Towards a tableau-based procedure for PLTL based on a multi-conclusion rule and logical optimizations

Mauro Ferrari[1], Camillo Fiorentini[2], Guido Fiorino[3]

[1] DiSTA, Univ. degli Studi dell'Insubria, Via Mazzini, 5, 21100, Varese, Italy
[2] DI, Univ. degli Studi di Milano, Via Comelico, 39, 20135 Milano, Italy
[3] DISCO, Univ. degli Studi di Milano-Bicocca, Viale Sarca, 336, 20126, Milano, Italy

Abstract. We present an ongoing work on a proof-search procedure for Propositional Linear Temporal Logic (PLTL) based on a one-pass tableau calculus with a multiple-conclusion rule. The procedure exploits logical optimization rules to reduce the proof-search space. We also discuss the performances of a Prolog prototype of our procedure.

1 Introduction

In recent years, we have introduced new tableau calculi and logical optimization rules for propositional Intuitionistic Logic [4] and propositional Gödel-Dummett Logic [7]. As an application of these results, we have implemented theorem provers for these logics [3, 7] which outperform their competitors. The above quoted calculi and optimizations are the result of a deep analysis of the Kripke semantics of the logic at hand. In this paper, we apply such a semantical analysis to PLTL. In particular, we present a refutation tableau calculus and some logical optimizations for PLTL and we briefly discuss a prototype Prolog implementation of the resulting proof-search procedure.

As for related work, our tableau calculus lies in the line of the one-pass calculi based on sequents and tableaux calculi [14, 2, 10], whose features are suitable for automated deduction. We also cite as related the approaches based on sequent calculi discussed in [12, 13] and the natural deduction based proof-search techniques discussed in [1]. The results in [8, 15] are based on resolution, thus they are related less to our approach.

2 Tableau calculus and replacement rules

We consider the language based on a denumerable set of propositional variables $\mathcal{V}$, the logical constants $\top$ (true), $\bot$ (false), $\neg$ and $\vee$ and the modal operators $\circ$ (next) and $\mathbf{U}$ (until). We define $\Box A$ as $\neg(\top \mathbf{U} \neg A)$. Given a set of formulas $\mathcal{S}$, we denote with $\circ S$ the set $\{\circ A \mid A \in \mathcal{S}\}$.

PLTL is semantically characterized by rooted linearly ordered Kripke models; formally, a PLTL-*model* is a structure $\mathcal{K} = \langle P, \leq, \rho, V \rangle$ where P is the set of

$$\frac{\mathcal{S}, A\mathbf{U}B}{\mathcal{S}, B \mid \mathcal{S}, A, \neg B, \circ(A\mathbf{U}B)}\mathbf{U} \qquad \frac{\mathcal{S}, \neg(A\mathbf{U}B)}{\mathcal{S}, \Box\neg B \mid \mathcal{S}, \neg A, \neg B \mid \mathcal{S}, A, \neg B, \circ\neg(A\mathbf{U}B)}\neg\mathbf{U}$$

$$\frac{\mathcal{S}, \neg\circ A}{\mathcal{S}, \circ\neg A}\neg\circ \quad \frac{\mathcal{S}, A \vee B}{\mathcal{S}, A \mid \mathcal{S}, B}\vee \quad \frac{\mathcal{S}, \neg(A \vee B)}{\mathcal{S}, \neg A, \neg B}\neg\vee \quad \frac{\mathcal{S}, \neg\neg A}{\mathcal{S}, A}\neg\neg \quad \frac{\mathcal{S}, \neg\top}{\mathcal{S}, \bot}\neg\top \quad \frac{\mathcal{S}, \neg\bot}{\mathcal{S}, \top}\neg\bot$$

$$\frac{\mathcal{T}, \circ\mathcal{A}, \circ\mathcal{B}}{\mathcal{A}, \mathcal{B}^+, \circ\mathcal{B} \mid \mathcal{A}, \mathcal{H}_1 \mid \ldots \mid \mathcal{A}, \mathcal{H}_m}\mathrm{Lin}$$

$\mathcal{T} \subseteq \mathcal{V} \cup \{\neg p \mid p \in \mathcal{V}\} \cup \{\top, \bot\}$, $\mathcal{A}$ is a possibly empty set,
$\mathcal{B} = \{U_1, \ldots, U_m\}$ is a possibly empty set, with $U_i = A_i\mathbf{U}B_i$ or $U_i = \neg(A_i\mathbf{U}B_i)$
$\mathcal{B}^+ = \{U_i^+ | U_i \in \mathcal{B}\}$, where

$$\begin{array}{rclrcl} (A\mathbf{U}B)^+ &=& A & \quad (A\mathbf{U}B)^- &=& B \\ (\neg(A\mathbf{U}B))^+ &=& \neg B & \quad (\neg(A\mathbf{U}B))^- &=& \neg A, \neg B \end{array}$$

$$\mathcal{H}_i = \{\circ U_1, U_1^+, \ldots, \circ U_{i-1}, U_{i-1}^+\} \cup \{U_i^-\} \cup \{U_{i+1}, \ldots, U_m\} \ (i = 1, \ldots, m)$$

Fig. 1. The tableau calculus for PLTL

worlds, $\leq$ is a linear well-order with minimum ρ and no maximum element, V is a function associating with every world $\alpha \in P$ the set of propositional variables satisfied in α. Given $\alpha \in P$, the *immediate successor* of α, denoted by α', is the minimum of the $<$-successors of α. The satisfiability of a formula A in a world α of $\mathcal{K}$, written $\mathcal{K}, \alpha \Vdash A$ (or simply $\alpha \Vdash A$), is defined as follows:

- for $p \in \mathcal{V}$, $\alpha \Vdash p$ iff $p \in V(\alpha)$; $\alpha \Vdash \top$; $\alpha \nVdash \bot$;
- $\alpha \Vdash \neg A$ iff $\alpha \nVdash A$; $\alpha \Vdash A \vee B$ iff $\alpha \Vdash A$ or $\alpha \Vdash B$;
- $\alpha \Vdash \circ A$ iff $\alpha' \Vdash A$;
- $\alpha \Vdash A\mathbf{U}B$ iff $\exists \beta \geq \alpha : \beta \Vdash B$ and $\forall \gamma : \alpha \leq \gamma < \beta$, $\gamma \Vdash A$.

The following properties can be easily proved. The latter one follows by the fact that $\leq$ is a well-order, hence, if B is satisfiable in some $\gamma \geq \alpha$, there exists the minimum $\gamma^* \geq \alpha$ satisfying B.

- $\alpha \Vdash \Box A$ iff $\forall \beta \geq \alpha, \beta \Vdash A$;
- $\alpha \nVdash A\mathbf{U}B$ iff $(\forall \gamma \geq \alpha, \gamma \nVdash B)$ or $(\exists \beta \geq \alpha : \beta \nVdash A$ and $\forall \gamma : \alpha \leq \gamma \leq \beta, \gamma \nVdash B)$.

A set of formulas $\mathcal{S}$ is *satisfiable in* α ($\mathcal{K}, \alpha \Vdash \mathcal{S}$) if every formula of $\mathcal{S}$ is satisfiable in α; $\mathcal{S}$ *is satisfiable* if it is satisfiable in some world of a PLTL-model. The rules of the tableau calculus $\mathbf{T}$ for PLTL are given in Fig. 1. The peculiar rule of $\mathbf{T}$ is the rule Lin inspired by the multiple-conclusion rule for Gödel-Dummett Logic DUM presented in [6, 7]. DUM is semantically characterized by intuitionistic linearly ordered Kripke models; the multi-conclusion rule for DUM simultaneously treats a set of implicative formulas while Lin simultaneously treats a set of modal formulas. We remark, that the number of conclusions of rule Lin depends on the number of formulas in $\mathcal{B}$; if $\mathcal{B}$ is empty, Lin has $\mathcal{A}$ as only conclusion.

The rules of $\mathbf{T}$ are sound is the sense that, if the premise of a rule is satisfiable then one of its conclusions is satisfiable. We briefly discuss, by means

$$\frac{\mathcal{S}, \Box A}{\mathcal{S}[\top/A], \Box A}\text{R-}\Box \qquad \frac{\mathcal{S}, \Box\neg A}{\mathcal{S}[\bot/A], \Box\neg A}\text{R-}\Box\neg$$

$$\frac{\mathcal{S}, A}{\mathcal{S}\{\top/A\}, A}\text{R-}cl \qquad \frac{\mathcal{S}, \neg A}{\mathcal{S}\{\bot/A\}, \neg A}\text{R-}cl\neg$$

$$\frac{\mathcal{S}}{\mathcal{S}[\top/p]} \preceq^{+} \text{ if } p\preceq^{+}\mathcal{S}$$

$$\frac{\mathcal{S}}{\mathcal{S}[\bot/p]} \preceq^{-} \text{ if } p\preceq^{-}\mathcal{S}$$

For for $l \in \{+, -\}$, $p\preceq^{l}\mathcal{S}$ iff $p\preceq^{l} H$ for every $H \in \mathcal{S}$ where, $p\preceq^{l} H$ is defined as follows:

- $p\preceq^{+} p$ and $p\preceq^{-} \neg p$ and $p\preceq^{l} H$, if $H \in (\mathcal{V} \setminus \{p\}) \cup \{\top, \bot\}$;
- $p\preceq^{l} (A \vee B)$ iff $p\preceq^{l} A$ and $p\preceq^{l} B$;
- $p\preceq^{l} (A\mathbf{U}B)$ iff $p\preceq^{l} A$ and p does not occur in B;
- $p\preceq^{l} \neg(A\mathbf{U}B)$ iff $p\preceq^{l} B$ and p does not occur in A;
- if $A \neq B\mathbf{U}C$, then $p\preceq^{+} \neg A$ iff $p\preceq^{-} A$ and $p\preceq^{-} \neg A$ iff $p\preceq^{+} A$;
- $p\preceq^{l} \circ A$ iff $p\preceq^{l} A$;

Fig. 2. Optimization rules for PLTL

of an example, the soundness of rule Lin. The application of rule Lin to $\circ\mathcal{B} = \{\circ(A_1\mathbf{U}B_1), \circ\neg(A_2\mathbf{U}B_2)\}$ generates as conclusions the sets:

$$\mathcal{C} = \{A_1, \neg B_2\} \cup \circ\mathcal{B}\ ,\ \mathcal{H}_1 = \{B_1, \neg(A_2\mathbf{U}B_2)\}\ ,\ \mathcal{H}_2 = \{\circ(A_1\mathbf{U}B_1), A_1, \neg A_2, \neg B_2\}.$$

Let us assume that $\alpha \Vdash \circ\mathcal{B}$; we show that at least one of the conclusions is satisfiable. We have $\alpha' \Vdash A_1\mathbf{U}B_1$ and $\alpha' \Vdash \neg(A_2\mathbf{U}B_2)$. Note that:

- $\alpha' \Vdash A_1\mathbf{U}B_1 \ \Rightarrow\ \exists\beta_1 \geq \alpha' : \beta_1 \Vdash B_1$ and $\forall\gamma : \alpha' \leq \gamma < \beta_1, \gamma \Vdash A_1$.
- $\alpha' \Vdash \neg(A_2\mathbf{U}B_2) \ \Rightarrow\ \begin{cases} \text{(i) } \forall\gamma \geq \alpha', \gamma \nVdash B_2 \text{ or} \\ \text{(ii) } \exists\beta_2 \geq \alpha' : \beta_2 \nVdash A_2 \text{ and } \forall\gamma : \alpha' \leq \gamma \leq \beta_2, \gamma \nVdash B_2 \end{cases}$

If (i) holds either $\alpha' < \beta_1$ and $\alpha' \Vdash \mathcal{C}$, or $\alpha' = \beta_1$ and $\alpha' \Vdash \mathcal{H}_1$. Now, let us suppose that (ii) holds; then:

- if $\alpha' < \beta_1$ and $\alpha' < \beta_2$, then $\alpha' \Vdash \mathcal{C}$;
- if $\alpha' = \beta_1$, then $\alpha' \Vdash \mathcal{H}_1$;
- if $\alpha' < \beta_1$ and $\alpha' = \beta_2$, then $\alpha' \Vdash \mathcal{H}_2$.

The notions of *proof-table* and *branch* are defined as usual. A set $\mathcal{S}$ of formulas is *closed* if it either contains $\bot$ or it contains a formula A and its negation. Branches of a proof-table are generated alternating *saturation phases*, where rules different from Lin are applied as long as possible, and applications of rule Lin. We remark that, at the end of a saturation phase, we get a set of formulas which only contains literals, $\top$, $\bot$ and formulas prefixed with $\circ$. If during the saturation phase a closed set is generated the construction of the branch is aborted. During branch construction loops can be generated, hence a loop-checking mechanism is needed to assure termination. A *loop* is a sequence of consecutive sets of formulas $\mathcal{S}_1, \ldots, \mathcal{S}_n$ in a branch such that $\mathcal{S}_1 = \mathcal{S}_n$ and $\mathcal{S}_i \neq \mathcal{S}_{i+1}$ for every $1 \leq i < n$. Whenever, during a branch construction, a loop is detected the branch construction is aborted. A loop is *closed* if there exist $i \in \{1, \ldots, n-1\}$ and $A\mathbf{U}B \in \mathcal{S}_i$ ($\neg(A\mathbf{U}B) \in \mathcal{S}_i$, respectively) such that $B \notin \mathcal{S}_j$ ($\{\neg A, \neg B\} \not\subseteq \mathcal{S}_j$, respectively) for every $1 \leq j < n$. A loop is *open* if it is not closed. A branch is *closed* if it contains a closed set of formulas or a closed loop and *open* otherwise. The proof of the completeness theorem for **T** is based on a procedure extracting a PLTL-model satisfying $\mathcal{S}$ from an open branch starting with $\mathcal{S}$.

Although multi-conclusion rules as Lin can generate a huge number of branches, as discussed in [7], theorem provers using these kind of rules can be effective.

To improve the performances of the above procedure we exploit the optimization rules depicted in Fig. 2 which are inspired by the rules presented in [11, 4]. In rules R-$\Box$ and R-$\Box\neg$ (R stands for Replacement), $\mathcal{S}[B/A]$ denotes formula substitution, that is the set of formulas obtained by replacing every occurrence of A in S with B. In rules R-*cl* and R-*cl*$\neg$, $\mathcal{S}\{B/A\}$ denotes *partial* formula substitution, that is the set of formulas obtained by replacing every occurrence of A in $\mathcal{S}$ which is not under the scope of a modal connective with B. As for rules $\preceq^+$ and $\preceq^-$, they can be applied if the propositional variable p has *constant polarity in* $\mathcal{S}$ ($p \preceq^l \mathcal{S}$). We remark that rules $\preceq^+$ and $\preceq^-$ are the PLTL version of the rules **T**-permanence and **T**$\neg$-permanence of [4].

All the rules of Fig. 2 have the property that the premise is satisfiable iff the conclusion is. In the proof search procedure we apply the optimization rules and the usual boolean simplification rules [11, 4] at every step of a saturation phase.

3 Implementations and performances

To perform some experiments on the benchmark formulas for PLTL, we have implemented β, a theorem prover written in Prolog[4]. At present β is a very simple prototype that implements **T** and the rules in Fig. 2. On the third column of the table in Fig. 4 we report the performances of β. For every family of formulas in the benchmark we indicate the number of formulas of the family solved within one minute. All tests were conducted on a machine with a 2.7GHz Intel Core i7 CPU with 8GB memory. All the optimizations rules we have described are effective in speeding-up the deduction. Indeed, without the described optimizations, timings of β are some order of magnitude greater and almost no formula is decided within one minute. In the fourth column of Fig. 4 we report the figures related to `PLTL`, an OCaml prover based on the one-pass tableau calculus of [14]. Although in general `PLTL` outperforms β, there are families where our prototype is more efficient than `PLTL` and this is encouraging for further research.

To conclude, we have presented our ongoing research on automated deduction for PLTL. In this note we have discussed a new proof-theoretical characterization of PLTL based on a multiple-conclusion rule and some optimization rules useful to cut the size of the proofs. As regards the future work, we aim to apply to the case of PLTL other optimizations introduced in the context of Intuitionistic logic as the permanence rules of [4] and the optimizations exploiting the notions of *local formula* [3] and *evaluation* [5].

[4] Available at `http://www2.disco.unimib.it/fiorino/beta.tgz`

Family	Status	β	PLTL
lift	sat.	76	42
anzu-amba	sat.	18	38
acacia-demo-v3	sat.	12	12
anzu-genbuf	sat.	28	26
rozier counters	sat.	35	65

Family	Status	β	PLTL
lift	unsat.	0	8
schuppan-O1	unsat.	27	10
schuppan-O2	unsat.	7	10
schuppan-phltl	unsat.	5	3

Fig. 3. Comparison between β and PLTL

References

1. A. Bolotov, O. Grigoriev, and V. Shangin. Automated natural deduction for propositional linear-time temporal logic. In *TIME (2007)*, pages 47–58. IEEE Computer Society, 2007.
2. K. Brünnler and M. Lange. Cut-free sequent systems for temporal logic. *Journal of Logic and Algebraic Programming*, 76(2):216–225, 2008.
3. M. Ferrari, C. Fiorentini, and G. Fiorino. fCube: An efficient prover for intuitionistic propositional logic. In C. G. Fermüller et al., editor, *LPAR-17*, volume 6397 of *LNCS*, pages 294–301. Springer, 2010.
4. M. Ferrari, C. Fiorentini, and G. Fiorino. Simplification rules for intuitionistic propositional tableaux. *ACM Transactions on Computational Logic (TOCL)*, 13(2):14:1–14:23, 2012.
5. M. Ferrari, C. Fiorentini, and G. Fiorino. An evaluation-driven decision procedure for G3i. *ACM Transactions on Computational Logic (TOCL)*, 6(1):8:1–8:37, 2015.
6. G. Fiorino. Tableau calculus based on a multiple premise rule. *Information Sciences*, 180(19):371–399, 2010.
7. G. Fiorino. Refutation in Dummett logic using a sign to express the truth at the next possible world. In T. Walsh, editor, *IJCAI 2011*, pages 869–874. IJCAI/AAAI, 2011.
8. M. Fisher, C. Dixon, and M. Peim. Clausal temporal resolution. *ACM Transactions on Computational Logic (TOCL)*, 2(1):12–56, 2001.
9. J. Gaintzarain, M. Hermo, P. Lucio, and M. Navarro. Systematic semantic tableaux for PLTL. *Electronic Notes in Theoretical Computer Science*, 206:59–73, 2008.
10. J. Gaintzarain, M. Hermo, P. Lucio, M. Navarro, and F. Orejas. Dual systems of tableaux and sequents for PLTL. *Journal of Logic and Algebraic Programming*, 78(8):701–722, 2009.
11. F. Massacci. Simplification: A general constraint propagation technique for propositional and modal tableaux. In Harrie de Swart, editor, *TABLEAUX'98*, volume 1397 of *LNCS*, pages 217–232. Springer-Verlag, 1998.
12. B. Paech. Gentzen-systems for propositional temporal logics. In E. Börger et al., editor, *CSL'88*, volume 385 of *LNCS*, pages 240–253. Springer, 1988.
13. R. Pliuskevicius. Investigation of finitary calculus for a discrete linear time logic by means of infinitary calculus. In J. Barzdins et al., editor, *Baltic Computer Science*, volume 502 of *LNCS*, pages 504–528. Springer, 1991.
14. S. Schwendimann. A new one-pass tableau calculus for PLTL. In H. C. M. de Swart, editor, *TABLEAUX'98*, volume 1397 of *LNCS*, pages 277–291. Springer, 1998.
15. M. Suda and C. Weidenbach. Labelled superposition for PLTL. In N. Bjørner et al., editor, *LPAR-18*, volume 7180 of *LNCS*, pages 391–405. Springer, 2012.