Combining Generalisation Into Instantiation Based Reasoning In EPR

Shasha Feng*
*The University of Manchester
Manchester, M13 9PL
fengs@cs.man.ac.uk

Abstract

Effectively propositional logic (EPR for short) is a fragment of first-order logic which can be effectively translated into propositional logic. Much attention has been drawn to reasoning in EPR because of its ability to represent real life applications such as bounded model checking and planning. In this paper we summarize some successul calculi in EPR reasoning and propose combining generalisation into instantiation-based reasoning.

1 Introduction

Effectively propositional logic (EPR for short) is a fragment of first-order logic whose formulae are those in the Bernays-Schønfinkel class. Much attention has been drawn to reasoning in EPR since several real life applications such as bounded model checking and planning can be naturally and succinctly encoded as EPR formulae.

When skolemised, EPR formulae contains no function symbols and thus have a finite Herbrand Universe, which allows one to translate them to propositional logic using *grounding*: substitutions of constants for variables of the formulae. Grounding and the subsequent use of SAT solvers remains one of the most successful approaches to checking the satisfiability of EPR formulae.

In this paper we summarize some successul calculi in EPR reasoning and propose combining generalisation into instantiation-based reasoning.

2 Existing Methods

2.1 Competent methods

Pérez and Voronkov (2008) analysed reasoning in EPR from the perspective of proof length. They show that propositional resolution for EPR may have exponentially longer refutations than resolution for EPR. It suggests that methods based on ground instantiation may be weaker than non-ground methods. Though their analysis based on resolution, which has so far been found not competitive on EPR formulae by the recent CASC competition, the conclusion finds some support from other calculi which are competent in EPR reasoning, as explained in the following two paragraph.

Model Evolution, proposed by Baumgartner and Tinelli (2003), is seen as a lift of DPLL to first-order logic. The calculus tries to build a model for input formulae. In the model, the presence of a predicate means that all of instances are satisfied, except those which are also instances of the predicate's instance, whose complement also appears in the model. For example, if one model only contains $P(x)$ and $\neg P(c)$, then it stands for a partial model of $\{P(a), P(b), P(d), \ldots\}$.

Instantiation based theorem proving, proposed by Ganzinger and Korovin (2003), is a cycle of generating instances and SAT checking, which instances generation is guided by resolution. SAT checking is performed on an abstraction of current problem. I. e. all the variables occurring are replace by $\perp$ Korovin (2008). So, in SAT checking, $P(x)$ and $P(a)$, abstracted to $P(\perp)$ and $P(a)$, are treated as different propositional variables. In some way, the $\perp$ in $P(\perp)$ can be seen as any constant in the domain. Thus, instantiation based theorem proving makes itself a calculus on the level of EPR instead of propositional logic.

2.2 Generalisation

Recently, Pérez and Voronkov (2008) introduces a generalisation rule and show that resolution for EPR may have exponentially longer proofs than resolution with generalisation. Suppose that in an EPR problem, the constants domain is $\{c_1, c_2, \ldots, c_8\}$, and we have $A[c_3], A[c_4], A[c_5], \ldots, A[c_8]$, where $A[x]$ is a quantifier-free formula with a free variable x. To generalise all the ground cases of $A[c_3], A[c_4], A[c_5], \ldots, A[c_8]$, it would be much convenient to add $\forall x A[x]$ to the formulae set. However, $\forall x A[x]$ only holds when $x \in \{c_3, c_4, \ldots, c_8\}$. So, we can add a constrained formula $\forall x A[x].C$, C represents the constraints about x. Resolution extends to constrained formulae naturally, with the resolvent's constraint being the intersection of the constraints from the two input formulae.

3 Combining Generalisation Into Instantiation Based Reasoning

We propose to combine generalisation rule into instantiation based reasoning. Constrained formulae generated by generalisation rule, together with instances generated by resolution rule, are added to the formulae set, which will be fed to a SAT solver. The constraints will be discarded from constrained formulae after some constraints checking. We illustrate by the following example.

Example 1 Suppose a formulae set is $\{P(x) \vee Q(x), \neg P(a), \neg P(b), \neg Q(b), R(x) \vee \neg Q(x), \neg R(b)\}$. Generalisation rule produces the following constrained formulae. $\neg P(x).x \in \{a, b\}, \neg Q(x).x \in \{b\}, \neg R(x).x \in \{b\}$. These constraints are satisfiable, as $x \in \{b\}$. Thus, we can add $\neg P(x), \neg Q(x), \neg R(x)$ to existing formulae set, waiting for a SAT solver, which will return unsatisfiable.

There are several unsolved problems in our method. One is what should be added when the constaints are unsatisfiable? One solution is to add only those constrained formulae whose constraints, put together, are satisfiable. Heuristic methods may be needed when chosing a subset of constrained formulae. Another solution is kind of splitting according to constants domain. For example, if the constraints formulae are $P(x).x \in S_1, Q(x).x \in S_2$, where $S_1 \cap S_2 = \phi$. We can get two duplications of existing formulae set, with all predicates subscribed by 1 and 2 respectively. Then replace existing formulae set with the union of duplications, and discard those $U_1(c)$ where $c \notin S_1$ and $V_2(d)$ where $d \in S_1$. For the constraints formulae $P(x).x \in S_1, Q(x).x \in S_2$, we add $P_1(x), Q_2(x)$.

Another problem is about lemma generating. In Ganzinger and Korovin (2003) method, when the problem is satisfiable, the SAT solver may return some clauses it derived, which can be fed into the problem as lemmas. With generalisation rule present, it is difficult to tell whether the returned clauses are universally valid or only on a specific subset.

4 Conclusions

We analyzed several successful calculi in EPR reasoning and proposed to combine generalisation into instantiation based reasoning. Some unsolved problems are described. Comparison with de Moura and Bjørner (2008) is also an interesting direction. Moreover, implementation is necessary to discovor how to balance over generalisation and instances generation on different problems.

References

Peter Baumgartner and Cesare Tinelli. The model evolution calculus. In *CADE*, pages 350–364, 2003.

Leonardo Mendonça de Moura and Nikolaj Bjørner. Deciding effectively propositional logic using dpll and substitution sets. In *IJCAR*, pages 410–425, 2008.

Harald Ganzinger and Konstantin Korovin. New directions in instantiation-based theorem proving. In *LICS*, pages 55–64, 2003.

Konstantin Korovin. iprover - an instantiation-based theorem prover for first-order logic (system description). In *IJCAR*, pages 292–298, 2008.

Juan Antonio Navarro Pérez and Andrei Voronkov. Proof systems for effectively propositional logic. In *IJCAR*, pages 426–440, 2008.