

CEGARTIX v2017-3-13: A SAT-Based Counter-Example Guided Argumentation Reasoning Tool

Wolfgang Dvořák¹, Matti Järvisalo², and Johannes P. Wallner¹

¹ TU Wien, Institute of Information Systems

² University of Helsinki, Department of Computer Science, HIIT

Abstract. We present CEGARTIX, version 2017-3-13, for the International Competition on Computational Models of Argumentation (IC-CMA) 2017. Main changes are the addition of support for more reasoning tasks, relying on variations of existing Boolean encodings. The tool CEGARTIX is now capable of deciding credulous and skeptical acceptance, and enumeration of all (or up to a pre-specified bound) extensions under the admissible, complete, preferred, stable, semi-stable, stage, and ideal semantics. For the NP-complete semantics, i.e., the admissible, complete, and stable semantics, we compute the reasoning tasks via a single SAT call (resp. enumerate all satisfying assignments of a formula). For ideal semantics, we implemented an existing algorithm based on computing the set of credulously accepted arguments under admissible semantics. The grounded extension is computed via a direct, and polynomial-time, algorithm. Reasoning tasks for the remaining semantics are solved, as in previous versions of CEGARTIX, via an (unchanged) counter-example guided abstraction refinement (CEGAR) algorithm, with the exception of enumeration of all (some) extensions under semi-stable and stage semantics, for which we adapted the corresponding variant under preferred semantics. We use miniSAT (v2.2.0), in an incremental mode, as the underlying SAT solver.

1 System Architecture and Algorithms

CEGARTIX (Counter-Example Guided Argumentation Reasoning Tool) [4, 5] is based on traversing the search space of the underlying reasoning tasks of a given argumentation framework (AF) [2] via a SAT solver.

In this version of CEGARTIX, a first component of the tool decides, based on the parameters, which algorithm has to be applied. We distinguish between algorithms for the grounded semantics, the ideal semantics, the NP-complete semantics (admissible, complete, and stable), and the semantics with complexity beyond NP (preferred, semi-stable, and stage), as well as the Dung’s triathlon. We give more details on each main component in the following, and illustrate the workflow in Figure 1.

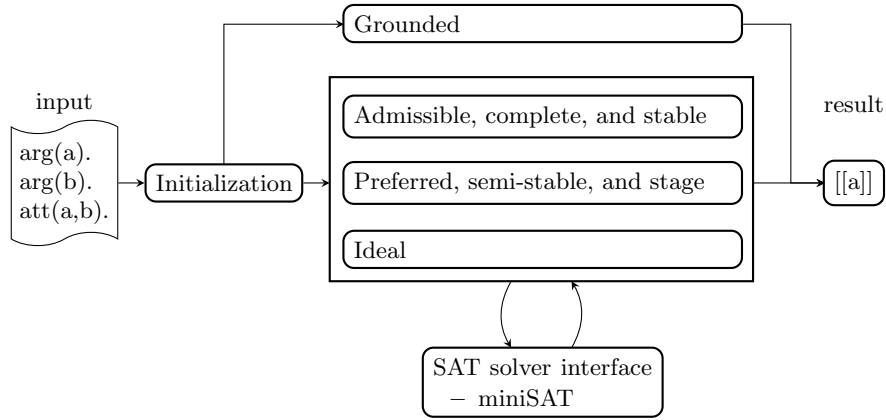


Fig. 1. CEGARTIX version 2017-3-13 system architecture

Grounded semantics. For this semantics, we apply a straightforward polynomial time algorithm to compute (part of) the grounded extension. During parsing, we assign each argument a positive integer equal to the number of attackers of that argument. This value indicates the number of non-counterattacked attackers of an argument. This integer may decrease during runtime. Further, for each argument we store a flag, which indicates whether this argument is attacked by the grounded extension. For each argument a with 0 non-counterattacked attackers we (i) add a to the output (partial grounded extension) and (ii) decrease the integers of the arguments where a contributes as a (new) defender. More concretely, if a is added to the grounded extension, we mark each argument b that is attacked by a . If such a b was not marked before, then we decrease the integer of each argument c attacked by b (a defends c against the attack from b). For instance, for the simple AF with three arguments, a , b , and c , and an attack from a to b and from b to c , we assign 0 to a , 1 to b , and 1 to c . After adding a to the output, we mark b and decrease the value of c to 0, since the newly added argument a defends against the attack (b, c). If an argument's value decreases to 0 during the process, we again apply steps (i) and (ii). For credulous or skeptical reasoning, we terminate already when the queried argument is added the output.

Ideal semantics. We implemented the algorithm of [3] for computing the ideal extension. This algorithm has two main components: (i) constructing the set of credulously accepted arguments under admissible semantics, and (ii) applying the so-called restricted characteristic function to iteratively remove undefended arguments. For (i), we query a SAT solver to compute an admissible set containing at least one argument from the set S_0 , initialized with the set of all arguments in the given framework. If the solver reports unsatisfiability, we terminate. Otherwise, we extract $C_0 \subseteq S_0$ which corresponds to the set of arguments contained in an admissible set. We then query the SAT solver again, this time asking for an admissible set containing at least one argument from the set $S_1 = S_0 \setminus C_0$.

Applying this procedure iteratively yields the set of all credulously accepted arguments under admissible semantics. For (ii), we apply a simple polynomial time algorithm: given a set of arguments and an AF, we construct the subset of the set of arguments that is defended by the given set.

Admissible, complete, and stable semantics. For these semantics we query a SAT solver, once for credulous or skeptical reasoning, and several times for enumeration. The encodings of the semantics are variants of the ones we utilize for preferred, semi-stable, and stage semantics from previous versions of CEGARTIX (see [7] for the system descriptions of the solvers submitted to ICCMA'15), which, themselves, are variants of the encodings defined by [1].

Preferred, semi-stable, and stage semantics. Reasoning tasks for these semantics remain unchanged as for previous version of CEGARTIX, with the only exception of enumeration of extensions under semi-stable and stage semantics, which was not supported before. For these tasks, we slightly adapt the algorithm of enumeration of all preferred extensions. That is, we iteratively construct larger sets of arguments w.r.t. the range of the sets that are admissible (conflict-free). When reaching a maximal element, we enumerate all admissible (conflict-free) sets of exactly that range (which are semi-stable or, resp., stage extensions). Afterwards we block the range and query the SAT solver again for a new admissible (conflict-free) set.

Dung's triathlon. This task requires enumeration of the grounded extension, all stable extensions, and all preferred extensions. For the triathlon, we apply the algorithms mentioned before with the sole exception that we do not re-compute stable extensions for the task of enumeration of preferred extensions. That is, we copy the computed stable extensions as a partial set of preferred extensions, and start enumeration of preferred with blocking the already found stable extensions (which are preferred extensions by definition).

1.1 Supported Reasoning Tasks

CEGARTIX v2017-3-13 supports the following reasoning tasks under admissible, complete, preferred, stable, semi-stable, stage, and ideal semantics:

- credulous acceptance,
- skeptical acceptance, and
- enumerating all (at most k) extensions.

For k the user may provide an integer, defining that k extensions of the chosen semantics shall be returned. If fewer exist for the given framework, then all will be returned. If $k = 0$, then all extensions are computed.

2 Competition Specific Settings

For this competition, we set the internal SAT solver to be miniSAT (v2.2.0) [6]. We use an incremental mode of miniSAT, whenever possible.

3 Web Access and License

The newest versions of CEGARTIX are available on the web under

<http://www.dbai.tuwien.ac.at/proj/argumentation/cegartix/>.

This tool is licensed under GNU public license v2. The tool CEGARTIX is based on miniSAT (v2.2.0) and several components of the boost library (v1.48). The license files for using miniSAT and boost are added, as specified by the licenses, to the archive containing the sources or binary. Nevertheless, CEGARTIX as a whole is licensed under GNU public license v2.

Acknowledgements

This work has been funded by the Austrian Science Fund (FWF) through projects I2854 and P30168-N31, and by Academy of Finland through grants 251170 COIN, 276412, and 284591.

References

1. Besnard, P., Doutre, S.: Checking the acceptability of a set of arguments. In: Delgrande, J.P., Schaub, T. (eds.) Proceedings of the Tenth International Workshop on Non-Monotonic Reasoning, NMR 2004. pp. 59–64 (2004)
2. Dung, P.M.: On the acceptability of arguments and its fundamental role in non-monotonic reasoning, logic programming and n-person games. *Artificial Intelligence* 77(2), 321–358 (1995)
3. Dunne, P.E.: The computational complexity of ideal semantics. *Artificial Intelligence* 173(18), 1559–1591 (2009)
4. Dvořák, W., Jarvisalo, M., Wallner, J.P., Woltran, S.: Complexity-Sensitive Decision Procedures for Abstract Argumentation. *Artificial Intelligence* 206, 53–78 (2014)
5. Dvořák, W., Jarvisalo, M., Wallner, J.P., Woltran, S.: Complexity-sensitive decision procedures for abstract argumentation (extended abstract). In: Yang, Q., Wooldridge, M. (eds.) Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015. pp. 4173–4177. AAAI Press / International Joint Conferences on Artificial Intelligence (2015)
6. Eén, N., Sörensson, N.: An extensible SAT-solver. In: Giunchiglia, E., Tacchella, A. (eds.) Proceedings of the Sixth International Conference on Theory and Applications of Satisfiability Testing, SAT 2003. LNCS, vol. 2919, pp. 502–518. Springer (2004)
7. Thimm, M., Villata, S.: System descriptions of the first international competition on computational models of argumentation (ICCMA’15). CoRR abs/1510.05373 (2015), <http://arxiv.org/abs/1510.05373>