2-2007

Cmodels: SAT-based Answer Set Programming System

Yuliya Lierler
University of Nebraska at Omaha, ylierler@unomaha.edu

Marco Maratea
Universtia di Genova

Cmodels: SAT-based Answer Set Programming System

Yuliya Lierler and Marco Maratea

Department of Computer Sciences, University of Texas at Austin, USA; and
DIST, University of Genova, Italy

CMODELS [1, 2] is an answer set programming [3] system that uses the same front-end LPARSE as answer set solver SMODELS (http://www.tcs.hut.fi/Software/smodels/). CMODELS main computational characteristics is that it computes answer sets using a SAT solver for search.

The use of SAT solvers for generating answer sets is based on the fact that for logic programs satisfying syntactic condition, *tightness*, the answer set semantics is equivalent to the Clark's completion semantics. In addition, [4] introduced concept of a loop formula, and demonstrated that the answer sets of a logic program are exactly the models of its completion that satisfy the loop formulas of all loops. These findings allowed to extend the applicability of SAT-based answer set solving to the case of nontight programs.

We can outline the basic execution steps of system CMODELS by the following. The system: (1) produces program's completion; (2) computes model of a completion using a SAT solver; (3) verifies if the model is indeed an answer set, if so returns a model, otherwise goes back to Step 2. The idea is thus to use a SAT solver for generating model candidates and then check if they are indeed the answer sets of a program. The way Step 3 is implemented depends on the class of a logic program. If a program is disjunctive (a head of some rule in the program contains a disjunction) another instance of a SAT solver is called for model verification as introduced in [5]; otherwise, it is sufficient to adopt the linear time Dowling-Gallier procedure.

One of the benefits of such architecture, in comparison to native procedures working directly on a logic program, like SMODELS and DLV (http://www.dlvsystem.com), is that SAT-based answer set programming systems can take advantage of the latest and continuous developments in the area of SAT. For instance, CMODELS supports the interface to the three state-of-the-art SAT solvers

- RELSAT (http://code.google.com/p/relsat/),
- SIMO (http://www.star.dist.unige.it/ sim/simo/), and
- ZCHAFF (http://www.princeton.edu/ chaff/zchaff.html).

On the other hand, in comparison with SAT-based system ASSAT (http://assat.cs.ust.hk/), CMODELS main advantages are that it (i) deals with programs that may contain disjunctive, choice, cardinality and weight constraint rules, and (ii) computes all solutions of the program.

CMODELS supports the input produced by the grounder LPARSE (see SMODELS link) that takes into account disjunctive, choice, cardinality and weight constraint rules that enhance compact encoding of the problems. In order to invoke CMODELS on program *P*, the following command line may be used:

```
lparse --dlp-choice P | cmodels .
```

where the directive "`--dlp-choice`" is to be used if P is a disjunctive program. The system also allows various flags to specify the details of its computation: considering they are continuously updated, refer to the system home page provided in the next paragraph.

Though CMODELS is a relatively young system, it has been already used in various application domains such as wire-routing, reconstruction of phylogenies, formal verification of abstract state machines and planning. We see main goal of CMODELS as a tool that has sufficient computational power in order to make answer set programming paradigm be widely used in applications.

Download and contact info CMODELS (source code) is publicly available at `http://www.cs.utexas.edu/users/tag/cmodels/`, and is continuously under development. The authors can be contacted by email at yuliya@cs.utexas.edu, and marco@dist.unige.it.

Acknowledgments We are grateful to Vladimir Lifschitz and Enrico Giunchiglia for their continuous support during the whole period of the development of CMODELS. This research has been partially supported by Texas Higher Education Coordinating Board under Grant 003658-0322-2001 and MIUR (Italian Ministry of Education, University and Research).

References

1. Giunchiglia, E., Lierler, Y., Maratea, M.: Answer set programming based on propositional satisfiability. Journal of Automated Reasoning (2007) `http://www.springerlink.com/content/q2826087v8172548/`.
2. Lierler, Y.: Cmodels: SAT-based disjunctive answer set solver. In: Proceedings of International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR). (2005) 447–452
3. Ferraris, P., Lifschitz, V.: Mathematical foundations of answer set programming. In: We Will Show Them! Essays in Honour of Dov Gabbay. King's College Publications (2005) 615–664
4. Lin, F., Zhao, Y.: ASSAT: Computing answer sets of a logic program by SAT solvers. In: Proceedings of National Conference on Artificial Intelligence (AAAI). (2002) 112–117
5. Koch, C., Leone, N., Pfeifer, G.: Using SAT checkers for disjunctive logic programming systems. Artificial Intelligence **151** (2003) 177–212