CEGARTIX v0.4: A SAT-Based Counter-Example Guided Argumentation Reasoning Tool

Wolfgang Dvořák[1], Matti Järvisalo[2], Johannes Peter Wallner[2], and Stefan Woltran[3]

[1] University of Vienna, Faculty of Computer Science
[2] University of Helsinki, Department of Computer Science, HIIT
[3] Vienna University of Technology, Institute of Information Systems

Abstract. We present CEGARTIX in version 0.4 for the International Competition on Computational Models of Argumentation (ICCMA) 2015. We describe the main parts of the software architecture, the main ideas behind the algorithm, ICCMA 2015 specific adaptations, and how to obtain CEGARTIX.

1 System Architecture

CEGARTIX (Counter-Example Guided Argumentation Reasoning Tool) [2] traverses the search space of a so-called base semantics of a given argumentation framework (AF) [1] to find preferred, semi-stable, or stage extensions. For preferred and semi-stable semantics the base semantics can either be complete or admissible semantics, while for stage semantics the base semantics comprises of the conflict-free sets. Each "step" in the traversal is delegated to a state-of-the-art complete Boolean satisfiability (SAT) solver. The main components of CEGARTIX's system architecture are shown in Fig. 1. The system takes as input an AF in the "ASPARTIX format" [4], i.e., a list of facts in the answer-set programming (ASP) language. The main procedure drives the overall algorithm and (i) translates the given AF and reasoning task to a Boolean formula, (ii) modifies the formula or generates new formulas based on previously found extensions of the base semantics, and (iii) calls the specified SAT solver with the current formula and reads the returned model if it exists. CEGARTIX incorporates miniSAT (v2.2.0) [3] and clasp (v2.0.5) [5] (utilizing the clasp library and interface), and is able to write the formula to an external SAT solver that adheres to standard input and output of such solvers according to the SAT solver competitions [6].

2 Algorithm

Algorithm 1 shows the underlying procedure ("main procedure" in Fig. 1) for skeptical acceptance under preferred semantics, which is prototypical also for the

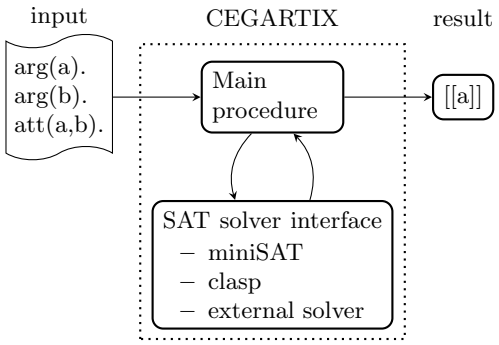

Fig. 1. CEGARTIX system architecture

Algorithm 1 $Skept_{prf}(F, \alpha)$

Require: AF $F = (A, R)$, $\alpha \in A$, $\sigma \in \{adm, com\}$
Ensure: *accept* iff α is skeptically accepted in F w.r.t. *prf*
1: $\mathcal{E} \leftarrow \emptyset$
2: **if** SAT-SOLVER($\exists E \in \sigma(F) : E$ attacking α) **then** *reject*
3: **while** $E \leftarrow$ SAT-SOLVER $(\exists E \in \sigma(F) : \alpha \notin E, (E \setminus E') \neq \emptyset \; \forall E' \in \mathcal{E})$ **do**
4: **while** $E' \leftarrow$ SAT-SOLVER $(\exists E' \in \sigma(F) : E \subset E')$ **do**
5: $E \leftarrow E'$
6: **if** $\alpha \notin E$ **then** *reject* **else** $\mathcal{E} \leftarrow \mathcal{E} \cup \{E\}$
7: *accept*

other variants. By σ we denote the base semantics, i.e. the semantics we traverse for finding a preferred extension not containing the queried argument α. After a shortcut for checking whether a σ-extension exists which attacks α, we begin with the main algorithm which consists of two while loops. We first generate a σ extension not contained in already visited extensions ($\mathcal{E}$). Then this σ extension is iteratively extended to a preferred extension in the inner while loop. If the inner while loop terminates, then we have found a preferred extension which is checked if it is a counterexample for skeptical acceptance of α. Termination of the outer while loop signifies that we have exhausted the search space. More details can be found in [2], including the precise Boolean formulas used. For finding all preferred extensions, we simply omit the queried argument from consideration.

2.1 Supported Reasoning Tasks

CEGARTIX v0.4 supports the following reasoning tasks:

- Credulous acceptance under semi-stable, and stage semantics,
- Skeptical acceptance under preferred, semi-stable, and stage semantics,
- returning an arbitrary preferred extension, and
- enumerating all preferred extensions.

3 Competition Specific Settings

For ICCMA 2015, we have adapted CEGARTIX as follows. First, we extended the reasoning tasks by enumeration of all preferred extensions, and finding a single preferred extension. This modification is straightforwardly obtained by essentially omitting the queried argument from the main Algorithm 1. Furthermore, we fixed the base semantics (σ in Algorithm 1) to complete semantics. In earlier tests, complete semantics outperformed admissible semantics on some test instances [2]. We pre-set the SAT-solver to be clasp for the competition.

4 Web Access and License

CEGARTIX v0.4 is available on the web under `http://www.dbai.tuwien.ac.at/research/project/argumentation/cegartix/`. Since the software incorporates clasp, the whole software is licensed under GNU public license v2. We accompany the package with a readme file that explains usage. Overall, this version of CEGARTIX adheres to the ICCMA 2015 input and output specification; the shell script has to be called, which in turn calls CEGARTIX.

Acknowledgements

This work has been funded by the Austrian Science Fund (FWF) through projects I1102 and Y698, and by Academy of Finland through grants 251170 COIN, 276412, and 284591.

References

1. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artif. Intell. 77(2), 321–358 (1995)
2. Dvořák, W., Järvisalo, M., Wallner, J.P., Woltran, S.: Complexity-Sensitive Decision Procedures for Abstract Argumentation. Artif. Intell. 206, 53–78 (2014)
3. Eén, N., Sörensson, N.: An extensible SAT-solver. In: Proc. SAT 2003. LNCS, vol. 2919, pp. 502–518. Springer (2004)
4. Egly, U., Gaggl, S.A., Woltran, S.: Answer-set programming encodings for argumentation frameworks. Argument and Computation 1(2), 147–177 (2010)
5. Gebser, M., Kaufmann, B., Schaub, T.: Conflict-driven answer set solving: From theory to practice. Artificial Intelligence 187-188, 52–89 (2012)
6. Järvisalo, M., Berre, D.L., Roussel, O., Simon, L.: The International SAT Solver Competitions. AI Magazine 33(1), 89–94 (2012)