Resolution-Based Model Construction for PLTL

Michel Ludwig and Ullrich Hustadt
Department of Computer Science
University of Liverpool, United Kingdom
{*Michel.Ludwig, U.Hustadt*}*@liverpool.ac.uk*

***Abstract*—With tableaux-based reasoning approaches or model checking techniques for propositional linear-time temporal logics, PLTL, it is easily possible to construct counter examples for formulae that are not valid. In contrast, only the information that a formula is satisfiable is usually available in resolution-based inference systems. In this paper we present a resolution-based approach for constructing models for satisfiable PLTL formulae. Our approach is based on using the standard model construction for sets of propositional clauses saturated under ordered resolution in the different time points of a temporal model. The temporal model construction procedure is also designed in such a way that it can be easily implemented in existing theorem provers for PLTL.**

***Keywords*-Propositional Linear-Time Temporal Logic; Resolution; Automated Model Construction**

I. Introduction

Temporal Logics are a powerful notational framework for specifying computational systems and associated properties in the area of formal verification. The field of formal verification is concerned with verifying that a specified system behaves correctly in all situations. In particular, propositional temporal logics have been successfully applied to the verification of reactive or concurrent systems [1] and to verification via model-checking [2].

In this paper we focus on temporal reasoning through clausal resolution-based methods. More specifically, we consider propositional linear-time temporal logic (PLTL) with finite past and infinite future. A clausal resolution calculus for this logic has been introduced in [3] and implemented, for example, in the theorem prover TSPASS [4]. Another type of proof methods for PLTL are, for instance, tableaux-based approaches [5] and an implementation of a one-pass tableau calculus [6] exists in the Logics Workbench [7]. In order to prove the validity of a formula φ both proof methods operate on the negated formula $\neg\varphi$. In the case of tableaux reasoning one essentially tries to construct a model for the formula $\neg\varphi$. If no model can be found, then one can conclude that the formula $\neg\varphi$ is unsatisfiable, which is equivalent to φ being valid. For resolution-based proof methods on the other hand the proof goal consists in deriving a contradiction from the formula $\neg\varphi$, from which one can conclude again that φ is valid.

It is therefore easy to see that formal verification by using tableaux-based systems bears the advantage that in case of a failure to prove the validity of a specific property a counter example demonstrating the erroneous behaviour has already been constructed. For clausal resolution-based reasoning a set of clauses on which every application of an inference rule will only derive redundant clauses, a so-called saturated set, will have typically been constructed in that case. If the empty clause is not contained in this saturated set, one can conclude that the formula $\neg\varphi$ is satisfiable, which implies that φ is not valid. Thus, only the knowledge that the specification does not satisfy the required property is generally available for clausal resolution-based verification.

A way of constructing a model satisfying a saturated set (under ordered resolution) both for propositional and first-order logic has been devised in [8]. The model construction algorithm involves ordering the clauses by using an extension of the ordering on propositional symbols that has been used in the saturation of the clause set. One positive (maximal) literal is then satisfied per clause, whenever necessary, starting from the smallest clause w.r.t. the considered ordering. A term model, or so-called Herbrand model, representing the satisfied literals will be constructed in this way.

In this paper we present a method that allows to construct a model for a satisfiable PLTL formula. Our approach is based on analysing the saturated clause set that has been computed under ordered fine-grained resolution with selection, which is in fact a sound and complete calculus for monodic first-order temporal logic. A temporal model is then obtained by constructing models for sets of (non-temporal) propositional clauses at the different time points. The sets of clauses considered for the individual points in the time line will be constructed dynamically during the model construction process by taking those clauses into account that allow to express constraints among different time points. The whole model construction procedure is designed in such a way that it can be easily incorporated into existing resolution-based theorem provers for PLTL.

The paper is organised as follows. In Section II we briefly define the variant of propositional temporal logic we are considering, whereas Section III recalls the calculus of ordered fine-grained resolution. We then describe the propositional model construction procedure in Section IV, and in Section V we introduce the resolution-based temporal model construction algorithm for PLTL and prove its correctness. In Section VI we consider practical aspects of the algorithm

and its complexity. We conclude with a brief overview of its implementation in the theorem prover TSPASS and present some experimental results in Section VII.

II. Propositional Linear Time Temporal Logic

The language of Propositional Linear Time Temporal Logic, PLTL, is an extension of classical propositional logic by temporal operators for a discrete linear model of time (i.e. isomorphic to $\mathbb{N}$). The signature of PLTL is composed of a countably infinite set of *propositional symbols* p, q, p_0, p_1, ..., the *propositional operators* $\top$, $\neg$, $\vee$, and the *temporal operators* $\Box$ ('always in the future'), $\Diamond$ ('eventually in the future'), $\bigcirc$ ('at the next moment'), U ('until') and W ('weak until') (see e.g. [9]). We also use $\bot$ (**false**), $\wedge$, and $\Rightarrow$ as additional operators, defined using $\top$ (**true**), $\neg$, and $\vee$ in the usual way. The set of PLTL formulae is defined as follows: $\top$ is a PLTL formula; any propositional symbol P is an *atomic* PLTL formula or *atom*; if φ and ψ are PLTL formulae, then so are $\neg\varphi$, $\varphi \vee \psi$, $\Box\varphi$, $\Diamond\varphi$, $\bigcirc\varphi$, $\varphi \,\mathsf{U}\, \psi$, and $\varphi \,\mathsf{W}\, \psi$. As usual, a *literal* is either an atomic formula or its negation. A propositional clause is a set of literals.

Formulae of this logic are interpreted over temporal structures $\mathfrak{M} = (D_n)_{n\in\mathbb{N}}$ that associate with each element n of $\mathbb{N}$, representing a moment in time, a propositional model (or valuation) D_n given by a set of propositional symbols. The definition of the *truth relation* $\mathfrak{M}_n \models \varphi$ is as follows:

$\mathfrak{M}_n \models \top$
$\mathfrak{M}_n \models p$ iff $p \in D_n$
$\mathfrak{M}_n \models \neg\varphi$ iff not $\mathfrak{M}_n \models \varphi$
$\mathfrak{M}_n \models \varphi \vee \psi$ iff $\mathfrak{M}_n \models \varphi$ or $\mathfrak{M}_n \models \psi$
$\mathfrak{M}_n \models \bigcirc\varphi$ iff $\mathfrak{M}_{n+1} \models \varphi$
$\mathfrak{M}_n \models \Diamond\varphi$ iff there exists $m \geq n$ such that $\mathfrak{M}_m \models \varphi$
$\mathfrak{M}_n \models \Box\varphi$ iff for all $m \geq n$, $\mathfrak{M}_m \models \varphi$
$\mathfrak{M}_n \models \varphi \,\mathsf{U}\, \psi$ iff there exists a $m \geq n$ such that $\mathfrak{M}_m \models \psi$ and $\mathfrak{M}_i \models \varphi$ for every $i, n \leq i < m$
$\mathfrak{M}_n \models \varphi \,\mathsf{W}\, \psi$ iff $\mathfrak{M}_n \models \varphi \,\mathsf{U}\, \psi$ or $\mathfrak{M}_n \models \Box\varphi$

A temporal structure $\mathfrak{M} = (D_n)_{n\in\mathbb{N}}$ is said to be a *model* for a formula φ if and only if it holds that $\mathfrak{M}_0 \models \varphi$. A formula is *satisfiable* if and only there exists a model for φ. A formula φ is *valid* if and only if every temporal structure $\mathfrak{M} = (D_n)_{n\in\mathbb{N}}$ is a model for φ.

We say that a set of formulae $\mathcal{F}$ *entails* a formula ψ, written $\mathcal{F} \vDash \psi$, if and only if every temporal structure $\mathfrak{M}$ that is a model for every formula $\varphi \in \mathcal{F}$ is a model for ψ (analogously for sets of propositional clauses).

Every PLTL formula can be transformed into an equi-satisfiable normal form, called *divided separated clausal normal form (DSCNF)*.

Definition 1. *A* propositional temporal problem P in divided separated clausal normal form (DSCNF) *is a quadruple* $\langle \mathcal{U}, \mathcal{I}, \mathcal{S}, \mathcal{E} \rangle$*, where*

(i) the universal part $\mathcal{U}$ and the initial part $\mathcal{I}$ are finite sets of propositional clauses;
(ii) the step part $\mathcal{S}$ is a finite set of clauses of the form $p \Rightarrow \bigcirc q$, where p is a propositional symbol and q is a propositional literal; and
(iii) the eventuality part $\mathcal{E}$ is a finite set of formulae of the form $\Diamond l$ (an eventuality *clause), where l is a propositional literal.*

We associate with each propositional temporal problem $\mathsf{P} = \langle \mathcal{U}, \mathcal{I}, \mathcal{S}, \mathcal{E} \rangle$ the PLTL formula $\mathcal{I} \wedge \Box\mathcal{U} \wedge \Box\mathcal{S} \wedge \Box\mathcal{E}$. When we talk about particular properties of a temporal problem (e.g., satisfiability, validity, logical consequences, etc.) we refer to properties of this associated formula.

The transformation to DSCNF is based on a renaming and unwinding technique which substitutes non-atomic subformulae by new propositional symbols and their definitions, and replaces temporal operators by their fixed point definitions as described, for example, in [3].

Theorem 1. *Any formula in propositional linear-time temporal logic can be transformed into an equi-satisfiable propositional temporal problem in DSCNF with at most a linear increase in the size of the problem.*

Proof: Follows from [10], Theorem 3.4. ∎

The main purpose of the divided separated clausal normal form is to cleanly separate different temporal aspects of a PLTL formula from each other. One has to note that step clauses of the form $C \Rightarrow \bigcirc D$, where C is a conjunction of propositional symbols and D a disjunction of propositional literals, can be derived by the calculus introduced in Section III.

In this paper we assume that propositional temporal problems in DSCNF contain at most one single eventuality. This is not a restrictive assumption as every propositional problem can be transformed in such a way that it contains at most one eventuality up to a linear increase in the size of the problem (see [11], Lemma 7).

Let $\{p_1 \Rightarrow \bigcirc q_1, \ldots, p_n \Rightarrow \bigcirc q_n\}$ be a set of step clauses in P. Then $(\bigwedge_{i=1}^{n} p_i) \Rightarrow \bigcirc(\bigwedge_{i=1}^{n} q_i)$ is called a *merged step clause* built from P.

In what follows, $\mathcal{A} \Rightarrow \bigcirc\mathcal{B}$ and $\mathcal{A}_i \Rightarrow \bigcirc\mathcal{B}_i$ denote merged step clauses, and $\mathcal{U}$ denotes the (current) universal part of a propositional temporal problem P.

In the next section we recall the propositional version of the ordered fine-grained resolution with selection calculus first presented in [12]. As the clauses we are considering are actually sets of literals instead of multisets, we do not have to introduce factoring rules.

III. Ordered Fine-Grained Resolution with Selection

We assume that we are given an *admissible ordering* $\succ$, that is, a strict partial ordering on propositional symbols that is well-founded and total, and a *selection function* S which maps any propositional clause C to a (possibly empty)

subset of its negative literals. The ordering $\succ$ is extended to literals by $\neg A \succ A$ and $(\neg)A \succ (\neg)B$ if and only if $A \succ B$. A literal L is called (strictly) maximal w.r.t. a clause C if and only if there is no literal $L' \in C$ with $L' \succ L$ ($L' \succeq L$). A literal L is *eligible* in a clause $L \vee C$ if either it is selected in $L \vee C$, or otherwise no literal is selected in C and L is maximal w.r.t. C. The admissible ordering $\succ$ and the selection function S are used to restrict the applicability of the deduction rules of fine-grained resolution as follows.

(i) *Ordered resolution with selection between two universal clauses*

$$\frac{C_1 \vee A \quad \neg A \vee C_2}{C_1 \vee C_2}$$

if A *is eligible in* $(C_1 \vee A)$*, and* $\neg A$ *is eligible in* $(\neg A \vee C_2)$. The result is a universal clause.

(ii) *Ordered resolution with selection between an initial and a universal clause and between two initial clauses*. These are defined in analogy to the two deduction rules above with the only difference that the result is an initial clause.

(iii) *Ordered fine-grained step resolution with selection*.

$$\frac{C_1 \Rightarrow \bigcirc(D_1 \vee A) \quad C_2 \Rightarrow \bigcirc(D_2 \vee \neg A)}{(C_1 \wedge C_2) \Rightarrow \bigcirc(D_1 \vee D_2)}$$

where $C_1 \Rightarrow \bigcirc(D_1 \vee A)$ and $C_2 \Rightarrow \bigcirc(D_2 \vee \neg A)$ are step clauses, A *is eligible in* $(D_1 \vee A)$*, and* $\neg A$ *is eligible in* $(D_2 \vee \neg A)$.

$$\frac{C_1 \Rightarrow \bigcirc(D_1 \vee A) \quad D_2 \vee \neg A}{C_1 \Rightarrow \bigcirc(D_1 \vee D_2)}$$

where $C_1 \Rightarrow \bigcirc(D_1 \vee A)$ is a step clause, $D_2 \vee \neg A$ is a universal clause, A *is eligible in* $(D_1 \vee A)$*, and* $\neg A$ *is eligible in* $(D_2 \vee \neg A)$.

$$\frac{D_1 \vee A \quad C_2 \Rightarrow \bigcirc(D_2 \vee \neg A)}{C_2 \Rightarrow \bigcirc(D_1 \vee D_2)}$$

where $D_1 \vee A$ is a universal clause, $C_2 \Rightarrow \bigcirc(D_2 \vee \neg A)$ is a step clause, A *is eligible in* $(D_1 \vee A)$*, and* $\neg A$ *is eligible in* $(D_2 \vee \neg A)$.

(iv) *Clause conversion*. A step clause of the form $C \Rightarrow \bigcirc\bot$ is rewritten to the universal clause $\neg C$.

(v) *Eventuality resolution rule w.r.t.* $\mathcal{U}$.

$$\frac{\mathcal{A}_1 \Rightarrow \bigcirc\mathcal{B}_1 \quad \cdots \quad \mathcal{A}_n \Rightarrow \bigcirc\mathcal{B}_n \qquad \Diamond l}{\bigwedge_{i=1}^{n} \neg\mathcal{A}_i} \ (\Diamond^{\mathcal{U}}_{res})$$

where $\mathcal{A}_i \Rightarrow \bigcirc\mathcal{B}_i$ are merged step clauses such that for every i, $1 \leq i \leq n$, the *loop* side conditions $\mathcal{U} \wedge \mathcal{B}_i \models \neg l$ and $\mathcal{U} \wedge \mathcal{B}_i \models \bigvee_{j=1}^{n} \mathcal{A}_j$ are valid. (In the case $\mathcal{U} \models \neg l$, the *degenerate clause*, $\top \Rightarrow \bigcirc\top$, can be considered as a premise of this rule, and the conclusion of the rule is then $\neg\top$.)

The set of full merged step clauses, satisfying the loop side conditions, is called a *loop in* $\Diamond l$ and the formula $\bigvee_{j=1}^{n} \mathcal{A}_j$ is called a *loop formula*.

Rules i to iii, also called rules of *fine-grained step resolution*, are either identical or closely related to the deduction rules of ordered propositional resolution with selection.

In contrast, rule v is much more complex, as it requires not just one or two premises, but an indeterminate (though finite) number of complex combinations of step clauses, which have to satisfy certain conditions. To find premises suitable for an application of the eventuality resolution rule we use a particular algorithm, called FG-BFS (for fine-grained breadth-first search), which conducts a so-called *loop search* (see e.g. [10] for more details). The algorithm internally uses the deduction rules i to iii and returns a loop formula $H = \bigvee_{j=1}^{n} A_j$, which allows to directly add $\neg H$ to the universal part of a temporal problem as the result of applying the eventuality resolution rule.

Let *ordered fine-grained resolution with selection* be the calculus consisting of the rules i to iv above, together with the eventuality resolution rule v. We denote this calculus by $\mathfrak{I}^{S,\succ}_{FG}$. The calculus can be extended by redundancy elimination rules, like for example, the deletion of subsumed clauses.

Definition 2 (Derivation)**.** *A (linear)* derivation D *(in* $\mathfrak{I}^{S,\succ}_{FG}$*) from a temporal problem* P *in DSCNF is a sequence of tuples*

$$D = \langle \mathcal{U}_1, \mathcal{I}_1, \mathcal{S}_1, \mathcal{E} \rangle, \langle \mathcal{U}_2, \mathcal{I}_2, \mathcal{S}_2, \mathcal{E} \rangle, \ldots$$

such that each tuple $\langle \mathcal{U}_{i+1}, \mathcal{I}_{i+1}, \mathcal{S}_{i+1}, \mathcal{E} \rangle$ *is obtained from* $\langle \mathcal{U}_i, \mathcal{I}_i, \mathcal{S}_i, \mathcal{E} \rangle$ *by adding the conclusion of an application of one of the inference rules of* $\mathfrak{I}^{S,\succ}_{FG}$ *to premises from one of the sets* $\mathcal{U}_i$*,* $\mathcal{I}_i$*,* $\mathcal{S}_i$ *to that set, with the other sets as well as* $\mathcal{E}$ *remaining unchanged*[1].

A derivation D *such that the empty clause is an element of a* $\mathcal{U}_i \cup \mathcal{I}_i$ *is called a (*$\mathfrak{I}^{S,\succ}_{FG}$*-)*refutation *of* $\langle \mathcal{U}_1, \mathcal{I}_1, \mathcal{S}_1, \mathcal{E} \rangle$.

A derivation D *is* fair *if and only if for each clause* C *which can be derived from premises in*

$$\langle \bigcup_{i \geq 1} \mathcal{U}_i, \bigcup_{i \geq 1} \mathcal{I}_i, \bigcup_{i \geq 1} \mathcal{S}_i, \mathcal{E} \rangle$$

there exists an index j *such that* C *occurs in* $\langle \mathcal{U}_j, \mathcal{I}_j, \mathcal{S}_j, \mathcal{E} \rangle$.

Ordered fine-grained resolution with selection is sound and complete for propositional temporal problems as stated in the following theorem.

Theorem 2 (see [12], Theorem 5)**.** *Let* P *be propositional temporal problem in DSCNF. Let* $\succ$ *be an admissible ordering and* S *a selection function. Then* P *is unsatisfiable iff there exists a* $\mathfrak{I}^{S,\succ}_{FG}$*-refutation of* P*. Moreover,* P *is unsatisfiable iff any fair* $\mathfrak{I}^{S,\succ}_{FG}$*-derivation is a refutation of* P*.*

IV. Propositional Model Construction

In this section we briefly recall the model construction procedure for satisfiable sets of (non-temporal) propositional

[1] In an application of the eventuality resolution rule, the set $\mathcal{U}$ in the definition of the rule refers to $\mathcal{U}_i$.

clauses as it was introduced in [8]. This model construction procedure uses an admissible ordering on propositional symbols again, which is then extended on propositional clauses as its (multi)set extension. The model is constructed by considering which literals have to be satisfied in a given clause, starting from the smallest clause w.r.t. the clause ordering.

Definition 3 (Propositional Model Construction)**.** *Let $\succ$ be an admissible ordering and S be a selection function. Additionally, let N be a set of propositional clauses.*

For a propositional clause $C \in N$ we inductively define a propositional model $I_{\succ,S}(C)$ and a set ε_C as follows.

Let $C \in N$ be a propositional clause. Then, we define $I_{\succ,S}(C) = \bigcup_{C \succ D} \varepsilon_D$, and if the clause C

(i) is of the form $C' \vee A$, where A is the maximal literal in C,
(ii) is false in $I_{\succ,S}(C)$, and
(iii) if no negative literal is selected in C,

we define $\varepsilon_C = \{A\}$; otherwise we set $\varepsilon_C = \emptyset$. Finally, we define $I_{\succ,S}(N) = \bigcup_{C \in N} \varepsilon_C$.

In line with the definition of the semantics for PLTL given in Section II propositional symbols not listed in a propositional model $I_{\succ,S}(N)$ will be set to '**false**'[2] in the model.

It can be shown that for an arbitrary admissible ordering, an arbitrary selection function and for an arbitrary saturated set of propositional clauses (w.r.t. to the given ordering) which does not contain the empty clause, the propositional model construction indeed constructs a model.

Theorem 3 (see [8], Theorem 3.16)**.** *Let $\succ$ be an admissible ordering and S be a selection function. Moreover, let N be a set of propositional clauses that is saturated under inferences by the rules of ordered (propositional) resolution with selection and let N not contain the empty clause. Then it holds that $I_{\succ,S}(N) \models N$.*

V. Temporal Model Construction

For a temporal problem $\mathsf{P} = \langle \mathcal{U}, \mathcal{I}, \mathcal{S}, \mathcal{E} \rangle$ the temporal model construction is based on using the regular propositional model construction for the different time points of a temporal model. For the initial time point 0 the regular propositional model construction will be performed over the set of universal clauses together with the set of initial clauses. For time points different from the initial point in time, the (merged) step clauses $C \Rightarrow \bigcirc D$ whose left-hand sides C were fulfilled at the previous moment in time have to be considered in addition to the set of universal clauses.

If the temporal problem P contains a single eventuality, i.e. $\mathcal{E} = \{\Diamond l\}$, special care has to be taken for allowing it to be satisfied infinitely often. We add the eventuality to the set of clauses used for the model construction in a specific time point if the newly-added eventuality unit clause does not lead to a contradiction. As a result, the constructed model will satisfy the eventuality in every time point in which the set of universal clauses and the right-hand sides of the step clauses whose left-hand sides were fulfilled at the previous time point do not imply the negated eventuality. Consequently, the only 'critical' merged step clauses $\mathcal{A} \Rightarrow \bigcirc \mathcal{B}$ are those with $\mathcal{U} \cup \{\mathcal{B}\} \models \neg l$ and $\mathcal{U} \not\models \neg \mathcal{A}$. In particular one has to avoid that the left-hand side of one of these 'critical' merged step clauses is constantly fulfilled from any given time point onwards. One way of ensuring this requirement consists in varying the ordering on propositional symbols that is used to construct the models for the different time points, which is also the approach that is taken in this paper.

For example, if we were to construct a temporal model as described above for the temporal problem $\mathsf{P}' = \langle \{p \vee q\}, \emptyset, \{p \Rightarrow \bigcirc \neg l\}, \{\Diamond l\} \rangle$, we have to ensure that the propositional symbol p is not satisfied at every time point as otherwise we would obtain the sequence of propositional models $\{p, l\}, \{p\}, \{p\}, \ldots$. The constructed sequence would obviously not satisfy the formula $\Box \Diamond l$.

In the next subsection we describe the model construction procedure in a formal way and give an example for the construction of a model, while we prove the correctness of the procedure in the subsequent subsection.

A. Construction Principle

Before we can introduce the model construction procedure, we still need to give a couple of auxiliary definitions.

First of all, for a temporal problem P we associate with every set of merged step clauses $\mathcal{C}$ (and with the power set $\mathscr{P}(\mathcal{C})$) a set $\mathcal{O}_\mathcal{C}$ of strict total orderings on $\mathrm{Symbols}(\mathsf{P})$.

Definition 4. *Let P be a propositional temporal problem in DSCNF and let $\mathcal{C} = \{\mathcal{A}_1 \Rightarrow \bigcirc \mathcal{B}_1, \ldots, \mathcal{A}_n \Rightarrow \bigcirc \mathcal{B}_n\}$ be a set of merged step clauses built from the temporal problem P, where $\mathcal{A}_i = \bigwedge_{j=1}^{m_i} a_j^i$ for $1 \leq i \leq n$ and $a_1^i, \ldots, a_{m_i}^i$ are propositional symbols for $1 \leq i \leq n$.*

We define $\mathcal{O}_\mathcal{C}$ to be the smallest set of admissible orderings on $\mathrm{Symbols}(\mathsf{P})$ which contains for every tuple $(i_1, \ldots, i_n) \in \{1, \ldots, m_1\} \times \ldots \times \{1, \ldots, m_n\}$ exactly one ordering $\succ \in \mathcal{O}_\mathcal{C}$ with $\mathrm{Symbols}(\mathsf{P}) \setminus \{a_{i_1}^1, \ldots, a_{i_n}^n\} \succ a_{i_1}^1, \ldots, a_{i_n}^n$.

For the power set $\mathscr{P}(\mathcal{C})$ of $\mathcal{C}$ we define that $\mathcal{O}_{\mathscr{P}(\mathcal{C})} = \bigcup_{S \in \mathscr{P}(\mathcal{C})} \mathcal{O}_S$, where $\mathcal{O}_\emptyset = \emptyset$.

The next definition introduces the set $R^S(\mathfrak{M})$ which contains the right-hand sides of step clauses whose left-hand sides are *triggered* by a propositional model $\mathfrak{M}$.

Definition 5. *Let $\mathsf{P} = \langle \mathcal{U}, \mathcal{I}, \mathcal{S}, \mathcal{E} \rangle$ be a propositional temporal problem such that $\mathcal{E} = \emptyset$ or $\mathcal{E} = \{\Diamond l\}$. Additionally, let S be a set of step clauses built from P and $\mathfrak{M}$ be a*

[2]More specifically, the terminology of "don't care" literals from SAT solvers does not apply here.

propositional model over $\mathrm{Symbols}(\mathsf{P})$*. Then we define:*

$$R^S(\mathfrak{M}) = \{\, l_1 \vee \ldots \vee l_m \mid (p_1 \wedge \ldots \wedge p_m) \Rightarrow \bigcirc (l_1 \vee \ldots \vee l_m) \in S \text{ and } \mathfrak{M} \models p_1 \wedge \ldots \wedge p_m \,\}$$

Next we define the set $L^{\mathcal{E}}(N)$, which adds to the set N the unit clause l if $\mathcal{E} = \{\Diamond l\}$ and $N \not\models \neg l$.

Definition 6. *Let* $\mathsf{P} = \langle \mathcal{U}, \mathcal{I}, \mathcal{S}, \mathcal{E} \rangle$ *be a propositional temporal problem such that* $\mathcal{E} = \emptyset$ *or* $\mathcal{E} = \{\Diamond l\}$*. Furthermore, let* N *be a set of propositional clauses over* $\mathrm{Symbols}(\mathsf{P})$*. Then we define:*

$$L^{\mathcal{E}}(N) = \begin{cases} N \cup \{l\} & \text{if } \mathcal{E} = \{\Diamond l\} \text{ and } N \not\models \neg l \\ N & \text{otherwise} \end{cases}$$

Finally, for a set of propositional clauses N we denote by $\mathrm{Res}_{\succ,S}(N)$ the set of all the clauses obtained by an application of the ordered resolution rule using the ordering $\succ$ to premises in N and the selection function S. We also define that $\mathrm{Res}^{\infty}_{\succ,S}(N) = \bigcup_{i \in \mathbb{N}} \mathrm{Res}^{i}_{\succ,S}(N)$, where $\mathrm{Res}^{0}_{\succ,S}(N) = N$.

We can now give the definition of the temporal model construction procedure.

Definition 7 (Temporal Model Construction)**.** *Let* $\mathsf{P} = \langle \mathcal{U}, \mathcal{I}, \mathcal{S}, \mathcal{E} \rangle$ *be a propositional temporal problem in DSCNF such that* $\bot \notin \mathcal{U} \cup \mathcal{I}$*, and* $\mathcal{E} = \emptyset$ *or* $\mathcal{E} = \{\Diamond l\}$*. Additionally, let* S *be a selection function, and if* $\mathcal{E} = \{\Diamond l\}$*, let* $\mathcal{C} = \{\mathcal{A}_1 \Rightarrow \bigcirc \mathcal{B}_1, \ldots, \mathcal{A}_n \Rightarrow \bigcirc \mathcal{B}_n\}$ *be the set of all the merged step clauses built from the temporal problem* P *such that for every* i*,* $1 \leq i \leq n$*:*

(i) $\mathcal{U} \cup \{\mathcal{B}_i\} \models \neg l$*, and*
(ii) $\mathcal{U} \not\models \neg \mathcal{A}_i$*, and*
(iii) for every $\mathcal{A} \Rightarrow \bigcirc \mathcal{B}$ *with* $\mathcal{A} \subsetneq \mathcal{A}_i$ *(and* $\mathcal{B} \subsetneq \mathcal{B}_i$*) it holds that* $\mathcal{U} \cup \{\mathcal{B}\} \not\models \neg l$*.*

The merged step clauses from the set $\mathcal{C}$ *will also be called* critical *merged step clauses for the temporal problem* P*.*

We then define a sequence of propositional models $H_0, H_1, \ldots$ *as follows:*

$$H_0 = I_{\succ_0,S}(\mathrm{Res}^{\infty}_{\succ_0,S}(L^{\mathcal{E}}(\mathcal{U} \cup \mathcal{I})))$$

and for $i \geq 1$*:*

$$H_i = I_{\succ_i,S}(\mathrm{Res}^{\infty}_{\succ_i,S}(L^{\mathcal{E}}(\mathcal{U} \cup R^S(H_{i-1}))))$$

where $\succ_i$ *(*$i \in \mathbb{N}$*) are admissible orderings on* $\mathrm{Symbols}(\mathsf{P})$ *such that for every* H_j*,* $j \geq 1$ *with* $H_j \models \bigvee_{k=1}^{n} \mathcal{A}_k$*, which occurs infinitely often,*

$$\mathcal{O}_{\mathscr{P}(\mathcal{C})} \subseteq \{\, \succ_{t+1} \mid t \geq j \text{ and } H_t = H_j \,\}.$$

Additionally, for every H_j*,* $j \geq 1$ *with* $H_j \not\models \bigvee_{k=1}^{n} \mathcal{A}_k$ *we have* $\succ_{j+1} = \succ_0$*.*

Let $\mathcal{H} = (H_0, H_1, \ldots)$ *denote the temporal model obtained in this way.*

First of all, one can observe that the process of ensuring that every ordering is used infinitely often corresponds to the notion of *fairness*, which is employed in the field of model checking [13].

Then, as explained above, the sets of initial and universal clauses are considered for the model construction in the time point 0. Additionally, the eventuality is added to the clause set used for model construction if its presence does not lead to a contradiction. The regular model construction is then performed through an initial ordering $\succ_0$ on $\mathrm{Symbols}(\mathsf{P})$ after the model construction clause set has been saturated under regular ordered resolution with selection using the ordering $\succ_0$. This saturation process is necessary in order to ensure the correctness of the propositional model construction.

For any time point other than the initial point of the time line, the universal clauses together with the right-hand side of any step clause whose left-hand was satisfied at the previous time point are used for the propositional model construction. Again, the eventuality is added to the considered set if does not lead to a contradiction. It is now important to note that the ordering on propositional symbols under which the propositional resolution and model construction is performed has to be varied for the temporal model construction to succeed. The variation of the orderings on propositional symbols ensures that a propositional model is found for a time point which does not trigger the left-hand side of any critical step clause,

For example, for the temporal problem $\mathsf{P}' = \langle \{p \vee q\}, \emptyset, \{p \Rightarrow \bigcirc \neg l\}, \{\Diamond l\} \rangle$ again, we cannot use the ordering $l \succ p \succ q$ at every time point as it would not lead to a correct temporal model. We have to use an ordering $\succ'$ with $q \succ' p$ at some time points instead.

It is important to note that in general different choices of orderings can lead to different models, which can greatly vary in size. However, it is not possible to construct every model of a temporal problem P though the model construction method introduced in this paper. For example, the model $\{p\}, \{p\}, \ldots$ cannot be obtained for the temporal problem $\langle \{\neg p \vee \neg q\}, \emptyset, \emptyset, \emptyset \rangle$.

We conclude this section by applying the temporal model construction procedure on a concrete example. We consider the temporal problem $\mathsf{P}'' = \langle \{p \vee q\}, \{p\}, \{p \Rightarrow \bigcirc q, q \Rightarrow \bigcirc p\}, \{\Diamond \neg p\} \rangle$. Saturating the problem P'' under ordered fine-grained resolution (with an empty selection function) using the ordering $p \succ q$ derives the universal clause $\neg p \vee \neg q$ (through loop search), the initial clause $\neg q$, and the step clause $q \Rightarrow \bigcirc \neg q$. The step clause $q \Rightarrow \bigcirc p$ is a critical step clause for the set of universal clauses as $\{p \vee q, \neg p \vee \neg q, p\} \models \neg \neg p$.

For the initial time point we hence consider the set of propositional clauses $\{\neg q, p, p \vee q, \neg p \vee \neg q\}$ for the propositional model construction procedure. With the symbol ordering $p \succ q$, we obtain the model $H_0 = \{p\}$.

Then, as the step clause $p \Rightarrow \bigcirc q$ has been triggered at the initial time point, we have to add the unit clause q to the considered clause set. As $\{q, p \vee q, \neg p \vee \neg q\} \not\models \neg \neg p$, we add the unit clause $\neg p$ and obtain the set $\{q, p \vee q, \neg p, \neg p \vee \neg q\}$, which is to be used for the propositional model construction. After saturation with the ordering $p \succ q$, the standard

propositional model construction yields the propositional model $H_1 = \{q\}$ in the time point 1.

Finally, as the step clauses $q \Rightarrow \bigcirc p$, $q \Rightarrow \bigcirc \neg q$ have been triggered in time point 1, the unit clauses p and $\neg q$ have to be added to the clause set used for the propositional model construction. Additionally, as the set $\{\neg q, p, p \vee q, \neg p, \neg p \vee \neg q\}$ is unsatisfiable, the set $\{\neg q, p, p \vee q, \neg p \vee \neg q\}$ has be to be considered for the propositional model construction, which results in the model $H_2 = \{p\}$ with the ordering $p \succ q$.

As $H_0 = H_2$ the temporal model construction procedure will now construct models for the remaining time points analogously to ones shown above.

B. Correctness

In this section we prove the correctness of the construction procedure introduced in Definition 7, i.e. we show that the constructed sequence of propositional models is indeed a model for the considered temporal problem. We only state the required lemmata and theorems; the full proofs can be found in [14]. First of all, we introduce three lemmata that will be required for the subsequent correctness theorem.

Lemma 4. *Let N be a set of propositional clauses over a set of propositional symbols $\mathcal{P}$ such that every clause contains at least one negative literal. Let $\succ$ be an admissible ordering on the propositional symbols $\mathcal{P}$ and S be a selection function.*

Then it holds that $I_{\succ,S}(N) = \emptyset$.

Lemma 5. *Let N be a satisfiable set of propositional clauses. Moreover, let $a_1, \ldots, a_n$ be propositional symbols and let $\succ$ be an admissible ordering on propositional symbols such that* $\mathrm{Symbols(N)} \setminus \{a_1, \ldots, a_n\} \succ a_1, \ldots, a_n$. *Finally, let S be a selection function. Then it holds that:*

$$I_{\succ,S}(\mathrm{Res}^{\infty}_{\succ,S}(N)) \models a_1 \vee \ldots \vee a_n \;\; \textit{iff} \;\; N \models a_1 \vee \ldots \vee a_n$$

Lemma 6. *Let P be a propositional temporal problem and let N be a satisfiable set of propositional clauses which only uses propositional symbols from P. Additionally, let $\mathcal{C} = \{\mathcal{A}_1 \Rightarrow \bigcirc\mathcal{B}_1, \ldots, \mathcal{A}_n \Rightarrow \bigcirc\mathcal{B}_n\}$ be a set of merged step clause built from the temporal problem P, and let S be a selection function. Then it holds that:*

$$N \models \textstyle\bigvee_{i=1}^{n} \mathcal{A}_i \textit{ iff } \forall \succ \in \mathcal{O}_{\mathcal{C}} \colon I_{\succ,S}(\mathrm{Res}^{\infty}_{\succ,S}(N)) \models \bigvee_{i=1}^{n} \mathcal{A}_i$$

We can now state the correctness theorem for the model construction procedure.

Theorem 7. *Let $\mathsf{P} = \langle \mathcal{U}, \mathcal{I}, \mathcal{S}, \mathcal{E} \rangle$ be a propositional temporal problem with $\mathcal{E} = \emptyset$ or $\mathcal{E} = \{\Diamond l\}$ which is saturated under ordered fine-grained resolution with selection and does not contain the empty clause. Additionally, let $\mathcal{H}$ be the corresponding sequence of propositional models obtained through temporal model construction. Then it holds that:*

$$\mathcal{H}_0 \models \mathcal{I} \wedge \Box\mathcal{U} \wedge \Box\mathcal{S} \wedge \Box\mathcal{E}$$

VI. Practical Considerations and Complexity

The temporal model construction as described in the previous section constructs an infinite sequence of propositional models, as suggested by the definition of the semantics for PLTL given in Section II. However, for practical applications, a finite representation of a temporal structure, as given by an ultimately periodic model is more useful.

Definition 8 (Ultimately Periodic Model)**.** *Let $\mathsf{P} = \langle \mathcal{U}, \mathcal{I}, \mathcal{S}, \mathcal{E} \rangle$ be a propositional temporal problem such that either $\mathcal{E} = \emptyset$ or $\mathcal{E} = \{\Diamond l\}$, and let $\mathcal{H} = (H_0, H_1, H_2, \ldots)$ be the an infinite sequence of propositional models over* $\mathrm{Symbols}(\mathsf{P})$. *Furthermore, let $I, J, L \in \mathbb{N}$ be indices such that $I \leq L < J$, $H_I = H_J$ and $H_L \models l$ if $\mathcal{E} = \{\Diamond l\}$, $I = L$ otherwise.*

We then define a sequence of propositional models $\mathcal{H}' = (H'_0, H'_1, \ldots)$ as follows:

(i) $H'_i = H_i$ *for every* $0 \leq i \leq J$

(ii) $H'_i = H_{I+(i-I) \bmod (J-I)}$ *for every* $i \geq J+1$

It can be shown that if the sequence $\mathcal{H}$ is a model for P, then the sequence $\mathcal{H}'$ is also a model for P [15].

More concretely, in an implementation of the temporal model construction procedure one has to keep track of the ordering that has been used for the saturations used in the different time points. Whenever a previously considered set of clauses is encountered again, the symbol ordering used for the model construction in the considered time point has to be changed cyclically. Finally, the construction procedure can terminate whenever a previously encountered valuation has been computed again and the possibly present eventuality has been satisfied in between those two time points.

Moreover, it easy to see that for a set $\mathcal{C} = \{\mathcal{A}_1 \Rightarrow \bigcirc\mathcal{B}_1, \ldots, \mathcal{A}_n \Rightarrow \bigcirc\mathcal{B}_n\}$ of critical merged step clauses for a temporal problem P the set $\mathcal{O}_{\mathscr{P}(\mathcal{C})}$ can be constructed from $\mathscr{P}(\cup_{i=1}^{n}\mathrm{Symbols}(\mathcal{A}_\mathrm{i}))$, the power set of all the propositional symbols occurring in left-hand sides of critical step clauses. Every ordering $\succ \in \mathcal{O}_{\mathscr{P}(\mathcal{C})}$ is characterised by the subset $\mathcal{P} \subseteq \cup_{i=1}^{n}\mathrm{Symbols}(\mathcal{A}_\mathrm{i})$ such that $\mathrm{Symbols}(\mathsf{P}) \setminus \mathcal{P} \succ p$ for every $p \in \mathcal{P}$. Thus, in an implementation it is sufficient to consider all the subsets of $\cup_{i=1}^{n}\mathrm{Symbols}(\mathcal{A}_\mathrm{i})$ in order to construct the required orderings.

Furthermore, it is also possible to eliminate redundant cycles in constructed temporal models. For example, if one has built a model for a temporal problem P with a single eventuality $\Diamond l$ and the constructed model contains a sequence of valuations $H_i, H_{i+1}, \ldots, H_j$ such that $H_i = H_j$ and $H_k \not\models l$ for every $i \leq k \leq j$, then the sequence $H_i, \ldots, H_{j-1}$ can be removed from the final model as it does not contribute to satisfying the eventuality.

It is important to note that the model construction procedure is completely deterministic, that is, neither the basic building blocks given by $I_{\succ,S}$ and $\mathrm{Res}^{\infty}_{\succ,S}$, nor the construction of the sequence of propositional models that form

	Time				LWB 'Model' Length	TSPASS Model Properties		
	LWB 'Model'	TSPASS	TSPASS Model Construction			Model Length	Constructed Time Points	Critical Merged Step Clauses
$\mathcal{C}^1_{ran}$ (n=5)	0.06s	0.03s	0.07s	(0.01s)	7	2 (1)	2	0
$\mathcal{C}^2_{ran}$ (n=5)	0.06s	0.06s	0.63s	(0.03s)	2	2 (1)	2	0
$\mathcal{C}^1_{ran}$ (n=12)	1.5s	0.04s	0.89s	(0.57s)	39	14 (7)	16	17
$\mathcal{C}^2_{ran}$ (n=12)	0.06s	1.13s	53.14s	(0.47s)	2	2 (1)	2	0

Table I
MEDIAN RESULTS (PER CLASS) FOR THE TSPASS AND LWB MODEL CONSTRUCTION PROCEDURES APPLIED ON THE BENCHMARK CLASSES

the ultimately periodic model involves any non-deterministic operation that in an implementation would force us to use a form of backtracking-search to find a model. On the other hand, just as standard tableaux-based model generation procedures for PLTL, there is no guarantee that we will produce a minimal, that is, shortest possible, ultimately periodic model for a temporal problem or PLTL formula.

The computational complexity of the temporal model construction procedure is determined mainly by the time required to compute the saturation $\mathrm{Res}^{\infty}_{\prec,S}(N)$ of a set N of clauses under ordered resolution, which is exponential in the size of N, the size of the $\mathrm{Res}^{\infty}_{\prec,S}(N)$, which is also exponential in the size of N, and the maximal length of the sequence of propositional models in an ultimately periodic model $\mathcal{H}'$ for a satisfiable temporal problem $\mathsf{P} = \langle \mathcal{U}, \mathcal{I}, \mathcal{S}, \mathcal{E} \rangle$, which is again exponential in the size of P. Overall, we obtain the following result.

Theorem 8. *Let $\mathsf{P} = \langle \mathcal{U}, \mathcal{I}, \mathcal{S}, \mathcal{E} \rangle$ be a satisfiable propositional temporal problem with $\mathcal{E} = \emptyset$ or $\mathcal{E} = \{\Diamond l\}$. Then an ultimately periodic model $\mathcal{H}$ for P can be constructed by the temporal model construction procedure in time exponential in the size of P.*

Since for a given PLTL formula φ an equi-satisfiable propositional problem P in DSCNF can be computed in polynomial time and space, this result also implies the we can construct an ultimately periodic model for φ in time exponential in the size of φ.

It is important to remember that while the satisfiability problem of PLTL is PSPACE-complete, given that ultimately periodic models can be of exponential size in the worst case, we cannot hope for a model construction procedure of better complexity.

VII. IMPLEMENTATION

The temporal model construction has been implemented as an extension of the theorem prover TSPASS[3] [4], which is a fair theorem prover for monodic first-order temporal logic based on ordered fine-grained resolution with selection. It is important to note that while the temporal problem is saturated by TSPASS, the critical merged step clauses for the considered temporal problem are also computed as part of the overall loop search process. Consequently, no further computation is required to obtain these step clauses.

We have compared the resolution-based model construction implemented in TSPASS 0.92-0.16 with the one-pass tableau calculus described in [6], which is implemented in the Logics Workbench (LWB) version 1.1 [7]. We have applied both systems to all the satisfiable PLTL formulae in the benchmark classes introduced in [16]. Two of the benchmark classes, $\mathcal{C}^1_{ran}$ ($n = 5$) and $\mathcal{C}^1_{ran}$ ($n = 12$), where n is the number of propositional symbols over which the formulae are constructed, are designed in such a way that they can be theoretically solved easily by resolution-based decision procedures, whereas the two other benchmark classes, $\mathcal{C}^2_{ran}$ ($n = 5$) and $\mathcal{C}^2_{ran}$ ($n = 12$), are designed so that the satisfiable formulae in them can be theoretically solved more easily by tableaux-based systems. In particular, in [16] the implementation of the one-pass tableau calculus in the LWB was indeed performing best on these formulae.

The experiments were run on a PC equipped with an Intel Core 2 E6400 CPU and 3 GB of main memory and an execution timeout of 5 minutes was imposed on each problem.

For TSPASS an empty selection function was used. The scheduling of the orderings on propositional symbols was done w.r.t. an increasing subset size and in such a way that a maximal number of different orderings were tried out: whenever a set of propositional clauses was encountered for the first time the next possible ordering was used for the propositional model construction instead of starting again from the initial ordering w.r.t. the critical symbols. For the $\mathcal{C}^1_{ran}$ classes TSPASS was instructed to perform matching replacement resolution and formulae $\bigcirc(\phi) \wedge \bigcirc(\psi)$ were rewritten to $\bigcirc(\phi \wedge \psi)$ in order to reduce the number of required renamings.

The median results for all the satisfiable formulae of each class are shown in Table I, with time values in the table being the average CPU time of three identical runs. We can observe that the number of generated clauses and the execution times increase for the model construction run of TSPASS, which is due to the transformation to single-eventuality problems and, as a result, an increased number of step clauses. Such a

[3] http://www.csc.liv.ac.uk/~michel/software/tspass/

transformation is not performed if no model construction is required. Additionally, one could observe that the time spent in the transformation to DSCNF is negligible. The numbers in brackets in the model construction time column indicate the amount of time actually spent on model construction w.r.t. the global execution time, and the numbers in brackets in the model length column represent the length of the periodic part. Finally, the median total number of constructed time points during the model construction in TSPASS is reproduced in the second last column, some of which are discarded during the elimination of redundant cycles.

The class $\mathcal{C}^1_{ran}$ (n=5) contains 2400 formulae in total of which 1217 formulae are satisfiable. On these satisfiable formulae, the model construction of TSPASS could solve all the problems, whereas the 'Model' function of the LWB did not finish on 26 problems within the given time limit. The class $\mathcal{C}^2_{ran}$ (n=5) contains 1400 formulae in total of which 955 are satisfiable. All the models constructed by TSPASS and the LWB for this class were at most of length 2. No timeouts were incurred either in the TSPASS or the LWB run. The class $\mathcal{C}^1_{ran}$ (n=12), then, contains 4000 formulae in total of which 2264 are satisfiable. The model construction of TSPASS did not finish on 30 satisfiable formulae in this class, whereas the 'Model' function of the LWB did not terminate within the given time limit on 284 of the satisfiable formulae. Finally, the class $\mathcal{C}^2_{ran}$ (n=12) contains 1900 formulae in total of which 1184 formulae are satisfiable. Again, no timeouts were incurred in either the TSPASS or LWB run, and all the models constructed by TSPASS and the LWB for this class were at most of length 2

As one might expect, the Logics Workbench can maintain its execution time advantage on $\mathcal{C}^2_{ran}$ ($n = 5$) and $\mathcal{C}^2_{ran}$ ($n = 12$). On the other hand, the model construction of TSPASS proves quite successful on $\mathcal{C}^1_{ran}$ ($n = 5$) and $\mathcal{C}^1_{ran}$ ($n = 12$), computing even a smaller median model length than the LWB.

VIII. Conclusion

We have presented a procedure for constructing models for satisfiable PLTL formula. The procedure is based on computing saturations under ordered fine-grained resolution with selection while using the standard model construction for propositional clauses to construct models for the different time points. It is important to observe that the temporal model construction procedure is not based on performing a search with backtracking but the construction is guaranteed to succeed once the appropriate symbol orderings have been considered, and that it can always produce finite, ultimately periodic models. We have proved the correctness of the model construction algorithm, analysed some of its practical aspects, and briefly introduced our implementation of the algorithm.

It is easily possible to extend the model construction method presented in this paper to CTL formulae, but an extension to first-order temporal logic will require greater efforts. In future work we intend to address the problem of reducing the number of renamings necessary for the transformation to single-eventuality problems and also try to construct even shorter, ideally minimal, ultimately periodic models for PLTL formulae.

References

[1] A. Pnueli, "The temporal logic of programs," in *Proc. FOCS'77*. IEEE Computer Society, 1977, pp. 46–57.

[2] E. Clarke, O. Grumberg, and D. A. Peled, *Model Checking*. MIT Press, 1999.

[3] M. Fisher, C. Dixon, and M. Peim, "Clausal temporal resolution," *ACM Transactions on Computational Logic*, vol. 2, no. 1, pp. 12–56, 2001.

[4] M. Ludwig and U. Hustadt, "Implementing a fair monodic temporal logic prover," *AI Communications*, To appear.

[5] P. Wolper, "Temporal logic can be more expressive," *Information and Control*, vol. 56, no. 1/2, pp. 72–99, 1983.

[6] S. Schwendimann, "A new one-pass tableau calculus for PLTL," in *Proc. TABLEAUX'98*, ser. LNCS, vol. 1397. Springer, 1998, pp. 277–292.

[7] A. Heuerding, G. Jäger, S. Schwendimann, and S. Michael, "The Logics Workbench LWB: A snapshot," *Euromath Bulletin*, vol. 2, no. 1, pp. 177–186, 1996.

[8] L. Bachmair and H. Ganzinger, "Resolution theorem proving," in *Handbook of Automated Reasoning*. Elsevier, 2001, vol. 1, ch. 2, pp. 19–99.

[9] E. A. Emerson, "Temporal and modal logic," in *Handbook of Theoretical Computer Science*. Elsevier, 1990, pp. 995–1072.

[10] A. Degtyarev, M. Fisher, and B. Konev, "Monodic temporal resolution," *ACM Transactions On Computational Logic*, vol. 7, no. 1, pp. 108–150, 2006.

[11] A. Degtyarev, M. Fisher, and B. Konev, "A simplified clausal resolution procedure for propositional linear-time temporal logic," in *Proc. TABLEAUX'2002*, ser. LNCS, vol. 2381. Springer, 2002, pp. 85–99.

[12] U. Hustadt, B. Konev, and R. A. Schmidt, "Deciding monodic fragments by temporal resolution," in *Proc. CADE-20*, ser. LNAI, vol. 3632. Springer, 2005, pp. 204–218.

[13] N. Francez, *Fairness*. New York, USA: Springer, 1986.

[14] M. Ludwig and U. Hustadt, "Resolution-based model construction for PLTL (Extended Version)," Dep. of Comp. Sci., Univ. of Liverpool, Tech. Rep. ULCS-09-008, 2009.

[15] A. P. Sistla and E. M. Clarke, "The complexity of propositional linear temporal logics," *J. ACM*, vol. 32, no. 3, pp. 733–749, 1985.

[16] U. Hustadt and R. A. Schmidt, "Scientific benchmarking with temporal logic decision procedures," in *Proc. KR'02*. Morgan Kaufmann, 2002, pp. 533–546.