Efficient Control of Temporal Reasoning

M.Carmen Fernández-Gago,
Centre for Agent Research and Development
Department of Computing and Mathematics,
Manchester Metropolitan University
Email: C.Gago@doc.mmu.ac.uk

1 Introduction

An important part of all logics is proof theory which is concerned with how statements in the logical language can be combined. A proof transforms statements in the language using only the inference rules and the axioms. Employing inference rules such as binary resolution and factoring, the resolution method is complete although it is not very efficient. For applications in Computer Science, we require that such methods become more effective. In particular, we require strategies. A strategy is a rule or set of rules that governs the use of inference rules. In resolution-based theorem-provers developed for classical logics, a particulary successful strategy for avoiding the generation of redundant information during proof has been the *set of support* strategy [8].

Temporal Logic is a variety of non-classical logic used in a wide variety of areas within Computer Science and Arificial Intelligence, for example, Robotics [1], Databases [2], Program Specification [3], Hardware verification [4], and Agent-Based Systems [5]. Proof in temporal logic is inherently complex and its implementation can be achieved by different appoaches. Within the Centre for Agent Research and Development a proof method for temporal logics has been developed. This method is based upon the use of clausal resolution [6, 7] and it has been defined, proved correct, implemented and extended in a number of ways. However, in a number of cases, the basic method leads to the generation of an unnecessarily large set of formulas during the proof. As some of these formulas are irrelevant, i.e., are not needed in the proof, it is clear that refinements are needed. In order to make the temporal reasoning more efficient, high-level strategies will be needed. Following its success within classical logics, it is our intention to develop a *set of support* strategy for temporal resolution.

2 Some Strategies for Temporal Resolution

The resolution method developed for linear temporal logics [6], besides the translation to a normal form, involves classical resolution between formulae that occur at the same moment in time, and temporal resolution over states.

In the first case, i.e, where the clauses contain no eventualities ($\Diamond l$, i.e, *l holds now or at sometime in the future*), we have found a method to apply the set of support strategy. Further we have proved that by choosing the set of support in a specific way, the resolution method is complete. That is, we choose the set of support as all the *rules* with positive right hand sides (analogously negative right hand sides). With this strategy the amount of irrelevant information during the proof is reduced.

In the second case, the resolution method is more complex and involves the detection of a set of rules known as a *loop*, that is, rules that together imply $\Box l$ (*l holds now and at all moments in the future*) for resolution with $\Diamond \neg l$. The resolution rule is as follows:

$$\frac{\begin{array}{rcl} \phi_1 & \Rightarrow & \bigcirc \psi_1 \\ \vdots & \vdots & \vdots \\ \phi_n & \Rightarrow & \bigcirc \psi_n \\ \chi & \Rightarrow & \Diamond \neg l \end{array}}{\chi \Rightarrow (\neg \bigvee_{i=1}^{n} \phi_i) \mathcal{W} \neg l}$$

where $\bigwedge_{i=1}^{n}(\phi_i \Rightarrow \bigcirc(l \wedge \bigvee_{i=1}^{n}\phi_j))$ ($\bigcirc$ means *in the next moment in time*). This condition ensures that the set of $\phi_i \Rightarrow \bigcirc\psi_i$ rules together imply $\bigvee_{i=1}^{n}\phi_i \Rightarrow \bigcirc\,\Box l$.

But the process of detecting the rules that characterise a loop is not easy or obvious, although it is crucial for the resolution method. Because of that, a new resolution rule is introduced [9] and it is proved sound and complete. The idea behind this new temporal resolution rule is that, rather than insisting that we have a loop, we derive a more complex resolvent, that allows for the possibility that such a loop does not exist. This new temporal resolution is:

$$\frac{\begin{array}{rcl}\phi_1 & \Rightarrow & \bigcirc\psi_1 \\ \vdots & \vdots & \vdots \\ \phi_n & \Rightarrow & \bigcirc\psi_n \\ \chi & \Rightarrow & \Diamond\neg l\end{array}}{\chi \Rightarrow \left[(\Diamond\neg\text{looping}) \vee (\neg \bigvee_{i=1}^{n}\phi_i)\,\mathcal{W}\neg l\right]}$$

This new resolvent includes the side condition (looping), needed for applying the traditional temporal resolution , in the resolvent. If such a loop is not obvious the resolvents produced will guide the detection of the right loop, after to apply step resolution. If such a loop exists, the new resolvent turns out to be equivalent form.

In our current work we are considering how to make the initial 'guess' of a loop and how to use output from the proof to improve our guess.

We hope to apply these results to the development of strategies for temporal resolution that allows us to reduce the search space. In particular, we are interested to incorporate the set of support strategy, hoping it will be so successfully like in the classical case.

References

[1] M. Shanahan, *Solving the Frame Problem.* MIT Press, 1997.

[2] J. Chomicki and G. Saake, eds, *Logics for Databases and Information Systems.* Kluwer, 1998

[3] Z. Manna and A. Pnueli, *The Temporal Logic of Reactive and Concurrent Systems: Specification.* Springer-Verlag, 1992.

[4] G. Holzmann.The Model-Checker SPIN. *IEEE Trans. on Software Engineering* 23(5), 1997.

[5] A. Rao and M. Georgeff. Modeling Rational Agents within a BDI-Architecture. In *Proc. Int. Conf. on Principles of Knowledge Representation and Reasoning.* Morgan Kaufmann Publishers, 1991.

[6] M. Fisher. A Resolution Method for Temporal Logic. In *Proc. Int. Joint Conf. on Artificial Intelligence.* Morgan Kauffmann Publishers, 1991.

[7] C. Dixon. Temporal Resolution using a Breadth-First Search Algorithm. *Annals of Mathematics and Artificial Intelligence, 22.* Baltzer Science Publishers, 1998.

[8] L. Wos, R. Overbeek, E.Lusk and J. Boyle. *Automated Reasoning: Introduction and Application*, Prentice-Hall International, 1984.

[9] M. Fisher and C. Dixon, Guiding Clausal Temporal Resolution. In *Advances in Temporal Logic*,Kluwer Academic Publishers,1999.

[10] W. Mc.Cune, Solution of the Robbins Problem. *Journal of Automated Reasoning* 19(3), 1997.

[11] C. Dixon and M. Fisher, The Set of Support Strategy in Temporal Resolution. *In Proc. Int. Workshop on Temporal Reasoning.* IEEE Computer Society Press, 1998.