ArgSemSAT-2017

Federico Cerutti[1], Mauro Vallati[2], Massimiliano Giacomin[3], and Tobia Zanetti[3]

[1] Cardiff University,
CeruttiF@cardiff.ac.uk

[2] University of Huddersfield,
m.vallati@hud.ac.uk

[3] University of Brescia,
massimiliano.giacomin@unibs.it
t.zanetti001@studenti.unibs.it

Abstract. In this paper we describe the system ArgSemSAT which includes algorithms that efficiently address several decision and enumeration problems — associated to various semantics — in abstract argumentation. A similar document for the ArgSemSAT version that participated in ICCMA15 is [5].

1 Introduction

Dung's abstract argumentation frameworks provides a fundamental reference in computational argumentation in virtue of its simplicity and ability to capture a variety of more specific approaches as special cases. An abstract argumentation framework (AF) consists of a set of arguments and an *attack* relation between them. The concept of *extension* plays a key role in this simple setting: intuitively, it is a set of arguments which can "survive the conflict together." Different notions of extensions and of the requirements they should satisfy correspond to alternative *argumentation semantics*. The main computational problems in abstract argumentation are related to extensions and can be partitioned into two classes: *decision* problems and *construction* problems.

In this paper we illustrate ArgSemSAT,[4] a collection of algorithms [2–7] for solving enumeration and sceptical–credulous acceptance problems for grounded, complete, preferred, stable, and semi-stable semantics. Differently from [5], we included (1) an efficient algorithm for semi-stable, and several technical improvements, including the use of MiniSAT [9] and AllSAT [10].

2 Background

An argumentation framework [8] consists of a set of arguments and a binary attack relation between them.

[4] https://sourceforge.net/projects/argsemsat/files/ArgSemSAT-2017.tar.bz2/download

Definition 1. *An* argumentation framework *(AF) is a pair $\Gamma = \langle \mathcal{A}, \mathcal{R} \rangle$ where $\mathcal{A}$ is a set of arguments and $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$. We say that $\boldsymbol{b}$* attacks *$\boldsymbol{a}$ iff $\langle \boldsymbol{b}, \boldsymbol{a} \rangle \in \mathcal{R}$, also denoted as $\boldsymbol{b} \to \boldsymbol{a}$. The set of attackers of an argument $\boldsymbol{a}$ will be denoted as $\boldsymbol{a}^- \triangleq \{\boldsymbol{b} : \boldsymbol{b} \to \boldsymbol{a}\}$.*

We also extend these notations to sets of arguments, i.e. given $E, S \subseteq \mathcal{A}$, $E \to \boldsymbol{a}$ iff $\exists \boldsymbol{b} \in E$ s.t. $\boldsymbol{b} \to \boldsymbol{a}$; $\boldsymbol{a} \to E$ iff $\exists \boldsymbol{b} \in E$ s.t. $\boldsymbol{a} \to \boldsymbol{b}$; $E \to S$ iff $\exists \boldsymbol{b} \in E, \boldsymbol{a} \in S$ s.t. $\boldsymbol{b} \to \boldsymbol{a}$; $E^- \triangleq \{\boldsymbol{b} \mid \exists \boldsymbol{a} \in E, \boldsymbol{b} \to \boldsymbol{a}\}$ and $E^+ \triangleq \{\boldsymbol{b} \mid \exists \boldsymbol{a} \in E, \boldsymbol{a} \to \boldsymbol{b}\}$.

The range of a set of arguments $S \subseteq \mathcal{A}$ is $S \cup S^+$.

The basic properties of conflict–freeness, acceptability, and admissibility of a set of arguments are fundamental for the definition of argumentation semantics.

Definition 2. *Given an AF $\Gamma = \langle \mathcal{A}, \mathcal{R} \rangle$:*

- *a set $S \subseteq \mathcal{A}$ is* conflict–free *if $\nexists\, \boldsymbol{a}, \boldsymbol{b} \in S$ s.t. $\boldsymbol{a} \to \boldsymbol{b}$;*
- *an argument $\boldsymbol{a} \in \mathcal{A}$ is* acceptable *with respect to a set $S \subseteq \mathcal{A}$ if $\forall \boldsymbol{b} \in \mathcal{A}$ s.t. $\boldsymbol{b} \to \boldsymbol{a}$, $\exists\, \boldsymbol{c} \in S$ s.t. $\boldsymbol{c} \to \boldsymbol{b}$;*
- *a set $S \subseteq \mathcal{A}$ is* admissible *if S is conflict–free and every element of S is acceptable with respect to S.*

An argumentation semantics σ prescribes for any AF Γ a set of *extensions*, denoted as $\mathcal{E}_\sigma(\Gamma)$, namely a set of sets of arguments satisfying some conditions dictated by σ.

Definition 3. *Given an AF $\Gamma = \langle \mathcal{A}, \mathcal{R} \rangle$:*

- *a set $S \subseteq \mathcal{A}$ is a* complete extension, *i.e. $S \in \mathcal{E}_{\mathcal{CO}}(\Gamma)$, iff S is admissible and $\forall \boldsymbol{a} \in \mathcal{A}$ s.t. $\boldsymbol{a}$ is acceptable w.r.t. S, $\boldsymbol{a} \in S$;*
- *a set $S \subseteq \mathcal{A}$ is a* preferred extension, *i.e. $S \in \mathcal{E}_{\mathsf{PR}}(\Gamma)$, iff S is a maximal (w.r.t. set inclusion) complete set;*
- *a set $S \subseteq \mathcal{A}$ is the* grounded extension, *i.e. $S \in \mathcal{E}_{\mathsf{GR}}(\Gamma)$, iff S is the minimal (w.r.t. set inclusion) complete set;*
- *a set $S \subseteq \mathcal{A}$ is a* stable extension, *i.e. $S \in \mathcal{E}_{\mathsf{ST}}(\Gamma)$, iff S is a complete set where $\forall \boldsymbol{a} \in \mathcal{A} \setminus S, \exists \boldsymbol{b} \in S$ s.t. $\boldsymbol{b} \to \boldsymbol{a}$;*
- *a set $S \subseteq \mathcal{A}$ is a* semi–stable extension *of Γ, i.e. $S \in \mathcal{E}_{\mathsf{SST}}(\Gamma)$, iff S is an admissible set where $S \cup S^+$ (i.e. its range) is maximal (w.r.t. set inclusion).*

Each extension implicitly defines a three-valued *labelling* of arguments (cf. Def. 4). In the light of this correspondence, argumentation semantics can equivalently be defined in terms of labellings rather than of extensions (see [1]). In particular, the notion of *complete labelling* [1] provides an equivalent characterization of complete semantics, in the sense that each complete labelling corresponds to a complete extension and vice versa. Complete labellings can be (redundantly) defined as follows.

Definition 4. *Let $\langle \mathcal{A}, \mathcal{R} \rangle$ be an argumentation framework. A total function $\mathcal{L}ab : \mathcal{A} \mapsto \{\mathtt{in}, \mathtt{out}, \mathtt{undec}\}$ is a* complete labelling *iff it satisfies the following conditions for any $\boldsymbol{a} \in \mathcal{A}$:*

- $\mathcal{L}ab(\boldsymbol{a}) = \mathtt{in} \Leftrightarrow \forall \boldsymbol{b} \in \boldsymbol{a}^- \mathcal{L}ab(\boldsymbol{b}) = \mathtt{out}$;
- $\mathcal{L}ab(\boldsymbol{a}) = \mathtt{out} \Leftrightarrow \exists \boldsymbol{b} \in \boldsymbol{a}^- : \mathcal{L}ab(\boldsymbol{b}) = \mathtt{in}$;
- $\mathcal{L}ab(\boldsymbol{a}) = \mathtt{undec} \Leftrightarrow \forall \boldsymbol{b} \in \boldsymbol{a}^- \mathcal{L}ab(\boldsymbol{b}) \neq \mathtt{in} \wedge \exists \boldsymbol{c} \in \boldsymbol{a}^- : \mathcal{L}ab(\boldsymbol{c}) = \mathtt{undec}$;

Algorithm 1 Enumeration of Preferred Extensions

Input: $\Gamma = \langle \mathcal{A}, \mathcal{R} \rangle$
Output: $E_p \subseteq 2^{\mathcal{A}}$
$E_p := \emptyset$
$cnf := \Pi_\Gamma \wedge \bigvee_{\mathbf{a} \in \mathcal{A}} I_{\phi^{-1}(\mathbf{a})}$
repeat
  $cnfdf := cnf$
  $prefcand := \emptyset$
  repeat
    $aCompl := SATSOLV(cnfdf)$
    if $aCompl \neq \varepsilon$ **then**
      $prefcand := aCompl$
      if $UNDECARGS(aCompl) \neq \emptyset$ **then**
        $cnfdf := cnfdf \wedge \bigwedge_{\mathbf{a} \in INARGS(aCompl)} I_{\phi^{-1}(\mathbf{a})} \wedge \bigwedge_{\mathbf{a} \in OUTARGS(aCompl)} O_{\phi^{-1}(\mathbf{a})} \wedge \bigvee_{\mathbf{a} \in UNDECARGS(aCompl)} I_{\phi^{-1}(\mathbf{a})}$
      end if
      $cnf := cnf \wedge \bigvee_{\mathbf{a} \in \mathcal{A} \setminus INARGS(aCompl)} I_{\phi^{-1}(\mathbf{a})}$
    end if
  until $(aCompl \neq \varepsilon \wedge UNDECARGS(aCompl) \neq \emptyset)$
  if $prefcand \neq \emptyset$ **then**
    $E_p := E_p \cup \{INARGS(prefcand)\}$
    $cnf := cnf \wedge \neg \left(\bigwedge_{\mathbf{a} \in INARGS(prefcand)} I_{\phi^{-1}(\mathbf{a})} \wedge \bigwedge_{\mathbf{a} \in OUTARGS(prefcand)} O_{\phi^{-1}(\mathbf{a})} \wedge \bigwedge_{\mathbf{a} \in UNDECARGS(prefcand)} U_{\phi^{-1}(\mathbf{a})} \right)$
  end if
until $(prefcand \neq \emptyset)$
if $E_p = \emptyset$ **then**
  $E_p = \{\emptyset\}$
end if
return E_p

It is proved [1] that:

- preferred extensions are in one-to-one correspondence with those complete labellings maximising the set of arguments labelled `in`;
- the grounded extension is in in one-to-one correspondence with the complete labelling maximising the set of arguments labelled `undec`;
- stable extensions are in one-to-one correspondence with those complete labellings with no argument labelled `undec`;
- semi-stable extensions are in one-to-one correspondence with those complete labellings that minimise the set of arguments labelled `undec`.

3 `ArgSemSAT`

`ArgSemSAT` is a set of search algorithms in the space of complete extensions to identify also preferred, stable and the grounded extensions (enumeration problems) as well as solving decisions problems associated to those semantics, namely credulous and skeptical acceptance of an argument. `ArgSemSAT` encodes the constraints corresponding to complete labellings of an AF as a SAT problem and then iteratively producing and solving modified versions of the initial SAT problem according to the needs of the search process. `ArgSemSAT` has been implemented in C++, and exploits the MiniSAT solver [9] as well as the AllSAT system presented in [10].

In Alg. 1, Π_{Γ} is a CNF representing the constraints for complete labellings; ϕ^{-1} : $\mathcal{A} \mapsto \mathbb{N}$; I_j (resp. O_j and U_j) is a SAT variable identifying the case that the j-th argument is `in` (resp. `out` an `undec`); $SATSOLV$ is a SAT solver which returns a satisfiable assignment of variables or ε if UNSAT; $INARGS$ (reps. $OUTARGS$ and $UNDECARGS$) is a function that takes as input a variable assignment and returns the set of arguments labelled as `in` (resp. `out` and `undec`) in such assignment.

Acknowledgement

The authors would like to acknowledge the use of the University of Huddersfield Queensgate Grid in carrying out this work.

The authors wish to thank Yinlei Yu, Pramod Subramanyan, Nestan Tsiskaridze, and Sharad Malik, for having shared their AllSAT implementation [10].

References

1. Caminada, M., Gabbay, D.M.: A logical account of formal argumentation. Studia Logica (Special issue: new ideas in argumentation theory) 93(2–3), 109–145 (2009)
2. Cerutti, F., Dunne, P.E., Giacomin, M., Vallati, M.: A SAT-based Approach for Computing Extensions in Abstract Argumentation. In: Second International Workshop on Theory and Applications of Formal Argumentation (TAFA-13) (2013)
3. Cerutti, F., Dunne, P.E., Giacomin, M., Vallati, M.: Computing Preferred Extensions in Abstract Argumentation: A SAT-Based Approach. In: Black, E., Modgil, S., Oren, N. (eds.) TAFA 2013, Lecture Notes in Computer Science, vol. 8306, pp. 176–193. Springer-Verlag Berlin Heidelberg (2014)
4. Cerutti, F., Giacomin, M., Vallati, M.: ArgSemSAT: Solving Argumentation Problems Using SAT. In: Parsons, S., Oren, N., Reed, C., Cerutti, F. (eds.) 5th Conference on Computational Models of Argument. pp. 455—-456 (2014)
5. Cerutti, F., Vallati, M., Giacomin, M.: ArgSemSAT-1.0: Exploiting SAT Solvers in Abstract Argumentation. System Descriptions of the First International Competition on Computational Models of Argumentation (ICCMA'15) pp. 4–7 (2015)
6. Cerutti, F., Vallati, M., Giacomin, M.: jArgSemSAT: an efficient off-the-shelf solver for abstract argumentation frameworks (2016)
7. Cerutti, F., Vallati, M., Giacomin, M.: An Efficient Java-Based Solver for Abstract Argumentation Frameworks: jArgSemSAT. International Journal of Artificial Intelligence Tools 26(2), in press (2017)
8. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming, and n-person games. Artificial Intelligence 77(2), 321–357 (1995)
9. Eén, N., Sörensson, N.: An Extensible SAT-solver, pp. 502–518. Springer Berlin Heidelberg, Berlin, Heidelberg (2004)
10. Yu, Y., Subramanyan, P., Tsiskaridze, N., Malik, S.: All-SAT Using Minimal Blocking Clauses. In: 2014 27th International Conference on VLSI Design and 2014 13th International Conference on Embedded Systems. pp. 86–91. IEEE (2014)