pi_{check1} \cup$ undefeated $(I) \cup$ in(I) unsat. then for each $u \in \mathscr{A}$ such that **undefeated** $(a) \in I$ do 7: 8: if $\pi_{check2} \cup$ undefeated(I) \cup {target(u)} \cup in(I) is unsatisfiable **then** flag := false; break if flag = true then $E := E \cup \{I\}$ 9: $\pi_{cand} := \pi_{cand} \cup \{constr(\mathbf{out}(I) \cup \mathbf{in}(I))\}$ 10: 11: **return** *E*

2 Experimental Results

Figure 1 shows the number of candidates found for instances which each version solved when considering the task of credulous acceptance under <-complete semantics. Similarly to finding one assumption set, using the stronger refinement for the CEGAR algorithm for <-complete semantics drastically reduces the number of iterations needed to find the solution. This supports the conclusion that reducing the number of candidates is at least a partial reason for the improvement in solving efficiency when using the stronger abstraction for <-complete semantics.

References

LEHTONEN, T., WALLNER, J. P., AND JÄRVISALO, M. 2021. Harnessing incremental answer set solving for reasoning in assumption-based argumentation. *Theory and Practice of Logic Programming*. DOI: 10.1017/S1471068421000296.





Fig. 1: Comparison of number of iterations using the weaker and stronger abstraction under *<-com* on the task of credulous reasoning.