Resolution-Based Model Construction for PLTL (Extended Version)

Michel Ludwig and Ullrich Hustadt[1]
Department of Computer Science, University of Liverpool, UK
{Michel.Ludwig, U.Hustadt}@liverpool.ac.uk

10th November 2009

Technical Report ULCS-09-008
Department of Computer Science
University of Liverpool

[1]Supported by
EPSRC Grant EP/D060451/1

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.

1 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 [12] 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 [6] and implemented, for example, in the theorem prover TSPASS [11]. Another type of proof methods for PLTL are, for instance, tableaux-based approaches [15] and an implementation of a one-pass tableau calculus [13] 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 [1]. 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 2 we briefly define the variant of propositional temporal logic we are considering, while Section 3 recalls the calculus of ordered fine-grained resolution. We then describe the propositional model construction procedure in Section 4, while in Section 5 we introduce the resolution-based temporal model con-

struction algorithm for PLTL and prove its correctness. In Section 6 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 7.

2 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. [5]). 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:

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

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 [6].

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 [4], 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.

In this paper we assume that propositional temporal problems in DSCNF contain at most one single eventuality. This is not a limiting 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 [3], 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 [8]. A version of the calculus without ordering restrictions and selection functions was introduced first in [10]. As the clauses we are considering are actually sets of literals instead of multisets, we do not have to introduce factoring rules.

3 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, 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. [4] 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, clause subsumption.

The version of the calculus without ordering restrictions and selection functions will be called *fine-grained resolution* and be denoted by $\mathfrak{I}_{FG}$.

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* $\mathfrak{I}_{FG}^{S,\succ}$ *to premises from one of the sets* $\mathcal{U}_i$, $\mathcal{I}_i$, $\mathcal{S}_i$ *to exactly one of the sets* $\mathcal{U}_i$, $\mathcal{I}_i$, $\mathcal{S}_i$, *with the other sets as well as* $\mathcal{E}$ *remaining unchanged*[2].

A derivation D *such that the empty clause is an element of a* $\mathcal{U}_i \cup \mathcal{I}_i$ *is called a* ($\mathfrak{I}_{FG}^{S,\succ}$-) 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 [8], 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* $\mathfrak{I}_{FG}^{S,\succ}$*-refutation of* P*. Moreover,* P *is unsatisfiable iff any fair* $\mathfrak{I}_{FG}^{S,\succ}$*-derivation is a refutation of* P*.*

4 Propositional Model Construction

In this section we briefly recall the model construction procedure for satisfiable sets of (non-temporal) propositional clauses as it was introduced in [1]. 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* $I_{\succ,S}(N) = \bigcup_{C \in N} \varepsilon_C$.

If it is clear from the context which selection function S *we are referring to, we also denote* $I_{\succ,S}(N)$ *by* $I_{\succ}(N)$.

A clause C is said to be *productive* and said to *produce the atom* A if and only if $\varepsilon_C = \{A\}$. 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.

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

Theorem 3 (see [1], 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 $I_{\succ,S}(N) \models N$.*

5 Temporal Model Construction

Before we define the model construction formally, we present two examples that should illustrate the basic ideas behind the model construction procedure. Let us first consider the construction of a temporal model $\mathcal{M}_1$ for the following satisfiable temporal problem P_1:

$$\mathsf{P}_1 = \langle \{d \vee e\}, \{a\}, \{a \Rightarrow \bigcirc b, b \Rightarrow \bigcirc c, c \Rightarrow \bigcirc a\}, \emptyset \rangle.$$

We observe that P_1 does not contain an eventuality and that it is already saturated under ordered fine-grained resolution w.r.t. any admissible ordering (and selection function). Additionally, for $\mathcal{M}_1$ to be a model of P_1, $\mathcal{M}_1$ has to fulfil the initial (unit) clause a and the universal clause $d \vee e$ at the initial point in time. Thus, if we apply the standard propositional model construction on the propositional clause set $\{a, d \vee e\}$ with an ordering $\succ$ given by $a \succ b \succ c \succ d \succ e$, we obtain the propositional model $H_0 = \{a, d\}$. Then, for constructing the propositional model in the time point 1 we have to consider the universal clause $d \vee e$ again together with the right-hand sides of those (merged) step clauses whose left-hand sides were triggered at the initial time point. In this case only the step clause $a \Rightarrow \bigcirc b$ was triggered by the model H_0. Consequently, we construct a propositional model for the clause set $\{d \vee e, b\}$ by using the ordering $\succ$ and obtain $H_1 = \{b, d\}$. Similarly, we can build the propositional model $H_2 = \{c, d\}$ for the time point 2. Now, we have to consider the clause set $\{d \vee e, a\}$ again for the time point 3, which results in the propositional model $H_3 = \{a, d\} = H_0$ through the standard propositional model construction with the ordering $\succ$. Hence, we can see that $\mathcal{M}_1 = (H_0, H_1, H_2, H_3, H_0, H_1, H_2, H_3, H_0, \dots)$ is a temporal model for P_1.

In the previous example one single ordering on propositional symbols was sufficient for constructing a temporal model. But as we will see in the following example, it can be necessary to change the ordering used for the propositional model construction. Let us consider the construction of a temporal model $\mathcal{M}_2$ for the following satisfiable temporal problem P_2:

$$\mathsf{P}_2 = \langle \{\neg f \vee a, a \vee p, \neg f \vee b, \neg d \vee \neg l \vee e, f \vee g\}, \{a\}, \{a \Rightarrow \bigcirc \neg l, b \Rightarrow \bigcirc d, c \Rightarrow \bigcirc \neg e\}, \{\Diamond l\} \rangle.$$

Here, the saturation of P_2 under ordered fine-grained resolution w.r.t. the ordering $\succ_0$ given by $a \succ_0 b \succ_0 c \succ_0 d \succ_0 e \succ_0 f \succ_0 g \succ_0 l \succ_0 p$ (and an empty selection function) will derive the merged step clauses $b \Rightarrow \bigcirc(\neg l \vee e)$ and $(b \wedge c) \Rightarrow \bigcirc \neg l$. There is no loop formula derivable from the problem P_2. We can see that the two (merged) step clauses $a \Rightarrow \bigcirc \neg l$ and $(b \wedge c) \Rightarrow \bigcirc \neg l$ imply the negation of the eventuality literal l at the next time point whenever their left-hand sides are fulfilled at the currently considered point of the time line. Now, if one wants to construct a model for a temporal problem that contains exactly one eventuality, then one has to ensure that the eventuality is satisfied infinitely often. The approach that we take in this paper consists in fulfilling the eventuality at a given time point whenever the clauses that have to be considered for this point in the time line do *not* imply the negated eventuality. In this way we can add the eventuality unit clause l to the clause set and saturate the enlarged clause set under propositional ordered resolution without deriving the empty clause.

Now, for the temporal problem P_2 we have to consider the clause set $\{f \vee g, \neg d \vee \neg l \vee e, \neg f \vee b, a, a \vee p, \neg f \vee a\}$ for the initial time point. As this clause set does not imply

the negated eventuality $\neg l$, we add the unit clause l and obtain the propositional model $H_0 = \{l, f, b, a\}$ by using the ordering $\succ_0$. Then, as the model H_0 triggers the merged step clauses $a \Rightarrow \bigcirc\neg l$, $b \Rightarrow \bigcirc d$ and $b \Rightarrow \bigcirc(e \vee \neg l)$, we have to additionally consider their right-hand sides for the propositional model construction in the time point 1, i.e. the clauses $\neg l$, d and $e \vee \neg l$. Consequently, as the clause set $\{\neg l, f \vee g, e \vee \neg l, d, \neg d \vee \neg l \vee e, \neg f \vee b, a \vee p, \neg f \vee a\}$ implies the negated eventuality $\neg l$, we do not add the unit clause l to the clause set. The propositional model construction with the ordering $\succ_0$ yields the model $H_1 = \{f, d, b, a\}$.

We can see that the model H_1 again triggers the left-hand side of the step clause $a \Rightarrow \bigcirc\neg l$. Additionally, due to the universal clause $a \vee p$, the ordering $\succ_0$ will enforce that the symbol a is fulfilled (and thus l cannot be satisfied at the next time point) whenever the propositional model construction is performed with the ordering $\succ_0$ (the symbol a does not occur negatively in the temporal problem). Thus, if we want the temporal model construction to succeed we have to use a different ordering for constructing propositional models in some points of the time line. As the model H_1 also triggers the step clauses $b \Rightarrow \bigcirc d$ and $b \Rightarrow (e \vee \neg l)$, we have to consider the clause set $\{d, \neg f \vee a, \neg f \vee b, f \vee g, \neg l, e \vee \neg l, \neg d \vee \neg l \vee e, a \vee p\}$ for the time point 2. If we now use the ordering $\succ_1$ given by $p \succ_1 l \succ_1 g \succ_1 f \succ_1 e \succ_1 d \succ_1 c \succ_1 b \succ_1 a$ for the propositional model construction, we first of all observe that the set is already saturated under ordered propositional resolution w.r.t. the ordering $\succ_0$. We hence obtain the model $H_2 = \{p, g, d\}$.

Finally, as H_2 does not trigger any of the step clauses, we only have to consider the clause set $\{l, f \vee g, \neg d \vee \neg l \vee e, \neg f \vee b, a \vee p, \neg f \vee a\}$, which contains the eventuality unit clause l, for the propositional model construction. By using the ordering $\succ_0$ again we obtain the model $H_3 = \{l, f, b, a\} = H_0$. We can thus conclude that $\mathcal{M}_2 = (H_0, H_1, H_2, H_3, H_0, H_1, H_2, H_3, H_0, \dots)$ is a temporal model for P_2.

As illustrated by these examples, the temporal model construction for a temporal problem $\mathsf{P} = \langle \mathcal{U}, \mathcal{I}, \mathcal{S}, \mathcal{E} \rangle$ 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 $\mathsf{P}_3 = \langle \{p \vee q\}, \emptyset, \{p \Rightarrow \bigcirc\neg l\}, \{\Diamond l\} \rangle$, 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 $\{p, l\}, \{p\}, \{p\}, \dots$. 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.

5.1 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 Symbols(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 Symbols(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* Symbols(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 $R^\mathcal{S}(\mathfrak{M})$ which contains the right-hand sides of step clauses contained in a set $\mathcal{S}$ whose left-hand sides are *triggered* by a propositional model $\mathfrak{M}$.

Definition 5. *Let $P = \langle \mathcal{U}, \mathcal{I}, \mathcal{S}, \mathcal{E} \rangle$ be a propositional temporal problem such that $\mathcal{E} = \emptyset$ or $\mathcal{E} = \{\Diamond l\}$. Additionally, let $\mathcal{S}'$ be a set of step clauses derived by $\mathfrak{I}_{FG}^{S,\succ}$ from P and $\mathfrak{M}$ be a propositional model over* Symbols(P)*. Then we define:*

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

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

Definition 6. *Let $P = \langle \mathcal{U}, \mathcal{I}, \mathcal{S}, \mathcal{E} \rangle$ 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* Symbols(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 with selection 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 $P = \langle \mathcal{U}, \mathcal{I}, \mathcal{S}, \mathcal{E} \rangle$ 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 and freed of duplicate propositional symbols such that for every i, $1 \leq i \leq n$:*

(i) $\mathcal{U} \cup \{\mathcal{B}_i\} \models \neg l$*, and*

(ii) $\mathcal{U} \not\models \neg\mathcal{A}_i$*.*

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 $i \geq 1$:

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

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

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

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.

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 Symbols(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 guarantee the correctness of the propositional model construction.

Then, 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 it 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.

For example, for the temporal problem $\mathsf{P}_3 = \langle \{p \vee q\}, \emptyset, \{p \Rightarrow \bigcirc \neg l\}, \{\Diamond l\} \rangle$ 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.

We conclude this section by applying the temporal model construction procedure on a concrete example. We consider the temporal problem $\mathsf{P}_4 = \langle \{p \vee q\}, \{p\}, \{p \Rightarrow \bigcirc q, q \Rightarrow \bigcirc p\}, \{\Diamond \neg p\} \rangle$. Saturating the problem P_4 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 $H_2 = \{p\}$ for 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.

5.2 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. 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 such that every clause contains at least one negative literal. Let $\succ$ be an arbitrary admissible ordering on* Symbols(N) *and let S be an arbitrary selection function.*

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

Proof. We show by induction on N with respect to the well-founded (and total) multiset extension of the ordering $\succ$ on clauses that for every clause $C \in N$ it holds that $\varepsilon_C = \emptyset$.

For the minimal clauses we have $I_{\succ,S}(C) = \emptyset$. As C contains a negative literal, C is true in $I_{\succ,S}(C)$, and therefore, $\varepsilon_C = \emptyset$. The proof for remaining clauses proceeds along the same line. □

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* 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:*

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

Proof. The implication "⇐" follows from Theorem 3. For the remaining implication "⇒", we assume that $I_{\succ,S}(\mathrm{Res}^{\infty}_{\succ,S}(N)) \models a_1 \vee \ldots \vee a_n$. As clauses which contain a literal p or $\neg p$ with $p \succ a_1 \succ \ldots \succ a_n$ cannot produce an atom a_i $(1 \leq i \leq n)$ in the interpretation $I_{\succ,S}(\mathrm{Res}^{\infty}_{\succ,S}(N))$, it follows that there exist clauses $C_1, \ldots, C_m \in \mathrm{Res}^{\infty}_{\succ,S}(N)$ that only contain atom symbols from $\{a_1, \ldots, a_n\}$, i.e. for every i, $1 \leq i \leq m$, there exists an index j, $1 \leq j \leq n$, such that $C_i = C' \vee a_j$, $C' \subseteq \{\, a_k \mid 1 \leq k \leq n \,\} \cup \{\, \neg a_k \mid 1 \leq k \leq n \,\}$ and $I_{\succ,S}(\mathrm{Res}^{\infty}_{\succ,S}(N)) \models a_j$. (Note that the index j could be the same for every i, $1 \leq i \leq m$.)

Then, if we assume that every clause C_i for $1 \leq i \leq m$ contains at least one negative literal, it would follow from Lemma 4 that $I_{\succ,S}(\{C_1, \ldots, C_m\}) = \emptyset$ and thus, $I_{\succ,S}(\mathrm{Res}^{\infty}_{\succ,S}(N)) \not\models a_1 \vee \ldots \vee a_n$, which contradicts with our assumptions. Thus, there exists a clause $C_i \in \mathrm{Res}^{\infty}_{\succ,S}(N)$ $(1 \leq i \leq m)$ such that C_i is positive and $C_i \subseteq a_1 \vee \ldots \vee a_n$. We can infer that $N \models a_1 \vee \cdots \vee a_n$. □

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:*

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

Proof. The implication "⇒" is obvious. For the implication "⇐", let $\mathcal{A}_i = \bigwedge_{j=1}^{m_i} a^i_j$ for $1 \leq i \leq n$ and propositional symbols $a^i_1, \ldots, a^i_{m_i}$ for $1 \leq i \leq n$. Then we have:

$$\bigvee_{i=1}^{n} \mathcal{A}_i \equiv \bigwedge_{(i_1,\ldots,i_n) \in \{1,\ldots,m_1\} \times \ldots \times \{1,\ldots,m_n\}} (a^1_{i_1} \vee \ldots \vee a^n_{i_n})$$

Furthermore, it follows from the assumptions that:

$$\forall \succ \in \mathcal{O}_{\mathcal{C}}\, \forall (i_1, \ldots, i_n) \in \{1, \ldots, m_1\} \times \ldots \times \{1, \ldots, m_n\} \colon I_{\succ,S}(\mathrm{Res}^{\infty}_{\succ,S}(N)) \models a^1_{i_1} \vee \ldots \vee a^n_{i_n}$$

Thus, as for every tuple $(i_1, \ldots, i_n) \in \{1, \ldots, m_1\} \times \ldots \times \{1, \ldots, m_n\}$ there exists an ordering $\succ \in \mathcal{O}_{\mathcal{N}}$ with $\mathrm{Symbols}(\mathcal{N}) \setminus \{a^1_{i_1}, \ldots, a^n_{i_n}\} \succ a^1_{i_1}, \ldots, a^n_{i_n}$, we obtain from Lemma 5:

$$\forall (i_1, \ldots, i_n) \in \{1, \ldots, m_1\} \times \ldots \times \{1, \ldots, m_n\} \colon N \models a^1_{i_1} \vee \ldots \vee a^n_{i_n}$$

We can therefore conclude that $N \models \bigvee_{i=1}^{n} \mathcal{A}_i$. □

We can now state and prove 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 $\mathcal{H} = (H_0, H_1, \ldots)$ be the corresponding sequence of propositional models obtained through temporal model construction. Then it holds that:*

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

Proof. Let S be the selection function used in the saturation. Then, first of all, as the set $\mathcal{U} \cup \mathcal{I}$ does not contain the empty clause, it is easy to see that $\bot \notin \mathrm{Res}^{\infty}_{\succ_0,S}(L^{\mathcal{E}}(\mathcal{U} \cup \mathcal{I}))$. We can thus conclude that $\mathcal{H}_0 \models \mathcal{I}$ and $\mathcal{H}_0 \models \mathcal{U}$.

We now show by induction on t that $\mathcal{H}_t \models \mathcal{S}$ and $\mathcal{H}_{t+1} \models \mathcal{U}$ for every $t \in \mathbb{N}$. For $t = 0$, we already have $\mathcal{H}_0 \models \mathcal{U}$, and if we assume that $\bot \in \mathrm{Res}^{\infty}_{\succ_1,S}(L^{\mathcal{E}}(\mathcal{U} \cup R^{\mathcal{S}}(H_0)))$, then it would follow that $\bot \in \mathrm{Res}^{\infty}_{\succ_1,S}(\mathcal{U} \cup R^{\mathcal{S}}(H_0))$. Thus, as $\bot \notin \mathrm{Res}^{\infty}_{\succ_1,S}(\mathcal{U})$ there would exist a derivation of a step clause $\mathcal{A} \Rightarrow \bigcirc\bot$ with $\mathcal{H}_0 \models \mathcal{A}$. Then, as the temporal problem P is saturated, we would have $\mathcal{U} \models \neg\mathcal{A}$ and hence, $\mathcal{H}_0 \not\models \mathcal{A}$, which is a contradiction. We can infer that $\bot \notin \mathrm{Res}^{\infty}_{\succ_1,S}(L^{\mathcal{E}}(\mathcal{U} \cup R^{\mathcal{S}}(H_0)))$, $\mathcal{H}_0 \models \mathcal{S}$ and $\mathcal{H}_1 \models \mathcal{U}$.

If $t > 0$, then it follows from the induction hypothesis that $\mathcal{H}_{t-1} \models \mathcal{S}$ and $\mathcal{H}_t \models \mathcal{U}$. Again, if we assume that $\bot \in \mathrm{Res}^{\infty}_{\succ_{t+1},S}(L^{\mathcal{E}}(\mathcal{U} \cup R^{\mathcal{S}}(H_t)))$, then there would exist a derivation of a clause $\mathcal{A} \Rightarrow \bigcirc\bot$ with $\mathcal{H}_t \models \mathcal{A}$. Additionally, as the temporal problem P is saturated, we would again have $\mathcal{U} \models \neg\mathcal{A}$ and $\mathcal{H}_t \not\models \mathcal{A}$, which is a contradiction. Thus, we obtain $\bot \notin \mathrm{Res}^{\infty}_{\succ_{t+1},S}(L^{\mathcal{E}}(\mathcal{U} \cup R^{\mathcal{S}}(H_t)))$, $\mathcal{H}_t \models \mathcal{S}$ and $\mathcal{H}_{t+1} \models \mathcal{U}$.

Finally, if $\mathcal{E} = \{\Diamond l\}$, let $t \in \mathbb{N}$. We still have to show that $\mathcal{H}_t \models \Diamond l$. If we assume for all $t' \in \mathbb{N}$ with $t' \geq t$ that $\mathcal{H}_{t'} \not\models l$, then for every $t' \geq t$ with $t' \geq 1$ it holds that $\mathcal{U} \cup R^{\mathcal{S}}(H_{t'-1}) \models \neg l$. It also holds that $\mathcal{U} \not\models \neg l$ as otherwise we could apply the eventuality resolution rule and have $\bot \in \mathcal{U}$. Additionally, for every $t' \geq t$ with $t' \geq 1$ there exists a merged clause $\mathcal{A}^{t'} \Rightarrow \bigcirc\mathcal{B}^{t'} \in \mathcal{C}$ with $\mathcal{U} \cup R^{\mathcal{S}}(H_{t'-1}) = \mathcal{U} \cup \{\mathcal{B}^{t'}\}$, $\mathcal{H}_{t'-1} \models \mathcal{A}^{t'}$ and $\mathcal{U} \cup \{\mathcal{B}^{t'}\} \models \neg l$, where $\mathcal{C}$ is the set of merged step clauses from Definition 7. Then, as there are only finitely many valuations H_i ($i \in \mathbb{N}$), it follows that there exists an index $T \geq \max(t, 1)$ such that every valuation H_i with $i \geq T$ occurs infinitely often in the sequence $H_i, H_{i+1}, H_{i+2}, \ldots$. Furthermore, as there are only finitely many merged step clauses which have been freed of duplicate propositional symbols, there exist merged step clauses $\mathcal{A}'_1 \Rightarrow \bigcirc\mathcal{B}'_1, \ldots, \mathcal{A}'_m \Rightarrow \bigcirc\mathcal{B}'_m \in \mathcal{C}$ such that

$$\{\mathcal{A}'_1 \Rightarrow \bigcirc\mathcal{B}'_1, \ldots, \mathcal{A}'_m \Rightarrow \bigcirc\mathcal{B}'_m\} = \{\, \mathcal{A}^{t'} \Rightarrow \bigcirc\mathcal{B}^{t'} \mid t' \geq T \,\}.$$

By Lemma 6 it holds for every $t' \geq T$ that there exists a subset $\{i_1 \ldots, i_k\} \subseteq \{1, \ldots, m\}$ such that $\mathcal{U} \cup \{\mathcal{B}^{t'}\} \models \bigvee_{j=1}^{k} \mathcal{A}'_{i_j}$, from which we can infer that $\mathcal{U} \cup \{\mathcal{B}^{t'}\} \models \bigvee_{i=1}^{m} \mathcal{A}'_i$. Consequently, we obtain for every i with $1 \leq i \leq m$ that $\mathcal{U} \cup \{\mathcal{B}'_i\} \models \bigvee_{i=1}^{m} \mathcal{A}'_i$ and $\mathcal{U} \cup \{\mathcal{B}'_i\} \models \neg l$. We could hence apply the eventuality resolution rule and derive the set of

universal clauses $\bigwedge_{j=1}^{n} \neg \mathcal{A}'_j$. Thus, as the temporal problem P is saturated under ordered fine-grained resolution with selection, we can infer that $\mathcal{H}_{T-1} \not\models \mathcal{A}^T$ holds, which is a contradiction. □

6 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 of PLTL given in Section 2. 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 $P = \langle \mathcal{U}, \mathcal{I}, \mathcal{S}, \mathcal{E} \rangle$ be a propositional temporal problem such that either $\mathcal{E} = \emptyset$ or $\mathcal{E} = \{\Diamond l\}$, and let $\mathcal{H} = (H_0, H_1, H_2, \dots)$ be the an infinite sequence of propositional models over* Symbols(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 $\mathcal{H}' = (H'_0, H'_1, \dots)$ 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 [14].

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 $\mathcal{C} = \{\mathcal{A}_1 \Rightarrow \bigcirc\mathcal{B}_1, \dots, \mathcal{A}_n \Rightarrow \bigcirc\mathcal{B}_n\}$ of critical merged step clauses for a temporal problem P the set $\mathcal{O}_{\mathscr{P}(\mathcal{C})}$ can be constructed from $\mathscr{P}(\cup_{i=1}^{n}\text{Symbols}(\mathcal{A}_i))$, 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 $\mathcal{P} \subseteq \cup_{i=1}^{n}\text{Symbols}(\mathcal{A}_i)$ such that $\text{Symbols}(\mathsf{P}) \setminus \mathcal{P} \succ p$ for every $p \in \mathcal{P}$. Thus, in an implementation it is sufficient to consider all the subsets of $\cup_{i=1}^{n}\text{Symbols}(\mathcal{A}_i)$ 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 $H_i, H_{i+1}, \dots, H_j$ such that $H_i = H_j$ and $H_k \not\models l$ for every $i \leq k \leq j$, then the sequence $H_i, \dots, H_{j-1}$ 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 $\text{Res}^{\infty}_{\succ,S}$, nor the construction of the sequence of propositional models that form 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 for 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 $\text{Res}^{\infty}_{\succ,S}(N)$ of a set N

of clauses under ordered resolution, which is exponential in the size of N, the size of the $\mathrm{Res}^{\infty}_{\succ,S}(N)$, 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.

7 Implementation

The temporal model construction has been implemented as an extension of the theorem prover TSPASS[3] [11], 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 minimal critical merged step clauses for the considered temporal problem are also computed as part of the overall loop search process. Independently of the model construction procedure, the loop search algorithm, which is used to find premises for applications of the eventuality resolution rule, computes all critical merged step clauses for the considered temporal problem. These clauses are kept by TSPASS and are easily identifiable as critical merged step clauses (by a particular marker literal). Consequently, the model construction procedure can just extract those clauses from the saturated clause set without the need for further computation.

	Clauses Generated		Time			Nr. of Critical Merged Step Clauses	TSPASS Model Length
	TSPASS	TSPASS Model Construction	LWB 'Model'	TSPASS	TSPASS Model Construction		
$\mathcal{C}^1_{ran}$ (n=5) E1	682	8779	11.75s	0.04s	0.41s (0.08s)	0	2
$\mathcal{C}^2_{ran}$ (n=5) E2	1263	15846	0.06s	0.06s	0.73s (0.03s)	0	2
$\mathcal{C}^1_{ran}$ (n=12) E3	606	110605	151.62s	0.05s	94.68s (34.78s)	552	348
$\mathcal{C}^2_{ran}$ (n=12) E4	31445	516454	0.06s	1.17s	65.81s (0.71s)	0	2

Table 1: TSPASS and LWB model construction procedures applied on selected examples

We have compared the resolution-based model construction implemented in TSPASS with the one-pass tableau calculus described in [13], 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 [9]. 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 two of the 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 [9] the implementation of the one-pass tableau calculus in the LWB was indeed performing best on these formulae.

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

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 formula.

Results for one formula taken from each class are shown in Table 1 with time values in the table being the average CPU time of three identical runs. The first two columns show the number of clauses generated during the initial saturation of the problem before the model construction procedure is started. We can observe that the number of generated clauses and the execution times increase for the model construction run, which is due to the transformation to single-eventuality problems. The numbers in brackets in the model construction time column for TSPASS indicate the amount of time actually spent on model construction.

As one might expect, the Logics Workbench can maintain its advantage on $\mathcal{C}^2_{ran}$ $(n = 5)$ and $\mathcal{C}^2_{ran}$ $(n = 12)$. For all satisfiable formulae in these two classes, the Logics Workbench can find a model of length 2. On the other hand, the model construction of TSPASS proves quite successful on $\mathcal{C}^1_{ran}$ $(n = 5)$ and $\mathcal{C}^1_{ran}$ $(n = 12)$. The runtime of TSPASS on E3 illustrate that a high number of critical merged step clauses complicates the model construction and leads to a considerable amount of time being spend on it (34.78s). Formula E4 is an example of a formula where the model construction only takes a trivial amount of time (0.71s), but the dominating factor is the initial saturation of the temporal problem (65.81s).

Overall, model construction in TSPASS appears to be a viable option for the construction of countermodels.

8 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. We have proved the correctness of the model construction algorithm, analysed some of its practical aspects, and briefly introduced our implementation of the algorithm. In future work we intend to address the problem of constructing short, ideally minimal, ultimately periodic models for PLTL formulae.

References

[1] L. Bachmair and H. Ganzinger. Resolution theorem proving. In *Handbook of Automated Reasoning*, volume 1, chapter 2, pages 19–99. Elsevier, 2001.

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

[3] A. Degtyarev, M. Fisher, and B. Konev. A simplified clausal resolution procedure for propositional linear-time temporal logic. In *Proc. TABLEAUX'2002*, volume 2381 of *LNCS*, pages 85–99. Springer, 2002.

[4] A. Degtyarev, M. Fisher, and B. Konev. Monodic temporal resolution. *ACM Transactions On Computational Logic*, 7(1):108–150, 2006.

[5] E. A. Emerson. Temporal and modal logic. In *Handbook of Theoretical Computer Science*, pages 995–1072. Elsevier, 1990.

[6] M. Fisher, C. Dixon, and M. Peim. Clausal temporal resolution. *ACM Transactions on Computational Logic*, 2(1):12–56, 2001.

[7] A. Heuerding, G. Jäger, S. Schwendimann, and S. Michael. The Logics Workbench LWB: A snapshot. *Euromath Bulletin*, 2(1):177–186, 1996.

[8] U. Hustadt, B. Konev, and R. A. Schmidt. Deciding monodic fragments by temporal resolution. In *Proc. CADE-20*, volume 3632 of *LNAI*, pages 204–218. Springer, 2005.

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

[10] B. Konev, A. Degtyarev, C. Dixon, M. Fisher, and U. Hustadt. Towards the implementation of first-order temporal resolution: the expanding domain case. In *Proc. TIME-ICTL 2003*, pages 72–82. IEEE Computer Society, 2003.

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

[12] A. Pnueli. The temporal logic of programs. In *Proc. FOCS'77*, pages 46–57. IEEE Computer Society, 1977.

[13] S. Schwendimann. A new one-pass tableau calculus for PLTL. In *Proc. TABLEAUX'98*, volume 1397 of *LNCS*, pages 277–292. Springer, 1998.

[14] A. P. Sistla and E. M. Clarke. The complexity of propositional linear temporal logics. *J. ACM*, 32(3):733–749, 1985.

[15] P. Wolper. Temporal logic can be more expressive. *Information and Control*, 56(1/2):72–99, 1983.