Efficient Decision Procedures for the Integration of Planning and Formal Verification in Advanced Systems

Marco Maratea *
Via F. Causa 13, 16145, Genova, Italy
phone +39-010-3532811; fax: +39-010-3532948
E-mail: marco@dist.unige.it

The autonomy and safety of critical systems is a crucial task that can be addressed via "Safe Planning". Safe Planning is the task of generating/validating plans that not only achieve the goal, but verify also a set of user-defined properties. A promising approach for Safe Planning is the result of the integration between planning and formal verification techniques and relies on a compilation into a propositional formula.

The contribution of this thesis in the area of Safe Planning is in particular in the design and in the implementation of specialized back-end solvers for deciding theories resulting from alternative approaches, that extend the one based on propositional satisfiability, defined in this thesis as well.

Keywords: propositional satisfiability (SAT), planning, formal verification

1. Introduction

The increasing complexity of the services requested to robotic devices results in a need for more and more sophisticated and autonomous systems. Planning is a research area in Artificial Intelligence aiming at the construction of systems – called planners – that enable a robot to autonomously synthesize a series of actions that will achieve its goals.

On the other hand, the same increasing complexity of the requested services causes an analogous increase in the complexity of the specifications and of the programs controlling the robotic devices. Model checking (MC) (see, e.g., [1]), is a research area in Computer Science devoted to the definition of procedures for the automatic verification of programs and specifications.

By Safe Planning we mean the task of generating/validating plans that not only achieve the goal, but verify also a set of other user-defined properties, e.g., safety properties. In particular, Safe Planning results in the integration of planning and formal verification techniques. A possible and promising approach for Safe Planning is based on reduction to a propositional formula, that has to be be solved by a satisfiability solver. The approach was introduced in [2].

Given a planning problem (expressed, e.g., in STRIPS/PDDL language) and the safety properties the plan has to comply with (expressed in Linear Temporal Logic), a procedure, relying on the "planning as satisfiability approach", [3,4] for generating "Safe Plans" was introduced. Each component of the planning problem (action/fluent) is mapped into a (series of) propositional variables, and the planning problem is translated into a propositional formula. Then, a satisfiability solver is called, and the solution (if any) is mapped back into a plan. The approach can leverage on the fact that modern satisfiability solvers can deal with problems having millions of variables in few seconds.

2. Contributions of the thesis

The contribution of this thesis is in the definition of three new approaches for Safe Planning and, in particular, in the implementation, design and testing of three back-end solvers for deciding the formulas resulting from the approaches. All the solvers presented are based on boolean reasoning (SAT-based approach) using the SIMO (Satis-

*I am indebted with Enrico Giunchiglia for his help and support during the whole period of the thesis. The work was partially supported by MIUR, ASI and a grant from Intel Corp.

AI Communications
ISSN 0921-7126, IOS Press. All rights reserved

fiability Internal Modulo Object-oriented) solver, an efficient new generation decision procedure for propositional satisfiability based on the Davis-Logemann-Loveland algorithm, that exploits the recent enhancements in the SAT field. SIMO is a contribution of the thesis as well.

Safe Planning via Separation Logic. The approach uses propositional logic enhanced with arithmetic constraints. In this thesis we restrict to constraints of the type $x - y \leq c$, where x, y are arithmetic variables, and c is a numeric constant. The resulting theory is known as Separation (or Difference) Logic (SL) in the area of FV. This theory is enough for our approach due to the way the encoding of actions (and fluents) is performed. SL is strictly more expressive than SAT and, despite its simplicity, it can be often used to encode interesting problems from the planning and scheduling domains. TSAT++ (Temporal SATisfiability) is a SAT-based decision procedure for solving formulas expressed in SL, and introduced in this thesis. TSAT++ uses a specialized reasoner, based on a modification of the Bellman-Ford algorithm, for checking (in polynomial time) the arithmetic consistency of the (candidate) solution. TSAT++ embeds both state-of-the-art and newly introduced techniques. Among the last ones, the most notables are two techniques for minimizing the number of arithmetic constraints in the satisfying assignments and in the set of constraints responsible for the conflicts. TSAT++'s strength is also due to the effective *combination* of techniques.

Safe Planning via Answer Set Programming. Answer set programming (ASP) [5] is a new declarative paradigm for solving search problems appearing in knowledge representation and reasoning. To solve a program, a programmer designs a logic program such that models of the program are solutions to the problem. The procedure and the approach presented in [2] are encoded as logic programs (to be solved under the Answer Set semantic). The approach relies on the strong link between ASP and SAT: A SAT formula can be translated into a logic program in a modular way. Despite the strong link, ASP and SAT are different is many ways: Among others, ASP is a non monotonic logic while SAT is monotonic; ASP allows in general for more "compact" representations, while SAT solvers are more optimized than ASP solvers; and the two formalism are best suited for different problem domains. CMODELS2 (Computing models) is a SAT-based decision procedure for finding AS of logic programs. It is a joint work with the University of Texas at Austin. It uses a new solving approach that has the advantage of being based on SAT (and thus it can leverage on the great amount of work done in the SAT area), in comparison with other ASP solvers not based on boolean reasoning, and many advantages (the most important that it works in polynomial space and can compute all the solutions) in comparison with the other SAT-based ASP solver.

Safe Planning via SAT Optimizations. The last approach is related to some optimization problems that can be applied to a propositional formula. Among them, there is the "Min-One" problem: "Given a propositional formula, find the satisfying assignment with fewer variables assigned to true". The approach relies on an encoding of a planning problem (+ safety properties) into a propositional formula (the approach presented in [2]) and a minimization of a functional cost (that defines the Min-ONE problem and the fact that the propositional satisfying assignment must be "minimal" w.r.t. this function). The functional cost is defined on the actions of the planning problem. This functional cost can be easily encoded into a propositional formula as well: Then, running a slightly modified satisfiability solver on the overall formula guarantees that the first satisfying assignment is "optimal", i.e., the solution corresponds to a plan with "minimal cost". OPTSAT (OPTimal SATisfiability) is a new decision procedure for optimization problems related to propositional satisfiability, implemented along the lines described here. It can actually work with other types of problems other then Min-ONE, like the Maximum SATisfiability (Max-SAT) problem, and the "weighted" (where each variable or clause is assigned a weight) versions of the problems, using the very same algorithm and minimal modifications. It uses both state-of-the-art encoding methods as well as optimization of these encodings. The main advantages w.r.t. rival systems, that are mainly based on branch-and-bound algorithms, is that it does not have to always look in the entire search space to find the solution.

As a result, SIMO, CMODELS2, TSAT++ and OPTSAT compare favorably with state-of-the-art systems. All the details about the work can be found in the thesis [6].

References

[1] E. Clarke, 0. Grumberg and D. Long. Model checking. Proc. Int'l Summer School on Deductive Program Design. 1994.

[2] L. Carlucci Aiello, A. Cesta, E. Giunchiglia and P. Traverso. Merging Planning and Verification Techniques in for 'Safe Planning' in Space Robotics. Proc. 6th Int. Symposium on Artificial Intelligence, Robotics and Automation in Space: A New Space Odyssey. 2001.

[3] H. Kautz and B. Selman. Planning as satisfiability. Proc. ECAI-92, 359–363. 1992.

[4] P. Traverso, A. Cesta and E. Giunchiglia. Interactive autonomy for space applications. Proc. Intl. Workshop on Planning and Scheduling for Space. March 2000.

[5] M. Gelfond and V. Lifschitz. The stable model semantics for logic programming. in Logic Programming: Proceedings of the Fifth International Conference and Symposium. 1070–1080. 1988.

[6] M. Maratea. Efficient Decision Procedures for the Integration of Planning and Formal Verification in Robotic Systems. PhD thesis. University of Genova, Italy. Available at http://www.star.dist.unige.it/∼marco/Data/phd.pdf.gz. April 2005.