A Trace Consistent Subset of PTL

P.S. Thiagarajan *
School of Mathematics
SPIC Science Foundation
92 G.N. Chetty Road
T. Nagar, Madras-600 017
India
E-mail: pst@ssf.ernet.in

Abstract. The Propositional Temporal logic of Linear time (PTL) is often interpreted over sequences (of states or actions) that can be grouped into equivalence classes under a natural partial order based semantics. It has been noticed in the literature that many PTL formulas are consistent with respect to such an equivalence relation. Either all members of an equivalence class satisfy the formula or none of them do. Such formulas can be often verified efficiently. One needs to test satisfiability of the concerned formula for just one member of each equivalence class. Here we identify an interesting subset of such equivalence consistent PTL formulas - denoted PTL | - by purely semantic means. We then use a partial order based linear time temporal logic denoted TrPTL | to obtain a simple syntactic characterization of PTL |. We also provide solutions to the satisfiability problem and an associated model checking problem for TrPTL | using networks of Büchi automata.

0 Introduction

The Propositional Temporal logic of Linear time (PTL) is often interpreted over sequences (of states or actions) capturing the behaviour of a distributed program. It is well known that such a collection of sequences can be grouped together into equivalence classes under a natural partial order based semantics; two sequences are equivalent just in case they are two interleavings of the same partially ordered stretch of behaviour. It has been noticed in the literature in different contexts (for instance in [V] and [KP]) that many of the dynamic properties of a distributed program that one would like to verify are consistent with respect to such a partial order based semantics. Either all the members of an equivalence class satisfy the property or none of them do. Such consistent properties can be often verified efficiently. One needs to verify the concerned property for just one member of each equivalence class of runs. This observation lies at the heart of so called partial order based verification methods reported for instance in [V], [KP], [P] and [GW].

With this as motivation, the question we address here is: When is a PTL formula amenable to partial order based verification techniques? In order to make

* This work has been partly supported by the IFCPAR Project 502-1.
the question precise, we choose the familiar setting consisting of a network of communicating automata that coordinate their behaviour by performing common actions together. In such a setting, there is a natural notion of causal independence of actions induced by the distribution of actions over the agents; two actions are independent if they are performed by disjoint sets of agents. This leads via standard notions in the theory of traces, to an equivalence relation, say $\approx$, over the runs of the network. The PTL formula $\psi$ may then said to be trace consistent if for any two runs $r_1$ and $r_2$ that are $\approx$-equivalent, $r_1$ satisfies $\psi$ if and only if $r_2$ satisfies $\psi$. The point is, trace consistent formulas are precisely those that are amenable to partial order based verification methods.

The general problem of characterizing the set of trace consistent PTL formulas appears to be hard. Hence as a partial solution we present here a characterization of an interesting subset of trace consistent formulas. We do so by semantically identifying a subset of PTL formulas denoted PTL$^\approx$. We then show that PTL$^\approx$ has the same expressive power as the linear time temporal logic called product TrPTL and denoted TrPTL$^\approx$. Unlike PTL which is interpreted over sequences, the logic TrPTL$^\approx$ is interpreted over Mazurkiewicz traces which may be viewed as restricted kinds of labelled partial orders. It has a very simple syntax. Moreover, our result showing that PTL$^\approx$ and TrPTL$^\approx$ are expressively equivalent yields the corollary that every formula of PTL$^\approx$ is semantically equivalent to a very easily constructed syntactic image of a TrPTL$^\approx$ formula. In this sense, we have also a syntactic characterization of PTL$^\approx$.

A second result that we establish is a strong relationship between PTL$^\approx$ and networks of communicating automata called product automata in this paper. This result consists of a decision procedure for TrPTL$^\approx$ in terms of product automata. This in turn leads to the solution of the model checking problem for TrPTL$^\approx$ when programs are modelled as product automata.

Our automata-theoretic solutions are an easy extension of the ones developed for PTL in terms of Büchi automata by Vardi and Wolper [VW]. Our decision procedure and the model checking procedure has the same time complexity as the corresponding procedures for PTL; the decision procedure runs in time $2^{O(n)}$ and the model checking procedure runs in time $O(m \cdot 2^{O(n)})$ where $n$ is the size of the input formula and $m$ is the size of the model of the input program. The logic is a strictly weaker subsystem of the logic TrPTL studied in [T]. TrPTL is also interpreted over Mazurkiewicz traces. However its decision procedure runs in time $2^{O(n^2 \log n)}$ and uses Büchi asynchronous automata which are strictly more expressive than product automata.

Our result concerning PTL$^\approx$ has been triggered by the work of Niebert and Penczek [NP]. They show that the set of trace consistent PTL formulas treated by Valamari [V1] and by Peled [P] can be easily translated into TrPTL$^\approx$. This subset of formulas is however identified by placing severe syntactic restrictions; for instance, the next state modality is not allowed. The work of Peled in [P] is very relevant because it defines the notion of trace consistent formulas (but called equivalence robust), points out their importance in the context of efficient model checking and proceeds to identify a subset of trace consistent formulas.