A Simplified Clausal Resolution Procedure for Propositional Linear-Time Temporal Logic*

Anatoly Degtyarev, Michael Fisher, and Boris Konev**

Logic and Computation Group, Department of Computer Science
University of Liverpool, Liverpool L69 7ZF, UK
{A.Degtyarev, M.Fisher, B.Konev}@csc.liv.ac.uk

Abstract. The clausal resolution method for propositional linear-time temporal logics is well known and provides the basis for a number of temporal provers. The method is based on an intuitive clausal form, called SNF, comprising three main clause types and a small number of resolution rules. In this paper, we show how the normal form can be radically simplified and, consequently, how a simplified clausal resolution method can be defined for this important variety of logic.

1 Introduction

As computational systems become more complex, it is increasingly important to be able to *verify* that the system behaves as required. While a computational system can be tested in many ways, it is only through *formal* verification that we have the possibility of establishing the correctness of the system in *all* possible situations. However, complex systems in turn require powerful formal notations, in particular logics such as *temporal logic*. Temporal logics are extensions of classical logic, with operators that deal with time. They have been used in a wide variety of areas within Computer Science and Artificial Intelligence, for example robotics [17], databases [18], hardware verification [10] and agent based systems [16]. In particular, propositional temporal logics have already made significant impact within Computer Science, having been applied to:

– the specification and verification of distributed or concurrent systems [14];
– the synthesis of programs from temporal specifications [15, 13];
– the semantics of executable temporal logic [9];
– algorithmic verification via model-checking [10, 2]; and
– knowledge representation and reasoning [6, 1, 20].

In developing such techniques, temporal proof is often required, and we base our work on practical proof techniques on the clausal resolution approach to temporal logic.

The clausal resolution method for propositional linear-time temporal logics provides the basis for a number of temporal provers. The method is based on an intuitive clausal form, called SNF, comprising three main clause types and a small number of resolution rules [7]. While the approach has been shown to be competitive [11, 12], we

* Work supported by both EPSRC (grant GR/L87491) and the University of Liverpool (RDF).

** On leave from Steklov Institute of Mathematics at St.Petersburg

here re-address the basic form of the resolution method. In particular, we here show that the normal form can be radically simplified and, following on from this, a simplified resolution method can be defined for this important variety of temporal logic. Thus, the main benefits of the reductions described in this paper are that they produce a temporal normal form that

– provides a cleaner separation between classical and temporal reasoning,
– ensures more streamlined use of simplified temporal resolvents (without the need for further transformation),
– is simpler, involving only one (unconditional) eventuality formula, and
– since there is only one eventuality, then no heuristics/strategy is needed for choosing which temporal formula to apply temporal resolution to.

It turns out that if a given problem contains only one conditional eventuality clause, then the simplified resolution can be applied immediately without any reductions. At the same time we show the necessity to reduce conditional eventuality clauses to unconditional ones if a problem contains more than one eventuality.

We believe that all of these factors, as well as simplifying the method itself, will have significant impact upon practical temporal resolution tools.

The structure of the paper is as follows. In §2, we provide an overview of the propositional temporal logic considered and the normal form used (see [7] for further details). We then proceed to describe and analyse two key reductions:

1. from *conditional* eventuality clauses to *unconditional* eventuality clauses (§4);
2. from *multiple* unconditional eventuality clauses to a *single* unconditional eventuality clause (§7).

These reductions not only radically simplify the normal form and the resolution calculus, but initial results indicate that they can improve the speed of practical resolution systems in certain cases.

The simplified clausal resolution procedure is given in §3 and §5. The case of one eventuality is considered in §6. The results of these sections refine those given in [3]; an extension of the simplified resolution calculus to fragments of first-order temporal logic has been considered in [4, 5].

2 Preliminaries

We define the temporal logic we use based on the following symbols:

– atomic propositions $Prop = a, b, c \ldots, p, q, r \ldots$;
– Boolean operators $\neg, \wedge, \Rightarrow, \equiv, \vee$, **true** (‘true’), **false** (‘false’);
– temporal operators **start** (‘at the initial moment of time’), $\square$ (‘always in the future’), $\Diamond$ (‘at sometime in the future’), $\bigcirc$ (‘at the next moment’), $\mathcal{S}$ (‘since’, a past-time operator).

For the interpretation of the formulas in the logic, we use discrete structures $\mathfrak{M} = \langle S, I \rangle$ where $S = s_0, s_1, s_2, \ldots$ is a linearly ordered infinite sequence of states such that each state, s_i $(0 \leq i)$, represents those elements of $Prop$ which are satisfied at the i^{th} moment of time, and I is an interpretation function $Prop \rightarrow 2^S$.

Below we define a relation '$\models$', which evaluates temporal formulas at an index $i \in \mathbb{N}$ in a model $\mathfrak{M}$ abbreviating with $\mathfrak{M}_I(p)$ a subset of S where p is true (we omit the standard definitions of the Boolean operators).

$$
\begin{array}{lll}
(\mathfrak{M}, i) \models p & \text{iff} & i \in \mathfrak{M}_I(p) \quad [\text{for } p \in Prop] \\
(\mathfrak{M}, i) \models \textbf{start} & \text{iff} & i = 0 \\
(\mathfrak{M}, i) \models \Box B & \text{iff} & \text{for each } j, \text{ if } i \leq j \text{ then } (\mathfrak{M}, j) \models B \\
(\mathfrak{M}, i) \models \Diamond B & \text{iff} & \text{there exists } j \text{ such that } i \leq j \text{ and } (\mathfrak{M}, j) \models B \\
(\mathfrak{M}, i) \models \bigcirc B & \text{iff} & (\mathfrak{M}, i+1) \models B \\
(\mathfrak{M}, i) \models A\,\mathcal{S}\,B & \text{iff} & \text{there exists a } k \in \mathbb{N}, \text{ such that } 0 \leqslant k < i \text{ and } (\mathfrak{M}, k) \models B \\
 & & \quad \text{and, for all } j \in \mathbb{N}, \text{ if } k \leqslant j < i \text{ then } (\mathfrak{M}, j) \models A
\end{array}
$$

Definition 1 (Satisfiability). *A formula R is satisfiable if, and only if, there exists a model $\mathfrak{M}$ such that $(\mathfrak{M}, 0) \models R$.*

Definition 2 (Validity). *A formula R is valid if, and only if, it is satisfiable in every possible model, i.e. for each $\mathfrak{M}$, $(\mathfrak{M}, 0) \models R$.*

Clausal temporal resolution, introduced in [8], operates on formulas in Separated Normal Form (SNF):

$$\Box \bigwedge_i A_i,$$

where each A_i is known as a *PLTL-clause* and must be one of the following forms with each particular k_a, k_b, l_c, l_d, and l representing a literal.

$$
\begin{array}{ll}
\textbf{start} \Rightarrow \bigvee_c l_c & \text{an } \textit{initial} \text{ PLTL-clause} \\
\bigwedge_a k_a \Rightarrow \bigcirc \bigvee_d l_d & \text{a } \textit{step} \text{ PLTL-clause} \\
\bigwedge_b k_b \Rightarrow \Diamond l & \text{a } \textit{eventuality (sometime)} \text{ PLTL-clause}
\end{array}
$$

(For convenience, the outer '$\Box$' and '$\wedge$' connectives are usually omitted.)
An eventuality PLTL-clause is called *unconditional* if it has the form $\Diamond l$.

It is known [7] that a PLTL-formula is satisfiable if, and only if, a set of temporal clauses is satisfiable. When a temporal formula is translated into the SNF form (see [7] for full details), we essentially apply a set of the transformation rules based upon the renaming of complex expressions by new propositions and upon the substitution of temporal operators by their fixpoint definitions.

3 Temporal Resolution for the unconditional eventuality case

We extend the notion of a PLTL-clause by allowing arbitrary Boolean combinations of propositions and giving a simplified normal form called *Divided Separated Normal Form (DSNF)*. Further, we consider unconditional eventuality PLTL-clauses only (and give a reduction to this case). We (ambiguously) refer to these new entities as *clauses*.

A *propositional temporal specification*, SP, is a triple consisting of:

1. an universal part, $\mathcal{U}$, given by a set of propositional formulas (clauses);
2. an initial part, $\mathcal{I}$, with the same form as the universal part; and
3. a step part, $\mathcal{S}$, given by a set of propositional step temporal clauses of the form:

$$P \Rightarrow \bigcirc Q \qquad (\textit{step}\text{ clause}),$$

where P and Q are Boolean combinations of propositional symbols [1].

(To relate these new clauses with the old ones, we note that the initial part corresponds to initial PLTL-clauses, step part corresponds to step clauses, and any clause C from the universal part can be represented by the pair: **start** $\Rightarrow C$, **true** $\Rightarrow \bigcirc C$.)

An *unconditional eventuality temporal problem*, P, whose satisfiability we are interested in, consists of a temporal specification SP with

4. an eventuality part, $\mathcal{E}$, given by a set of unconditional eventuality clauses of the form $\Diamond l$, where l is a literal.

This combination is denoted $\mathsf{P} = \mathsf{SP} \cup \mathcal{E}$.

A literal l from an eventuality clause is called an *eventuality literal*. Step clauses will also be referred to as *step rules*. Without loss of generality, we can assume that there are no two different temporal step clauses with the same left-hand sides.

In what follows we will not distinguish between a finite set of formulas $\mathcal{X}$ and the conjunction $\bigwedge \mathcal{X}$ of formulas in it. To each unconditional eventuality temporal problem, we associate the formula

$$\mathcal{I} \wedge \Box\mathcal{U} \wedge \Box\mathcal{S} \wedge \Box\mathcal{E}.$$

When we talk about particular properties of temporal problems (e.g., satisfiability, validity, logical consequences etc) we mean properties of the associated formula. The similar agreement takes place for specifications.

The inference system we use consists of an (implicit) *merging operation*

$$\frac{P_1 \Rightarrow \bigcirc Q_1, \ldots, P_n \Rightarrow \bigcirc Q_n}{\bigwedge\limits_{j=1}^{n} P_i \Rightarrow \bigcirc \bigwedge\limits_{j=1}^{n} Q_i} \ ,$$

(whose result is a logical consequence of its premises) and the following inference rules[2]. Due to our understanding of the temporal problem, the premises and conclusion of the rules are (implicitly) closed under $\Box$ operator.

Let $A \Rightarrow \bigcirc B$, $A_i \Rightarrow \bigcirc B_i$ be merged step rules, $\mathcal{U}$ be the (current) universal part of the problem.

– *Step resolution rule w.r.t.* $\mathcal{U}$: $\dfrac{A \Rightarrow \bigcirc B}{\neg A} \ (\bigcirc^{\mathcal{U}}_{res})$, where $\mathcal{U} \cup \{B\} \vdash \perp$.

---

[1] We could still restrict ourselves (e.g., for implementation purposes) to formulas in clausal form: $(p_1 \wedge p_2 \wedge \ldots \wedge p_k) \Rightarrow \bigcirc(q_1 \vee q_2 \vee \ldots \vee q_l)$.

[2] Note that, if the premises of the rules are given in clausal form, the result of applying these rules is a clause (or set of clauses for the sometime resolution rule).

– *Sometime resolution rule w.r.t.* $\mathcal{U}$

$$\frac{A_1 \Rightarrow \bigcirc B_1, \quad \ldots, \quad A_n \Rightarrow \bigcirc B_n \qquad \Diamond l}{(\bigwedge_{i=1}^{n} \neg A_i)} \; (\Diamond^{\mathcal{U}}_{res}),$$

where $A_i \Rightarrow \bigcirc B_i$ are *merged* step rules such that the *loop* side conditions

$$\mathcal{U} \cup \{B_i, l\} \vdash \bot \quad \text{and} \quad \mathcal{U} \cup \{B_i, \bigwedge_{j=1}^{n} \neg A_j\} \vdash \bot \quad \text{for all} \quad i \in \{1, \ldots, n\}$$

are satisfied. (The side conditions imply validity of $\bigvee A_j \Rightarrow \Box\bigcirc\neg l$. Indeed, $\bigvee A_j \Rightarrow \bigvee \bigcirc B_j \equiv \bigcirc \bigvee B_j \Rightarrow \bigcirc \neg l$ and $\bigvee A_j \Rightarrow \bigvee \bigcirc B_j \equiv \bigcirc \bigvee B_j \Rightarrow \bigcirc \bigvee A_j$; the formula $\bigvee A_j$ can be considered as an *invariant formula*.)

– *Sometime termination rule w.r.t.* $\mathcal{U}$
The contradiction $\bot$ is derived and the derivation is (successfully) terminated if $\mathcal{U} \cup \{l\} \vdash \bot$, where l is an eventuality literal.

– *Initial termination rule w.r.t.* $\mathcal{U}$
The contradiction $\bot$ is derived and the derivation is (successfully) terminated if $\mathcal{U} \cup \mathcal{I} \vdash \bot$.

Successful termination means that a given problem is unsatisfiable.

Note 1. All clauses generated by our inference rules are universal. Hence, the proof procedure does not change the Initial, Step and Eventuality parts of the temporal problem. As to the Universal part, it is extended step by step until one of termination rules is applied.

Note 2. The *sometime resolution rule* above can be thought of as two separate rules:

– *Induction rule w.r.t.* $\mathcal{U}$

$$\frac{A_1 \Rightarrow \bigcirc B_1, \quad \ldots, \quad A_n \Rightarrow \bigcirc B_n}{(\bigvee_{i=1}^{n} A_i) \Rightarrow \bigcirc \Box \neg l} \; (ind^{\mathcal{U}}),$$

(with the same side conditions as the sometime resolution rule above).

– *Pure sometime resolution*[3]

$$\frac{(\bigvee_{i=1}^{n} A_i) \Rightarrow \bigcirc \Box \neg l \qquad \Diamond l}{\neg(\bigvee_{i=1}^{n} A_i)} \; (\Diamond_{res}).$$

[3] We could as well formulate this rule in a more "traditional" form, with $\bigcirc\Diamond l$ as the second premise of the rule. However, note that $\Phi \wedge \Box\bigcirc\Diamond l$ is satisfiable if, and only if, $\Phi \wedge \Box\Diamond l$ is satisfiable for any temporal formula Φ.

4 Reduction to the unconditional eventuality case

Suppose we are interested in satisfiability of $\Phi \cup \{\,\Box(P \Rightarrow \Diamond q)\}$, where Φ is a set of propositional temporal formulas. Let us consider two clauses:

$$\Box((P \wedge \neg q) \Rightarrow waitforQ) \tag{1}$$

$$\Box((waitforQ \wedge \bigcirc\neg q) \Rightarrow \bigcirc\, waitforQ) \tag{2}$$

where $waitforQ$ is a new propositional symbol. The first clause is universal, the second is translated into a step clause $waitforQ \Rightarrow \bigcirc(q \vee waitforQ)$. Let us note that clauses (1) and (2) are logical consequences of a formula $\Box(q \equiv \neg waitforQ)$.

Theorem 1. *$\Phi \cup \{\,\Box(P \Rightarrow \Diamond q)\}$ is satisfiable if, and only if, $\Phi \cup \{(1),(2)\} \cup \{\,\Box\Diamond\neg waitforQ\}$ is satisfiable.*

Proof. ($\Rightarrow$) Let $\mathfrak{M}$ be a model of $\Phi \cup \{\,\Box(P \Rightarrow \Diamond q)\}$. Let us extend this model by a new proposition $waitforQ$ such that, in the extended model, $\mathfrak{M}'$, formulas (1), (2) and $\Box\Diamond\neg waitforQ$ would be true. In order to define the truth value of $waitforQ$, in n-th moment, $n \in \mathbb{N}$, we consider two cases depending on whether $\mathfrak{M} \models \Box\Diamond P$ or $\mathfrak{M} \models \Diamond\Box\neg P$.

– Assume $\mathfrak{M} \models \Box\Diamond P$. Together with $\Box(P \Rightarrow \Diamond q)$, this implies that $\mathfrak{M} \models \Box\Diamond q$. For every $n \in \mathbb{N}$ let us put

$$(\mathfrak{M}', n) \models \neg waitforQ \;\Leftrightarrow\; (\mathfrak{M}', n) \models q \qquad (\Leftrightarrow\; (\mathfrak{M}, n) \models q).$$

– Assume $\mathfrak{M} \models \Diamond\Box\neg P$. There are two possibilities:
  - $\mathfrak{M} \models \Box\neg P$. In this case let us put $(\mathfrak{M}', n) \models \neg waitforQ$ for all $n \in \mathbb{N}$.
  - There exists $m \in \mathbb{N}$ such that $(\mathfrak{M}, m) \models P$ and, for all $n > m$, $(\mathfrak{M}, n) \models \neg P$. These conditions imply, in particular, that there is $k \geq m$ such that $(\mathfrak{M}, k) \models q$ if the formula is satisfiable. Now we define $waitforQ$ in $\mathfrak{M}'$ as follows:

$$\begin{array}{lll} (\mathfrak{M}', n) \models \neg waitforQ \;\Leftrightarrow\; (\mathfrak{M}', n) \models q & \text{if} & n < k, \\ (\mathfrak{M}', n) \models \neg waitforQ & \text{if} & n \geq k. \end{array}$$

It is easy to see that $\mathfrak{M}'$ is a required model.

($\Leftarrow$) Let us show that $\Box(P \Rightarrow \Diamond q)$ is a logical consequence of $\Phi \cup \{(1),(2)\} \cup \{\,\Box\Diamond\neg waitforQ\}$.
Let $\mathfrak{M}'$ be a model of $\Phi \cup \{(1),(2)\} \cup \{\,\Box\Diamond\neg waitforQ\}$. By contradiction, suppose $\mathfrak{M}' \not\models \Box(P \Rightarrow \Diamond q)$, that is, $\mathfrak{M}' \models \Diamond(P \wedge \Box\neg q)$. Let $m \in \mathbb{N}$ be an index such that $(\mathfrak{M}', m) \models P$ and for all $n \geq m$, $(\mathfrak{M}', n) \models \neg q)$. Then from (1) and (2) we conclude that for all $n \geq m$ $(\mathfrak{M}', n) \models waitforQ)$ holds. However, this conclusion contradicts the formula $\Box\Diamond\neg waitforQ$ which is true in $\mathfrak{M}'$.

Lemma 1. *The growth in size of the problem following the reduction from a conditional to an unconditional eventuality temporal problem is linear in the number of conditional eventualities occurring in the given problem.*

Proof. Follows from the proof of Theorem 1.

Example 1. Consider the following set of formulas containing two eventuality literals:

1. $a \wedge \neg l_1 \wedge \neg l_2$
2. $\Box(a \Rightarrow \bigcirc(\neg a \wedge (l_1 \vee l_2) \wedge (\neg l_1 \vee \neg l_2)))$
3. $\Box((\neg a \wedge l_1 \wedge \neg l_2) \Rightarrow \bigcirc(\neg a \wedge l_1 \wedge \neg l_2))$
4. $\Box((\neg a \wedge \neg l_1 \wedge l_2) \Rightarrow \bigcirc(\neg a \wedge \neg l_1 \wedge l_2))$
5. $\Box(a \Rightarrow \Diamond l_1)$
6. $\Box(a \Rightarrow \Diamond l_2)$

We reduce it to an unconditional eventuality problem as given by Theorem 1.

$$\mathcal{I} = \left\{ 1.\ a \wedge \neg l_1 \wedge \neg l_2 \right\}$$

$$\mathcal{U} = \left\{ \begin{array}{l} 9.\ a \wedge \neg l_1 \Rightarrow wl_1 \\ 10.\ a \wedge \neg l_2 \Rightarrow wl_2 \end{array} \right\}$$

$$\mathcal{E} = \left\{ \begin{array}{l} 11.\ \Diamond \neg wl_1 \\ 12.\ \Diamond \neg wl_2 \end{array} \right\}$$

$$\mathcal{S} = \left\{ \begin{array}{l} 2.\ a \Rightarrow \bigcirc(\neg a \wedge (l_1 \vee l_2) \wedge (\neg l_1 \vee \neg l_2)) \\ 3.\ (\neg a \wedge l_1 \wedge \neg l_2) \Rightarrow \bigcirc(\neg a \wedge l_1 \wedge \neg l_2) \\ 4.\ (\neg a \wedge \neg l_1 \wedge l_2) \Rightarrow \bigcirc(\neg a \wedge \neg l_1 \wedge l_2) \\ 7.\ wl_1 \Rightarrow \bigcirc(l_1 \vee wl_1) \\ 8.\ wl_2 \Rightarrow \bigcirc(l_2 \vee wl_2) \end{array} \right\}$$

The derivation given below involves the following merged step clauses:

13. $(a \wedge wl_1 \wedge wl_2) \Rightarrow \bigcirc((\neg a \wedge \neg l_1 \wedge l_2 \wedge wl_1) \vee (\neg a \wedge l_1 \wedge \neg l_2 \wedge wl_2))$
14. $(\neg a \wedge l_1 \wedge \neg l_2 \wedge wl_2) \Rightarrow \bigcirc(\neg a \wedge l_1 \wedge \neg l_2 \wedge wl_2)$
15. $(\neg a \wedge \neg l_1 \wedge l_2 \wedge wl_1) \Rightarrow \bigcirc(\neg a \wedge \neg l_1 \wedge l_2 \wedge wl_1)$

(Clause 13 is obtained by merging clauses 2, 7 and 8, clause 14 by merging 3 and 8, and clause 15 by merging 4 and 7.)

Clause 14 gives a loop for resolution with 12, and clause 15 gives a loop for resolution with 11 resulting in two new universal clauses:

16. $a \vee \neg l_1 \vee l_2 \vee \neg wl_2$ [sometime resolution 14 and 12]
17. $a \vee l_1 \vee \neg l_2 \vee \neg wl_1$ [sometime resolution 15 and 11]

Let $\mathcal{U}_1$ be $\mathcal{U} \cup \{16, 17\}$. Then the step resolution of 13 with respect to $\mathcal{U}_1$ can be applied:

18. $\neg a \vee \neg wl_1 \vee \neg wl_2$ [step resolution 13 w.r.t $\mathcal{U}_1$]

Let $\mathcal{U}_2$ be $\mathcal{U}_1 \cup \{18\}$. Because $\mathcal{U}_2 \cup \mathcal{I} \vdash \perp$, the initial termination rule can be applied and the derivation is terminated. It follows that the given set of formulas is unsatisfiable.

5 Completeness of simplified resolution

From consideration of the models, it straightforwardly follows that:

Theorem 2 (soundness). *Temporal resolution rules preserve satisfiability.*

To show completeness of the simplified system we adapt the completeness proof of the original system [7] as follows.

Notation We consider interpretations (or valuations) of a set of propositional symbols (or atoms) $\mathcal{L}$ as Boolean functions over $\mathcal{L}$, that is, functions $I\colon \mathcal{L} \rightarrow \{0, 1\}$. A proposition $p \in \mathcal{L}$ is called *true* in I if, and only if, $I(p) = 1$ and *false* otherwise. This notion of truth and falsehood is extended in the usual way to literals and formulas built over $\mathcal{L}$. If E is an atom, literal, or formula such that E is true in I, then we write $I \models E$, and we write $I \not\models E$ if E is false in I.

Definition 3 (behaviour graph). *Given a specification $\mathsf{SP} =< \mathcal{U}, \mathcal{I}, \mathcal{S} >$ over a set of propositional symbols $\mathcal{L}$, we construct a finite directed graph G as follows. The nodes of G are interpretations of $\mathcal{L}$, and an interpretation, I, is a node of G if $I \models \mathcal{U}$.*

For each node, I, we construct an edge in G to a node I' if, and only if, the following condition is satisfied:

– *For every step rule $(P \Rightarrow \bigcirc Q) \in \mathcal{S}$, if $I \models P$ then $I' \models Q$.*

A node, I, is designated an initial node of G if $I \models \mathcal{I} \cup \mathcal{U}$. The behavior graph *$H$ of SP is the maximal subgraph of G given by the set of all nodes reachable from initial nodes.*

It is easy to see the following relation between behavior graphs of two temporal problems when one of them is obtained by extending the universal part of the other.

Lemma 2. *Let $\mathsf{SP}_1 =< \mathcal{U}_1, \mathcal{I}, \mathcal{S} >$ and $\mathsf{SP}_2 =< \mathcal{U}_2, \mathcal{I}, \mathcal{S} >$ be two specifications over the same set of propositional symbols such that $\mathcal{U}_1 \subseteq \mathcal{U}_2$. Then the behavior graph H_2 of SP_2 is a subgraph of the behavior graph H_1 of SP_1.*

Proof. The graph H_2 is the maximal subgraph of H_1 given by the set of all nodes whose interpretations satisfy $\mathcal{U}_2$ and that are reachable from the initial nodes of H_1 whose interpretations also satisfy $\mathcal{U}_2$. $\square$

Definition 4. *Let I, I' be nodes of a graph H. We denote the relation "I' is an immediate successor of I" by $I \rightarrow I'$, and the relation "I' is a successor of I" by $I \rightarrow^{+} I'$.*

Lemma 3 (existence of a model). *Let $\mathsf{P} = \mathsf{SP} \cup \mathcal{E}$ be an unconditional eventuality temporal problem. Let H be the behavior graph of SP such that both the set of initial nodes of H is non-empty and the following condition is satisfied:*

$$\forall I \forall l \exists I' (I \rightarrow^{+} I' \wedge I' \models l), \tag{3}$$

where I,I' are nodes of H and $\Diamond l \in \mathcal{E}$. Then P has a model.

Proof. It follows from the conditions of the lemma that all paths through H are infinite. We can construct a model for P as follows. Let I_0 be an initial node of H and $l_1, \ldots, l_m$ be all eventuality literals of $\mathcal{E}$. Let π be the infinite path $I_0, I_1, \ldots, I_{k_1}, I_{k_1+1}, \ldots, I_{k_2}, \ldots$, where for all $i \geq 0$ and $j \geq 1$, $I_{k_{mi+j}} \models l_j$. It follows by the construction of the behavior graph that the sequence of interpretations given by π is a model for P.

Indeed, all nontemporal clauses and all step clauses of P are satisfied on this sequence immediately by the definition of the behavior graph of SP. Now, let us take an eventuality clause $\Diamond l_j$ and a node I_ν on π. By construction of π, there is a node $I_{k_{mi+j}}$ such $I_\nu \rightarrow^{+} I_{k_{mi+j}}$ and $I_{k_{mi+j}} \models l_j$. It implies that $\Diamond l_j$ is satisfied at the moment ν. $\square$

Note This lemma remains valid in the case when a temporal problem does not contain eventualities. In this case the (sufficient) condition assumes the form

$$\forall I \exists I' (I \to^{+} I'), \tag{4}$$

which simply says that P has a model if all paths through H are infinite.

Theorem 3 (completeness). *If an unconditional eventuality problem $\mathsf{P} = \mathsf{SP} \cup \mathcal{E}$ is unsatisfiable then the temporal resolution procedure will derive a contradiction when applied to it.*

Proof. The proof proceeds by induction on the number of nodes in the behavior graph H of SP, which is finite. Let $\mathsf{SP} =< \mathcal{U}, \mathcal{I}, \mathcal{S} >$. If H is empty then the set $\mathcal{U} \cup \mathcal{I}$ is unsatisfiable. In this case the derivation is successfully terminated by the initial termination rule.

Now suppose H is not empty. Let I be a node of H which has no successors. In this case there exists a set of step rules $\{P_1 \Rightarrow \bigcirc Q_1, \ldots, P_m \Rightarrow \bigcirc Q_m\}$, $m > 0$, such that for all $1 \leq i \leq m$, $I \models P_i$ but the set $\mathcal{U} \cup \{Q_1, \ldots Q_m\}$ is unsatisfiable. So, we can derive by the step resolution rule a new clause $\neg P_1 \vee \ldots \vee \neg P_m$. Adding this clause to the set $\mathcal{U}$ results in removing the node I because $I \not\models \neg P_1 \vee \ldots \vee \neg P_m$. Let us note that if $m = 0$ the set $\mathcal{U}$ would be unsatisfiable in contradiction to the supposition H is not empty.

Now we consider another possibility when all nodes of H have a successor. Note that in this case $\mathcal{E}$ cannot be empty. Because P is unsatisfiable the following condition (the negation of condition (3) concerning the existence of a model given in lemma 3) holds:

$$\exists I \exists l \forall I' (I \to^{+} I' \Rightarrow I' \not\models l), \tag{5}$$

where I, I' are nodes of H and $l \in \mathcal{E}$.

Let I_0 be a node defined by the first quantifier in condition (5), and l_0 be an eventuality literal defined by the second one.

Let $\mathcal{I}$ be a finite nonempty set of indexes such that $\{I_n \mid n \in \mathcal{I}\}$ is the set of all successors of I_0. (It is possible that $0 \in \mathcal{I}$.) Let $I_{n_1}, \ldots I_{n_k}$ be the set of all immediate successors of I_0.

Let R_0 (R_n) be the set of all step rules of $\mathcal{S}$ whose left-hand sides are satisfied by I_0 (I_n). Let $A_0 \Rightarrow \bigcirc B_0$ ($A_n \Rightarrow \bigcirc B_n$) be the result of applying the merging operation to all clauses in R_0 (R_n) simultaneously.

Consider the following two cases depending on the emptiness of either R_0 or any $R_n, n \in \mathcal{I}$.

1. Let R_0 be empty. It implies, that $\mathcal{U} \vdash \neg l$. Indeed, since $I_{n_1}, \ldots I_{n_k}$ is the set of all immediate successors of I_0, it holds that $I_{n_1}, \ldots I_{n_k}$ are all possible models of $\mathcal{U}$. Because for all $j \in \{n_1, \ldots, n_k\}$ it holds $I_j \not\models l_0$, we can conclude that $\mathcal{U} \vdash \neg l_0$. Now, we can apply the sometime termination rule as this contradicts $\Diamond l_0$.
   The same argument holds if any of the sets R_n, $n \in \mathcal{I}$, is empty.
2. Let R_0 and R_n (for every $n \in \mathcal{I}$) be non empty. Then we have:

(a) $\mathcal{U} \cup \{B_0\} \vdash \neg l_0$ and $\mathcal{U} \cup \{B_n\} \vdash \neg l_0$ for all $n \in \mathcal{I}$.
Indeed, by arguments similar to given above at (1) we conclude that $I_{n_1}, \ldots, I_{n_k}$ are all interpretations of $\mathcal{U} \cup \{B_0\}$. Since $I_{n_1} \not\models l_0, \ldots, I_{n_k} \not\models l_0$ it follows that $\mathcal{U} \cup \{B_0\} \vdash \neg l_0$.
The same holds for every node I_n and every conjunction B_n, $n \in \mathcal{I}$.

(b) $\mathcal{U} \cup \{B_n\} \vdash \bigvee_{j \in \{0\} \cup \mathcal{I}} A_j$ for all $n \in \{0\} \cup \mathcal{I}$.
Again, consider the case $n = 0$ (for other indexes arguments are similar). Since $I_{n_1}, \ldots, I_{n_k}$ are all possible interpretations of $\mathcal{U} \cup \{B_0\}$ and for every $j \in \{n_1, \ldots, n_k\}$ $I_j \vdash A_j$ holds we can conclude that $\mathcal{U} \cup \{B_0\} \vdash \bigvee_{j \in \{n_1, \ldots, n_k\}} A_j$.

Therefore, the sometime resolution rule

$$\frac{\{A_j \Rightarrow \bigcirc B_j \mid j \in \{0\} \cup \mathcal{I}\} \qquad \Diamond l_0}{(\bigwedge_{j \in \{0\} \cup \mathcal{I}} \neg A_j)} \; (\Diamond^{\mathcal{U}}_{res}).$$

can be applied. Then, the node I_0 will is removed from H (recall that $I_0 \vdash A_0$ by construction of A_0) together with the set of its successors.

$\square$

6 Conditional single eventuality

Our simplified resolution technique relies on the translation from conditional eventualities to unconditional ones (Theorem 1). Here we show that, if a temporal problem is given in DSNF with only one conditional eventuality rule of the form [4]

$$P \Rightarrow \Diamond \bigcirc l,$$

then we do not actually need to carry out *any translation.*
Instead of the sometime termination rule, we now use

– *Sometime negation rule for single eventuality w.r.t.* $\mathcal{U}$

$$\frac{P \Rightarrow \bigcirc \Diamond l}{\neg P} \; (\Diamond^{\mathcal{U}}_{neg})$$

where $\mathcal{U} \cup \{l\} \vdash \bot$ (or $\mathcal{U} \vdash \neg l$).

The modified sometime resolution rule now takes the following form

[4] This is not the exact DSNF form—we here extend it to the conditional eventuality case. Note further the following equivalence $(P \Rightarrow \Diamond l) \equiv (P \Rightarrow (l \vee \Diamond \bigcirc l)) \equiv ((P \wedge \neg l) \Rightarrow \Diamond \bigcirc l)$. If we have more than one eventuality rule sharing the same eventuality literal, e.g., $P_1 \Rightarrow \bigcirc \Diamond l$, $P_2 \Rightarrow \bigcirc \Diamond l$, we replace them with the combined rule $((P_1 \vee P_2) \Rightarrow \bigcirc \Diamond l)$, which is equivalent w.r.t. satisfiability to the given pair of eventuality rules.

– *Sometime resolution rule for single eventuality w.r.t.* $\mathcal{U}$

$$\frac{A_1 \Rightarrow \bigcirc B_1, \quad \ldots, \quad A_n \Rightarrow \bigcirc B_n \quad P \Rightarrow \bigcirc\Diamond l}{(\bigwedge_{i=1}^{n} \neg A_i) \vee \neg P} \ (\Diamond^{\mathcal{U}}_{s-res})$$

with the usual loop side conditions.

Theorem 4. *Temporal resolution rules for the single eventuality case preserve satisfiability.*

Proof. Follows straightforwardly from consideration of the models. □

Lemma 4 (existence of a model: single eventuality). *Let* $\mathsf{P} = \mathsf{SP} \cup \{P \Rightarrow \bigcirc\Diamond l\}$ *be a single eventuality temporal problem. Let* H *be the behavior graph of* SP *such that all paths through* H *are infinite and the following condition is satisfied:*

$$\forall I(I \models P \Rightarrow \exists I'(I \rightarrow^{+} I' \wedge I' \models l)), \tag{6}$$

where I,I' *are nodes of* H. *Then* P *has a model.*

Proof. Similar to the proof of Lemma 3. □

Theorem 5 (completeness: single eventuality). *If a problem* $\mathsf{P} = \mathsf{SP} \cup \{P \Rightarrow \bigcirc\Diamond l\}$ *is unsatisfiable, then the temporal resolution procedure will derive a contradiction when applied to it.*

Proof. The proof is obtained by analysing the proof of Theorem 3 given above. It remains the same for the case when H contains nodes with no successor.

If all nodes in H have a successor, because of Lemma 4, the counterpart of the condition (6) now has the following form:

$$\exists I(I \models P \wedge \forall I'(I \rightarrow^{+} I' \Rightarrow I' \not\models l)). \tag{7}$$

Let I_0 be a node of H determined by the first quantifier of (7).

If case (1) of the previous proof holds (i.e. $\mathcal{U} \models \neg l$), node I_0 will be deleted from the graph because of the sometime negation rule (recall that $I_0 \models P$).

If case (2) holds, node I_0 will be deleted because of the conclusion of the sometime resolution rule for single eventuality: $(\bigwedge_{i=1}^{n} \neg A_i) \vee \neg P$. □

Example 2. Let us replace the two eventuality clauses of the example 1 by a single eventuality clause and show that the resulting problem is still unsatisfiable.

1. $a \wedge \neg l_1 \wedge \neg l_2$
2. $\Box(a \Rightarrow \bigcirc(\neg a \wedge (l_1 \vee l_2) \wedge (\neg l_1 \vee \neg l_2)))$
3. $\Box((\neg a \wedge l_1 \wedge \neg l_2) \Rightarrow \bigcirc(\neg a \wedge l_1 \wedge \neg l_2))$
4. $\Box((\neg a \wedge \neg l_1 \wedge l_2) \Rightarrow \bigcirc(\neg a \wedge \neg l_1 \wedge l_2))$
5. $\Box(a \Rightarrow \Diamond\bigcirc(\neg l_1 \wedge \neg l_2))$

The following DSNF corresponds to this problem[5]:

$$\mathcal{I} = \{\, 1.\ a \wedge \neg l_1 \wedge \neg l_2 \,\} \qquad \mathcal{S} = \left\{ \begin{array}{l} 2.\ a \Rightarrow \bigcirc(\neg a \wedge (l_1 \vee l_2) \wedge (\neg l_1 \vee \neg l_2)) \\ 3.\ (\neg a \wedge l_1 \wedge \neg l_2) \Rightarrow \bigcirc(\neg a \wedge l_1 \wedge \neg l_2) \\ 4.\ (\neg a \wedge \neg l_1 \wedge l_2) \Rightarrow \bigcirc(\neg a \wedge \neg l_1 \wedge l_2) \end{array} \right\}$$

$$\mathcal{U} = \{\, 6.\ l_{\neg l_1 \wedge \neg l_2} \Rightarrow (\neg l_1 \wedge \neg l_2) \,\}$$

$$\mathcal{E} = \{\, 7.\ a \Rightarrow \Diamond \bigcirc l_{\neg l_1 \wedge \neg l_2} \,\}$$

We see that step clauses $2, 3, 4$, taken together with the universal clause 6, form a loop for the single eventuality temporal resolution with clause 7. The resulting universal clause, $\neg a$, contradicts the initial clause.

Example 3 (Example 1 cont.). We show now that if the given DSNF contains more than one eventuality clause, reduction to the unconditional case is necessary. (I.e. the inference system described in this section is not complete for the general case.)

Consider the original set of temporal formulas from Example 1. The step resolution rule cannot be applied to the problem. Clauses 3 and 4 form a loop for eventuality rules 6 and 5 respectively; however, the temporal resolvents (by the rule $\Diamond^{\mathcal{U}}_{s-res}$) are tautologies: $a \vee \neg l_1 \vee l_2 \vee \neg a$ (resp., $a \vee l_1 \vee \neg l_2 \vee \neg a$).

Note 3. Instead of an *eventuality literal* we could introduce a notion of an *eventuality expression* giving eventuality rules the form $P \Rightarrow \Diamond \bigcirc Q$ where P, Q are arbitrary Boolean combinations of propositional symbols. It is not difficult to check that our inference system is adapted to such reformulation straightforwardly—we do not distinguish between eventuality expressions Q_1 and Q_2 if $\mathcal{U} \vdash (Q_1 \equiv Q_2)$, i.e. they are equivalent with respect a given universal part. Let us remind that during the derivation the universal part of a given problem is not narrowed. Alternatively, we could rename these eventuality expressions taking into consideration the equivalence, and introducing the same name for equivalent expressions.

7 Reduction to the single eventuality problem

We reduce now a temporal problem with several unconditional eventualities to a single eventuality temporal problem (first, in the language with past-time operator '$\mathcal{S}$').

Lemma 5. *$\mathsf{SP} \cup \{\Box \Diamond Q_i\}_{i \in I}$ is satisfiable if, and only if, $\mathsf{SP} \cup \{l \wedge \Box(l \Rightarrow \Diamond \bigcirc(\bigwedge_{i \in I} (\neg l\, \mathcal{S}\, Q_i) \wedge l))\}$ is satisfiable, where l is a new propositional symbol.*

Proof. Let us reformulate the given problem in a two-sorted temporal language with variables over $\mathbb{N}$ for the temporal sort:

$$\mathsf{SP} \cup \{\forall n \exists m (n \leq m \wedge Q_i(m))\}_{i \in I}$$

(meaning that each $Q_i, i \in I$, is satisfied infinitely often). This problem is equivalent with respect to satisfiability to the following (this can easily be checked by considering possible models):

$$\mathsf{SP} \cup \{\forall n \exists m (m > n \wedge \bigwedge_{i \in I} \exists k_i (n \leq k_i < m \wedge Q_i(k_i)))\} \tag{8}$$

[5] When introducing a new name for the positive occurrence of the subformula $\neg l_1 \wedge \neg l_2$, we use implication rather than equivalence; this technique goes back to [19].

which states, informally, that for each moment of time, n, there is a moment $m > n$, such that all eventualities $Q_i, i \in I$, are satisfied "after n and before m".

We prove that given a model for (8) it is possible to find a model for

$$\mathsf{SP} \cup \{l \wedge \Box(l \Rightarrow \Diamond\bigcirc(\bigwedge_{i\in I}(\neg l\,\mathcal{S}\,Q_i) \wedge l))\} \tag{9}$$

and vice versa.

First, consider a model $\mathfrak{M}$ for (8). We construct a model $\mathfrak{M}'$ for (9) by extending $\mathfrak{M}$ with a new proposition l and defining its value as follows. Formula (8) states that for each moment of time, n, there exists a future moment, m, when a certain property holds, defining thus a function $m(n)$. Let us construct a sequence of times defined by (8) starting from 0, i.e. $m_0 = 0$, $m_2 = m(0), \ldots, m_{j+1} = m(m_j)$; and let us also define y in $\mathfrak{M}'$ to be **true** at those times and **false** everywhere else. Note that, for all $i \in I$ and $j \geq 0$, there exists a moment $k_i : m_j \leq k_i < m_{j+1}$ such that $Q_i(k_i)$. Therefore, $(\mathfrak{M}, m_{j+1}) \models \bigwedge_{i\in I}(\neg y\,\mathcal{S}\,Q_i)$; hence, $(\mathfrak{M}, m_j) \models \Diamond\bigcirc(\bigwedge_{i\in I}(\neg l\,\mathcal{S}\,Q_i) \wedge l)$, making (9) **true** in $\mathfrak{M}'$.

Let $\mathfrak{M}$ be a model for (9); we show that it is also a model for (8). It is enough to show that for infinitely many n's there exists an m such that $(m > n)$ and $\bigwedge_{i\in I}\exists k_i(n \leq k_i < m \wedge Q_i(k_i))$ holds. For $j \geq 1$, let us consider the sequence m_j ($m_j > 0$) of all moments such that $(\mathfrak{M}, m_j) \models \bigwedge_{i\in I}(\neg l\,\mathcal{S}\,Q_i) \wedge l$ (note that there are infinitely many such moments); let $m_0 = 0$. We can see that for all $j \geq 0$, $n = m_j$, and $m = m_{j+1}$, the formula $\bigwedge_{i\in I}\exists k_i(n \leq k_i < m \wedge Q_i(k_i))$ is true in $\mathfrak{M}$. Indeed, $(\mathfrak{M}, n) \models l$, $(\mathfrak{M}, m) \models l$; by semantics of the operator "since", $(\mathfrak{M}, m) \models \bigwedge_{i\in I}(\neg l\,\mathcal{S}\,Q_i)$ means that $(\mathfrak{M}, m) \models \bigwedge_{i\in I}\exists k_i(n \leq k_i < m \wedge Q_i(k_i))$. □

Lemma 6. *Formula $\Box(A \Rightarrow (B\,\mathcal{S}\,C))$ is satisfiable if, and only if, the temporal specification*

$$(\neg s) \quad \wedge \quad \Box(A \Rightarrow s) \quad \wedge \quad \Box((C \vee (B \wedge s)) \equiv \bigcirc s)$$

is satisfiable, where s is a new propositional symbol. (The first clause goes into the initial part, the second into the universal part, and the third can be represented by two step clauses).

Proof. Follows straightforwardly from consideration of possible models. □

Corollary 1. *Any propositional temporal problem with an arbitrary number of eventuality clauses is equivalent, by satisfiability, to a single eventuality propositional temporal problem.*

Lemma 7. *The growth in size of the problem following the reduction from DSNF to a single eventuality temporal problem is linear in the number of eventualities occurring in the DSNF form.*

Proof. Follows from the above transformation. □

Example 4 (Example 1 cont.). We reduce now the given set of formulas to a single eventuality problem.

$$\mathcal{I} = \left\{ \begin{array}{l} 1.\ a \wedge \neg l_1 \wedge \neg l_2 \\ 18. l \\ 19.\ \neg s_1 \\ 20.\ \neg s_2 \end{array} \right\}$$

$$\mathcal{U} = \left\{ \begin{array}{l} 13.\ a \wedge \neg l_1 \Rightarrow wl_1 \\ 14.\ a \wedge \neg l_2 \Rightarrow wl_2 \\ 15.\ Q_1 \Rightarrow s_1 \\ 16.\ Q_2 \Rightarrow s_2 \\ 17.\ l \wedge \neg Q \Rightarrow wQ \end{array} \right\}$$

$$\mathcal{E} = \{21.\ \Diamond \neg wQ\}$$

$$\mathcal{S} = \left\{ \begin{array}{l} 2.\ a \Rightarrow \bigcirc(\neg a \wedge (l_1 \vee l_2) \wedge (\neg l_1 \vee \neg l_2)) \\ 3.\ (\neg a \wedge l_1 \wedge \neg l_2) \Rightarrow \bigcirc(\neg a \wedge l_1 \wedge \neg l_2) \\ 4.\ (\neg a \wedge \neg l_1 \wedge l_2) \Rightarrow \bigcirc(\neg a \wedge \neg l_1 \wedge l_2) \\ 5.\ wl_1 \Rightarrow \bigcirc(l_1 \vee wl_1) \\ 6.\ wl_2 \Rightarrow \bigcirc(l_2 \vee wl_2) \\ 7.\ Q \Rightarrow \bigcirc(Q_1 \wedge Q_2 \wedge l) \\ 8.\ (\neg wl_1 \vee \neg l \wedge s_1) \Rightarrow \bigcirc s_1 \\ 9.\ wl_1 \wedge (l \vee \neg s_1) \Rightarrow \bigcirc \neg s_1 \\ 10.\ (\neg wl_2 \vee \neg l \wedge s_2) \Rightarrow \bigcirc s_2 \\ 11.\ wl_2 \wedge (l \vee \neg s_2) \Rightarrow \bigcirc \neg s_2 \\ 12.\ wQ \Rightarrow \bigcirc(Q \vee wQ) \end{array} \right\}$$

The derivation of a contradiction is rather lengthy for this example; we omit it due to lack of space. We note that it enjoys the following property: Instead of two loops needed for Example 1, one is enough. However, the following example shows that it is not always the case.

Example 5. The following single unconditional eventuality temporal problem

$$\mathcal{I} = \{ 1.\ a \wedge \neg l \} \qquad \mathcal{U} = \emptyset \qquad \mathcal{E} = \{ \Diamond l \}$$

$$\mathcal{S} = \left\{ \begin{array}{l} 2.\ a \wedge \neg l \Rightarrow \bigcirc((a \wedge \neg l) \vee (a \wedge l)) \\ 3.\ a \wedge l \Rightarrow \bigcirc(\neg a \wedge \neg l) \\ 4.\ \neg a \wedge \neg l \Rightarrow \bigcirc(\neg a \wedge \neg l) \end{array} \right\}$$

requires two applications of the sometime resolution rule.

Indeed, the behavior graph for this problem consists of three vertices, I_0, I_1, I_2 (see Fig. 1). One application of the sometime resolution rule deletes the node I_2; then, the node I_1 can be deleted by the step resolution rule; after that, one more application of the sometime resolution is needed to delete the node I_0.

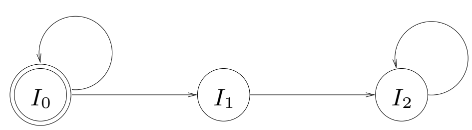

Fig. 1. Behavior graph for the problem. $I_0 = \{a, \neg l\}$; $I_1 = \{a, l\}$; $I_2 = \{\neg a, \neg l\}$.

8 Conclusion

In this paper, we have addressed the problem of simplifying, still further, the clausal resolution approach described in [7]. We have shown how to reduce conditional eventualities, i.e formulas of the form $\Box(P \Rightarrow \Diamond q)$, to unconditional eventualities, i.e. $\Box\Diamond q'$, and how to reduce problems containing multiple formulate of the form $\Diamond r$, to problems containing just one. This not only allows us to simplify the normal form required (from that defined in [7]) to a more streamlined version, but has also allowed us to introduce a set of simplified resolution rules. For example, in [7], the resolvent generated by applying temporal resolution to the formula $A \Rightarrow \bigcirc \square \neg l$ and $\lozenge l$ will be $(\neg A) \mathcal{U}\, l$, while we have here shown that the resolvent can be as simple as $\neg A$.

In addition to providing both a much simpler normal form for temporal formula, and a streamlined resolution process, the reductions described in this paper can, we believe, form the basis of temporal resolution provers with greatly improved efficiency. Thus, our future work in this area mainly involves the incorporation of the techniques described here to develop improved temporal provers.

References

1. A. Artale and E. Franconi. Temporal description logics. In *Handbook of Temporal Reasoning in Artificial Intelligence*. To appear.
2. E. Clarke, O. Grumberg and D. A. Peled. *Model Checking*. MIT Press, 2000.
3. A. Degtyarev and M. Fisher. Propositional temporal resolution revisited. In *Proc. of 7th UK Workshop on Automated Reasoning (ARW)*, London, July 2000.
4. A. Degtyarev and M. Fisher. An Extension of Propositional Temporal Resolution. In *Proc. UK Workshop on Automated Reasoning (ARW)*, York, March 2001.
5. A. Degtyarev and M.Fisher. Towards First-Order Temporal Resolution. In Proc. *KI 2001*, volume 2174 of *LNCS*. Springer Verlag, 2001.
6. R. Fagin, J. Halpern, Y. Moses, and M. Vardi. *Reasoning About Knowledge*. MIT Press, 1996.
7. M. Fisher, C. Dixon, and M. Peim. Clausal temporal resolution. *ACM Transactions on Computation Logic*, 2(1), January 2001.
8. M. Fisher. A resolution method for temporal logic. In *Proc. 12th Int. Joint Conf. on Artificial Intelligence (IJCAI)*. Morgan Kaufman, 1991.
9. M. Fisher. An introduction to executable temporal logics. *Knowledge Engineering Review*, 11:43–56, 1996.
10. G. J. Holzmann. The Model Checker Spin. *IEEE Trans. on Software Engineering*, 23(5):279–295, May 1997. Special Issue on Formal Methods in Software Practice.
11. U. Hustadt and R. Schmidt. Scientific Benchmarking with Temporal Logic Decision Procedures *Proc. Int. Conf. on Principles of Knowledge Representation & Reasoning*, 2002.
12. U. Hustadt and R. Schmidt. Formulae which Highlight Differences between Temporal Logic and Dynamic Logic Provers In *Issues in the Design and Experimental Evaluation of Systems for Modal and Temporal Logics*, Italy, 2001.
13. O. Kupferman and M. Vardi. Synthesis with incomplete informatio. In *Advances in Temporal Logic*. pp.109-128. Kluwer, 2000.
14. Z. Manna and A. Pnueli. *The Temporal Logic of Reactive and Concurrent Systems: Specification*. Springer, 1992.
15. A. Pnueli and R. Rosner. On the synthesis of a reactive module. In *Proc. 16th ACM Symp. Princip. Prog. Lang.*, pp.179–190, 1989.
16. A. S. Rao and M. P. Georgeff. Decision procedures for BDI logics. *Journal of Logic and Computation*, 8(3):293–342, 1998.
17. M. P. Shanahan. *Solving the Frame Problem*. MIT Press, 1997.
18. A. Tansel, editor. *Temporal Databases: theory, design, and implementation*. Benjamin/Cummings, 1993.
19. G. Tseitin. On the complexity of derivations in propositional calculus. In *Studies in Constructive Mathematics and Mathematical Logic*, 1968.
20. F. Wolter and M. Zakharyaschev. Temporalizing description logics. In *Frontiers of Combining Systems II*, pp.379–401. Research Studies Press LTD, England, 2000.