Searching for Invariants Using Temporal Resolution

James Brotherston\(^1\), Anatoli Degtyarev\(^2\), Michael Fisher\(^2\), and Alexei Lisitsa\(^2\)

\(^1\) Division of Informatics, University of Edinburgh, Edinburgh EH1 1HN, U.K.
jjb@dai.ed.ac.uk

\(^2\) Department of Computer Science, University of Liverpool, Liverpool L69 7ZF, U.K.
\{A.Degtyarev,M.Fisher,A.Lisitsa\}@csc.liv.ac.uk

Abstract. In this paper, we show how the clausal temporal resolution technique developed for temporal logic provides an effective method for searching for invariants, and so is suitable for mechanising a wide class of temporal problems. We demonstrate that this scheme of searching for invariants can be also applied to a class of multi-predicate induction problems represented by mutually recursive definitions. Completeness of the approach, examples of the application of the scheme, and overview of the implementation are described.

1 Introduction

The identification of invariants within complex, often inductive, system descriptions, is a vital component within the area of program verification. However, identifying such invariants is often complex. We are here concerned with finding invariants in a class of multi-predicate recursive definitions by translation of the problem to first-order temporal logic followed by application of a clausal temporal resolution method. It has been known for some time that first-order temporal logic over the Natural numbers (FOLTL, in short) is incomplete [Sza86]; that is, there exists no finitistic inference system which is sound and complete for this logic or, equivalently, the set of valid formulae of the logic is not recursively enumerable. The complete Gentzen-like proof systems for FOLTL contain the \(\omega\)-type infinitary rule \([Kaw87]\):

\[
\frac{\Gamma \rightarrow \Delta, \psi_1; \ldots; \Gamma \rightarrow \Delta, \psi_n;}{\Gamma \rightarrow \Delta, \Box \psi} \quad (\rightarrow \Box \omega)
\]

However, in some cases (in particular, in the propositional case [Pae88]), instead of the \(\omega\)-type rule \((\rightarrow \Box \omega)\) the following finitary rule can be used:

\[
\frac{\Gamma \rightarrow \Delta, I; \quad I \rightarrow \Box I; \quad I \rightarrow \psi}{\Gamma \rightarrow \Delta, \Box \psi} \quad (\rightarrow \Box)
\]

This rule corresponds to the induction axiom within temporal logic: \(\psi \land \Box (\psi \supset \Box \psi) \Rightarrow \Box \psi\). The formula \(I\) is called an invariant formula and has a close relation with invariant

\(^*\) Work supported by EPSRC grants GR/M46624, GR/M46631 and GR/R45367.

\(^1\) Intuitively, ‘\(\Box\)’ here stands for “in the next moment of time” and ‘\(\Box\)’ stands for “always in the future”; see \([2]\) for the definitions.
formulae in the logic of programs. Even in the propositional case, the search for such invariants can be very expensive. It is quite a usual situation (e.g. in Hoare logic for the partial correctness of while-programs) that the invariant has to be stronger than the desired conclusion suggests.

To illustrate the difficulties in searching for invariants let us consider an example. The sequent \( P(c), \Box \forall x (P(x) \supset \Box P(f(x))) \rightarrow \Box \exists y P(y) \) can be proved using as an invariant the formula \( I = \Box (\exists x P(x) \supset \Box \exists x P(f(x))) \land \exists x P(x) \). At the same time the most plausible conjecture is that there is no invariant for the sequent \( P(c), \forall x (P(x) \supset P(f(x))), \forall x (P(f(x)) \supset \Box P(x)) \rightarrow \Box \exists y P(y) \). In both these cases our arguments are heuristic since both sequents lie outside of any known complete fragment of FOLTL.

Recently, the interesting monodic fragment of first-order temporal logic has been investigated [HWZ00]. This has a quite transparent (and intuitive) syntactic definition and a finite Hilbert-like inference system [WZ01]. In [DF01] a clausal temporal resolution procedure has been developed covering a special subclass of the monodic fragment, namely the subclass of ground eventuality monodic problems. In this paper we apply this clausal resolution method in order to give a sound and complete scheme for searching for invariants for sequents of the form \( SP \rightarrow \Box \psi \) where \( SP \) is a monodic temporal specification and \( \psi \) is a ground first-order formula.

There is some similarity between linear temporal logic over the Natural numbers and Peano arithmetic. The induction axiom of Peano arithmetic \( \varphi(0) \land \forall n (\varphi(n) \supset \varphi(s(n))) \Rightarrow \forall n \varphi(n) \) corresponds to the induction axiom within temporal logic, and there is a complete and consistent Gentzen-like proof system for Peano arithmetic where the induction axiom is replaced by an \( \omega \)-type inference rule \( \rightarrow \forall \omega \) similar to \( \rightarrow \Box \omega \). Because of that we will refer to the temporal problem \( SP \rightarrow \Box \psi \) mentioned above as a (ground) induction problem (taking into account that the formula \( \psi \) under \( \Box \) is ground).

An important aspect of this paper is that we particularly consider a class of induction problems over the Natural numbers with recursive predicate definitions. Such recursion is difficult for many systems to work with effectively, often leading to quite complex and non-trivial induction schemes (see, for example, [BS00] where the use of mutually recursive definitions has been investigated and several heuristic multi-predicate induction schemes have been developed in order to make implementations of such definitions useful). If such a problem with mutually recursive definitions is translated into a monodic ground induction problem then we can automate its proof, using our invariant scheme. This aspect is demonstrated in examples later in the paper.

Structure of the paper. We split our presentation into two main parts: the first essentially concerns propositional (discrete, linear) temporal logic; the second targets a fragment of monodic first-order temporal logic [HWZ00,DF01]. While the propositional part is clearly included within the first-order part, we have chosen to introduce this separately in order to give the reader a simpler introduction to the techniques involved. Thus, in §3 we consider this propositional temporal fragment, providing formal justification and a simple example. Then, in §4 we consider first-order monodic ground induction problems, providing both completeness arguments and examples, and, in §5 outline the current state of the implementation. Finally, in §6 we provide concluding remarks. Some technical proofs in §4 are omitted due to lack of space and can be found in the full version of this paper, which is available as a technical report [BDFL02].