Tractable Temporal Reasoning–Extended Abstract

Clare Dixon[⋆] Michael Fisher[⋆] Boris Konev[⋆]

[⋆]Department of Computer Science
University of Liverpool
Liverpool L69 3BX, UK
{clare,michael,konev}@csc.liv.ac.uk

Abstract

We outline a tractable fragment of PTL, and related resolution calculus, which makes the central use of XOR sets. Such sets enforce the restriction that exactly one proposition from each XOR set must hold at each moment in time.

1 Introduction

Temporal logics have been used to describe a wide variety of systems, from both Computer Science and Artificial Intelligence. The basic idea of proof, within propositional, discrete temporal logics, is also both intuitive and appealing. However the complexity of satisfiability for such logics is high. For example, the complexity of satisfiability for propositional linear time temporal logic (PTL) is PSPACE-complete Sistla and Clarke (1985).

Often temporal problems involve an underlying structure, such as an automaton, where a key property is that the automaton can be in exactly one state at each moment. Such problems frequently involve several processes or agents, each with underlying automaton-like structures, and we are interested in properties relating to how the agents progress under particular models of concurrency such as synchrony, asynchrony etc., or particular coordination or cooperation actions. In this extended abstract we consider a new fragment of PTL that incorporates the use of XOR operators, denoted $(q_1 \oplus q_2 \oplus \ldots \oplus q_n)$ meaning that exactly one q_i holds for $1 \leq i \leq n$.

2 XOR Temporal Logic

The logic we consider is called "TLX", and its syntax and semantics essentially follow that of PTL Gabbay et al. (1980), with models (isomorphic to the Natural Numbers, $\mathbb{N}$) of the form: $\sigma = t_0, t_1, t_2, t_3, \ldots$ where each state, t_i, is a set of proposition symbols, representing those propositions which are satisfied in the i^{th} moment in time. The notation $(\sigma, i) \models A$ denotes the truth (or otherwise) of formula A in the model σ at state index $i \in \mathbb{N}$. This leads to semantic rules:

$$\begin{array}{lll} (\sigma,i) \models \bigcirc A & \text{iff} & (\sigma,i+1) \models A \\ (\sigma,i) \models \Diamond A & \text{iff} & \exists k \in \mathbb{N}.\ (k \geqslant i) \text{ and } (\sigma,k) \models A \\ (\sigma,i) \models \Box A & \text{iff} & \forall j \in \mathbb{N}.\ \text{if } (j \geqslant i) \text{ then } (\sigma,j) \models A \end{array}$$

For any formula A, model σ, and state index $i \in \mathbb{N}$, then either $(\sigma, i) \models A$ holds or $(\sigma, i) \models A$ does not hold, denoted by $(\sigma, i) \not\models A$. If there is some σ such that $(\sigma, 0) \models A$, then A is said to be *satisfiable*. If $(\sigma, 0) \models A$ for all models, σ, then A is said to be *valid* and is written $\models A$.

The main novelty in TLX is that it is parameterised by XOR-sets $\mathcal{P}_1$, $\mathcal{P}_2$,..., and the formulae of TLX($\mathcal{P}_1, \mathcal{P}_2, \ldots$) are constructed under the restrictions that *exactly* one proposition from every set $\mathcal{P}_i$ is true in every state. For example, if we consider just one set of propositions $\mathcal{P}$, we have $\Box(p_1 \oplus p_2 \oplus \ldots \oplus p_n)$ for all $p_i \in \mathcal{P}$. Furthermore, we assume that there exists a set of propositions in addition to those defined by the parameters, and that these propositions are unconstrained as normal. Thus, TLX() is essentially a standard propositional, linear temporal logic, while TLX($\mathcal{P}$,$\mathcal{Q}$,$\mathcal{R}$) is a temporal logic containing at *least* the propositions $\mathcal{P} \cup \mathcal{Q} \cup \mathcal{R}$, where $\mathcal{P} = \{p_1, p_2, \ldots, p_l\}$, $\mathcal{Q} = \{q_1, q_2, \ldots, q_m\}$, and $\mathcal{R} = \{r_1, r_2, \ldots, r_n\}$ where $\mathcal{P}$, $\mathcal{Q}$ and $\mathcal{R}$ are disjoint, but also satisfying

$$\Box[(p_1 \oplus p_2 \oplus \ldots \oplus p_l) \ \wedge (q_1 \oplus q_2 \oplus \ldots \oplus q_m) \ \wedge (r_1 \oplus r_2 \oplus \ldots \oplus r_n)]$$

3 A Normal Form and Resolution Calculus for TLX

First we define a normal form to which we apply a set of resolution rules. Any TLX formula can be transformed into this normal form. Assume we have n sets of XOR propositions $P_1 = \{p_{11}, \ldots p_{1N_1}\}$, ..., $P_n = \{p_{n1}, \ldots p_{nN_n}\}$ and a set of additional propositions $A = \{a_1, \ldots a_{N_a}\}$. In the following:

- $\hat{P}_{ij}^{-}$ ($\overset{\vee}{P}_{ij}^{+}$) denotes a conjunction (disjunction) of negated (positive) XOR propositions from the set P_i;
- $\hat{A}_i$ ($\overset{\vee}{A}_i$) denotes a conjunction (disjunction) of non-XOR literals;

A normal form for TLX is of the form $\square \bigwedge_i C_i$ where each C_i is one of the following.

$$\begin{array}{rcll} \textbf{start} & \Rightarrow & \overset{\vee}{P}_{1i}^{+} \vee \ldots \vee \overset{\vee}{P}_{ni}^{+} \vee \overset{\vee}{A}_i & \text{Initial clause} \\ \hat{P}_{1j}^{-} \wedge \ldots \hat{P}_{nj}^{-} \wedge \hat{A}_j & \Rightarrow & \bigcirc(\overset{\vee}{P}_{1j}^{+} \vee \ldots \vee \overset{\vee}{P}_{nj}^{+} \vee \overset{\vee}{A}_j) & \text{Step clause} \\ \textbf{true} & \Rightarrow & \Diamond(\overset{\vee}{P}_{1k}^{+} \vee \ldots \vee \overset{\vee}{P}_{nk}^{+} \vee \overset{\vee}{A}_k) & \text{Sometime clause} \end{array}$$

We have developed a sound, complete and terminating resolution calculus for the logic TLX. There are a number of resolution rules applied between initial clauses, step clauses, and sometimes clauses with sets of step clauses. For more details see Dixon et al. (2007)

Theorem 1 *[Dixon et al. (2007)] A set of clauses is unsatisfiable if an only if it has a refutation by the temporal resolution procedure given.*

Theorem 2 *(Termination) [Dixon et al. (2007)] The resolution procedure terminates.*

Theorem 3 *(Complexity) [Dixon et al. (2007)] If a set of temporal clauses is unsatisfiable, temporal resolution will deduce a contradiction in time polynomial in $N_1 \times N_2 \times \cdots \times N_n \times 2^{N_a}$ (where N_i is the number of propositions in $\mathcal{P}_i$ and N_a is the number of unconstrained propositions).*

4 Concluding Remarks

In this extended abstract we have described a tractable sub-class of temporal logic, based on the central use of XOR operators. This work extends the fragment defined in Dixon et al. (2006). TLX can be decided, tractably, via clausal temporal resolution. Importantly, multiple XOR fragments can be combined. This new approach to temporal reasoning provides a framework in which tractable temporal logics can be engineered by intelligently combining appropriate XOR fragments. Further, this has the potential to provide a deductive approach, with a similar complexity to model checking, thus obtaining a practical verification method. In addition, this approach has the potential to be extended to first-order temporal logics which can deal with infinite state systems.

The complexity result means that TLX is more amenable to efficient implementation than other similar temporal logics. Moreover, since no two propositions from the same XOR set can occur in the right- (or left-) hand side of any temporal clause, one can efficiently represent disjunctions of (positive) propositions (and conjunctions of negated propositions) as bit vectors and the rules of temporal resolution as bit-wise operations on such bit vectors. Thus, temporal reasoning in TLX can be efficient not only in theory, but also in practice.

Acknowledgements

The work of the first and last authors was partially supported by EPRSC grant number GR/S63182/01 "Dynamic Ontologies: a Framework for Service Descriptions". This extended abstract is a summary of the work in Dixon et al. (2007).

References

C. Dixon, M. Fisher, and B. Konev. Is There a Future for Deductive Temporal Verification? In *Proceedings of TIME 2006 the Thirteenth International Symposium on Temporal Representation and Reasoning*, Budapest, Hungary, June 2006. IEEE Computer Society Press.

C. Dixon, M. Fisher, and B. Konev. Tractable Temporal Reasoning. In *Proc. International Joint Conference on Artificial Intelligence (IJCAI)*, pages 318–23. AAAI Press, 2007.

D. Gabbay, A. Pnueli, S. Shelah, and J. Stavi. The Temporal Analysis of Fairness. In *Proceedings of the Seventh ACM Symposium on the Principles of Programming Languages*, pages 163–173, Las Vegas, Nevada, January 1980.

A. P. Sistla and E. M. Clarke. Complexity of Propositional Linear Temporal Logics. *ACM Journal*, 32(3):733–749, July 1985.