The Pyglaf Argumentation Reasoner*

Mario Alviano

Department of Mathematics and Computer Science, University of Calabria, Italy
`alviano@mat.unical.it`

Abstract

The PYGLAF reasoner takes advantage of circumscription to solve computational problems of abstract argumentation frameworks. In fact, many of these problems are reduced to circumscription by means of linear encodings, and a few others are solved by means of a sequence of calls to an oracle for circumscription. Within PYGLAF, Python is used to build the encodings and to control the execution of the external circumscription solver, which extends the SAT solver GLUCOSE and implements an algorithm based on unsatisfiable core analysis.

1998 ACM Subject Classification I.2.4 Knowledge Representation Formalisms and Methods

Keywords and phrases abstract argumentation frameworks, propositional circumscription, minimal model enumeration, incremental solving

Digital Object Identifier 10.4230/OASIcs.ICLP.2017.2

1 Introduction

Circumscription [9] is a nonmonotonic logic formalizing common sense reasoning by means of a second order semantics, which essentially enforces to minimize the extension of some predicates. With a little abuse on the definition of circumscription, the minimization can be imposed on a set of literals, so that a set of negative literals can be used to encode a maximization objective function. Since many semantics of abstract argumentation frameworks are based on a preference relation that essentially amount to inclusion relationships, PYGLAF (`http://alviano.com/software/pyglaf/`) uses circumscription as a target language to solve computational problems of abstract argumentation frameworks.

PYGLAF is implemented in Python and uses CIRCUMSCRIPTINO [1], a circumscription solver extending the SAT solver GLUCOSE [7] with the unsatisfiable core based algorithm ONE [6] enhanced by *reiterated progression shrinking* [3], native support for cardinality constraints as in WASP [4, 5, 8], and polyspace model enumeration [2]. Linear reductions are used for all considered semantics. The communication between PYGLAF and CIRCUMSCRIPTINO is handled in the simplest possible way, that is, via stream processing. In fact, the communication is limited to a single invocation of the circumscription solver.

2 From Argumentation Frameworks to Circumscription

Let $\mathcal{A}$ be a fixed, countable set of *atoms* including $\bot$. A *literal* is an atom possibly preceded by the connective $\neg$. For a literal ℓ, let $\overline{\ell}$ denote its *complementary literal*, that is, $\overline{p} = \neg p$ and $\overline{\neg p} = p$ for all $p \in \mathcal{A}$; for a set L of literals, let $\overline{L}$ be $\{\overline{\ell} \mid \ell \in L\}$. *Formulas* are defined as

* The paper has been partially supported by the Italian Ministry for Economic Development (MISE) under project "PIUCultura – Paradigmi Innovativi per l'Utilizzo della Cultura" (n. F/020016/01-02/X27), and under project "Smarter Solutions in the Big Data World (S2BDW)" (n. F/050389/01-03/X32) funded within the call "HORIZON2020" PON I&C 2014-2020, and by Gruppo Nazionale per il Calcolo Scientifico (GNCS-INdAM).

usual by combining atoms and the connectives $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$. A *theory* is a set T of formulas including $\neg\bot$; the set of atoms occurring in T is denoted by $atoms(T)$. An *assignment* is a set A of literals such that $A \cap \overline{A} = \emptyset$. An *interpretation* for a theory T is an assignment I such that $(I \cup \overline{I}) \cap \mathcal{A} = atoms(T)$. Relation $\models$ is defined as usual. I is a *model* of a theory T if $I \models T$. Let $models(T)$ denote the set of models of T.

Circumscription applies to a theory T and a set P of literals subject to minimization. Formally, relation $\leq^P$ is defined as follows: for I, J interpretations of T, $I \leq^P J$ if $I \cap P \subseteq J \cap P$. $I \in models(T)$ is a *preferred model* of T with respect to $\leq^P$ if there is no $J \in models(T)$ such that $I \not\leq^P J$ and $J \leq^P I$. Let $CIRC(T, P)$ denote the set of preferred models of T with respect to $\leq^P$.

An *abstract argumentation framework* (AF) is a directed graph G whose nodes $arg(G)$ are arguments, and whose arcs $att(G)$ represent an attack relation. An *extension* E is a set of arguments. The *range* of E in G is $E_G^+ := E \cup \{x \mid \exists yx \in att(G) \text{ with } y \in E\}$. In the following, several AF semantics are characterized by means of circumscription.

For each argument x, an atom a_x is possibly introduced to represent that x is attacked by some argument that belongs to the computed extension E, and an atom r_x is possibly introduced to enforce that x belongs to the range E_G^+:

$$attacked(G) := \left\{ a_x \leftrightarrow \bigvee_{yx \in att(G)} y \;\middle|\; x \in arg(G) \right\} \tag{1}$$

$$range(G) := \left\{ r_x \rightarrow x \vee \bigvee_{yx \in att(G)} y \;\middle|\; x \in arg(G) \right\} \tag{2}$$

The following set of formulas characterize semantics not based on preferences:

$$conflict\text{-}free(G) := \{\neg\bot\} \cup \{\neg x \vee \neg y \mid xy \in att(G)\} \tag{3}$$

$$admissible(G) := conflict\text{-}free(G) \cup attacked(G) \cup \{x \rightarrow a_y \mid yx \in att(G)\} \tag{4}$$

$$complete(G) := admissible(G) \cup \left\{ \left(\bigwedge_{yx \in att(G)} a_y \right) \rightarrow x \;\middle|\; x \in arg(G) \right\} \tag{5}$$

$$stable(G) := complete(G) \cup range(G) \cup \{r_x \mid x \in arg(G)\} \tag{6}$$

Note that in (4) truth of an argument x implies that all arguments attacking x are actually attacked by some true argument. In (5), instead, whenever all attackers of an argument x are attacked by some true argument, argument x is forced to be true. Finally, in (6) all atoms of the form r_x are forced to be true, so that the range of the computed extension has to cover all arguments.

Below are several AF semantics with natural characterization in circumscription:

$$co(G) := CIRC(complete(G), \emptyset) \tag{7}$$

$$st(G) := CIRC(stable(G), \emptyset) \tag{8}$$

$$gr(G) := CIRC(complete(G), arg(G)) \tag{9}$$

$$pr(G) := CIRC(complete(G), \overline{arg(G)}) \tag{10}$$

$$sst(G) := CIRC(complete(G) \cup range(G), \{\neg r_x \mid x \in arg(G)\}) \tag{11}$$

$$stg(G) := CIRC(conflict\text{-}free(G) \cup range(G), \{\neg r_x \mid x \in arg(G)\}) \tag{12}$$

All of the above semantics are supported by PYGLAF, which provides a uniform developing platform for reasoning on argumentation frameworks.

References

1 Mario Alviano. Model enumeration in propositional circumscription via unsatisfiable core analysis. *TPLP*, 17, 2017. To appear. URL: `https://arxiv.org/abs/1707.01423`.

2 Mario Alviano and Carmine Dodaro. Answer set enumeration via assumption literals. In Giovanni Adorni, Stefano Cagnoni, Marco Gori, and Marco Maratea, editors, *AI*IA 2016: Advances in Artificial Intelligence - XVth International Conference of the Italian Association for Artificial Intelligence, Genova, Italy, November 29 - December 1, 2016, Proceedings*, volume 10037 of *Lecture Notes in Computer Science*, pages 149–163. Springer, 2016. `doi:10.1007/978-3-319-49130-1_12`.

3 Mario Alviano and Carmine Dodaro. Anytime answer set optimization via unsatisfiable core shrinking. *TPLP*, 16(5-6):533–551, 2016. `doi:10.1017/S147106841600020X`.

4 Mario Alviano, Carmine Dodaro, Wolfgang Faber, Nicola Leone, and Francesco Ricca. WASP: A native ASP solver based on constraint learning. In Pedro Cabalar and Tran Cao Son, editors, *Logic Programming and Nonmonotonic Reasoning, 12th International Conference, LPNMR 2013, Corunna, Spain, September 15-19, 2013. Proceedings*, volume 8148 of *Lecture Notes in Computer Science*, pages 54–66. Springer, 2013. `doi:10.1007/978-3-642-40564-8_6`.

5 Mario Alviano, Carmine Dodaro, Nicola Leone, and Francesco Ricca. Advances in WASP. In Francesco Calimeri, Giovambattista Ianni, and Miroslaw Truszczynski, editors, *Logic Programming and Nonmonotonic Reasoning - 13th International Conference, LPNMR 2015, Lexington, KY, USA, September 27-30, 2015. Proceedings*, volume 9345 of *Lecture Notes in Computer Science*, pages 40–54. Springer, 2015. `doi:10.1007/978-3-319-23264-5_5`.

6 Mario Alviano, Carmine Dodaro, and Francesco Ricca. A maxsat algorithm using cardinality constraints of bounded size. In Qiang Yang and Michael Wooldridge, editors, *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 2677–2683. AAAI Press, 2015. URL: `http://ijcai.org/Abstract/15/379`.

7 Gilles Audemard and Laurent Simon. Predicting learnt clauses quality in modern SAT solvers. In Craig Boutilier, editor, *IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, California, USA, July 11-17, 2009*, pages 399–404, 2009. URL: `http://ijcai.org/Proceedings/09/Papers/074.pdf`.

8 Carmine Dodaro, Mario Alviano, Wolfgang Faber, Nicola Leone, Francesco Ricca, and Marco Sirianni. The birth of a WASP: preliminary report on a new ASP solver. In Fabio Fioravanti, editor, *Proceedings of the 26th Italian Conference on Computational Logic, Pescara, Italy, August 31 - September 2, 2011*, volume 810 of *CEUR Workshop Proceedings*, pages 99–113. CEUR-WS.org, 2011. URL: `http://ceur-ws.org/Vol-810/paper-l06.pdf`.

9 John McCarthy. Circumscription - A form of non-monotonic reasoning. *Artif. Intell.*, 13(1-2):27–39, 1980. `doi:10.1016/0004-3702(80)90011-9`.