Implementing Tractable Temporal Logics

Lan Zhang[⋆] Clare Dixon[⋆] Ullrich Hustadt[⋆] [*]

[⋆] Department of Computer Science,
University of Liverpool
Liverpool, L69 7ZF, UK
{lan,clare,ullrich}@csc.liv.ac.uk

1 Introduction

In (Dixon, Fisher, and Konev, 2006) a fragment of propositional linear-time logic (PLTL), called PLTL-X $_{\mathcal{A}}$, has been defined. It is sufficiently expressive to capture Büchi automata (Büchi, 1962) using formulae in PLTL with boolean XOR operators. The formulae are of polynomial size in the size of the Büchi automata captured. The satisfiability problem is of polynomial complexity. The paper also defined a tractable temporal resolution calculus for PLTL-X $_{\mathcal{A}}$. This calculus provides an efficient theorem proving method for this fragment as well as providing an alternative way to check the emptiness of Büchi automata.

In this abstract we present the prover XA (XOR clauses representing Automata), an implementation of the calculus of (Dixon, Fisher, and Konev, 2006). We compare our system with TRP++, a full PLTL resolution theorem prover, and then present our conclusions.

2 The Temporal Logic PLTL-X$_{\mathcal{A}}$

The temporal logic PLTL-X$_{\mathcal{A}}$ (Dixon, Fisher, and Konev, 2006) is a sub-class of PLTL (Gabbay, Pnueli, Shelah, and Stavi, 1980) and is defined via a normal form, called *SNFX*$_{\mathcal{A}}$ (Dixon, Fisher, and Konev, 2006), shown in Figure 1. All clauses are implicitly in the scope of the temporal $\square$ operator ('always'). One key aspect of PLTL-X $_{\mathcal{A}}$ is that the set $\mathcal{P}$ of propositions over which formulae are constructed, is partitioned into two disjoint sets, $\mathcal{S} = \{q_1 \dots q_n\}$ and $\mathcal{L} = \{l_1 \dots l_n\}$. We can use $\mathcal{S}$ and $\mathcal{L}$ to represent *states* and *labels* of Büchi automata respectively. Also, in addition to the standard boolean and temporal connectives, the signature of the language of PLTL-X $_{\mathcal{A}}$ includes the boolean XOR operator '$\oplus$'. Its semantics is defined in the standard way, i.e., $\varphi_1 \oplus \cdots \oplus \varphi_m$ is true (at a particular moment in time) iff exactly one φ_j, $1 \leq j \leq m$ is true (at that moment in time). We call N a *set of SNFX*$_{\mathcal{A}}$ *clauses* iff (a) N contains only clauses of the form (1) to (6) in Figure 1, (b) N contains at most one *sometime* clause and exactly one *initial* clause, (c) for every $q_i \in \mathcal{S}$ and every $l_j \in \mathcal{L}$ there is at most one clause of the form $(q_i \wedge l_j) \Rightarrow \bigcirc(q_{i_1} \vee \dots \vee q_{i_r})$ in N, and (d) if **true** $\Rightarrow \neg q_i \vee \neg l_j$ is in N for some $q_i \in \mathcal{S}$ and some $l_j \in \mathcal{L}$, then $(q_i \wedge l_j) \Rightarrow \bigcirc(q_{i_1} \vee \dots \vee q_{i_r})$ is not in N.

Figure 2 shows a sound and complete resolution calculus (Dixon, Fisher, and Konev, 2006) for SNFX $_{\mathcal{A}}$. Angled brackets $\langle \dots \rangle$ around a premise indicate that the conclusion of the rule replaces that premise in the clause set. Further, since the number of SNFX$_{\mathcal{A}}$ clauses which can be formed over the finite set $\mathcal{P}$ of propositional variables is finite, we can guarantee termination of proof conducted by this calculus. In addition, Dixon, Fisher, and Konev (2006) establish that a Büchi automaton has an accepting run if and only if the PLTL-X $_{\mathcal{A}}$ formula corresponding to it is satisfiable.

1. **start**	$\Rightarrow (q_{i_1} \vee \dots \vee q_{i_k})$	(*initial*)	4. **true**	$\Rightarrow \Diamond(q_{i_1} \vee \dots \vee q_{i_s})$	(*sometime*)
2. $(q_i \wedge l_j)$	$\Rightarrow \bigcirc(q_{i_1} \vee \dots \vee q_{i_r})$	(*step*)	5. **true**	$\Rightarrow q_1 \oplus q_2 \oplus \dots \oplus q_n$	(*XOR-*$\mathcal{S}$)
3. **true**	$\Rightarrow R_c$	(*universal*)	6. **true**	$\Rightarrow l_1 \oplus l_2 \oplus \dots \oplus l_m$	(*XOR-*$\mathcal{L}$)

Here $q_{i_k}, q_{i_r}, q_{i_s} \in \mathcal{S}, k, r, s \geq 0, m, n \geq 1$ and $l_j \in \mathcal{L}$, and where R_c must be one of $\neg q_i$, or $(\neg q_i \vee \neg l_j)$.

Figure 1: Types of SNFX$_{\mathcal{A}}$ clauses

[*]The first and third author are supported by EPSRC grant EP/D060451/1.

$$(SU)\ \frac{\begin{array}{c}\langle q_i \wedge l_j \Rightarrow \bigcirc(Q \vee q_k)\rangle \\ \textbf{true} \Rightarrow \neg q_k\end{array}}{q_i \wedge l_j \Rightarrow \bigcirc Q}$$

$$(IR)\ \frac{\begin{array}{c}\langle \textbf{start} \Rightarrow Q \vee q_i\rangle \\ \textbf{true} \Rightarrow \neg q_i\end{array}}{\textbf{start} \Rightarrow Q}$$

$$(SR)\ \frac{\langle q_i \wedge l_j \Rightarrow \bigcirc\ \textbf{false}\ \rangle}{\textbf{true} \Rightarrow \neg q_i \vee \neg l_j}$$

$$(HR)\ \frac{\begin{array}{c}\langle \textbf{true} \Rightarrow \neg q_k \vee \neg l_1\rangle \\ \vdots \\ \langle \textbf{true} \Rightarrow \neg q_k \vee \neg l_m\rangle \\ \textbf{true} \Rightarrow l_1 \oplus \cdots \oplus l_m\end{array}}{\textbf{true} \Rightarrow \neg q_k}$$

$$(TR)\ \frac{\begin{array}{c}\langle (q_{i_0^1} \wedge l_{j_0^1}) \Rightarrow \bigcirc(q_{i_1^1} \vee \ldots \vee q_{i_{r_1}^1})\rangle \\ \vdots \\ \langle (q_{i_0^n} \wedge l_{j_0^n}) \Rightarrow \bigcirc(q_{i_1^n} \vee \ldots \vee q_{i_{r_n}^n})\rangle \\ \textbf{true} \Rightarrow \Diamond \bigvee_k q_k\end{array}}{\textbf{true} \Rightarrow \bigwedge_{s=1}^n \neg q_{i_0^s}}$$

where N_r^u is the set of right-hand sides of all universal, XOR-$\mathcal{S}$, and XOR-$\mathcal{L}$ clauses in a set of SNFX$_{\mathcal{A}}$ clauses and
$\bigwedge_{s=1}^n (N_r^u \wedge (q_{i_1^s} \vee \ldots \vee q_{i_{r_s}^s}) \Rightarrow \bigwedge_k \neg q_k)$ and
$\bigwedge_{s=1}^n (N_r^u \wedge (q_{i_1^s} \vee \ldots \vee q_{i_{r_s}^s}) \Rightarrow \bigvee_{t=1}^n (q_{i_0^t} \wedge l_{j_0^t}))$.

Figure 2: The resolution calculus $\mathcal{C}_{\mathrm{PLTL-X}_{\mathcal{A}}}$ for PLTL-X$_{\mathcal{A}}$

3 Evaluation of XA

XA implements the resolution calculus described in Section 2 and is implemented in Java. To test and evaluate XA we have used a collection of PLTL-X$_{\mathcal{A}}$ formulae corresponding to Büchi automata with the number of states varying between 3 and 500. We have compared XA to TRP++ (Hustadt and Konev, 2004), a clausal resolution-based prover for full PLTL, on this collection of PLTL-X$_{\mathcal{A}}$ formulae. As the language used by TRP++ does not include the XOR operator, an XOR clause with n propositions has to be represented by $1 + \frac{n(n-1)}{2}$ universal clauses for the purpose of applying TRP++ to a PLTL-X$_{\mathcal{A}}$ formula. Therefore, in all our tests, the input to TRP++ is bigger than the input to XA. In all our experiments, XA requires fewer inference steps than TRP++. The difference in the number of inference steps is greater than what might be expected simply based on the difference in the number of input clauses. The reason is that our calculus utilises the XOR clauses within the resolution rules themselves and the restricted format of clauses means that the conclusion of all resolution rules subsumes one or more of the parents. This keeps both the number of clauses generated small and the clause set at any moment small, which makes the process of theorem proving efficient and succinct.

4 Conclusions and Future Work

The comparison between XA and TRP++ shows the advantage of this calculus: XA's search space is polynomial, on the other hand, TRP++'s search space is exponential. Note, however XA is specialised for a fragment of PLTL whereas TRP++ is more general and therefore can deal with a larger class of formulae.

The original calculus has recently been extended (Dixon, Fisher, and Konev, 2007). It allows more than two XOR clauses as well as a set of 'normal' non-XOR propositions and is therefore able to capture a wider range of problems than PLTL-X$_{\mathcal{A}}$. Further, the complexity of the calculus is still polynomial if the number of non-XOR proposition is low. Adapting our system to the calculus (Dixon, Fisher, and Konev, 2007) is future work.

References

J. R. Büchi. On a Decision Method in Restricted Second Order Arithmetic. *Proc. Int. Congress on Logic, Methodology and Philosophy of Science 1960*, pages 1–11, Stanford University Press, 1962.

C. Dixon, M. Fisher, and B. Konev. Is there a future for deductive temporal verification? In *Proc. TIME 2006*, pages 11–18. IEEE, 2006.

C. Dixon, M. Fisher, and B. Konev. Tractable temporal resolution. In *Proc. IJCAI-2007*, pages 318–323. AAAI Press, 2007.

D. Gabbay, A. Pnueli, S. Shelah, and J. Stavi. The Temporal Analysis of Fairness. In *Proc. POPL '80*, pages 163–173. ACM, 1980.

U. Hustadt and B. Konev. TRP++: A temporal resolution prover. In *Collegium Logicum*, pages 65–79. Kurt Gödel Society, 2004.