Monodic Temporal Resolution

ANATOLY DEGTYAREV
Department of Computer Science, King's College London, London, UK.
and
MICHAEL FISHER and BORIS KONEV
Department of Computer Science, University of Liverpool, Liverpool, UK.

Until recently, First-Order Temporal Logic (FOTL) has been only partially understood. While it is well known that the full logic has no finite axiomatisation, a more detailed analysis of fragments of the logic was not previously available. However, a breakthrough by Hodkinson et.al., identifying a finitely axiomatisable fragment, termed the *monodic* fragment, has led to improved understanding of FOTL. Yet, in order to utilise these theoretical advances, it is important to have appropriate proof techniques for this monodic fragment.

In this paper, we modify and extend the clausal temporal resolution technique, originally developed for propositional temporal logics, to enable its use in such monodic fragments. We develop a specific normal form for monodic formulae in FOTL, and provide a complete resolution calculus for formulae in this form. Not only is this clausal resolution technique useful as a practical proof technique for certain monodic classes, but the use of this approach provides us with increased understanding of the monodic fragment. In particular, we here show how several features of monodic FOTL can be established as corollaries of the completeness result for the clausal temporal resolution method. These include definitions of new decidable monodic classes, simplification of existing monodic classes by reductions, and completeness of clausal temporal resolution in the case of monodic logics with expanding domains, a case with much significance in both theory and practice.

Categories and Subject Descriptors: F.4.1 [**MATHEMATICAL LOGIC AND FORMAL LANGUAGES**]: Mathematical Logic—*Temporal logic*; F.4.1 [**MATHEMATICAL LOGIC AND FORMAL LANGUAGES**]: Mathematical Logic—*Mechanical theorem proving*; I.2.3 [**ARTIFICIAL INTELLIGENCE**]: Deduction and Theorem Proving—*Resolution*

General Terms: Theory, Algorithms, Verification

Additional Key Words and Phrases: Temporal Logic, Automated Theorem Proving, Resolution

1. INTRODUCTION

Temporal Logic has achieved a significant role in Computer Science, in particular, within the formal specification and verification of concurrent and distributed systems [Pnueli 1977; Manna and Pnueli 1992; Holzmann 1997]. While First-Order Temporal Logic (FOTL) is a very powerful and expressive formalism in which the specification of many algorithms, protocols and computational systems can be given at the natural level of abstraction, most of the temporal logics used remain essentially propositional. The reason for this is that it is easy to show that FOTL is, in general, incomplete (that is, not recursively-enumerable [Szalas and Holenderski 1988]). In fact, until recently, it has been difficult to find *any* non-trivial fragment of FOTL that has reasonable properties. A breakthrough by Hodkinson *et. al.* [Hodkinson et al. 2000] showed that *monodic* fragments of FOTL could be complete, even decidable. (In spite of this, the addition of equality or function symbols can again lead to the loss of recursively enumerability from these monodic fragments [Wolter and Zakharyaschev 2002a; Degtyarev et al. 2002].)

Following the definition of the monodic fragment, work analysing and extending this fragment has continued rapidly, and holds great promise for increasing the power of logic-based formal methods. However, until recently, there were no proof techniques for monodic fragments of FOTLs. Although a tableaux based approach was proposed in [Kontchakov et al. 2004], we here provide a complete resolution calculus for monodic FOTL, based on our work on clausal temporal resolution over a number of years [Fisher 1991; Fisher et al. 2001; Degtyarev and Fisher 2001; Degtyarev et al. 2002; 2003b]. The clausal resolution technique has been shown to be one of the most effective proof techniques for propositional temporal logics [Hustadt and Konev 2003], and we have every reason to believe that it will be as least as successful in the case of FOTL; this paper provides the key formal background for this approach.

The structure of the paper is as follows. After a brief introduction to FOTL (Section 2), we define a normal form that will be used as the basis of the resolution technique and show that any monodic temporal problem can be transformed into the normal form (Section 3). In Section 4 we present the temporal resolution calculus and, in Section 5, we provide detailed completeness results.

In Sections 6 and 7, we adapt the resolution technique to a number of variations of monodic FOTL, whose completeness follows from the corresponding adaptation of the completeness results given in Section 5. Thus, in Section 6, we provide an extension of the monodic fragment (as defined in [Hodkinson et al. 2000]) and, in Section 7, we restrict first-order quantification in a number of ways to provide sub-classes which admit simplified clausal resolution techniques.

In the penultimate part of the paper, we examine results relating to the practical use of the clausal resolution calculus. The first such aspect concerns decidability, which we consider in Section 8. An appropriate *loop search* algorithm is required for implementation of the clausal resolution technique, and the definition and completeness of such an algorithm is examined in Section 9. In order to develop a practical clausal resolution system, as well as examining a fragment with important applications and a simplified normal form, we present results relating to resolution over the monodic fragment with *expanding domains* in Section 10. This provides the basis for the system currently being implemented [Konev et al. 2003b].

Finally, in Section 12, we present conclusions and outline our future work.

2. FIRST-ORDER TEMPORAL LOGIC

First-Order (linear time) Temporal Logic, FOTL, is an extension of classical first-order logic with operators that deal with a linear and discrete model of time (isomorphic to $\mathbb{N}$, and the most commonly used model of time).

2.1 Syntax of FOTL

The first-order temporal language is constructed in a standard way [Fisher 1997; Hodkinson et al. 2000] from:

- *predicate symbols* $P_0, P_1, \ldots$ each of which is of some fixed arity (N.B., null-ary predicate symbols are called *propositions*);
- *individual variables* $x_0, x_1, \ldots$;
- *individual constants* $c_0, c_1, \ldots$ (N.B., there is no equality operator defined and, while constants are present, no other function symbols are allowed in this FOTL language);
- *boolean operators* $\wedge$, $\neg$, $\vee$, $\Rightarrow$, $\equiv$ **true** ('true'), **false** ('false'); *quantifiers* $\forall$ and $\exists$; together with
- *temporal operators* $\square$ ('always in the future'), $\lozenge$ ('sometime in the future'), $\bigcirc$ ('at the next moment'), U (until), and W (weak until).

Definition 2.1 *Atomic Formulae and Literals*. An *atomic formula* of FOTL is defined as $P(t_1, \ldots, t_n)$, where P is a predicate symbol with arity n, and each t_i is either an individual constant or an individual variable. A *literal* is either an atomic formula or the negation of an atomic formula.

Definition 2.2 *Well-Formed Formulae*. The set of *well-formed formulae* of FOTL, WFF_{FOTL} is defined as follows:

- **false**, **true** and any atomic formula is in WFF_{FOTL};
- if A is in WFF_{FOTL} then so are $\neg A$, $\lozenge A$, $\square A$, and $\bigcirc A$;
- if A is in WFF_{FOTL} and x is an individual variable, then $\forall x A$ and $\exists x A$ are also in WFF_{FOTL};
- if A and B are in WFF_{FOTL} then so are $A \vee B$, $A \wedge B$, $A \Rightarrow B$, $A \equiv B$, $A \mathsf{U} B$, and $A \mathsf{W} B$.

For a given formula, ϕ, $\mathrm{const}(\phi)$ denotes the set of constants occurring in ϕ. We write $\phi(x)$ to indicate that $\phi(x)$ has *at most one* free variable x (if not explicitly stated otherwise). As usual, a *closed formulae* is one with no free variables.

From now on, we deal exclusively with well-formed formulae of FOTL.

2.2 Semantics of FOTL

Formulae in FOTL are interpreted in *first-order temporal structures* of the form $\mathfrak{M} = \langle D, I \rangle$, where D is a non-empty set, the *domain* of $\mathfrak{M}$, and I is a function associating with every moment of time, $n \in \mathbb{N}$, an interpretation of predicate and constant symbols over D. We require that the interpretation of constants is *rigid*. Thus, for every constant c and all moments of time $i, j \geq 0$, we have $I_i(c) = I_j(c)$. The interpretation of predicate symbols is flexible.

A *(variable) assignment* $\mathfrak{a}$ over D is a function from the set of individual variables to D. For every moment of time, n, there is a corresponding *first-order* structure $\mathfrak{M}_n = \langle D, I_n \rangle$, where $I_n = I(n)$. Intuitively, FOTL formulae are interpreted in sequences of *worlds*, $\mathfrak{M}_0, \mathfrak{M}_1, \ldots$ with truth values in different worlds being connected by means of temporal operators.

The *truth* relation $\mathfrak{M}_n \models^{\mathfrak{a}} \phi$ in a structure $\mathfrak{M}$, for an assignment $\mathfrak{a}$, is defined inductively in the usual way under the following understanding of temporal operators:

$$\begin{array}{lll}
\mathfrak{M}_n \models^{\mathfrak{a}} \textbf{true},\ \mathfrak{M}_n \not\models^{\mathfrak{a}} \textbf{false} & & \\
\mathfrak{M}_n \models^{\mathfrak{a}} P(t_1,\dots,t_m) & \text{iff} & \langle I_n^{\mathfrak{a}}(t_1),\dots I_n^{\mathfrak{a}}(t_m)\rangle \in I_n(P),\ \text{where} \\
 & & I_n^{\mathfrak{a}}(t_i) = I_n(t_i),\ \text{if } t_i \text{ is a constant, and} \\
 & & I_n^{\mathfrak{a}}(t_i) = \mathfrak{a}(t_i),\ \text{if } t_i \text{ is a variable} \\
\mathfrak{M}_n \models^{\mathfrak{a}} \neg\phi & \text{iff} & \mathfrak{M}_n \not\models^{\mathfrak{a}} \phi \\
\mathfrak{M}_n \models^{\mathfrak{a}} \phi \wedge \psi & \text{iff} & \mathfrak{M}_n \models^{\mathfrak{a}} \phi \text{ and } \mathfrak{M}_n \models^{\mathfrak{a}} \psi \\
\mathfrak{M}_n \models^{\mathfrak{a}} \phi \vee \psi & \text{iff} & \mathfrak{M}_n \models^{\mathfrak{a}} \phi \text{ or } \mathfrak{M}_n \models^{\mathfrak{a}} \psi \\
\mathfrak{M}_n \models^{\mathfrak{a}} \phi \Rightarrow \psi & \text{iff} & \mathfrak{M}_n \models^{\mathfrak{a}} (\neg\phi \vee \psi) \\
\mathfrak{M}_n \models^{\mathfrak{a}} \phi \equiv \psi & \text{iff} & \mathfrak{M}_n \models^{\mathfrak{a}} ((\phi \Rightarrow \psi) \wedge (\psi \Rightarrow \phi)) \\
\mathfrak{M}_n \models^{\mathfrak{a}} \bigcirc\phi & \text{iff} & \mathfrak{M}_{n+1} \models^{\mathfrak{a}} \phi; \\
\mathfrak{M}_n \models^{\mathfrak{a}} \Diamond\phi & \text{iff} & \text{there exists } m \geq n \text{ such that } \mathfrak{M}_m \models^{\mathfrak{a}} \phi; \\
\mathfrak{M}_n \models^{\mathfrak{a}} \Box\phi & \text{iff} & \text{for all } m \geq n,\ \mathfrak{M}_m \models^{\mathfrak{a}} \phi; \\
\mathfrak{M}_n \models^{\mathfrak{a}} (\phi \mathsf{U} \psi) & \text{iff} & \text{there exists } m \geq n, \text{ such that } \mathfrak{M}_m \models^{\mathfrak{a}} \psi, \\
 & & \quad \text{and for all } i \in \mathbb{N},\ n \leq i < m \text{ implies } \mathfrak{M}_m \models^{\mathfrak{a}} \phi; \\
\mathfrak{M}_n \models^{\mathfrak{a}} (\phi \mathsf{W} \psi) & \text{iff} & \mathfrak{M}_n \models^{\mathfrak{a}} (\phi \mathsf{U} \psi) \text{ or } \mathfrak{M}_n \models^{\mathfrak{a}} \Box\phi.
\end{array}$$

$\mathfrak{M}$ is a *model* for a formula ϕ (or ϕ is *true* in $\mathfrak{M}$) if there exists an assignment $\mathfrak{a}$ such that $\mathfrak{M}_0 \models^{\mathfrak{a}} \phi$. A formula is *satisfiable* if it has a model. A formula is *valid* if it is true in any temporal structure under any assignment. We say that a formula ψ is a *logical consequence* of formula ϕ, denoted $\phi \models \psi$, if for every structure $\mathfrak{M}$ such that $\mathfrak{M} \models \phi$ we also have $\mathfrak{M} \models \psi$.

This logic is complex. It is known that even “small” fragments of FOTL, such as the *two-variable monadic* fragment (all predicates are unary), are not recursively enumerable [Merz 1992; Hodkinson et al. 2000]. However, the set of valid *monodic* formulae is known to be finitely axiomatisable [Wolter and Zakharyaschev 2002a].

Definition 2.3 *Monodic Formula*. An FOTL-formula ϕ is called *monodic* if any subformulae of the form $\mathcal{T}\psi$, where $\mathcal{T}$ is one of $\bigcirc$, $\Box$, $\Diamond$ (or $\psi_1\mathcal{T}\psi_2$, where $\mathcal{T}$ is one of U, W), contains at most one free variable.

Example 2.4. The formulae

$$\forall x \Box \exists y P(x,y) \quad \text{and} \quad \forall x \Box P(x,c)$$

are monodic, whereas the formula

$$\forall x, y(P(x,y) \Rightarrow \Box P(x,y))$$

is non-monodic.

The addition of either equality or function symbols to the monodic fragment leads to the loss of recursive enumerability [Wolter and Zakharyaschev 2002a]. Moreover, it was proved in [Degtyarev et al. 2002] that the *two variable monadic monodic fragment with equality* is not recursively enumerable. However, in [Hodkinson 2002] it was shown that the *guarded monodic fragment with equality* is decidable.

3. DIVIDED SEPARATED NORMAL FORM (DSNF)

As in the case of classical resolution, our method works on temporal formulae transformed into a normal form. The normal form we use follows the spirit of Separated Normal Form

(SNF) [Fisher 1991; Fisher et al. 2001] and First-Order Separated Normal Form (SNF$_f$) [Fisher 1992; 1997], but is refined even further.

The development of SNF/SNF$_f$ was partially devised in order to separate past, present and future time temporal formula (inspired by Gabbay's separation result [Gabbay 1987]). Thus, formulae in SNF/SNF$_f$ comprise implications with present-time formulae on the left-hand side and (present or) future formulae on the right-hand side. The transformation of temporal formulae into separated form is based upon the well-known *renaming* technique [Tseitin 1983; Plaisted and Greenbaum 1986], which preserves satisfiability and admits the extension to temporal logic in (Renaming Theorems [Fisher 1997]).

Another aim with SNF/SNF$_f$ was to reduce the variety of temporal operators used to a simple core set. To this end, the transformation to SNF/SNF$_f$ involves the removal of temporal operators represented as *maximal* fixpoints, that is, $\square$ and W (Maximal Fixpoint Removal Theorems [Fisher 1997]). Note that the U operator can be represented as a combination of operators based upon maximal fixpoints and the $\diamondsuit$ operator (which is retained within SNF/SNF$_f$). This transformation is based upon the simulation of fixpoints using QPTL [Wolper 1982; Kesten and Pnueli 1995].

In the first-order context, we now add one further aim, namely to divide the temporal part of a formula and its (classical) first-order part in such way that the temporal part is as simple as possible. The modified normal form is called Divided Separated Normal Form or DSNF for short.

Definition 3.1 *Temporal Step Clauses*. A *temporal step clause* is a formula either of the form $l \Rightarrow \bigcirc m$, where l and m are propositional literals, or $(L(x) \Rightarrow \bigcirc M(x))$, where $L(x)$ and $M(x)$ are unary literals. We call a clause of the the first type an (original) *ground* step clause, and of the second type an (original) *non-ground* step clause[1]. (Note that the term 'original' here is used to distinguish these clauses from other that are introduced later.)

Definition 3.2 *DSNF*. A *monodic temporal problem in Divided Separated Normal Form (DSNF)* is a quadruple $\langle \mathcal{U}, \mathcal{I}, \mathcal{S}, \mathcal{E} \rangle$, where

(1) the universal part, $\mathcal{U}$, is a finite set of arbitrary closed first-order formulae;

(2) the initial part, $\mathcal{I}$, is, again, a finite set of arbitrary closed first-order formulae;

(3) the step part, $\mathcal{S}$, is a finite set of original (ground and non-ground) temporal step clauses; and

(4) the eventuality part, $\mathcal{E}$, is a finite set of eventuality clauses of the form $\diamondsuit L(x)$ (a *non-ground* eventuality clause) and $\diamondsuit l$ (a *ground eventuality* clause), where l is a propositional literal and $L(x)$ is a unary non-ground literal.

The intuition here is that the initial part describes the initial state of the temporal model, the universal part describes the properties of *all* states, the step part describes the required transitions from one state to the next, and the eventuality part describes properties of some future state.

Note that, in a monodic temporal problem, we disallow two different temporal step clauses with the same left-hand sides. This requirement can be easily guaranteed by renaming. For

[1] We could also allow arbitrary Boolean combinations of propositional and unary literals in the right hand side of ground and non-ground step clauses, respectively, and all results of this paper would hold. We restrict ourselves with literals for simplicity of the presentation.

example, if we have two step clauses

$$\begin{array}{lcl} P & \Rightarrow & \bigcirc Q \\ P & \Rightarrow & \bigcirc R \end{array}$$

then we can rename '$Q \wedge R$' by a new predicate 'S', add the formula '$S \Rightarrow (Q \wedge R)$' to $\mathcal{U}$ and replace the above step clauses by just

$$P \;\Rightarrow\; \bigcirc S$$

In what follows, we will not distinguish between a finite set of formulae $\mathcal{X}$ and the conjunction $\bigwedge \mathcal{X}$ of formulae within the set. With each monodic temporal problem, we associate the formula

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

Now, when we talk about particular properties of a temporal problem (e.g., satisfiability, validity, logical consequences etc) we mean properties of the associated formula.

Arbitrary monodic first-order temporal formula can be transformed into DSNF. We present the transformation as a two stage reduction.

Reduction to conditional DSNF. We first give a reduction from monodic FOTL to a normal form where, in addition to the parts above, *conditional* eventuality clauses of the form

$$P(x) \Rightarrow \Diamond L(x) \text{ and } p \Rightarrow \Diamond l$$

are allowed. The reduction is based on using a renaming technique to substitute non-atomic subformulae and replacing temporal operators by their fixed point definitions described e.g. in [Fisher et al. 2001]. The translation can be described as a number of steps.

(1) Translate a given monodic formula to negation normal form. (To assist understanding of the translation, we list here some equivalent FOTL formulae.)

$$\begin{array}{lcl} \forall x(\neg \bigcirc \phi(x) & \equiv & \bigcirc \neg \phi(x)); \\ \forall x(\neg \Box \phi(x) & \equiv & \Diamond \neg \phi(x)); \\ \forall x(\neg \Diamond \phi(x) & \equiv & \Box \neg \phi(x); \\ \forall x(\neg(\phi(x) \mathsf{U} \psi(x)) & \equiv & \neg \psi(x) \mathsf{W} (\neg \phi(x) \wedge \neg \psi(x)))); \\ \forall x(\neg(\phi(x) \mathsf{W} \psi(x)) & \equiv & \neg \psi(x) \mathsf{U} (\neg \phi(x) \wedge \neg \psi(x))). \end{array}$$

If the transformations above are applied in a straightforward way, the size of the result may grow exponentially; we may have to use *renaming* [Tseitin 1983; Plaisted and Greenbaum 1986; Nonnengart and Weidenbach 2001] in order to keep it linear.

(2) Recursively rename innermost temporal subformulae, $\bigcirc \phi(x)$, $\Diamond \phi(x)$, $\Box \phi(x)$, $\phi(x) \mathsf{U} \psi(x)$, $\phi(x) \mathsf{W} \psi(x)$ by a new unary predicate $P(x)$. Since subformulae have positive polarity then, as in the classical case [Tseitin 1983; Plaisted and Greenbaum 1986; Nonnengart and Weidenbach 2001], renaming introduces implications $P(x)$ of the following form [Fisher et al. 2001]:

$$\begin{array}{llcl} (a) & \Box \forall x(P(x) & \Rightarrow & \bigcirc \phi(x)); \\ (b) & \Box \forall x(P(x) & \Rightarrow & \Diamond \phi(x)); \\ (c) & \Box \forall x(P(x) & \Rightarrow & \Box \phi(x)); \\ (d) & \Box \forall x(P(x) & \Rightarrow & \phi(x) \mathsf{U} \psi(x)); \\ (e) & \Box \forall x(P(x) & \Rightarrow & \phi(x) \mathsf{W} \psi(x)). \end{array}$$

Assuming that any required (first-order) renaming of the complex expression $\phi(x)$ can be carried out[2], then formulae of the form (a) and (b) are already in the normal form, while formulae of the form (c), (d), and (e) require extra reduction by removing the temporal operators using their fixed point definitions.

(3) Use fixed point definitions
$\Box\forall x(P(x) \Rightarrow \Box\phi(x))$ is satisfiability equivalent [Kaivola 1995; Fisher et al. 2001] to

$$\begin{array}{l} \Box\forall x(P(x) \Rightarrow R(x)) \\ \wedge \Box\forall x(R(x) \Rightarrow \bigcirc R(x)) \\ \wedge \Box\forall x(R(x) \Rightarrow \phi(x)), \end{array}$$

$\Box\forall x(P(x) \Rightarrow (\phi(x)\mathsf{U}\psi(x)))$ is equivalent (w.r.t. satisfiability) to

$$\begin{array}{l} \Box\forall x(P(x) \Rightarrow \Diamond\psi(x)) \\ \wedge \Box\forall x(P(x) \Rightarrow \phi(x) \vee \psi(x)) \\ \wedge \Box\forall x(P(x) \Rightarrow S(x) \vee \psi(x)) \\ \wedge \Box\forall x(S(x) \Rightarrow \bigcirc(\phi(x) \vee \psi(x))) \\ \wedge \Box\forall x(S(x) \Rightarrow \bigcirc(S(x) \vee \psi(x))), \end{array}$$

and $\Box\forall x(P(x) \Rightarrow (\phi(x)\mathsf{W}\psi(x)))$ is equivalent (w.r.t. satisfiability) to

$$\begin{array}{l} \Box\forall x(P(x) \Rightarrow \phi(x) \vee \psi(x)) \\ \wedge \Box\forall x(P(x) \Rightarrow S(x) \vee \psi(x)) \\ \wedge \Box\forall x(S(x) \Rightarrow \bigcirc(\phi(x) \vee \psi(x))) \\ \wedge \Box\forall x(S(x) \Rightarrow \bigcirc(S(x) \vee \psi(x))), \end{array}$$

where $R(x)$ and $S(x)$ are new unary predicates.

Conditional problems to unconditional problems. In the second stage, we replace any formula $\Box\forall x(P(x) \Rightarrow \Diamond L(x))$ by

$$\Box\forall x(((P(x) \wedge \neg L(x)) \Rightarrow \mathit{waitforL}(x))) \tag{1}$$

$$\Box\forall x((\mathit{waitforL}(x) \wedge \bigcirc\neg L(x)) \Rightarrow \bigcirc\, \mathit{waitforL}(x)) \tag{2}$$

$$\Box\forall x(\Diamond\neg \mathit{waitforL}(x)) \tag{3}$$

where $\mathit{waitforL}(x)$ is a new unary predicate. Note that formula (2) can easily be transformed into the required form by moving the $\bigcirc\neg L(x)$ subformula across the implication.

LEMMA 3.3. *$\Phi \cup \{\Box\forall x(P(x) \Rightarrow \Diamond L(x))\}$ is satisfiable if, and only if, $\Phi \cup \{(1),(2),(3)\}$ is satisfiable.*

Proof $(\Rightarrow)$ Let $\mathfrak{M}$ be a model of $\Phi \cup \{\Box\forall x(P(x) \Rightarrow \Diamond L(x))\}$. Let us extend this model by a new predicate $\mathit{waitforL}$ such that, in the extended model, $\mathfrak{M}'$, formulae (1), (2), and (3) would be true.

Let d be an arbitrary element of the domain D. We define the truth value of $\mathit{waitforL}(d)$ in n-th moment, $n \in \mathbb{N}$, depending on whether $\mathfrak{M} \models \Box\Diamond P(d)$ or $\mathfrak{M} \models \Diamond\Box\neg P(d)$.

[2] The new 'renaming' formulae are added to the universal part; this kind of first-order renaming will be used implicitly later in this section.

—Assume $\mathfrak{M} \models \Box\Diamond P(d)$. Together with $\mathfrak{M} \models \Box\forall x(P(x) \Rightarrow \Diamond L(x))$, and the fact that $\Diamond\Diamond P \Rightarrow \Diamond P$ is an axiom, then the above implies that $\mathfrak{M} \models \Box\Diamond L(d)$.

For every $n \in \mathbb{N}$ let us put

$$\mathfrak{M}'_n \models \neg \mathit{waitforL}(d) \Leftrightarrow \mathfrak{M}'_n \models L(d) \qquad (\Leftrightarrow \mathfrak{M}_n \models L(d)).$$

—Assume $\mathfrak{M} \models \Diamond\Box\neg P(d)$. There are two possibilities:

—$\mathfrak{M} \models \Box\neg P(d)$. In this case let us put $\mathfrak{M}'_n \models \neg \mathit{waitforL}(d)$ for all $n \in \mathbb{N}$.

—There exists $m \in \mathbb{N}$ such that $\mathfrak{M}_m \models P(d)$ and, for all $n > m$, $\mathfrak{M}_n \models \neg P(d)$. These conditions imply, in particular, that there is $l \geq m$ such that $\mathfrak{M}_l \models L(d)$ if the formula is satisfiable. Now we define $\mathit{waitforL}(d)$ in $\mathfrak{M}'$ as follows:

$$\begin{array}{lll} \mathfrak{M}'_n \models \neg \mathit{waitforL}(d) \Leftrightarrow \mathfrak{M}'_n \models L(d) & \text{if} & 0 \leq n < l, \\ \mathfrak{M}'_n \models \neg \mathit{waitforL}(d) & \text{if} & n \geq l. \end{array}$$

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

($\Leftarrow$) Let us show that $\Box\forall x(P(x) \Rightarrow \Diamond L(x))$ is a logical consequence of $\Phi \cup \{(1),(2),(3)\}$.

Let $\mathfrak{M}'$ be a model of $\Phi \cup \{(1),(2),(3)\}$. By contradiction, suppose $\mathfrak{M}' \not\models \Box\forall x(P(x) \Rightarrow \Diamond L(x))$, that is, $\mathfrak{M}' \models \Diamond\exists x(P(x) \wedge \Box\neg L(x))$. Let $m \in \mathbb{N}$ be an index and $e \in D_m$ be a domain element such that $\mathfrak{M}'_m \models P(e)$ and for all $n \geq m$, $\mathfrak{M}'_n \models \neg L(e))$. Then from (1) and (2) we conclude that for all $n \geq m$, we have $\mathfrak{M}'_n \models \mathit{waitforL}(e))$. However, this conclusion contradicts the formula $\Box\forall x\Diamond\neg \mathit{waitforL}(x)$ which is true in $\mathfrak{M}'$. ❑

This leads us to the following theorem.

THEOREM 3.4 TRANSFORMATION. *Every monodic first-order temporal formula can be transformed, in a satisfiability equivalence preserving way, to DSNF with at most a linear increase in size of the problem.*

Note 3.5. Furthermore, if ϕ is a formula and P is a problem in DSNF obtained from ϕ by the transformations given above, then every model of ϕ can be *expanded* to a model of P, and every model of P can be *reducted* to a model of ϕ, where the notions of an expansion and reduct are analogous to the once used in classical first-order logic [Gallier 1986].

Example 3.6. Let us consider the temporal formula $\exists x \Box\Diamond\forall y\forall z\exists u\, \Phi(x,y,z,u)$ where $\Phi(x,y,z,u)$ does not contain temporal operators and reduce it to DSNF. First, we rename the innermost temporal subformula by a new predicate,

$$\exists x \Box P_1(x) \wedge \Box\forall x[P_1(x) \Rightarrow \Diamond\forall y\forall z\exists u\, \Phi(x,y,z,u)].$$

Now, we rename the first '$\Box$'-formula and the subformula under the '$\Diamond$' operator,

$$\begin{array}{rl} \exists x P_3(x) & \wedge \;\; \Box\forall x[P_1(x) \Rightarrow \Diamond P_2(x)] \\ & \wedge \;\; \Box\forall x[P_2(x) \Rightarrow \forall y\forall z\exists u\, \Phi(x,y,z,u)] \\ & \wedge \;\; \Box\forall x[P_3(x) \Rightarrow \Box P_1(x)], \end{array}$$

"unwind" the '$\Box$' operator

$$\begin{array}{rcl} \exists x P_3(x) & \wedge & \Box\forall x[P_1(x) \Rightarrow \Diamond P_2(x)] \\ & \wedge & \Box\forall x[P_2(x) \Rightarrow \forall y \forall z \exists u\, \Phi(x,y,z,u)] \\ & \wedge & \Box\forall x[P_3(x) \Rightarrow P_4(x)] \\ & \wedge & \Box\forall x[P_4(x) \Rightarrow \bigcirc P_4(x)] \\ & \wedge & \Box\forall x[P_4(x) \Rightarrow P_1(x)], \end{array}$$

and, finally, reduce the conditional eventuality to an unconditional one.

$$\begin{array}{rcl} \exists x P_3(x) & \wedge & \Box\forall x[P_2(x) \Rightarrow \forall y \forall z \exists u\, \Phi(x,y,z,u)] \\ & \wedge & \Box\forall x[P_3(x) \Rightarrow P_4(x)] \\ & \wedge & \Box\forall x[P_4(x) \Rightarrow \bigcirc P_4(x)] \\ & \wedge & \Box\forall x[P_4(x) \Rightarrow P_1(x)] \\ & \wedge & \Box\forall x[(P_1(x) \wedge \neg P_2(x)) \Rightarrow \mathit{waitforP_2}(x)] \\ & \wedge & \Box\forall x[(\mathit{waitforP_2}(x) \wedge \bigcirc\neg P_2(x)) \Rightarrow \bigcirc\, \mathit{waitforP_2}(x)] \\ & \wedge & \Box\forall x \Diamond\neg \mathit{waitforP_2}(x). \end{array}$$

The parts of this formula form the following monodic temporal problem (we also rename the complex $P_2(x) \vee \mathit{waitforP_2}(x)$ expression by $P_5(x)$):

$$\begin{array}{rcl} \mathcal{I} & = & \left\{ \exists x P_3(x) \right\}, \\ \mathcal{U} & = & \left\{ \begin{array}{l} \forall x(P_2(x) \Rightarrow \forall y \forall z \exists u\, \Phi(x,y,z,u)), \\ \forall x(P_3(x) \Rightarrow P_4(x)), \\ \forall x(P_4(x) \Rightarrow P_1(x)), \\ \forall x((P_1(x) \wedge \neg P_2(x)) \Rightarrow \mathit{waitforP_2}(x)), \\ \forall x(P_5(x) \Rightarrow P_2(x) \vee \mathit{waitforP_2}(x)) \end{array} \right\}, \\ \mathcal{S} & = & \left\{ \begin{array}{l} P_4(x) \Rightarrow \bigcirc P_4(x), \\ \mathit{waitforP_2}(x) \Rightarrow \bigcirc P_5(x) \end{array} \right\} \\ \mathcal{E} & = & \left\{ \Diamond\neg \mathit{waitforP_2}(x) \right\}. \end{array}$$

4. TEMPORAL RESOLUTION

As in the propositional case [Fisher 1991; Degtyarev et al. 2002], our calculus works with *merged step clauses*, but here the notion of a merged step clause is much more complex. This is, of course, because of the first-order nature of the problem and the fact that skolemisation is not allowed under temporal operators. In order to build towards the calculus, we first provide some important definitions.

While the formal definitions of various different forms of clause are given below, it is useful to consider a simple example. Imagine we have, amongst out original set of step clauses, the three step clauses:

$$\begin{array}{rcl} P(x) & \Rightarrow & \bigcirc Q(x) \\ R(y) & \Rightarrow & \bigcirc S(y) \\ T(z) & \Rightarrow & \bigcirc U(z) \end{array}$$

From these clauses we can derive the ground step clauses

$$\begin{array}{rcl}\forall v(P(v) \vee R(v) \vee T(v)) & \Rightarrow & \bigcirc \forall w(Q(w) \vee S(w) \vee U(w)) \\ \exists v(P(v) \wedge R(v) \wedge T(v)) & \Rightarrow & \bigcirc \exists w(Q(w) \wedge S(w) \wedge U(w))\end{array}$$

Since we know the set of constants that can be used in the problem, we can also derive clauses of the form

$$P(c) \;\Rightarrow\; \bigcirc Q(c)$$

The above three types of clause are called *derived clauses*. We can then combine (conjoin) these derived clauses both with each other and with a conjunction of original ground step clauses. Such combinations are called *merged derived step clauses*. Finally, combining (again, conjoining) merged derived step clauses together with a conjunction of original step clauses gives us *full merged step clauses*. It is these that we will work with in general.

Definition 4.1 *Derived Step Clauses*. Let P be a monodic temporal problem, and let

$$P_{i_1}(x) \Rightarrow \bigcirc M_{i_1}(x), \dots, P_{i_k}(x) \Rightarrow \bigcirc M_{i_k}(x) \tag{4}$$

be a subset of the set of its original non-ground step clauses. Then

$$\forall x(P_{i_1}(x) \vee \cdots \vee P_{i_k}(x)) \;\Rightarrow\; \bigcirc \forall x(M_{i_1}(x) \vee \cdots \vee M_{i_k}(x)), \tag{5}$$

$$\exists x(P_{i_1}(x) \wedge \cdots \wedge P_{i_k}(x)) \;\Rightarrow\; \bigcirc \exists x(M_{i_1}(x) \wedge \cdots \wedge M_{i_k}(x)), \tag{6}$$

$$P_{i_j}(c) \;\Rightarrow\; \bigcirc M_{i_j}(c) \tag{7}$$

are *derived* step clauses, where $c \in \mathrm{const}(\mathsf{P})$ and $j = 1 \dots k$.

A derived step clause is a logical consequence of its premises obtained by “dividing” and bounding left-hand and right-hand sides.

Definition 4.2 *Merged Derived Step Clauses*. Let $\{\Phi_1 \Rightarrow \bigcirc \Psi_1, \dots, \Phi_n \Rightarrow \bigcirc \Psi_n\}$ be a set of derived step clauses or original *ground* step clauses. Then

$$\bigwedge_{i=1}^{n} \Phi_i \Rightarrow \bigcirc \bigwedge_{i=1}^{n} \Psi_i$$

is called a *merged derived step clause*.

Note that the left-hand and right-hand sides of any merged derived step clause are closed formulae.

Definition 4.3 *Full Merged Step Clauses*. Let $\mathcal{A} \;\Rightarrow\; \bigcirc \mathcal{B}$ be a merged derived step clause, $P_1(x) \Rightarrow \bigcirc M_1(x), \dots, P_k(x) \Rightarrow \bigcirc M_k(x)$ be original step clauses, and $A(x) \overset{\mathrm{def}}{=} \bigwedge_{i=1}^{k} P_i(x)$, $B(x) \overset{\mathrm{def}}{=} \bigwedge_{i=1}^{k} M_i(x)$. Then

$$\forall x(\mathcal{A} \wedge A(x) \Rightarrow \bigcirc(\mathcal{B} \wedge B(x)))$$

is called a *full merged step clause*. In the case $k = 0$, the conjunctions $A(x)$, $B(x)$ are empty, that is, their truth value is **true**, and the merged step clause is just a merged derived step clause.

Definition 4.4 *Constant Flooding*. Let P be a monodic temporal problem, $\mathsf{P}^c = \mathsf{P} \cup \{\Diamond L(c) \mid \Diamond L(x) \in \mathcal{E}, c \in \mathrm{const}(\mathsf{P})\}$ is the *constant flooded form*[3] of P.

Evidently, P^c is satisfiability equivalent to P.

Example 4.5. Let us consider a temporal problem given by

$$
\begin{array}{rcl}
\mathcal{I} & = & \left\{ \, i1. \quad Q(c) \, \right\}, \\
\mathcal{U} & = & \left\{ \begin{array}{ll} u1. & \exists x(P_1(x) \wedge P_2(x)) \\ u2. & \forall x(Q(x) \wedge \exists y(\neg P_1(y) \wedge \neg P_2(y)) \Rightarrow L(x)) \end{array} \right\}, \\
\mathcal{S} & = & \left\{ \begin{array}{lrcl} s1. & P_1(x) & \Rightarrow & \bigcirc \neg P_1(x) \\ s2. & P_2(x) & \Rightarrow & \bigcirc \neg P_2(x) \\ s3. & Q(x) & \Rightarrow & \bigcirc Q(x) \end{array} \right\} \\
\mathcal{E} & = & \left\{ \, e1. \quad \Diamond \neg L(x) \, \right\},
\end{array}
$$

Then

$$
\begin{array}{lrcl}
d1. & P_1(c) & \Rightarrow & \bigcirc \neg P_1(c), \\
d2. & \exists y P_1(y) & \Rightarrow & \bigcirc \exists y \neg P_1(y), \\
d3. & \forall y P_1(y) & \Rightarrow & \bigcirc \forall y \neg P_1(y), \\
d4. & \exists y(P_1(y) \wedge P_2(y)) & \Rightarrow & \bigcirc \exists y(\neg P_1(y) \wedge \neg P_2(y)) \\
d5. & \forall y(P_1(y) \vee P_2(y)) & \Rightarrow & \bigcirc \forall y(\neg P_1(y) \vee \neg P_2(y))
\end{array}
$$

are examples of derived step clauses. Every derived step clause is also a merged derived step clause. In addition,

$$
\begin{array}{lrcl}
m1. & P_1(c) \wedge \exists y P_1(y) & \Rightarrow & \bigcirc(\neg P_1(c) \wedge \exists y \neg P_1(y)), \\
m2. & \exists y P_1(y) \wedge \forall y P_1(y) & \Rightarrow & \bigcirc(\exists y \neg P_1(y) \wedge \forall y \neg P_1(y))
\end{array}
$$

are other examples of merged derived step clauses. Finally,

$$
\begin{array}{lrcl}
fm1. & \forall x(P_2(x) \wedge P_1(c) & \Rightarrow & \bigcirc(\neg P_2(x) \wedge \neg P_1(c))), \\
fm2. & \forall x(Q(x) \wedge \exists y(P_1(y) \wedge P_2(y)) & \Rightarrow & \bigcirc(Q(x) \wedge \exists y(\neg P_1(y) \wedge \neg P_2(y)))), \\
fm3. & \forall x(P_1(x) \wedge \exists y P_1(y) \wedge \forall y P_1(y) & \Rightarrow & \bigcirc(Q(x) \wedge \exists y \neg P_1(y) \wedge \forall y \neg P_1(y)))
\end{array}
$$

are examples of full merged step clauses.

Note that, constant flooding adds to the problem the eventuality $\Diamond \neg L(c)$.

Inference Rules. The inference system we use consists of the following inference rules. (Recall that the premises and conclusion of these rules are (implicitly) closed under the $\Box$ operator.) The conclusion of every rule is a first-order formula to be added to the universal part (see the definition of a derivation, Definition 4.11 below), where neither of the initial, step, or eventuality parts is changed by our rules.

In what follows, $\mathcal{A} \Rightarrow \bigcirc \mathcal{B}$ and $\mathcal{A}_i \Rightarrow \bigcirc \mathcal{B}_i$ denote merged derived step clauses, $\forall x(\mathcal{A} \wedge A(x) \Rightarrow \bigcirc(\mathcal{B} \wedge B(x)))$ and $\forall x(\mathcal{A}_i \wedge A_i(x) \Rightarrow \bigcirc(\mathcal{B}_i \wedge B_i(x)))$ denote full merged step clauses, and $\mathcal{U}$ denotes the (current) universal part of the problem.

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

[3] Strictly speaking, P^c is not in DSNF: we have to rename ground eventualities by propositions. Rather than "flooding", we could have introduced special inference rules to deal with constants.

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

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

$$\frac{\begin{array}{c}\forall x(\mathcal{A}_1 \wedge A_1(x) \Rightarrow \bigcirc(\mathcal{B}_1 \wedge B_1(x)))\\ \dots \\ \forall x(\mathcal{A}_n \wedge A_n(x) \Rightarrow \bigcirc(\mathcal{B}_n \wedge B_n(x)))\end{array} \qquad \Diamond L(x)}{\forall x \bigwedge_{i=1}^{n} (\neg \mathcal{A}_i \vee \neg A_i(x))} \ (\Diamond^{\mathcal{U}}_{res}),$$

where $\Diamond L(x)$ is a non-ground eventuality from $\mathcal{E}$ and $\forall x(\mathcal{A}_i \wedge A_i(x) \Rightarrow \bigcirc \mathcal{B}_i \wedge B_i(x))$ are full merged step clauses such that for all $i \in \{1, \dots, n\}$, the *loop* side conditions

$$\forall x(\mathcal{U} \wedge \mathcal{B}_i \wedge B_i(x) \Rightarrow \neg L(x)) \quad \text{and} \quad \forall x(\mathcal{U} \wedge \mathcal{B}_i \wedge B_i(x) \Rightarrow \bigvee_{j=1}^{n} (\mathcal{A}_j \wedge A_j(x))$$

are both valid.
The set of merged step clauses, satisfying the loop side conditions, is called a *loop in* $\Diamond L(x)$ and the formula $\bigvee\limits_{j=1}^{n} (\mathcal{A}_j(x) \wedge A_j(x))$ is called a *loop formula*.

(4) *Eventuality termination rule w.r.t. $\mathcal{U}$:* The contradiction $\perp$ is derived and the derivation is (successfully) terminated if $\mathcal{U} \models \forall x \neg L(x)$, where $\Diamond L(x) \in \mathcal{E}$[4].

(5) *Ground eventuality resolution rule w.r.t. $\mathcal{U}$:*

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

where $\mathcal{A}_i \Rightarrow \bigcirc \mathcal{B}_i$ are *merged grounded* step clauses such that the *ground* loop side conditions

$$\mathcal{U} \wedge \mathcal{B}_i \models \neg l \quad \text{and} \quad \mathcal{U} \wedge \mathcal{B}_i \models \bigvee_{j=1}^{n} \mathcal{A}_j \quad \text{for all} \quad i \in \{1, \dots, n\}$$

are satisfied.

(6) *Ground eventuality termination rule w.r.t. $\mathcal{U}$:*
The contradiction $\perp$ is derived and the derivation is (successfully) terminated if $\mathcal{U} \models \neg l$, where $\Diamond l \in \mathcal{E}$.

Note 4.6. In principle, the eventuality resolution and eventuality termination rules could handle both ground and non-ground eventualities. However, we consider their ground counterparts explicitly. Note that the ground eventuality resolution rule does not use *full* merged step clauses and can be considered, thus, as a specific *strategy* for the general eventuality resolution rule.

For a temporal problem P, by $\mathrm{TRes}(\mathsf{P})$ we denote the set of all formulae which can be obtained from P applying the inference rules above.

[4] In the case $\mathcal{U} \models \forall x \neg L(x)$, the *degenerate clause*, **true** $\Rightarrow \bigcirc$**true**, can be considered as a premise of the eventuality resolution rule; the conclusion of the rule is then $\neg$**true** and the derivation successfully terminates.

Note 4.7. The *eventuality resolution rule* above can be thought of as two separate rules: an induction rule to extract a formula of the form $\forall x(P(x) \Rightarrow \bigcirc \Box \neg L(x))$ and a resolution rule to resolve this with $\forall y \Diamond L(y)$, that is,

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

$$\frac{\begin{array}{c}\forall x(\mathcal{A}_1 \wedge A_1(x) \Rightarrow \bigcirc(\mathcal{B}_1 \wedge B_1(x))) \\ \dots \\ \forall x(\mathcal{A}_n \wedge A_n(x) \Rightarrow \bigcirc(\mathcal{B}_n \wedge B_n(x)))\end{array}}{\forall x(\bigvee_{i=1}^{n}(\mathcal{A}_i \wedge A_i(x)) \Rightarrow \bigcirc \Box \neg L(x))}\ (ind^{\mathcal{U}})\,,$$

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

The formula $\bigvee_{i=1}^{n}(\mathcal{B}_i \wedge B_i(x))$ can be considered as an *invariant formula* since, within the loop detected, this formula is always true.

—*Pure eventuality resolution:*

$$\frac{\forall x(\bigvee_{i=1}^{n}(\mathcal{A}_i \wedge A_i(x)) \Rightarrow \bigcirc \Box \neg L(x)) \qquad \Diamond L(x)}{\forall x \bigwedge_{i=1}^{n}(\neg \mathcal{A}_i \vee \neg A_i(x))}\ (\Diamond_{res})\,.$$

We see here that a classical first-order formula is generated; this is added to $\mathcal{U}$.

The *ground eventuality resolution rule* can be split into two parts in a similar way.

Example 4.8 *Example 4.5 contd..* We apply temporal resolution to the (unsatisfiable) temporal problem from Example 4.5. It can be immediately checked that the loop side conditions are valid for the full merged step clause $fm2$,

$$fm2. \quad \forall x(Q(x) \wedge \exists y(P_1(y) \wedge P_2(y)) \ \Rightarrow\ \bigcirc(Q(x) \wedge \exists y(\neg P_1(y) \wedge \neg P_2(y)))),$$

that is,

$$\begin{array}{ll}\exists y(\neg P_1(y) \wedge \neg P_2(y)) \wedge Q(x) \Rightarrow L(x) & \text{(see } u2\text{)}, \\ \exists y(\neg P_1(y) \wedge \neg P_2(y)) \wedge Q(x) \Rightarrow \exists y(P_1(y) \wedge P_2(y)) \wedge Q(x) & \text{(see } u1\text{)}.\end{array}$$

We apply the eventuality resolution rule to $e1$ and $m1$ and derive a new universal clause

$$nu1. \quad \Box \forall x(\neg(\exists y(P_1(y) \wedge P_2(y))) \vee \neg Q(x))$$

which contradicts clauses $u1$ and $i1$ (the initial termination rule is applied).

Example 4.9. The need for constant flooding can be demonstrated by the following example. None of the rules of temporal resolution can be applied directly to the (unsatisfiable) temporal problem given by

$$\begin{array}{ll}\mathcal{I} = \{P(c)\}, & \mathcal{S} = \{q \Rightarrow \bigcirc q\}, \\ \mathcal{U} = \{q \equiv P(c)\}, & \mathcal{E} = \{\Diamond \neg P(x)\}.\end{array}$$

If, however, we add to the problem an eventuality clause $\Diamond l$ and a universal clause $l \Rightarrow \neg P(c)$, the step clause $q \Rightarrow \bigcirc q$ will be a loop in $\Diamond l$, and the eventuality resolution rule would derive $\neg$**true**[5].

Correctness of the presented calculi is straightforward.

THEOREM 4.10 SOUNDNESS OF TEMPORAL RESOLUTION. *The rules of temporal resolution preserve satisfiability.*

Proof Considering models for FOTL formulae, it can be shown that the temporal resolution rules preserve satisfiability. Let $\mathfrak{M} = \langle D, I \rangle$ be a temporal structure and $\mathfrak{a}$ be a variable assignment. We assume that a temporal problem P is true in $\mathfrak{M}$ under the assignment $\mathfrak{a}$ and show that P, extended with the conclusion of a temporal resolution rule, is true in $\mathfrak{M}$ under $\mathfrak{a}$. We do this by considering cases of the inference rule used, as follows.

—Consider the step resolution rule. Let $\mathcal{A} \Rightarrow \bigcirc\mathcal{B}$ be a merged derived clause and assume that $\mathfrak{M}_0 \models^{\mathfrak{a}} \Box(\mathcal{A} \Rightarrow \bigcirc\mathcal{B})$, $\mathcal{U} \cup \mathcal{B} \models \perp$, but for some $i \geq 0$, $\mathfrak{M}_i \not\models^{\mathfrak{a}} \neg\mathcal{A}$. Then $\mathfrak{M}_{i+1} \models^{\mathfrak{a}} \mathcal{B}$ in contradiction with the side condition of the rule.

—Consider now the eventuality resolution rule. Let $\forall x(\mathcal{A}_i \wedge A_i(x) \Rightarrow \bigcirc\mathcal{B}_i \wedge B_i(x))$, $i \in \{1, \ldots, n\}$, be full merged step clauses and $\Diamond L(x)$ be an eventuality such that $\mathfrak{M}_0 \models^{\mathfrak{a}} \bigwedge\limits_{i=1}^{n} \forall x(\mathcal{A}_i \wedge A_i(x) \Rightarrow \bigcirc\mathcal{B}_i \wedge B_i(x))$, $\mathfrak{M}_0 \models^{\mathfrak{a}} \Box\forall x \Diamond L(x)$, and the loop side conditions $\forall x(\mathcal{U} \wedge \mathcal{B}_i \wedge B_i(x) \Rightarrow \neg L(x))$ and $\forall x(\mathcal{U} \wedge \mathcal{B}_i \wedge B_i(x) \Rightarrow \bigvee\limits_{j=1}^{n} (\mathcal{A}_j \wedge A_j(x))$ are both valid, but for some $k \geq 0$, $\mathfrak{M}_k \not\models^{\mathfrak{a}} \forall x \bigwedge\limits_{i=1}^{n} (\neg\mathcal{A}_i \vee \neg A_i(x))$. It follows there exists a domain element $d \in D$ such that $\mathfrak{M}_k \models^{\mathfrak{a}} (\mathcal{A}_j \wedge A_j(d))$. It is not hard to see that, by validity of the loop side conditions and by the fact that the full merged clauses are true in $\mathfrak{M}$ under $\mathfrak{a}$, $\mathfrak{M}_l \models^{\mathfrak{a}} \neg L(d)$ for all $l > k$, that is, $\mathfrak{M}_{k+1} \models^{\mathfrak{a}} \Box\neg L(d)$ in contradiction with the eventuality.

—Correctness of the initial termination and eventuality termination rules is obvious.

—Correctness of the ground counterparts of the eventuality resolution and eventuality termination rules can be proved in a similar way.

❏

Similarly to classical first-order resolution, temporal resolution is a refutationally complete saturation-based theorem proving method, i.e., a contradiction can be deduced from any unsatisfiable problem, and the search for a contradiction proceeds by saturation the universal part of a given problem.

Definition 4.11 *Derivation*. A *derivation* is a sequence of universal parts, $\mathcal{U} = \mathcal{U}_0 \subset \mathcal{U}_1 \subset \mathcal{U}_2 \subset \ldots$, extended little by little by the conclusions of the inference rules. The $\mathcal{I}$, $\mathcal{S}$ and $\mathcal{E}$ parts of the temporal problem are not changed during a derivation.

A derivation *terminates* if, and only if, either the contradiction is derived, in which case we say that the derivation *successfully terminates*, or if no new formulae can be derived by further inference steps. Note that since there exist only finitely many different full

[5] Note that the non-ground eventuality $\Diamond\neg P(x)$ is not used. We show in Section 7 that if all step clauses are ground, for constant flooded problems we can neglect non-ground eventualities.

merged step clauses, the number of different conclusions of the inference rules of temporal resolution is finite. Therefore, every derivation is finite. If a (finite) derivation does not terminate, we call it *partial*. Any partial derivation can be continued yielding a terminating derivation.

We adopt the notion of a fair derivation from [Bachmair and Ganzinger 2001].

Definition 4.12 *Fair derivation*. A derivation $\mathcal{U} = \mathcal{U}_0 \subset \mathcal{U}_1 \subset \mathcal{U}_2 \subset \cdots \subset \mathcal{U}_n$ is called *fair* if for any $i \geq 0$ and formula $\phi \in \mathrm{TRes}(\langle \mathcal{U}_i, \mathcal{I}, \mathcal{S}, \mathcal{E} \rangle)$, there exists $j \geq i$ such that $\phi \in \mathcal{U}_j$.

We formulate now the completeness result and prove it in Section 5, which is entirely devoted to this issue.

THEOREM 4.13 COMPLETNESS OF TEMPORAL RESOLUTION. *Let an arbitrary monodic temporal problem* P *be unsatisfiable. Then any* fair *derivation by temporal resolution from* P^c *successfully terminates.*

5. COMPLETENESS OF TEMPORAL RESOLUTION

In short, the proof of Theorem 4.13 proceeds by building a graph associated with a monodic temporal problem, then showing that there is a correspondence between properties of the graph and of the problem, and that equivalent properties are captured by the rules of the proof system. Therefore, if the problem is unsatisfiable, eventually our rules will discover it.

First, we introduce additional concepts. Let $\mathsf{P} = \langle \mathcal{U}, \mathcal{I}, \mathcal{S}, \mathcal{E} \rangle$ be a monodic temporal problem. Let $\{P_1, \ldots, P_N\}$ and $\{p_1, \ldots, p_n\}$, $N, n \geq 0$, be the sets of all (monadic) predicate symbols and all propositional symbols, respectively, occurring in $\mathcal{S} \cup \mathcal{E}$.

A *predicate colour* γ is a set of unary literals such that for every $P_i(x) \in \{P_1(x), \ldots, P_N(x)\}$, either $P_i(x)$ or $\neg P_i(x)$ belongs to γ. A *propositional colour* θ is a set of propositional literals such that for every $p_i \in \{p_1, \ldots, p_n\}$, either p_i or $\neg p_i$ belongs to θ. Let Γ be a set of predicate colours, θ be a propositional colour, and ρ be a map from the set of constants, $\mathrm{const}(\mathsf{P})$, to Γ. A triple $\langle \Gamma, \theta, \rho \rangle$ is called a *colour scheme*, and ρ is called a *constant distribution*. We write sometime $\gamma \in \mathcal{C}$ when $\gamma \in \Gamma$ and $\mathcal{C} = \langle \Gamma, \theta, \rho \rangle$.

Note 5.1. The notion of colour scheme came, of course, from the well known concept used in the decidability proof for the monadic class in classical first-order logic (see, for example, [Börger et al. 1997]). In our case, Γ is the quotient domain (a subset of all possible equivalence classes of predicate values), θ is a propositional valuation, and ρ is a standard interpretation of constants in the domain Γ. We construct quotient structures based only on the predicates and propositions which occur in the temporal part of the problem, since only these symbols are really responsible for the satisfiability (or unsatisfiability) of temporal constraints. In addition, we have to consider so-called constant distributions because, unlike in the classical case, we cannot eliminate constants replacing them by existentially bound variables since in doing this the monodicity property would be lost.

For every colour scheme $\mathcal{C} = \langle \Gamma, \theta, \rho \rangle$ let us construct the formulae $\mathcal{F}_{\mathcal{C}}$, $\mathcal{A}_{\mathcal{C}}$, $\mathcal{B}_{\mathcal{C}}$ in the following way. For every $\gamma \in \Gamma$ and for every θ, introduce the conjunctions:

$$F_\gamma(x) = \bigwedge_{L(x) \in \gamma} L(x); \qquad F_\theta = \bigwedge_{l \in \theta} l.$$

Let

$$\begin{array}{lcl} A_\gamma(x) & = & \bigwedge\{L(x) \mid L(x) \Rightarrow \bigcirc M(x) \in \mathcal{S},\ L(x) \in \gamma\}, \\ B_\gamma(x) & = & \bigwedge\{M(x) \mid L(x) \Rightarrow \bigcirc M(x) \in \mathcal{S},\ L(x) \in \gamma\}, \\ A_\theta & = & \bigwedge\{l \mid l \Rightarrow \bigcirc m \in \mathcal{S},\ l \in \theta\}, \\ B_\theta & = & \bigwedge\{m \mid l \Rightarrow \bigcirc m \in \mathcal{S},\ l \in \theta\}. \end{array}$$

(Recall that there are no two different step clauses with the same left-hand side.)
Now $\mathcal{F_C}$, $\mathcal{A_C}$, $\mathcal{B_C}$ are of the following forms:

$$\begin{array}{rcl} \mathcal{F_C} & = & \bigwedge\limits_{\gamma\in\Gamma} \exists x F_\gamma(x) \wedge F_\theta \wedge \bigwedge\limits_{c\in\mathrm{const}(P)} F_{\rho(c)}(c) \wedge \forall x \bigvee\limits_{\gamma\in\Gamma} F_\gamma(x), \\ \mathcal{A_C} & = & \bigwedge\limits_{\gamma\in\Gamma} \exists x A_\gamma(x) \wedge A_\theta \wedge \bigwedge\limits_{c\in\mathrm{const}(P)} A_{\rho(c)}(c) \wedge \forall x \bigvee\limits_{\gamma\in\Gamma} A_\gamma(x), \\ \mathcal{B_C} & = & \bigwedge\limits_{\gamma\in\Gamma} \exists x B_\gamma(x) \wedge B_\theta \wedge \bigwedge\limits_{c\in\mathrm{const}(P)} B_{\rho(c)}(c) \wedge \forall x \bigvee\limits_{\gamma\in\Gamma} B_\gamma(x). \end{array}$$

We can consider the formula $\mathcal{F_C}$ as a "categorical" formula specification of the quotient structure given by a colour scheme. In turn, the formula $\mathcal{A_C}$ represents the part of this specification which is "responsible" just for "transferring" requirements from the current world (quotient structure) to its immediate successors, and $\mathcal{B_C}$ represents the result of transferal.

Example 5.2. Consider a monodic temporal problem, P, given by

$$\begin{array}{lcllcl} \mathcal{I} & = & \emptyset, & \mathcal{S} & = & \{P(x) \Rightarrow \bigcirc P(x)\}, \\ \mathcal{U} & = & \{l \Rightarrow \exists x P(x)\}, & \mathcal{E} & = & \{\Diamond\neg P(x), \Diamond l\}. \end{array}$$

For this problem, there exist two predicate colours, $\gamma_1 = [P(x)]$ and $\gamma_2 = [\neg P(x)]$; two propositional colours $\theta_1 = [l]$ and $\theta_2 = [\neg l]$; and six colour schemes (we omit the empty constant distribution for readability),

$$\begin{array}{lcllcl} \mathcal{C}_1 & = & (\{\gamma_1\}, \theta_1), & \mathcal{C}_4 & = & (\{\gamma_1\}, \theta_2), \\ \mathcal{C}_2 & = & (\{\gamma_2\}, \theta_1), & \mathcal{C}_5 & = & (\{\gamma_2\}, \theta_2), \\ \mathcal{C}_3 & = & (\{\gamma_1, \gamma_2\}, \theta_1), & \mathcal{C}_6 & = & (\{\gamma_1, \gamma_2\}, \theta_2). \end{array}$$

The categorical formulae for these colour schemes are the following:

$\mathcal{F}_{\mathcal{C}_1} = \exists x P(x) \wedge \forall x P(x) \wedge l$	$\mathcal{A}_{\mathcal{C}_1} = \exists x P(x) \wedge \forall x P(x)$	$\mathcal{B}_{\mathcal{C}_1} = \exists x P(x) \wedge \forall x P(x)$
$\mathcal{F}_{\mathcal{C}_2} = \exists x \neg P(x) \wedge \forall x \neg P(x) \wedge l$	$\mathcal{A}_{\mathcal{C}_2} =$ **true**	$\mathcal{B}_{\mathcal{C}_2} =$ **true**
$\mathcal{F}_{\mathcal{C}_3} = \exists x P(x) \wedge \exists x \neg P(x) \wedge l$	$\mathcal{A}_{\mathcal{C}_3} = \exists x P(x)$	$\mathcal{B}_{\mathcal{C}_3} = \exists x P(x)$
$\mathcal{F}_{\mathcal{C}_4} = \exists x P(x) \wedge \forall x P(x) \wedge \neg l$	$\mathcal{A}_{\mathcal{C}_4} = \exists x P(x) \wedge \forall x P(x)$	$\mathcal{B}_{\mathcal{C}_4} = \exists x P(x) \wedge \forall x P(x)$
$\mathcal{F}_{\mathcal{C}_5} = \exists x \neg P(x) \wedge \forall x \neg P(x) \wedge \neg l$	$\mathcal{A}_{\mathcal{C}_5} =$ **true**	$\mathcal{B}_{\mathcal{C}_5} =$ **true**
$\mathcal{F}_{\mathcal{C}_6} = \exists x P(x) \wedge \exists x \neg P(x) \wedge \neg l$	$\mathcal{A}_{\mathcal{C}_6} = \exists x P(x)$	$\mathcal{B}_{\mathcal{C}_6} = \exists x P(x)$

Definition 5.3 *Canonical Merged Derived Step Clauses*. Let P be a first-order temporal problem, $\mathcal{C}$ be a colour scheme for P. Then the clause

$$(\mathcal{A_C} \Rightarrow \bigcirc \mathcal{B_C}),$$

is called a *canonical merged derived step clause* for P.

If all conjunctions in $\mathcal{A_C}$ are empty, which implies all conjunctions in $\mathcal{B_C}$ are empty and vice versa, the truth value of both $\mathcal{A_C}$ and $\mathcal{B_C}$ is **true**, and the clause $(\mathcal{A_C} \Rightarrow \bigcirc \mathcal{B_C})$ *degenerates* to (**true** $\Rightarrow \bigcirc$**true**). If a conjunction $A_\gamma(x)$, $\gamma \in \Gamma$, is empty (which also implies the conjunction $B_\gamma(x)$ is empty and vice versa) then the formula $\forall x \bigvee_{\gamma\in\Gamma} A_\gamma(x)$

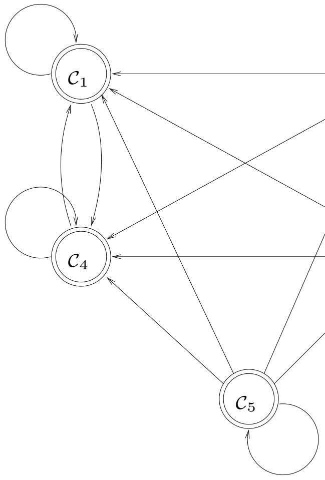

Fig. 1. Behaviour graph for the problem $\mathcal{I} = \emptyset, \mathcal{U} = \{l \Rightarrow \exists x P(x)\}$, $\mathcal{S} = \{P(x) \Rightarrow \bigcirc P(x)\}$, $\mathcal{E} = \{\Diamond\neg P(x), \Diamond l\}$ (Example 5.6).

(and $\forall x \bigvee_{\gamma\in\Gamma} B_\gamma(x)$) disappears from $\mathcal{A}_\mathcal{C}$ (from $\mathcal{B}_\mathcal{C}$ respectively). In the propositional case, the clause $(\mathcal{A}_\mathcal{C} \Rightarrow \bigcirc\mathcal{B}_\mathcal{C})$ reduces to $(A_\theta \Rightarrow \bigcirc B_\theta)$.

Definition 5.4 *Canonical Merged Step Clause*. Let $\mathcal{C}$ be a colour scheme, $\mathcal{A}_\mathcal{C} \Rightarrow \bigcirc\mathcal{B}_\mathcal{C}$ be a canonical merged derived step clause, and $\gamma \in \mathcal{C}$.

$$\forall x(\mathcal{A}_\mathcal{C} \wedge A_\gamma(x) \Rightarrow \bigcirc(\mathcal{B}_\mathcal{C} \wedge B_\gamma(x)))$$

is called a *canonical merged step clause*.

If the truth value of the conjunctions $A_\gamma(x)$, $B_\gamma(x)$ is **true**, the canonical merged step clause is just a canonical merged derived step clause.

Definition 5.5 *Behaviour Graph*. Now, given a temporal problem $\mathsf{P} = \langle\mathcal{U}, \mathcal{I}, \mathcal{S}, \mathcal{E}\rangle$ we define a finite directed graph G as follows. Every vertex of G is a colour scheme $\mathcal{C}$ for P such that $\mathcal{U} \cup \mathcal{F}_\mathcal{C}$ is satisfiable. For each vertex $\mathcal{C} = \langle\Gamma, \theta, \rho\rangle$, there is an edge in G to $\mathcal{C}' = \langle\Gamma', \theta', \rho'\rangle$, if $\mathcal{U} \wedge \mathcal{F}_{\mathcal{C}'} \wedge \mathcal{B}_\mathcal{C}$ is satisfiable. They are the only edges originating from $\mathcal{C}$.
A vertex $\mathcal{C}$ is designated as an *initial* vertex of G if $\mathcal{I} \wedge \mathcal{U} \wedge \mathcal{F}_\mathcal{C}$ is satisfiable.
The *behaviour graph* H of P is the subgraph of G induced by the set of all vertices reachable from the initial vertices.

Example 5.6 *Example 5.2 contd.*. Let us construct the behaviour graph for the problem given in Example 5.2. Note that $\mathcal{F}_{\mathcal{C}_2} \wedge \mathcal{U} \models\perp$, so the vertex $\mathcal{C}_2$ is not in the graph. The behaviour graph for P, given in Fig. 1, consists of five vertices; all of them are initial.

There is an edge in the graph from the node $\mathcal{C}_3$ to the node $\mathcal{C}_1$ since the formula $\mathcal{U} \wedge \mathcal{F}_{\mathcal{C}_1} \wedge \mathcal{B}_{\mathcal{C}_3}$,

$$\underbrace{l \Rightarrow \exists x P(x)}_{\mathcal{U}} \wedge \underbrace{\exists x P(x) \wedge \forall x P(x) \wedge l}_{\mathcal{F}_{\mathcal{C}_1}} \wedge \underbrace{\exists x P(x)}_{\mathcal{B}_{\mathcal{C}_3}},$$

is satisfiable. There is no edge from $\mathcal{C}_1$ to $\mathcal{C}_3$ since the formula $\mathcal{U} \wedge \mathcal{F}_{\mathcal{C}_3} \wedge \mathcal{B}_{\mathcal{C}_1}$,

$$\underbrace{l \Rightarrow \exists x P(x)}_{\mathcal{U}} \wedge \underbrace{\exists x P(x) \wedge \exists x \neg P(x) \wedge l}_{\mathcal{F}_{\mathcal{C}_3}} \wedge \underbrace{\exists x P(x) \wedge \forall x P(x)}_{\mathcal{B}_{\mathcal{C}_1}}$$

is unsatisfiable. Other edges are considered in a similar way.

LEMMA 5.7. *Let $\mathsf{P}_1 = \langle \mathcal{U}_1, \mathcal{I}, \mathcal{S}, \mathcal{E} \rangle$ and $\mathsf{P}_2 = \langle \mathcal{U}_2, \mathcal{I}, \mathcal{S}, \mathcal{E} \rangle$ be two problems over the same set of symbols, such that $\mathcal{U}_1 \subset \mathcal{U}_2$. Then the behaviour graph of P_2 is a subgraph of the behaviour graph of P_1.*

Proof Satisfiability of $\mathcal{U}_2$ implies satisfiability of $\mathcal{U}_1$. ❑

Definition 5.8 *Path; Path Segment*. A *path*, π, through a behaviour graph, H, is a function from $\mathbb{N}$ to the vertices of the graph such that for any $i \geq 0$ there is an edge $\langle \pi(i), \pi(i+1) \rangle$ in H. In a similar way, we define a *path segment* as a function from $[m, n]$, $m < n$, to the vertices of H with the same property.

Recall that vertices of the behaviour graph of a problem, P, are quotient representations of "intermediate" interpretations $\mathfrak{M}_n$ in possible models of P. Intuitively, if a pair of vertices, or of colour schemes, $\mathcal{C}$ and $\mathcal{C}'$ is suitable, then this pair can represent adjacent interpretations $\mathfrak{M}_i$ and $\mathfrak{M}_{i+1}$ in a model of P. The definition of predicate colour suitability given below expresses the condition when a pair of predicate colours specify an element in adjacent interpretations with regard to the step part of P. A similar intuition is behind the notions of suitable propositional colours and suitable constant distributions.

Definition 5.9 *Suitability*. For $\mathcal{C} = \langle \Gamma, \theta, \rho \rangle$ and $\mathcal{C}' = \langle \Gamma', \theta', \rho' \rangle$, let $(\mathcal{C}, \mathcal{C}')$ be an ordered pair of colour schemes for a temporal problem P.
An ordered pair of predicate colours $(\gamma, \gamma\,')$ where $\gamma \in \Gamma$, $\gamma\,' \in \Gamma'$ is called *suitable* if the formula $\mathcal{U} \wedge \exists x (F_{\gamma\,'}(x) \wedge B_\gamma(x))$ is satisfiable;
Similarly, an ordered pair of propositional colours (θ, θ') is suitable if $\mathcal{U} \wedge F_{\theta'} \wedge B_\theta$ is satisfiable; and
an ordered pair of constant distributions (ρ, ρ') is suitable if, for every $c \in C$, the pair $(\rho(c), \rho'(c))$ is suitable.

When the graph is clear from the context, we denote suitable pairs by connecting them with an arrow, for example, if a pair of predicate colours (γ, γ') is suitable, we denote it by $\gamma \rightarrow \gamma'$.

Note that the satisfiability of $\exists x (F_{\gamma\,'}(x) \wedge B_\gamma(x))$ implies $\models \forall x (F_{\gamma\,'}(x) \Rightarrow B_\gamma(x))$ as the conjunction $F_{\gamma\,'}(x)$ contains a valuation at x of *all* predicates occurring in $B_\gamma(x)$.

LEMMA 5.10. *Let H be the behaviour graph for the problem $\mathsf{P} = \langle \mathcal{U}, \mathcal{I}, \mathcal{S}, \mathcal{E} \rangle$ with an edge from a vertex $\mathcal{C} = \langle \Gamma, \theta, \rho \rangle$ to a vertex $\mathcal{C}' = \langle \Gamma', \theta', \rho' \rangle$. Then*

(*1*) *for every $\gamma \in \Gamma$ there exists a $\gamma\,' \in \Gamma'$ such that the pair $(\gamma, \gamma\,')$ is suitable;*

(*2*) *for every $\gamma\,' \in \Gamma'$ there exists a $\gamma \in \Gamma$ such that the pair $(\gamma, \gamma\,')$ is suitable;*

(*3*) *the pair of propositional colours (θ, θ') is suitable;*

(*4*) *the pair of constant distributions (ρ, ρ') is suitable.*

Proof From the definition of a behaviour graph it follows that $\mathcal{U} \wedge \mathcal{F}_{\mathcal{C}'} \wedge \mathcal{B}_{\mathcal{C}}$ is satisfiable. Now to prove the first item it is enough to note that satisfiability of the expression

$\mathcal{U} \wedge \mathcal{F}_{\mathcal{C}'} \wedge \mathcal{B}_{\mathcal{C}}$ implies satisfiability of $\mathcal{U} \wedge (\forall x \bigvee_{\gamma' \in \Gamma'} F_{\gamma'}(x)) \wedge \exists x B_\gamma(x)$. This, in turn, implies satisfiability of its logical consequence $\mathcal{U} \wedge \bigvee_{\gamma' \in \Gamma'} \exists x (F_{\gamma'}(x) \wedge B_\gamma(x))$. So, one of the members of this disjunction must be satisfiable. The second item follows from the satisfiability of $\mathcal{U} \wedge (\forall x \bigvee_{\gamma \in \Gamma} B_\gamma(x)) \wedge \exists x F_{\gamma'}(x)$. Other items are similar. ❑

Example 5.11 *Example 5.6 cont.*. Let us consider suitability of predicate and propositional colours from Example 5.2.
Since the formula $\mathcal{U} \wedge \exists x (F_{\gamma_1}(x) \wedge B_{\gamma_2}(x))$, where $\mathcal{U} = \{l \Rightarrow \exists x P(x)\}$, $F_{\gamma_1} = P(x)$, and $B_{\gamma_2} =$ **true**, is satisfiable, the pair (γ_1, γ_2) is suitable.
Since the formula $\mathcal{U} \wedge \exists x (F_{\gamma_2}(x) \wedge B_{\gamma_1}(x))$, where $\mathcal{U} = \{l \Rightarrow \exists x P(x)\}$, $F_{\gamma_2} = \neg P(x)$, and $B_{\gamma_1} = P(x)$, is unsatisfiable, the pair (γ_2, γ_1) is not suitable.
In a similar way, it can be easily checked that the pairs of predicate colours

$$(\gamma_1, \gamma_1) \quad \text{and} \quad (\gamma_2, \gamma_2),$$

and the pairs of propositional colours

$$(\theta_1, \theta_1), \quad (\theta_1, \theta_2), \quad (\theta_2, \theta_1), \text{ and } \quad (\theta_2, \theta_2)$$

are suitable.

Let H be the behaviour graph for a temporal problem $\mathsf{P} = \langle \mathcal{U}, \mathcal{I}, \mathcal{S}, \mathcal{E} \rangle$ and $\pi = \mathcal{C}_0, \ldots, \mathcal{C}_n, \ldots$ be a path in H where $\mathcal{C}_i = \langle \Gamma_i, \theta_i, \rho_i \rangle$. Let $\mathcal{G}_0 = \mathcal{I} \cup \{\mathcal{F}_{\mathcal{C}_o}\}$ and $\mathcal{G}_n = \mathcal{F}_{\mathcal{C}_n} \wedge \mathcal{B}_{\mathcal{C}_{n-1}}$ for $n \geq 1$. According to the definition of a behaviour graph, the set $\mathcal{U} \cup \{\mathcal{G}_n\}$ is satisfiable for every $n \geq 0$.

From classical model theory, since the language $\mathcal{L}$ is countable and does not contain equality, the following lemma holds.

LEMMA 5.12. *Let κ be a cardinal, $\kappa \geq \aleph_0$. For every $n \geq 0$, if the set $\mathcal{U} \cup \{\mathcal{G}_n\}$ is satisfiable then there exists an $\mathcal{L}$-model $\mathfrak{M}_n = \langle D, I_n \rangle$ of $\mathcal{U} \cup \{\mathcal{G}_n\}$ such that for every $\gamma \in \Gamma_n$ the set $D_{(n,\gamma)} = \{a \in D \mid \mathfrak{M}_n \models F_\gamma(a)\}$ is of cardinality κ.*

Definition 5.13 *Run/E-Run*. Let π be a path through a behaviour graph H of a temporal problem P, and $\pi(i) = \langle \Gamma_i, \theta_i, \rho_i \rangle$. By a *run in* π we mean a function $r(n)$ from $\mathbb{N}$ to $\bigcup_{i \in \mathbb{N}} \Gamma_i$ such that for every $n \in \mathbb{N}$, $r(n) \in \Gamma_n$ and the pair $(r(n), r(n+1))$ is suitable. In a similar way, we define a *run segment* as a function from $[m, n]$, $m < n$, to $\bigcup_{i \in \mathbb{N}} \Gamma_i$ with the same property.

A run r is called an *e-run* if for all $i \geq 0$ and for every non-ground eventuality $\Diamond L(x) \in \mathcal{E}$ there exists $j > i$ such that $L(x) \in r(j)$.

Let π be a path, the set of all runs in π is denoted by $\mathcal{R}(\pi)$, and the set of all e-runs in π is denoted by $\mathcal{R}_e(\pi)$. If π is clear, we may omit it.

Here $(\mathcal{C}, \gamma) \to^+ (\mathcal{C}', \gamma')$ denotes that there exists a path π from $\mathcal{C}$ to $\mathcal{C}'$ such that γ and γ' belong to a run in π; and $\mathcal{C} \to^+ \mathcal{C}'$ denotes that there exists a path from $\mathcal{C}$ to $\mathcal{C}'$.

Example 5.14. $\pi = \mathcal{C}_3, \mathcal{C}_6, \mathcal{C}_3, \mathcal{C}_6, \ldots$ is a path through the behaviour graph given in Fig. 1. $r_1 = \gamma_1, \gamma_1, \gamma_1, \ldots$ and $r_2 = \gamma_1, \gamma_2, \gamma_1, \gamma_2, \ldots$ are both runs in π. r_2 is an e-run, but r_1 is not.

We now relate properties of the behaviour graph for a problem to the satisfiability of the problem.

THEOREM 5.15 EXISTENCE OF A MODEL. *Let $\mathsf{P} = \langle \mathcal{U}, \mathcal{I}, \mathcal{S}, \mathcal{E} \rangle$ be a temporal problem. Let H be the behaviour graph of P, let $\mathcal{C}$ and $\mathcal{C}'$ be vertices of H such that $\mathcal{C} = \langle \Gamma, \theta, \rho \rangle$ and $\mathcal{C}' = \langle \Gamma', \theta', \rho' \rangle$. If both the set of initial vertices of H is non-empty and the following conditions hold*

(1) *For every vertex $\mathcal{C}$, predicate colour $\gamma \in \Gamma$, and non-ground eventuality $\Diamond L(x) \in \mathcal{E}$ there exist a vertex $\mathcal{C}'$ and a predicate colour $\gamma' \in \Gamma'$ such that*

$$\big((\mathcal{C}, \gamma) \to^{+} (\mathcal{C}', \gamma') \wedge L(x) \in \gamma' \big) ;$$

(2) *For every vertex $\mathcal{C}$, constant $c \in \mathrm{const}(\mathsf{P})$, and non-ground eventuality $\Diamond L(x) \in \mathcal{E}$, there exists a vertex $\mathcal{C}'$ such that*

$$\big(\mathcal{C} \to^{+} \mathcal{C}' \wedge L(x) \in \rho'(c) \big) ;$$

(3) *For every vertex $\mathcal{C}$ and ground eventuality $\Diamond l \in \mathcal{E}$, there exists a vertex $\mathcal{C}'$ such that*

$$\big(\mathcal{C} \to^{+} \mathcal{C}' \wedge l \in \theta' \big)$$

then P has a model.

The proof proceeds as follows. First, we provide a lemma showing that, under the conditions of Theorem 5.15, there exists a path through the behaviour graph satisfying certain properties, and then we show that, if such a path exists, then the problem has a model.

LEMMA 5.16. *Under the conditions of Theorem 5.15, there exists a path π through H where:*

(a) *$\pi(0)$ is an initial vertex of H;*

(b) *for every colour scheme $\mathcal{C} = \pi(i),\ i \geq 0$, and every ground eventuality literal $\Diamond l \in \mathcal{E}$ there exists a colour scheme $\mathcal{C}' = \pi(j)$, $j > i$, such that $l \in \theta'$;*

(c) *for every colour scheme $\mathcal{C} = \pi(i),\ i \geq 0$ and every predicate colour γ from the colour scheme there exists an e-run $r \in \mathcal{R}_e(\pi)$ such that $r(i) = \gamma$; and*

(d) *for every constant $c \in \mathcal{L}$, the function $r_c(n)$ defined by $r_c(n) = \rho_n(c)$, where ρ_n is the constant distribution from $\pi(n)$, is an e-run in π.*

Proof [of Lemma 5.16] Let $\Diamond L_1(x), \ldots, \Diamond L_k(x)$ be all non-ground eventuality literals from $\mathcal{E}$; $\Diamond l_1, \ldots, \Diamond l_p$ be all ground eventuality literals from $\mathcal{E}$; and $c_1, \ldots, c_q$ be all constants of P. Let $\mathcal{C}_0$ be an initial vertex of H. We construct the path π as follows. Let $\{\gamma_1, \ldots, \gamma_{s_0}\}$ be all predicate colours from $\Gamma_{\mathcal{C}_0}$. By condition (1) there exists a vertex $\mathcal{C}_0^{(\gamma_1, L_1)}$ and a predicate colour $\gamma_1^{(1)} \in \Gamma_{\mathcal{C}_0^{(\gamma_1, L_1)}}$ such that $(\mathcal{C}_0, \gamma_1) \to^{+} (\mathcal{C}_0^{(\gamma_1, L_1)}, \gamma_1^{(1)})$ and $L_1(x) \in \gamma_1^{(1)}$. In the same way, there exists a vertex $\mathcal{C}_0^{(\gamma_1, L_2)}$ and a predicate colour $\gamma_1^{(2)} \in \Gamma_{\mathcal{C}_0^{(\gamma_1, L_2)}}$ such that $(\mathcal{C}_0^{(\gamma_1, L_1)}, \gamma_1^{(1)}) \to^{+} (\mathcal{C}_0^{(\gamma_1, L_2)}, \gamma_1^{(2)})$ and $L_2(x) \in \gamma_1^{(2)}$. And so on. Finally, there exists a vertex $\mathcal{C}_0^{(\gamma_1, L_k)}$ and a predicate colour $\gamma_1^{(k)} \in \Gamma_{\mathcal{C}_0^{(\gamma_1, L_k)}}$ such that $(\mathcal{C}_0^{(\gamma_1, L_{k-1})}, \gamma_1^{(k-1)}) \to^{+} (\mathcal{C}_0^{(\gamma_1, L_k)}, \gamma_1^{(k)})$ and $L_k(x) \in \gamma_1^{(k)}$. Clearly, γ_1, $\ldots$,$\gamma_1^{(1)}$,$\ldots$,$\gamma_1^{(2)}$,$\ldots$,$\gamma_1^{(k)}$ forms a segment of a run and every non-ground eventuality is satisfied along this segment.

Now, let $\gamma_2^{(0)}$ be any successor of γ_2 in $\Gamma_{\mathcal{C}_0^{(\gamma_1, L_k)}}$. As above, there exists a sequence of vertices $\mathcal{C}_0^{(\gamma_2, L_1)}$,$\ldots$, $\mathcal{C}_0^{(\gamma_2, L_k)}$ and a sequence of predicate colours $\gamma_2^{(1)} \in \Gamma_{\mathcal{C}_0^{(\gamma_2, L_1)}}$,$\ldots$,

$\gamma_2^{(k)} \in \Gamma_{\mathcal{C}_0^{(\gamma_2, L_k)}}$ such that $\gamma_2, \ldots, \gamma_2^{(0)}, \ldots, \gamma_2^{(1)}, \ldots, \gamma_2^{(k)}$ forms a segment of a run and every non-ground eventuality is satisfied along this segment. Continue this construction. At a certain point we construct a segment of a path from $\mathcal{C}_0$ to a vertex $\mathcal{C}_0^{(\gamma_{s_0}, L_k)}$ such that for every $\gamma \in \mathcal{C}_0$ there exists $\gamma' \in \mathcal{C}_0^{(\gamma_{s_0}, L_k)}$ such that all eventualities are satisfied on the run-segment from γ to γ'.

In a similar way we can construct a vertex $\mathcal{C}_0^{(c_1, L_1)}$ such that $\mathcal{C}_0^{(\gamma_{s_0}, L_k)} \to^+ \mathcal{C}_0^{(c_1, L_1)}$ and $L_1(x) \in \rho_{\mathcal{C}_0^{(c_1, L_1)}}(c_1)$. And so on. As above, at some point we will have constructed a segment to a vertex such that all eventualities are satisfied on the run-segment. Then we can construct a vertex $\mathcal{C}_0^{(l_1)}$ such that $\mathcal{C}_0^{(c_q, L_k)} \to^+ \mathcal{C}_0^{(l_1)}$ and $l_1 \in \theta_{\mathcal{C}_0^{(l_1)}}$. And so on.

Finally, we construct a vertex $\mathcal{C}_0' = \mathcal{C}_0^{(l_p)}$ such that $\mathcal{C}_0 \to^+ \mathcal{C}_0'$ and on this path segment all conditions of the theorem hold for $\mathcal{C} = \mathcal{C}_0$. Let us denote this path segment as λ_0, and let $\mathcal{C}_1$ be any successor of $\mathcal{C}_0'$.

By analogy, we can construct a vertex $\mathcal{C}_1'$ and a path segment λ_1 from $\mathcal{C}_1$ to $\mathcal{C}_1'$ such that all conditions of the theorem hold for $\mathcal{C} = \mathcal{C}_1$. An so forth. Eventually, we construct a sequence $\mathcal{C}_0, \mathcal{C}_1, \ldots, \mathcal{C}_j$ such that there exists n, $0 \le n < j$ and $\mathcal{C}_n = \mathcal{C}_j$ because there are only finitely many different colour schemes. Let $\pi_1 = \lambda_0, \ldots, \lambda_{n-1}$, $\pi_2 = \lambda_n, \ldots \lambda_{j-1}$. Now, we define our path π as $\pi_1(\pi_2)^*$. Properties (a) and (b) evidently hold on π.

Let $\mathcal{C} = \pi(i)$ and $\gamma \in \Gamma_{\mathcal{C}}$. Clearly, there exist $\gamma' \in \mathcal{C}_0$ and $\gamma'' \in \mathcal{C}_n$ such that $(\mathcal{C}_0, \gamma') \to^+ (\mathcal{C}, \gamma)$ and $(\mathcal{C}, \gamma) \to^+ (\mathcal{C}_n, \gamma'')$. Since for every $\gamma'' \in \mathcal{C}_n$ there exists $\gamma''' \in \mathcal{C}_n^{(\gamma_{s_n}, L_k)}$ such that all eventualities are satisfied on the run-segment from γ'' to γ''' and there exists $\gamma^{(4)} \in \mathcal{C}_n$, $(\mathcal{C}_n^{(\gamma_{s_n}, L_k)}, \gamma''') \to^+ (\mathcal{C}_n, \gamma^{(4)})$, then there is an e-run, r, such that $r(i) = \gamma$, that is, property (c) holds.

Note that, for every constant c of P the sequence $r_c(n)$ is a run in π. By construction, for every $\Diamond L(x) \in \mathcal{E}$ there is a vertex $\mathcal{C}_n^{(c,L)}$ in π_2 such that $L(x) \in \rho_{\mathcal{C}_n^{(c,L)}}(c)$. Therefore, $r_c(n)$ is an e-run in π and property (d) holds. ❏

Proof [of Theorem 5.15] Following [Hodkinson et al. 2000; Degtyarev and Fisher 2001] take a cardinal $\kappa \ge \aleph_0$ exceeding the cardinality of the set $\mathcal{R}_e$. Let us define a domain $D = \{\langle r, \xi \rangle \mid r \in \mathcal{R}_e, \xi < \kappa\}$. Then for every $n \in \mathbb{N}$ we have

$$D = \bigcup_{\gamma \in \Gamma_n} D_{(n,\gamma)}, \text{ where } D_{(n,\gamma)} = \{\langle r, \xi \rangle \mid r(n) = \gamma\} \text{ and } \left| D_{(n,\gamma)} \right| = \kappa.$$

Hence, by Lemma 5.12, for every $n \in \mathbb{N}$ there exists an $\mathcal{L}$-structure $\mathfrak{M}_n = \langle D, I_n \rangle$ satisfying $\mathcal{U} \cup \{\mathcal{G}_n\}$ such that $D_{(n,\gamma)} = \{\langle r, \xi \rangle \in D \mid \mathfrak{M}_n \models F_\gamma(\langle r, \xi \rangle)\}$. Moreover, we can suppose that $c^{I_n} = \langle r_c, 0 \rangle$ for every constant $c \in \mathrm{const}(\mathsf{P})$. A potential first order temporal model is $\mathfrak{M} = \langle D, I \rangle$, where $I(n) = I_n$ for all $n \in \mathbb{N}$. To be convinced of this we have to check validity of step and eventuality clauses. (Recall that satisfiability of $\mathcal{I}$ and $\mathcal{U}$ in $\mathfrak{M}_0$ is implied by satisfiability of $\mathcal{G}_0$ in $\mathfrak{M}_0$ and definition of a behaviour graph.)

Let $\Box \forall x (P_i(x) \Rightarrow \bigcirc R_i(x))$ be an arbitrary step clause; we show that it is true in $\mathfrak{M}$. Namely, we show that for every $n \ge 0$ and every $\langle r, \xi \rangle \in D$, if $\mathfrak{M}_n \models P_i(\langle r, \xi \rangle)$ then $\mathfrak{M}_{n+1} \models R_i(\langle r, \xi \rangle)$. Suppose $r(n) = \gamma \in \Gamma_n$ and $r(n+1) = \gamma' \in \Gamma'$, where (γ, γ') is a suitable pair in accordance with the definition of a run. It follows that $\langle r, \xi \rangle \in D_{(n,\gamma)}$ and $\langle r, \xi \rangle \in D_{(n+1,\gamma')}$, in other words $\mathfrak{M}_n \models F_\gamma(\langle r, \xi \rangle)$ and $\mathfrak{M}_{n+1} \models F_{\gamma'}(\langle r, \xi \rangle)$. Since $\mathfrak{M}_n \models P_i(\langle r, \xi \rangle)$ then $P_i(x) \in \gamma$. It follows that $R_i(x)$ is a conjunctive member of $B_\gamma(x)$. Since the pair (γ, γ') is suitable, it follows that the conjunction $\exists x (F_{\gamma'}(x) \wedge B_\gamma(x))$ is satisfiable and, moreover, $\models \forall x (F_{\gamma'}(x) \Rightarrow B_\gamma(x))$. Together with $\mathfrak{M}_{n+1} \models F_{\gamma'}(\langle r, \xi \rangle)$

this implies that $\mathfrak{M}_{n+1} \models R_i(\langle r, \xi \rangle)$. Propositional step clauses are treated in a similar way.

Let $(\Box \forall x) \Diamond L(x)$ be an arbitrary eventuality clause. We show that for every $n \geq 0$ and every $\langle r, \xi \rangle \in D$, $r \in \mathcal{R}_e, \xi < \kappa$, there exists $m > n$ such that $\mathfrak{M}_m \models L(\langle r, \xi \rangle)$. Since r is an e-run, there exists $\mathcal{C}' = \pi(m)$ for some $m > n$ such that $r(m) = \gamma' \in \Gamma'$ and $L(x) \in \gamma'$. It follows that $\langle r, \xi \rangle \in D_{(m, \gamma')}$, that is $\mathfrak{M}_m \models F_{\gamma'}(\langle r, \xi \rangle)$. In particular, $\mathfrak{M}_m \models L(\langle r, \xi \rangle)$. Propositional eventuality clauses are considered in a similar way. ❑

Note 5.17. For *constant flooded* temporal problems condition 3 of Theorem 5.15 implies condition 2.

LEMMA 5.18. *Let $\mathfrak{M}$ be a first-order temporal structure. Then there exists a colour scheme $\mathcal{C}$ such that $\mathfrak{M} \models \mathcal{F}_\mathcal{C}$.*

Proof Let $\mathfrak{M} = \langle D, I \rangle$. For every $a \in D$, let $\gamma_{(a)}$ be the set of unary literals such that for every predicate $P_i(x)$, $0 \leq i \leq N$,

$$\begin{array}{lll} P_i(x) & \in \gamma_{(a)} & \text{if} \quad \mathfrak{M} \models P_i(a) \\ \neg P_i(x) & \in \gamma_{(a)} & \text{if} \quad \mathfrak{M} \not\models P_i(a). \end{array}$$

Similarly, let θ be the set of propositional literals such that for every proposition p_j, $0 \leq j \leq n$,

$$\begin{array}{lll} p_j & \in \theta & \text{if} \quad \mathfrak{M} \models p_j \\ \neg p_j & \in \theta & \text{if} \quad \mathfrak{M} \not\models p_j. \end{array}$$

We define Γ as $\{\gamma_{(a)} \mid a \in D\}$, and $\rho(c)$ as $\gamma_{(c^I)}$. Clearly, $\mathfrak{M} \models \mathcal{F}_\mathcal{C}$. ❑

Proof [**Theorem 4.13: completeness of temporal resolution**] To simplify denotation, we assume that the temporal problem $\mathsf{P} = \langle \mathcal{U}, \mathcal{I}, \mathcal{S}, \mathcal{E} \rangle$ is already in the constant flooded form. Recall that according to our definitions, a fair derivation for the problem P is a finite sequence of universal parts, $\mathcal{U} = \mathcal{U}_0 \subset \mathcal{U}_1 \subset \mathcal{U}_2 \subset \cdots \subset \mathcal{U}_n$ such that for any $i \geq 0$ and formula $\phi \in \text{TRes}(\langle \mathcal{U}_i, \mathcal{I}, \mathcal{S}, \mathcal{E} \rangle)$, there exists $j \geq i$ such that $\phi \in \mathcal{U}_j$. In particular, for any formula $\phi \in \text{TRes}(\langle \mathcal{U}_n, \mathcal{I}, \mathcal{S}, \mathcal{E} \rangle)$ we have $\phi \in \mathcal{U}_n$.

The proof of the theorem proceeds by consideration of the number of vertices in the behaviour graph H for $\mathsf{P}_n = \langle \mathcal{U}_n, \mathcal{I}, \mathcal{E}, \mathcal{S} \rangle$, which is finite. If H is empty, then by Lemma 5.18 the set $\mathcal{U}_n \cup \mathcal{I}$ is unsatisfiable, and $\mathcal{U}_n$ contains the contradiction due to the initial termination rule.

Now suppose H is not empty. In the following we show that there exists an inference rule of temporal resolution such that when $\mathcal{U}_n$ is extended with the conclusion of the rule yielding $\mathcal{U}'_n$, the behaviour graph for the resulting temporal problem $\mathsf{P}' = \langle \mathcal{U}'_n, \mathcal{I}, \mathcal{S}, \mathcal{E} \rangle$ contains at least one vertex less than H. By lemma 5.7 this means, however, that $\mathcal{U}'_n \not\subseteq \mathcal{U}_n$ in contradiction with our assumption that $\mathcal{U}_n$ is the last member of the fair derivation.

Suppose there exists a vertex $\mathcal{C}$ of H which has no successors. In this case the set $\mathcal{U}_n \cup \mathcal{B}_\mathcal{C}$ is unsatisfiable. Indeed, suppose $\mathcal{U}_n \cup \{\mathcal{B}_\mathcal{C}\}$ is true in a model $\mathfrak{M}$. By lemma 5.18, we can define a colour scheme $\mathcal{C}'$ such that $\mathfrak{M} \models \mathcal{F}_{\mathcal{C}'}$. As $\mathcal{B}_\mathcal{C} \wedge \mathcal{F}_{\mathcal{C}'}$ is satisfiable, there exists an edge from the vertex $\mathcal{C}$ to the vertex $\mathcal{C}'$ in the contradiction with the choice of C as having no successor. The conclusion of the step resolution rule, $\neg \mathcal{A}_\mathcal{C}$, is added to the set $\mathcal{U}_n$; this implies removing the vertex $\mathcal{C}$ from the behaviour graph because the set $\{\mathcal{F}_{\mathcal{C}}, \neg\mathcal{A}_{\mathcal{C}}\}$ is not satisfiable.

Next, we check the possibility where H is not empty and every vertex H has a successor. Since the problem, P, is unsatisfiable, at least one condition of Theorem 5.15 is violated. By Note 5.17, it is enough to consider only two cases of violation of the conditions of Theorem 5.15.

First condition of Theorem 5.15 does not hold. Then, there exist a vertex $\mathcal{C}_0$, predicate colour γ_0, and eventuality $\Diamond L_0(x)$ such that for every vertex $\mathcal{C}'$ and predicate colour $\gamma \in \Gamma'$,

$$(\mathcal{C}_0, \gamma_0) \to^{+} (\mathcal{C}', \gamma\,') \Rightarrow L_0(x) \notin \gamma\,'. \tag{8}$$

Let $\mathfrak{I}$ be a finite nonempty set of indexes such that $\{\mathcal{C}_i \mid i \in \mathfrak{I}\}$ is the set of *all* successors of $\mathcal{C}_0$ (possibly including $\mathcal{C}_0$ itself); and let $\mathfrak{J}_i$, for $i \in \mathfrak{I}$, be finite nonempty sets of indexes such that $\{\gamma_{i,j} \in \Gamma_i \mid i \in \mathfrak{I},\ j \in \mathfrak{J}_i,\ \gamma_0 \to^{+} \gamma_{i,j}\}$ is the set of *all* predicate colours $\gamma_{i,j}$ such that there exists a run going through γ_0 and the colour $\gamma_{i,j}$. (To unify notation, if $0 \notin \mathfrak{I}$, we define $\mathfrak{J}_0$ as $\{0\}$, and $\gamma_{0,0}$ as γ_0; and if $0 \in \mathfrak{I}$, we add the index of γ_0 to $\mathfrak{J}_0$. Therefore, $\mathfrak{J}_0$ is always defined and without loss of generality we may assume that $\gamma_{0,0} = \gamma_0$.)

Let $\mathcal{C}_{i_1}, \ldots, \mathcal{C}_{i_k}$ be the set of all immediate successors of $\mathcal{C}_0$. To simplify the proof, we will represent canonical merged derived step clauses $\mathcal{A}_{\mathcal{C}_i} \Rightarrow \bigcirc\mathcal{B}_{\mathcal{C}_i}$ (and $\mathcal{A}_{\mathcal{C}_{i_l}} \Rightarrow \bigcirc\mathcal{B}_{\mathcal{C}_{i_l}}$) simply as $\mathcal{A}_i \Rightarrow \bigcirc\mathcal{B}_i$ (and $\mathcal{A}_{i_l} \Rightarrow \bigcirc\mathcal{B}_{i_l}$, resp.), and formulae $\mathcal{F}_{\mathcal{C}_i}$ (and $\mathcal{F}_{\mathcal{C}_{i_l}}$) simply as $\mathcal{F}_i$ (and $\mathcal{F}_{i_l}$, resp.).

Consider two cases depending on whether the canonical merged derived step clause $\mathcal{A}_0 \Rightarrow \bigcirc\mathcal{B}_0$ (or any of $\mathcal{A}_i \Rightarrow \bigcirc\mathcal{B}_i$, $i \in \mathfrak{I}$) degenerates or not.

(1) Let $\mathcal{A}_0 = \mathcal{B}_0 =$ **true**. It follows that $\mathcal{U}_n \models \forall x \neg L_0(x)$. Indeed, suppose $\mathcal{U}_n \cup \{\exists x L_0(x)\}$ has a model, $\mathfrak{M}$. Then we can construct a colour scheme $\mathcal{C}'$ such that $\mathfrak{M} \models \mathcal{F}_{\mathcal{C}'}$. Since $\mathcal{C}_{i_1}, \ldots, \mathcal{C}_{i_k}$ is the set of all immediate successors of $\mathcal{C}_0$ and $\mathcal{B}_0 =$ **true**, it holds that there exists $j, 1 \leq j \leq k$, such that $\mathcal{C}_{i_j} = \mathcal{C}'$. Since $B_{\gamma_0}(x) =$ **true**, every pair $(\gamma_0, \gamma\,')$, where $\gamma\,' \in \Gamma'$, is suitable; hence $\neg L_0(x) \in \gamma\,'$ for every $\gamma\,' \in \Gamma'$, and $\mathcal{F}_{\mathcal{C}'} \models \forall x \neg L_0(x)$ leading to a contradiction. Therefore, $\mathcal{U}_n \models \forall x \neg L_0(x)$ and the eventuality termination rule can be applied. The same holds if any one of $\mathcal{A}_i \Rightarrow \bigcirc\mathcal{B}_i$ degenerates.

(2) Let none of the $\mathcal{A}_i \Rightarrow \bigcirc\mathcal{B}_i$ degenerate. We are going to prove that the eventuality resolution rule can be applied. First, we have to check the side conditions for such an application.

(a) $\forall x(\mathcal{U}_n \wedge \mathcal{B}_i \wedge B_{\gamma_{i,j}}(x) \Rightarrow \neg L_0(x))$ for all $i \in \mathfrak{I} \cup \{0\}$, $j \in \mathfrak{J}_i$.
Consider the case when $i = j = 0$ (for other indexes the arguments are similar). We show that

$$\forall x(\mathcal{U}_n \wedge \mathcal{B}_0 \wedge B_{\gamma_0}(x) \Rightarrow \bigvee_{l \in \{1,\ldots,k\},\ \gamma\,' \in \Gamma_{i_l},\ \gamma \to \gamma\,'} F_{\gamma\,'}(x))$$

is valid (it follows, in particular, that $\forall x(\mathcal{U}_n \wedge \mathcal{B}_0 \wedge B_{\gamma_0}(x) \Rightarrow \neg L_0(x))$ is valid). Suppose $\mathfrak{M}$ is a model for

$$\exists x(\mathcal{U}_n \wedge \mathcal{B}_0 \wedge B_{\gamma_0}(x) \wedge \bigwedge_{l \in \{1,\ldots,k\},\ \gamma\,' \in \Gamma_{i_l},\ \gamma \to \gamma\,'} \neg F_{\gamma\,'}(x)).$$

Then there exists a colour scheme $\mathcal{C}'$ such that $\mathfrak{M} \models \mathcal{F}_{\mathcal{C}'}$. Since $\mathfrak{M} \models \mathcal{B}_0 \wedge \mathcal{F}_{\mathcal{C}'}$, we conclude that $\mathcal{C}'$ is among $\mathcal{C}_{i_1}, \ldots, \mathcal{C}_{i_k}$. Note that $\mathfrak{M} \models \mathcal{F}_{\mathcal{C}'}$ follows, in particular, $\mathfrak{M} \models \forall x \bigvee_{\gamma'' \in \Gamma'} F_{\gamma''}(x)$ and, hence, $\mathfrak{M} \models \forall x(B_{\gamma_0}(x) \Rightarrow \bigvee_{\gamma'' \in \Gamma'} F_{\gamma''}(x))$. Together with the fact that $\mathfrak{M} \models \mathcal{U}_n \wedge \exists x(B_{\gamma_0}(x) \wedge F_{\gamma''}(x))$ implies $\gamma_0 \to \gamma''$, we have $\mathfrak{M} \models \forall x(B_{\gamma_0}(x) \Rightarrow \bigvee_{\gamma'' \in \Gamma',\ \gamma_0 \to \gamma''} F_{\gamma''}(x))$. This contradicts the choice of the structure $\mathfrak{M}$.

(b) $\forall x(\mathcal{U}_n \wedge \mathcal{B}_i \wedge B_{\gamma_{i,j}}(x) \Rightarrow \bigvee_{k \in \mathfrak{I} \cup \{0\},\ l \in \mathfrak{I}_k} (\mathcal{A}_k \wedge A_{\gamma_{k,l}}(x)))$ for all $i \in \mathfrak{I} \cup \{0\}$, $j \in \mathfrak{I}_i$.
Again, consider the case $i = j = 0$. Suppose

$$\mathcal{U}_n \wedge \mathcal{B}_0 \wedge \exists x(B_{\gamma_0}(x) \wedge \bigwedge_{k \in \mathfrak{I} \cup \{0\},\ l \in \mathfrak{I}_k} (\neg(\mathcal{A}_k \wedge A_{\gamma_{k,l}}(x))))$$

is satisfied in a structure $\mathfrak{M}$. Let $\mathcal{C}'$ be a colour scheme such that $\mathfrak{M} \models \mathcal{F}_{\mathcal{C}'}$. By arguments similar to the ones given above, there is a vertex $\mathcal{C}_{i_l}$, $1 \leq l \leq k$, which is an immediate successor of $\mathcal{C}_0$, such that $\mathcal{C}_{i_l} = \mathcal{C}'$, and hence $\mathfrak{M} \models \mathcal{A}'$. It suffices to note that

$$\mathfrak{M} \models \forall x(B_{\gamma_0}(x) \Rightarrow \bigvee_{\gamma'' \in \Gamma',\ \gamma_0 \to \gamma''} A_{\gamma''}(x)).$$

(As in the case 2(a) above, $\mathfrak{M} \models \forall x(B_{\gamma_0}(x) \Rightarrow \bigvee_{\gamma'' \in \Gamma',\ \gamma_0 \to \gamma''} F_{\gamma''}(x))$, and for all $\gamma'' \in \Gamma'$, the formula $\forall x(F_{\gamma''}(x) \Rightarrow A_{\gamma''}(x))$ is valid.)

After applying the eventuality resolution rule we add to $\mathcal{U}_n$ its conclusion:

$$\forall x \bigwedge_{i \in \mathfrak{I} \cup \{0\},\ j \in \mathfrak{I}_i} (\neg \mathcal{A}_i \vee \neg A_{\gamma_{i,j}}(x)).$$

Then, the vertex $\mathcal{C}_0$ will be removed from the behaviour graph (recall that $\mathcal{F}_0 \models \mathcal{A}_0 \wedge \exists x A_{\gamma_0}(x)$).

Third condition of Theorem 5.15 does not hold. This case is analogous to the previous one; we only sketch the proof. There exist a vertex $\mathcal{C}_0$ and eventuality $\Diamond l_0$ such that for every vertex $\mathcal{C}'$ and predicate colour $\gamma \in \Gamma'$,

$$\mathcal{C}_0 \to^+ \mathcal{C}' \Rightarrow l_0 \notin \theta'. \tag{9}$$

Let $\mathfrak{I}$ be a finite nonempty set of indexes, $\{\mathcal{C}_i \mid i \in \mathfrak{I}\}$ be the set of all successors of $\mathcal{C}_0$ (possibly including $\mathcal{C}_0$ itself). As in the previous case, one can show that

—If any of $\mathcal{A}_i \Rightarrow \bigcirc \mathcal{B}_i$ (where $i \in \mathfrak{I}$) degenerates then $\mathcal{U}_n \models \neg l$, and the ground eventuality termination rule can be applied.

—If none of the canonical merged derived step clauses degenerate then the following conditions hold
—for all $i \in \mathfrak{I} \cup \{0\}$ $\quad \mathcal{U}_n \cup \mathcal{B}_i \models l_0$
—for all $i \in \mathfrak{I} \cup \{0\}$ $\quad \mathcal{U}_n \cup \mathcal{B}_i \models \bigvee_{j \in \mathfrak{I} \cup \{0\}} \mathcal{A}_j$
and so the ground eventuality resolution rule can be applied.

❑

Example 5.19 *example 5.6 contd.*. We illustrate the proof of Theorem 4.13 on the temporal problem introduced in Example 5.6. The behaviour graph of the problem is not empty; every vertex has a successor. It is not hard to see that the first condition of Theorem 5.15 does not hold, and, following the proof, we can choose as $\mathcal{C}_0$, γ_0, and L_0, for example, $\mathcal{C}_1$, γ_1, and $\neg P(x)$, respectively. Then for every vertex $\mathcal{C}'$ and predicate colour $\gamma' \in \Gamma'$,

$$(\mathcal{C}_0, \gamma_0) \to^+ (\mathcal{C}', \gamma') \Rightarrow L_0(x) \notin \gamma'.$$

The set of all (and all immediate) successors of $\mathcal{C}_1$ is $\{\mathcal{C}_1, \mathcal{C}_4\}$. Note that the canonical full merged step clauses corresponding to $\mathcal{C}_1$ and $\mathcal{C}_4$ are identical, and none of them degenerates. For $i \in \{1, 4\}$, the loop side conditions,

$$\forall x((\underbrace{(l \Rightarrow \exists x P(x))}_{\mathcal{U}_i} \wedge \underbrace{(\exists x P(x) \wedge \forall x P(x))}_{\mathcal{B}_i} \wedge \underbrace{P(x)}_{B_{\gamma_1}(x)}) \Rightarrow P(x))$$

and

$$\forall x((\underbrace{(l \Rightarrow \exists x P(x))}_{\mathcal{U}_i} \wedge \underbrace{(\exists x P(x) \wedge \forall x P(x))}_{\mathcal{B}_i} \; \wedge \; \underbrace{P(x)}_{B_{\gamma_1}(x)}) \Rightarrow$$

$$\bigvee_{j \in \{1,4\}} (\underbrace{\exists x P(x) \wedge \forall x P(x)}_{\mathcal{A}_j} \wedge \underbrace{P(x)}_{A_{\gamma_1}(x)}))$$

hold. Therefore, we can apply the eventuality resolution rule whose conclusion,

$$\forall x(\bigwedge_{j \in \{1,4\}} (\neg(\exists x P(x) \wedge \forall x P(x))) \wedge \neg P(x)),$$

can be simplified to $\exists x \neg P(x)$. After the conclusion of the rule is added to $\mathcal{U}$, vertices $\mathcal{C}_1$ and $\mathcal{C}_4$ and edges leading to and from them are deleted from the behaviour graph.

For the temporal problem with the new universal part, again the first condition of Theorem 5.15 does not hold and, for example, for $\mathcal{C}_0 = \mathcal{C}_3$, $\gamma_0 = \gamma_1$, and $L_0(x) = \neg P(x)$, and for every colour scheme $\mathcal{C}'$ and every predicate colour $\gamma' \in \Gamma'$,

$$(\mathcal{C}_0, \gamma_0) \to^+ (\mathcal{C}', \gamma') \Rightarrow L_0(x) \notin \gamma'.$$

(Note that γ_2 is never a successor of γ_1.) The set of all (and all immediate) successors of $\mathcal{C}_3$ is $\{\mathcal{C}_3, \mathcal{C}_6\}$. The canonical full merged step clauses corresponding to $\mathcal{C}_3$ and $\mathcal{C}_6$ are identical, and none of them degenerates. In a similar way, the loop side conditions hold and the conclusion of the eventuality resolution rule simplifies to $\forall x \neg P(x)$. This time, vertices $\mathcal{C}_3$ and $\mathcal{C}_6$ are deleted from the behaviour graph.

For the new problem, the third condition of Theorem 5.15 does not hold for $\mathcal{C}_0 = \mathcal{C}_5$, $l_0 = l$. Then for any vertex $\mathcal{C}'$,

$$\mathcal{C}_0 \to^+ \mathcal{C}' \Rightarrow l_0 \notin \theta'.$$

As the canonical full merged step clause degenerates (and $\mathcal{U} \models \neg l$), the ground eventuality termination rule can be applied.

Note that if, in the beginning, instead of $\mathcal{C}_1$ we selected $\mathcal{C}_3$ (or $\mathcal{C}_6$) as $\mathcal{C}_0$, vertices $\mathcal{C}_1$, $\mathcal{C}_3$, $\mathcal{C}_4$, and $\mathcal{C}_6$ would be deleted after the first application of the eventuality resolution rule.

6. EXTENSION OF THE MONODIC FRAGMENT

In this, and the subsequent, section we adapt the resolution technique to a number of variations of monodic FOTL, whose completeness follows from the corresponding adaptation of the completeness results given in Section 5. We here consider an extension of monodic temporal problems allowing an additional *extended part* $\mathcal{X}$ given by a set of arbitrary FOTL in the language without function symbols and with the *only* temporal operator being '$\bigcirc$'. Since this temporal operator can be "moved inside" classical quantifiers, we can assume, without loss of generality, that $\mathcal{X}$ is given by a set of first-order formulae constructed from *temporal atoms* of the form $\bigcirc^i P(t_1, t_2, \ldots, t_n)$, where $P(t_1, t_2, \ldots, t_n)$ is a first-order atom[6]. Such an extension permits more complex step formulae to be employed while restricting the allowed temporal operators.

Example 6.1. A set of formulae XP given by

$$\begin{aligned}
&\mathcal{X} = \{\forall x \forall y (P(x,y) \Rightarrow \bigcirc\bigcirc P(x,y))\},\\
&\mathcal{I} = \{\exists x \exists y P(x,y)\},\\
&\mathcal{U} = \{\forall x \forall y (P(x,y) \Rightarrow R(x))\},\\
&\mathcal{S} = \{R(x) \Rightarrow \bigcirc R(x)\},\\
&\mathcal{E} = \{\Diamond \neg R(x)\}
\end{aligned}$$

is an example of an extended monodic problem.

We are going to show that an extended monodic temporal problem can be translated (*with a linear growth in size*) into a monodic temporal problem while preserving satisfiability. Essentially, we encode a few initial states of a temporal model as a first-order formula and ensure that this encoding is consistent with the rest of the model.

Reduction Let $\mathsf{XP} = \mathsf{P} \cup \mathcal{X}$ be an extended monodic temporal problem. Let $\mathsf{P} = \langle \mathcal{I}, \mathcal{U}, \mathcal{S}, \mathcal{E} \rangle$. Let k be the maximal number of nested applications of $\bigcirc$ in $\mathcal{X}$, that is, the maximal i such that $\bigcirc^i P(t_1, t_2, \ldots, t_n)$ occurs in $\mathcal{X}$ for some predicate symbol P. For every predicate, P, occurring in XP, we introduce $k+1$ new predicates $P^0, P^1, \ldots, P^k$ of the same arity. Let ϕ be a first-order formula in the language of XP. We denote by $[\phi]^i$, $0 \le i \le k$, the result of substitution of all occurrences of predicates in ϕ with their i-th counterparts; (e.g., $P(x_1, x_2)$ is replaced with $P^i(x_1, x_2)$).

We define the monodic problem $\mathsf{P}' = \langle \mathcal{I}', \mathcal{U}', \mathcal{S}', \mathcal{E}' \rangle$ as follows. Let $\mathcal{U}' = \mathcal{U}$, $\mathcal{S}' = \mathcal{S}$, $\mathcal{E}' = \mathcal{E}$. As for $\mathcal{I}'$, we take the following set of formulae.

(1) For every $\phi \in \mathcal{I}$, the formula $[\phi]^0$ is in $\mathcal{I}'$.

(2) For every $\phi \in \mathcal{U}$, the formula $\bigwedge_{i=0}^{k} [\phi]^i$ is in $\mathcal{I}'$.

(3) For every $P(x) \Rightarrow \bigcirc Q(x) \in \mathcal{S}$, the formula $\bigwedge_{i=0}^{k-1} \big(\forall x (P^i(x) \Rightarrow Q^{i+1}(x)\big)$ is in $\mathcal{I}'$.

(4) For every $\psi \in \mathcal{X}$, the formula ψ', the result of replacing all occurrences of temporal atoms $\bigcirc^i P(\bar{t})$, $i \ge 0$, in ψ with $P^i(\bar{t})$, is in $\mathcal{I}'$.

(5) For every n-ary predicate P in the language of XP, the formula $\forall x_1, \ldots x_n (P(x_1, \ldots, x_n) \equiv P^k(x_1, \ldots, x_n))$ is in $\mathcal{I}'$.

[6] Decidability of this extension of the monodic fragment was suggested in a private communication by M. Zakharyaschev.

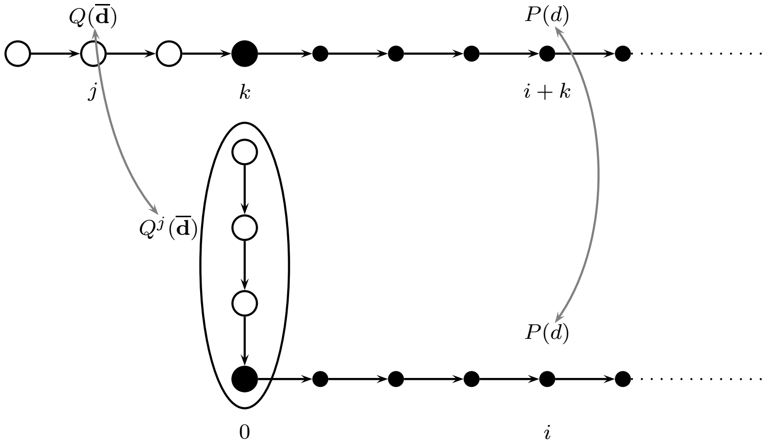

Fig. 2. Model transformation

(6) No other formulae are in $\mathcal{I}'$.

Example 6.2 *Example 6.1 contd*. We give the reduction, $\mathsf{P} = \langle \mathcal{U}, \mathcal{I}, \mathcal{S}, \mathcal{E} \rangle$, of the extended temporal problem XP from Example 6.1. The universal, step and eventuality parts of P' are the same as of XP. The initial part, $\mathcal{I}$, consists of the following formulae

$$\begin{array}{l}
\exists x \exists y P^0(x,y), \\
\forall x \forall y (P^0(x,y) \Rightarrow R^0(x)), \\
\forall x \forall y (P^1(x,y) \Rightarrow R^1(x)), \\
\forall x \forall y (P^2(x,y) \Rightarrow R^2(x)), \\
\forall x (R^0(x) \Rightarrow R^1(x)), \\
\forall x (R^1(x) \Rightarrow R^2(x)), \\
\forall x \forall y (P^0(x,y) \Rightarrow P^2(x,y)), \\
\forall x \forall y (P(x,y) \equiv P^2(x,y)), \\
\forall x (R(x) \equiv R^2(x)).
\end{array}$$

THEOREM 6.3 REDUCTION OF EXTENDED PROBLEMS. *XP is satisfiable if, and only if, P' is satisfiable.*

Proof We prove that given a model for XP it is possible to find a model for P' and vice versa. The transformation of models is depicted in Fig. 2.

First, consider a model $\mathfrak{M} = \langle D, I \rangle$ for XP and construct a model $\mathfrak{M}' = \langle D, I' \rangle$ as follows. The interpretation of constants in the language of XP in $\mathfrak{M}'$ is the same as in $\mathfrak{M}$ (recall that constants are *rigid*).

For every n-ary predicate P in the language of XP (in the initial signature), every n-tuple $(d_1, \ldots, d_n) \in D$, and every $i \geq 0$, we define

$$\mathfrak{M}'_i \models P(d_1, \ldots, d_n) \quad \text{iff} \quad \mathfrak{M}_{i+k} \models P(d_1, \ldots, d_n).$$

For every n-ary predicate P^i, $0 \leq i \leq k$, in the extension of the initial language (that is, in the language of P' but not in the language of XP) we define

$$\mathfrak{M}'_0 \models P^i(d_1, \ldots, d_n) \quad \text{iff} \quad \mathfrak{M}_i \models P(d_1, \ldots, d_n),$$

and P^i is false in $\mathfrak{M}'$ for all other tuples and moments of time[7]. This definition is consistent with formulae from part 5 of $\mathcal{I}'$; therefore $\mathfrak{M}'$ is defined correctly.

Since truth values of all predicates from P are not changed but "shifted", clearly, $\mathfrak{M}' \models \mathcal{U}$ and $\mathfrak{M}' \models \mathcal{S}$. Since all our eventualities are unconditional, that is, are of the form $\Box\Diamond l$ and $\Box\forall x \Diamond L(x)$, the truth value of $L(x)$ in the first $k+1$ states of $\mathfrak{M}$ does not affect the truth value of $\mathcal{E}'$ in $\mathfrak{M}'$; so $\mathfrak{M}' \models \mathcal{E}'$. The fact that $\mathfrak{M}' \models \mathcal{I}'$ can be established by considering step by step the definition of $\mathcal{I}'$. Indeed:

(1) Let a formula $[\phi]^0$ be in $\mathcal{I}'$, where $\phi \in \mathcal{I}$. Then $\mathfrak{M}'_0 \models [\phi]^0$ because for every predicates P and P^0,

$$\mathfrak{M}_0 \models P(d_1,\dots,d_n) \quad \text{iff} \quad \mathfrak{M}'_0 \models P^0(d_1,\dots,d_n)$$

holds and $\mathfrak{M}_0 \models \phi$.

(2) Let a formula $[\phi]^i$, $0 \leq i \leq k$, be in $\mathcal{I}'$, where $\phi \in \mathcal{U}$. Then $\mathfrak{M}'_0 \models [\phi]^i$ because for all predicates P and P^i,

$$\mathfrak{M}_i \models P(d_1,\dots,d_n) \quad \text{iff} \quad \mathfrak{M}'_0 \models P^i(d_1,\dots,d_n)$$

holds and $\mathfrak{M}_i \models \phi$.

(3) Let $\mathfrak{M}'_0 \models P^i(d_1,\dots,d_n)$, $0 \leq i \leq k$. Then $\mathfrak{M}_i \models P(d_1,\dots,d_n)$, and because of $P(x) \Rightarrow \bigcirc Q(x) \in \mathcal{S}$, we have $\mathfrak{M}_{i+1} \models Q(d_1,\dots,d_n)$. It follows $\mathfrak{M}'_0 \models Q^{i+1}(d_1,\dots,d_n)$.

(4) Let $\psi \in \mathcal{X}$, that is, $\mathfrak{M}_0 \models \psi$. For every subformula $\bigcirc^i P(d_1,\dots,d_n)$ of ψ, $\mathfrak{M}_0 \models \bigcirc^i P(d_1,\dots,d_n)$ holds if, and only if, $\mathfrak{M}'_0 \models P^i(d_1,\dots,d_n)$. So, $\mathfrak{M}'_0 \models \psi'$.

(5) In accordance with the definition of $\mathfrak{M}'$, $\mathfrak{M}'_0 \models P(d_1,\dots,d_n)$ if, and only if, $\mathfrak{M}_k \models P(d_1,\dots,d_n)$ if, and only if, $\mathfrak{M}'_0 \models P^k(d_1,\dots,d_n)$.

Let $\mathfrak{M}'$ be a model for P'. We construct a model $\mathfrak{M}$ for XP. The interpretation of constants in the language of XP in $\mathfrak{M}$ is the same as in $\mathfrak{M}'$. For every n-ary predicate P in the language of XP and every n-tuple $(d_1,\dots,d_n) \in D$ we define for every $i \geq k$

$$\mathfrak{M}_i \models P(d_1,\dots,d_n) \quad \text{iff} \quad \mathfrak{M}'_{i-k} \models P(d_1,\dots,d_n),$$

and for every i such that $0 \leq i \leq k$

$$\mathfrak{M}_i \models P(d_1,\dots,d_n) \quad \text{iff} \quad \mathfrak{M}'_0 \models P^i(d_1,\dots,d_n).$$

Note that $\mathfrak{M}'_0 \models \mathcal{I}'$ and, in particular, formulae from part 5 of $\mathcal{I}'$; therefore, $\mathfrak{M}$ is defined correctly. Indeed, in the case $i = k$ we obtain

$$\mathfrak{M}'_0 \models P(d_1,\dots,d_n) \quad \text{iff} \quad \mathfrak{M}'_0 \models P^k(d_1,\dots,d_n).$$

Evidently, for $i \geq k$, $\mathfrak{M}_i \models \mathcal{U}$ and $\mathfrak{M}_i \models \mathcal{S}$. Again, since our eventualities are unconditional, evaluation of $\mathcal{E}$ does not depend on a finite number of initial states, and $\mathfrak{M} \models \mathcal{E}$. It is enough to show that $\mathfrak{M}_i \models \mathcal{U}$ and $\mathfrak{M}_i \models \mathcal{S}$ for $i \in [0,(k-1)]$, and $\mathfrak{M}_0 \models \mathcal{I}$. Again, this can be done by analysing the definition of $\mathcal{I}'$.

The first claim, $\mathfrak{M}_i \models \mathcal{U}$, follows from item 2 of the definition of $\mathcal{I}'$, from the relation

$$\mathfrak{M}_i \models P(d_1,\dots,d_n) \quad \text{iff} \quad \mathfrak{M}'_0 \models P^i(d_1,\dots,d_n)$$

[7] Note that all new predicates occur only in $\mathcal{I}'$.

and the fact that $\mathfrak{M}'_0 \models [\phi]^i$ for every $\phi \in \mathcal{U}$, $0 \leq i \leq k$.

The second claim, $\mathfrak{M}_i \models \mathcal{S}$, follows from item 3 of the definition of $\mathcal{I}'$ and from the relation

$$\mathfrak{M}_i \models P(d_1, \ldots, d_n) \quad \text{iff} \quad \mathfrak{M}'_0 \models P^i(d_1, \ldots, d_n).$$

The last claim, $\mathfrak{M}_0 \models \mathcal{I}$, follows immediately from item 1 of the definition of $\mathcal{I}'$ and from the relation

$$\mathfrak{M}_0 \models P(d_1, \ldots, d_n) \quad \text{iff} \quad \mathfrak{M}'_0 \models P^0(d_1, \ldots, d_n)$$

given above. ❑

7. GROUNDING TEMPORAL PROBLEMS

In this section we adapt the core temporal resolution calculus given in Section 4 to a variation of monodic FOTL where sub-parts of the temporal problem are *grounded*. Not only does this characterise an important class of formulae, but this variation admits simplified clausal resolution techniques (in particular, simplified DSNF).

Definition 7.1 *Groundedness.* A temporal problem P is called *grounded* if all the step clauses and the eventuality clauses of P are ground. (Correspondingly, a monodic temporal formula is called grounded if it can be transformed to a grounded temporal problem.) A temporal problem P is called a *ground eventuality* problem if all the eventualities of P are ground. A temporal problem P is called a *ground next-time* problem if all the step clauses of P are ground.

If P is a ground eventuality problem then only the ground versions of the eventuality resolution and eventuality termination rules are needed.

THEOREM 7.2 REDUCING A GROUND EVENTUALITY PROBLEM. *Every ground eventuality monodic temporal problem can be reduced to a satisfiability equivalent grounded monodic problem with an exponential growth in size of the given problem.*

Proof Note that the ground eventuality resolution rule, step resolution rule, and initial termination rule operate on merged derived step clauses. So, if instead of original step clauses we consider step clauses given by formulae (7), (6), and (5) (and strictly speaking, rename by propositions closed first-order formulae in the right- and left-hand sides), we obtain a satisfiability equivalent grounded temporal problem. ❑

Example 7.3. Consider an unsatisfiable formula

$$\Diamond \exists x (P(x) \wedge \bigcirc \neg P(x)) \wedge \Box (P(x) \Rightarrow \bigcirc P(x)).$$

In DSNF we have (note that $\mathcal{I}$ is empty throughout),

$$\mathcal{S} = \left\{ \begin{array}{l} P(x) \Rightarrow \bigcirc P(x) \\ Q(x) \Rightarrow \bigcirc \neg P(x) \end{array} \right\}, \qquad \begin{array}{l} \mathcal{U} = \emptyset, \\ \mathcal{E} = \{\Diamond \exists x (P(x) \wedge Q(x))\}. \end{array}$$

According to our reduction, this problem is satisfiability equivalent to the following

$$\begin{array}{l}\mathcal{U} = \emptyset,\\ \mathcal{S} = \left\{\begin{array}{l}\exists x P(x) \Rightarrow \bigcirc \exists x P(x)\\ \forall x P(x) \Rightarrow \bigcirc \forall x P(x)\\ \exists x Q(x) \Rightarrow \bigcirc \exists x \neg P(x)\\ \forall x Q(x) \Rightarrow \bigcirc \forall x \neg P(x)\\ \exists x (P(x) \wedge Q(x)) \Rightarrow \bigcirc \exists x (P(x) \wedge \neg P(x))\\ \forall x (P(x) \vee Q(x)) \Rightarrow \bigcirc \forall x (P(x) \vee \neg P(x))\end{array}\right\},\\ \mathcal{E} = \{\Diamond \exists x (P(x) \wedge Q(x))\}.\end{array}$$

The last step clause is a tautology which can be eliminated immediately, the next to last can be moved to the universal part by an application of step resolution. .

$$\begin{array}{l}\mathcal{U} = \{\forall x (\neg P(x) \vee \neg Q(x))\},\\ \mathcal{S} = \left\{\begin{array}{l}\exists x P(x) \Rightarrow \bigcirc \exists x P(x)\\ \forall x P(x) \Rightarrow \bigcirc \forall x P(x)\\ \exists x Q(x) \Rightarrow \bigcirc \exists x \neg P(x)\\ \forall x Q(x) \Rightarrow \bigcirc \forall x \neg P(x)\end{array}\right\},\\ \mathcal{E} = \{\Diamond \exists x (P(x) \wedge Q(x))\}.\end{array}$$

Now the ground eventuality termination rule can be applied.

Together with Theorem 7.2 the following theorem shows that for any problem P, if either all the step clauses are ground or all the eventuality clauses are ground, then it can be reduced to a grounded problem.

THEOREM 7.4 REDUCING A GROUND NEXT-TIME PROBLEM.
Let $\mathsf{P} = \langle \mathcal{I}, \mathcal{U}, \mathcal{S}, \mathcal{E} \rangle$ be a temporal problem such that all step rules of P are ground. Let $\mathcal{E}^{\exists}$ be obtained from $\mathcal{E}$ as follows: every eventuality clause of the form $\Diamond L(x)$ (in the meaning of $\forall x \Diamond L(x)$) is replaced with its ground *consequence $\exists x \Diamond L(x)$ (equivalent to $\Diamond \exists x L(x)$). Let $\mathsf{P}' = \langle \mathcal{I}, \mathcal{U}, \mathcal{S}, \mathcal{E}^{\exists} \cup \{\Diamond L(c) \mid \Diamond L(x) \in \mathcal{E},\ c \in \mathrm{const}(\mathsf{P})\} \rangle$. Then P is satisfiable if and only if P' is satisfiable.*

Proof (Sketch) Evidently, if P' is unsatisfiable, then P is unsatisfiable. Suppose now P is unsatisfiable, then there exists a successfully terminating temporal resolution derivation from P^c. Note that the added eventualities of the form $\Diamond L(c)$ exactly correspond to the eventualities added by reduction to constant-flooded form.

Suppose the eventuality resolution rule is applied to a non-ground eventuality $\forall x \Diamond L(x)$. The validity of the side conditions implies the validity of the formula

$$\Box \forall x (\mathcal{U} \wedge \bigvee_{j=1}^{n} \mathcal{A}_i \Rightarrow \bigcirc \Box \neg L(x)) \tag{10}$$

for a set $\{\mathcal{A}_i \Rightarrow \bigcirc \mathcal{B}_i, 1 \leq i \leq\}$ of *ground* merged derived step rules. (10) is resolved with the formula $\Box \forall x \Diamond L(x)$ giving the conclusion $(\bigwedge_{i=1}^{n} \neg \mathcal{A}_i)$. However (10) or, equivalent to (10),

$$\Box (\mathcal{U} \wedge \bigvee_{j=1}^{n} \mathcal{A}_i \Rightarrow \bigcirc \Box \forall x \neg L(x)) \tag{11}$$

can be resolved with a "weaker" formula $\Box\exists x\Diamond L(x)$ giving the same result.

If the eventuality termination rule is applied to $\forall x\Diamond L(x)$, its side condition, $\mathcal{U} \models \forall x\neg L(x)$, equally contradicts to the ground eventuality $\exists x\Diamond L(x)$. So, we can conclude that replacing non-ground eventualities of the form $\forall x\Diamond L(x)$ with ground eventualities $\exists x\Diamond L(x)$ (equivalent to $\Diamond\exists x L(x)$) does not affect (un)satisfiability. ❑

Example 7.5.

$$\begin{array}{ll} \mathcal{I} = \{l\}, & \mathcal{U} = \{\forall x(l \Rightarrow Q(x))\}, \\ \mathcal{S} = \{l \Rightarrow \bigcirc l\}, & \mathcal{E} = \{\Diamond\neg Q(x)\}. \end{array}$$

Evidently, the initial, universal, and step parts imply $\Box\forall x Q(x)$ which also contradicts to $\Box\forall x\Diamond\neg Q(x)$ and $\Box\exists x\Diamond\neg Q(x)$.

8. DECIDABILITY BY TEMPORAL RESOLUTION

Temporal resolution provides a decision procedure for a class of monodic temporal formulae provided that there exists a first-order decision procedure for side conditions of all inference rules. Direct examination of the side conditions shows that we are interested in the satisfiability of the conjunction of the (current) universal part and sets of monadic formulae built from predicate symbols which occur in the temporal part. At the same time, the current universal part of a derivation is obtained by extending the initially given universal part by monadic formulae from the conclusions of the inference rules. So, after imposing appropriate restrictions on the form of the universal part of a given temporal problem, we can guarantee its decidability (the addition of monadic formulae usually does not affect decidability).

To reflect our "rename and unwind" transformation to the normal form, we define decidable fragments in terms of *surrogates* [Hodkinson et al. 2000]. Let us reserve for every formula ϕ, whose main connective is a temporal operator, a unary predicate $P_\phi(x)$, and for every sentence ψ, whose main connective is a temporal operator, a propositional variable p_ψ. $P_\phi(x)$ and p_ψ are called surrogates. Given a monodic temporal formula ϕ, we denote by $\overline{\phi}$ the formula that results from ϕ by replacing all of its subformulae whose main connective is a temporal operator and which is not within a scope of another temporal operator with their surrogates.

Such an approach allows us to define decidable monodic classes based on the properties of surrogates analogously to the classical first-order decision problem [Börger et al. 1997]. Note however, that it is necessary to take into consideration occurrences of temporal operators as the following example shows.

Example 8.1. The first-order formula $\exists x\forall y\forall z\exists u\Phi(x, y, z, u)$, where Φ is quantifier free, belongs to the classical decidable fragment $\exists^*\forall^2\exists^*$. Let us consider the temporal formula $\exists x\Box\Diamond\forall y\forall z\exists u\Phi(x, y, z, u)$ with the same Φ. It is not hard to see that after the translation into DSNF (see Example 3.6), the first formula from $\mathcal{U}$ does not belong to $\exists^*\forall^2\exists^*$ any more. (Formally, it belongs to the undecidable Surányi [Börger et al. 1997] class $\forall^3\exists$.)

The following definition takes into account the considerations above.

Definition 8.2 *Temporalisation by Renaming*. Let $\mathfrak{C}$ be a class of first-order formulae. Let ϕ be a monodic temporal formula in Negation Normal Form (that is, the only boolean connectives are conjunction, disjunction and negation, and negations are only applied to atoms). We say that ϕ belongs to the class $\mathcal{T}_{ren}\mathfrak{C}$ if

(1) $\overline{\phi}$ belongs to $\mathfrak{C}$ and

(2) for every subformula of the form $\mathcal{T}\psi$, where $\mathcal{T}$ is a temporal operator (or of the form $\psi_1\mathcal{T}\psi_2$ if $\mathcal{T}$ is binary), either $\overline{\psi}$ is a closed formula belonging to $\mathfrak{C}$ or the formula $\forall x(P(x) \Rightarrow \overline{\psi})$, where P is a new unary predicate symbol, belongs to $\mathfrak{C}$ (analogous conditions for ψ_1, ψ_2).

Note that the formulae indicated in the first and second items of the definition exactly match the shape of the formulae contributing to $\mathcal{U}$ when we reduce a temporal formula to the normal form by renaming the complex expressions and replacing temporal operators by their fixpoint definitions.

THEOREM 8.3 DECIDABILITY BY TEMPORAL RESOLUTION. *Let $\mathfrak{C}$ be a decidable class of first-order formulae which does not contain equality and functional symbols, but possibly contains constants, such that*

—$\mathfrak{C}$ is closed under conjunction;

—$\mathfrak{C}$ contains monadic formulae.

Then $\mathcal{T}_{ren}\mathfrak{C}$ is decidable.

Proof After reduction to DSNF, all formulae from $\mathcal{U}$ belong to $\mathfrak{C}$. The (monadic) formulae from side conditions and the (monadic) formulae generated by temporal resolution rules belong to $\mathfrak{C}$. Theorem 4.13gives the decision procedure. ❑

Theorem 8.3 provides the possibility of using temporal resolution to confirm decidability of all temporal monodic classes listed in [Hodkinson et al. 2000; Wolter and Zakharyaschev 2002a]: *monadic, two-variable, fluted, guarded and loosely guarded*. Moreover, combining the constructions from [Hodkinson 2002] and the saturation-based decision procedure for the guarded fragment with equality [Ganzinger and De Nivelle 1999], it is possible to build a temporal resolution decision procedure for the monodic guarded and loosely guarded fragments with equality [Degtyarev et al. 2003a].

In addition, using the above theorem, we also obtain decidability of some monodic *prefix-like* classes.

COROLLARY 8.4 TEMPORALISED GÖDEL CLASS. *The class $\mathcal{T}_{ren}\exists^*\forall^2\exists^*$ is decidable.*

Proof Every monadic formula can be reduced, in a satisfiability equivalence preserving way, to a conjunction of formulae of the form $\forall x(l_1 \vee \cdots \vee l_p \vee L_1(x) \vee \cdots \vee L_q(x))$, $p, q \geq 0$ or $\exists x(L_1(x) \wedge \cdots \wedge L_r(x))$, $r \geq 0$, where l_j are ground literals and $L_j(x)$ are non-ground literals. Obviously, every conjunct is in $\exists^*\forall^2\exists^*$. Satisfiability of a conjunction of formulae belonging to $\exists^*\forall^2\exists^*$ is decidable, e.g. by the resolution-based technique (see clause set class $\mathcal{S}^+$ in [Fermüller et al. 2001]). ❑

COROLLARY 8.5 TEMPORALISED MASLOV CLASS. *The class $\mathcal{T}_{ren}K$ is decidable (where K is the Maslov class [Maslov 1968]).*

Proof Again, monadic formulae can be rewritten as a conjunction of Maslov formulae; satisfiability of a conjunction of Maslov formulae is decidable as shown in [Hustadt and Schmidt 1999]. ❑

```
Input. A temporal problem P and an eventuality clause $\Diamond L(x) \in \mathcal{E}$.
Output. A formula $H(x)$ with at most one free variable.
Method:. (1) Let $H_0(x) = \textbf{true}$; $N_0 = \emptyset$; $i = 0$.
         (2) Let $N_{i+1} = \{\forall x(\mathcal{A}_j^{(i+1)}(x) \Rightarrow \bigcirc \mathcal{B}_j^{(i+1)}(x))\}_{j=1}^{k}$ be the set of all full merged step clauses such that for every $j \in \{1 \dots k\}$, $\forall x(\mathcal{U} \wedge \mathcal{B}_j^{(i+1)}(x) \Rightarrow (\neg L(x) \wedge H_i(x)))$ holds. (The set $N_{i+1}$ possibly includes the degenerate clause $\textbf{true} \Rightarrow \bigcirc \textbf{true}$ in the case $\mathcal{U} \models \forall x(\neg L(x) \wedge H_i(x))$.)
         (3) If $N_{i+1} = \emptyset$, return false; else let $H_{i+1}(x) = \bigvee_{j=1}^{k} (\mathcal{A}_j^{(i+1)}(x))$.
         (4) If $\forall x(H_i(x) \Rightarrow H_{i+1}(x))$ return $H_{i+1}(x)$.
         (5) $i = i + 1$; goto 2.
```

Fig. 3. Breadth First Search algorithm.

Example 8.6. Let us consider the temporal formula $\Psi = \exists x \Box \Diamond \forall y \forall z \exists u \Phi(x, y, z, u)$ from example 8.1. In case when $\Phi(x, y, z, u)$ is a literal, the first formula from $\mathcal{U}$ (see Example 3.6) belongs to the Maslov class, and, thus, Ψ belongs to $\mathcal{T}_{ren}K$. If, however, Φ is a complex formula, for example, $\Phi(x, y, z, u) = Q_1(x, y) \vee Q_2(y, z) \vee Q(x, y, z, u)$, the first formula from $\mathcal{U}$ does not belong to K any more.

9. LOOP SEARCH ALGORITHM

The notion of a full merged step clause given in Section 5 is quite involved and the search for appropriate merging of simpler clauses is computationally hard. Finding *sets* of such full merged clauses needed for the temporal resolution rule is even more difficult. In Fig. 3 we present a search algorithm that finds a *loop formula* (cf. page 12) — a disjunction of the left-hand sides of full merged step clauses that together with an eventuality literal form the premises for the temporal resolution rule. The algorithm is based on Dixon's loop search algorithm for the propositional case [Dixon 1996]. For simplicity, in what follows we consider non-ground eventualities only. The algorithm and the proof of its properties for the ground case can be obtained by considering merged derived step clauses instead of the general case and by deleting the parameter "x" and quantifiers. We are going to show now that the algorithm terminates (Lemma 9.2), its output is a loop formula (lemmas 9.3 and 9.4), and temporal resolution is complete if we consider only the loops generated by the algorithm (Theorem 9.5).

LEMMA 9.1. *The formulae $H_i(x)$, $i \geq 0$, constructed by the BFS algorithm, satisfy the following property: $\forall x(H_{i+1}(x) \Rightarrow H_i(x))$.*

Proof By induction. In the base case $i = 0$, we have $H_0(x) \equiv$ **true** and, obviously, $\forall x(H_1(x) \Rightarrow$ **true**$)$. The induction hypothesis is that $\forall x(H_i(x) \Rightarrow H_{i-1}(x))$. In the induction step, let $N_{i+1} \neq \emptyset$ (otherwise, $H_{i+1}(x) \equiv$ **false** and, evidently, $\forall x($**false** $\Rightarrow H_i(x))$ holds). Let $N_{i+1} = \{\forall x(\mathcal{A}_j^{(i+1)}(x) \Rightarrow \bigcirc \mathcal{B}_j^{(i+1)}(x))\}_{j=1}^{k}$. For every $j \in \{1 \dots k\}$ we have $\forall x(\mathcal{B}_j^{(i+1)}(x) \Rightarrow (\neg L(x) \wedge H_i(x)))$. By the induction hypothesis, $\forall x(H_i(x) \Rightarrow H_{i-1}(x))$ and, therefore, $\forall x(\mathcal{B}_j^{(i+1)}(x) \Rightarrow (\neg L(x) \wedge H_{i-1}(x)))$, that is, $N_{i+1} \subset N_i$. It follows that $\forall x(H_{i+1}(x) \Rightarrow H_i(x))$. ❑

LEMMA 9.2. *The BFS algorithm terminates.*

Proof There are only finitely many different $H_i(x)$. Therefore, either there exists k such that $H_k(x) \equiv$ **false** and the algorithm terminates by step 3, or there exist $l, m : l < m$ such that $\forall x(H_l(x) \equiv H_m(x))$. In the latter case, by Lemma 9.1, we have $\forall x(H_{m-1}(x) \Rightarrow H_l(x))$, that is $\forall x(H_{m-1}(x) \Rightarrow H_m(x))$. By step 4, the algorithm terminates. ❏

LEMMA 9.3. *Let $H(x)$ be a formula produced by the BFS algorithm. Then $\forall x(\mathcal{U} \wedge H(x) \Rightarrow \bigcirc \Box \neg L(x))$.*

Proof If $H(x) =$ **false**, the lemma holds. Otherwise, consider the last computed set N_{i+1} (that is, $H(x) = H_{i+1}(x)$). Let $N_{i+1} = \{\forall x(\mathcal{A}_j^{(i+1)}(x) \Rightarrow \bigcirc \mathcal{B}_j^{(i+1)}(x))\}_{j=1}^{k}$. Note that for all $j \in \{1 \ldots k\}$, it holds $\forall x(\mathcal{U} \wedge \mathcal{B}_j^{(i+1)}(x) \Rightarrow \neg L(x))$ and, since $\forall x(H_i(x) \Rightarrow H_{i+1}(x))$, we also have $\forall x(\mathcal{U} \wedge \mathcal{B}_j^{(i+1)}(x) \Rightarrow H_{i+1}(x))$, that is, N_{i+1} is a loop and $H_{i+1}(x)$ is its loop formula. ❏

LEMMA 9.4. *Let P be a monodic temporal problem, $\mathcal{L}$ be a loop in $\Diamond L(x) \in \mathcal{E}$, and $\mathbf{L}(x)$ be its loop formula. Then for the formula $H(x)$, produced by the BFS algorithm on $\Diamond L(x)$, the following holds: $\forall x(\mathbf{L}(x) \Rightarrow H(x))$.*

Proof We show by induction that for all sets of full merged step clauses N_{i+1}, constructed by the algorithm, $\mathcal{L} \subset N_{i+1}$. In the base case $i = 0$, $H_0(x) \equiv$ **true** and for every full merged step clause $\forall x(\mathcal{A}(x) \Rightarrow \mathcal{B}(x)) \in \mathcal{L}$, we have $\forall x(\mathcal{U} \wedge \mathcal{B}(x) \Rightarrow (\neg L(x) \wedge \textbf{true}))$; therefore, $\mathcal{L} \subset N_1$.

Our induction hypothesis is that $\mathcal{L} \subset N_i$, that is, $N_i = \mathcal{L} \cup N_i'$. Then $H_i(x) = \mathbf{L}(x) \vee H_i'(x)$. Let $\forall x(\mathcal{A}(x) \Rightarrow \mathcal{B}(x))$ be any full merged step clause from $\mathcal{L}$. By the definition of a loop, $\forall x(\mathcal{U} \wedge \mathcal{B}(x) \Rightarrow (\neg L(x) \wedge \mathbf{L}(x)))$, hence, $\forall x(\mathcal{U} \wedge \mathcal{B}(x) \Rightarrow ((\neg L(x) \wedge \mathbf{L}(x)) \vee (\neg L(x) \wedge H_i'(x))))$, that is, $\forall x(\mathcal{U} \wedge \mathcal{B}(x) \Rightarrow (\neg L(x) \wedge H_i(x)))$. Since the set N_{i+1} consists of all full merged step clauses, $\forall x(\mathcal{A}_j^{(i+1)}(x) \Rightarrow \bigcirc \mathcal{B}_j^{(i+1)}(x))$, such that $\forall x(\mathcal{U} \wedge \mathcal{B}_j^{(i+1)}(x) \Rightarrow (\neg L(x) \wedge H_i(x)))$ holds, we have $\forall x(\mathcal{A}(x) \Rightarrow \mathcal{B}(x)) \in N_{i+1}$. As $\forall x(\mathcal{A}(x) \Rightarrow \mathcal{B}(x))$ is an arbitrary full merged step clause from $\mathcal{L}$, it means that $\mathcal{L} \subset N_{i+1}$.

It follows that $\forall x(\mathbf{L}(x) \Rightarrow H(x))$. ❏

The proof of the completeness theorem goes by showing that there exists an eventuality $\Diamond L(x) \in \mathcal{E}$ and a loop $\mathcal{L} = \{\forall x(\mathcal{A}_i(x) \Rightarrow \bigcirc \mathcal{B}_i(x))\}_{i=1}^{k}$ such that the application of the eventuality resolution rule to $\Diamond L(x)$ and $\mathcal{L}$ leads to the deletion of some vertices from the eventuality graph. A vertex $\mathcal{C}$ is deleted from the graph if the categorical formula, $\mathcal{F}_{\mathcal{C}}$, together with the universal part, $\mathcal{U}$, is satisfiable, but $\mathcal{F}_{\mathcal{C}} \wedge \forall x \neg \bigvee_{j=1}^{k} \mathcal{A}_j(x) \wedge \mathcal{U}$ is unsatisfiable.

THEOREM 9.5 RELATIVE COMPLETNESS. *Temporal resolution is complete if we restrict ourselves to loops found by the BFS algorithm.*

Proof Let $H(x)$ be the output of the BFS algorithm, let $\mathbf{L}(x) \overset{\text{def}}{=} \bigvee_{j=1}^{k} \mathcal{A}_j(x)$. By Lemma 9.4, $\forall x(L(x) \Rightarrow H(x))$ holds; therefore, $H(x)$ is not **false**. From the proof of Lemma 9.3 it follows that the last computed set N_{i+1} (that is, $H(x) = H_{i+1}(x)$) is a loop in $\Diamond L(x)$ and $H(x)$ is its loop formula. Since $\forall x(\mathbf{L}(x) \Rightarrow H(x))$, the formula $\mathcal{F}_{\mathcal{C}} \wedge \forall x \neg H(x) \wedge \mathcal{U}$ is unsatisfiable as well and the application of the eventuality resolution rule to $\Diamond L(x)$ and N_{i+1} leads to deletion of at least the same vertices from the eventuality graph. ❏

Note 9.6. The need to include *all* full merged step clauses satisfying some particular conditions into N_{i+1} might lead to quite extensive computations. Note however that due to the trivial fact that if $\forall x(A(x) \Rightarrow B(x))$ then $\forall x((A(x) \vee B(x)) \equiv B(x))$, we can restrict the choice to only those full merged step clauses whose left-hand sides do not imply the left-hand side of any other clause in N_{i+1} yielding a formula $H'_{i+1}(x)$ equivalent to the original formula $H_{i+1}(x)$.

Example 9.7. Let us consider an unsatisfiable monodic temporal problem, P, given by

$$\begin{array}{rcl} \mathcal{I} & = & \{\exists x A(x)\}, \\ \mathcal{U} & = & \{\forall x(B(x) \Rightarrow A(x) \wedge \neg L(x))\}, \\ \mathcal{S} & = & \{A(x) \Rightarrow \bigcirc B(x)\}, \\ \mathcal{E} & = & \{\Diamond L(x)\} \end{array}$$

and apply the BFS algorithm to $\Diamond L(x)$.

The set of all full merged step clauses, N_1, whose right-hand sides imply $\neg L(x)$, is:

$$(\forall y A(y)) \;\Rightarrow\; \bigcirc(\forall y B(y)), \tag{12}$$

$$(A(x) \wedge \forall y A(y)) \;\Rightarrow\; \bigcirc(B(x) \wedge \forall y B(y)), \tag{13}$$

$$(A(x) \wedge \exists y A(y)) \;\Rightarrow\; \bigcirc(B(x) \wedge \exists y B(y)). \tag{14}$$

Note that $\forall x(\forall y A(y) \Rightarrow A(x) \wedge \forall y A(y))$ and $\forall x(A(x) \wedge \forall y A(y)) \Rightarrow A(x) \wedge \exists y A(y))$; therefore, clauses (12) and (13) can be deleted from N_1 yielding

$$N'_1 = \{(A(x) \wedge \exists y A(y)) \Rightarrow \bigcirc(B(x) \wedge \exists y B(y))\} \quad \text{and} \quad H'_1(x) = (A(x) \wedge \exists y A(y)).$$

The set of all full merged step clauses N_2 whose right-hand sides imply $L(x) \wedge H'_1(x)$ coincides with N_1 and the output of the algorithm is $H'_2(x) \equiv H'_1(x)$. The conclusion of the eventuality resolution rule, $\forall x \neg A(x) \vee \neg\exists y A(y)$, simplified to $\forall x \neg A(x)$, contradicts the initial part of the problem.

Note that all full merged step clauses from N_1 are loops in $\Diamond L(x)$, but both conclusions of the eventuality resolution rule, applied to the loops (12) and (13), can be simplified to $\exists x \neg A(x)$ which does not contradict the initial part.

10. SEMANTICS WITH EXPANDING DOMAINS

So far, we have been considering temporal formulae interpreted over models with the *constant domain assumption*. In this section we consider another important case, namely models that have *expanding domains*. Although it is known that satisfiability over expanding domains can be reduced to satisfiability over constant domains [Wolter and Zakharyaschev 2001], we here provide a procedure that can be applied directly to expanding domain problems. Our interest in such problems is partly motivated by the fact that the expanding domain assumption leads to a simpler calculus, more amenable to practical implementation [Konev et al. 2003b], and partly by the correspondence between expanding domain problems and important applications, such as spatio-temporal logics [Wolter and Zakharyaschev 2002b; Gabelaia et al. 2003] and temporal description logics [Artale et al. 2002]. In addition, the way we refine the calculus of Section 4 to the expanding domain case constitutes, we believe, an elegant and significant simplification.

We begin by presenting the expanding domain semantics and proceed to give the give the resolution calculus for the expanding domain case.

Under expanding domain semantics, formulae of FOTL are interpreted in first-order temporal structures of the form $\mathfrak{M} = \langle D_n, I_n \rangle$, $n \in \mathbb{N}$, where every D_n is a non-empty set such that whenever $n < m$, $D_n \subseteq D_m$, and I_n is an interpretation of predicate and constant symbols over D_n. Again, we require that the interpretation of constants is rigid. A (variable) assignment $\mathfrak{a}$ is a function from the set of individual variables to $\cup_{n \in \mathbb{N}} D_n$; the set of all assignments is denoted by $\mathfrak{V}$.

For every moment of time n, the corresponding first-order structure, $\mathfrak{M}_n = \langle D_n, I_n \rangle$; the corresponding set of variable assignments $\mathfrak{V}_n$ is a subset of the set of all assignments, $\mathfrak{V}_n = \{\mathfrak{a} \in \mathfrak{V} \mid \mathfrak{a}(x) \in D_n$ for every variable $x\}$; clearly, $\mathfrak{V}_n \subseteq \mathfrak{V}_m$ if $n < m$.

Then, the *truth* relation $\mathfrak{M}_n \models^{\mathfrak{a}} \phi$ in a structure $\mathfrak{M}$ is defined inductively in the same way as in the constant domain case, but *only for those assignments* $\mathfrak{a}$ *that satisfy the condition* $\mathfrak{a} \in \mathfrak{V}_n$.

Example 10.1. The formula $\forall x P(x) \wedge \Box(\forall x P(x) \Rightarrow \bigcirc \forall x P(x)) \wedge \Diamond \exists y \neg P(y)$ is unsatisfiable over both expanding and constant domains; the formula $\forall x P(x) \wedge \Box(\forall x (P(x) \Rightarrow \bigcirc P(x))) \wedge \Diamond \exists y \neg P(y)$ is unsatisfiable over constant domains but has a model with an expanding domain.

It can be seen that our earlier reduction to DSNF holds for the expanding domain case (the only difficulty is Lemma 3.3 where, in defining $waitforL(d)$, we must consider cases where $\mathfrak{M}_k \models \Box\Diamond P(d)$ or $\mathfrak{M}_k \models \Diamond\Box\neg P(d)$ where k is the moment when d "appears").

The calculus itself coincides with the calculus given in Section 4; the only difference occurs in the merging operation. As Example 10.1 shows, the derived step clause (5) is not a logical consequence of (4) in the expanding domain case. Surprisingly, if we omit derived step clauses of this form, we not only obtain a correct calculus, but also a complete calculus for the expanding domain case!

Definition 10.2 *Derived Step Clauses: Expanding Domains*. Let P be a monodic temporal problem, and let

$$P_{i_1}(x) \Rightarrow \bigcirc M_{i_1}(x), \ldots, P_{i_k}(x) \Rightarrow \bigcirc M_{i_k}(x)$$

be a subset of the set of its original non-ground step clauses. Then

$$\exists x (P_{i_1}(x) \wedge \cdots \wedge P_{i_k}(x)) \Rightarrow \bigcirc \exists x (M_{i_1}(x) \wedge \cdots \wedge M_{i_k}(x)),$$
$$P_{i_j}(c) \Rightarrow \bigcirc M_{i_j}(c)$$

are *e-derived* step clauses, where c is a constant occurring in P.

The notions of a merged derived and full step clause as well as the calculus itself are exactly the same as in Section 4.

Correctness of this calculus is again straightforward. As for completeness, we have to slightly modify the proof of Section 5.

The proof of Theorem 4.13 relies on the theorem on existence of a model, Theorem 5.15, and it can be seen that if we prove an analog of Theorem 5.15 for the expanding domain case, the given proof of completeness holds for this case.

We outline here how to modify the proof of Theorem 5.15 for the case of expanding domains. All the definitions and properties from Section 5 are transfered here with the following exceptions.

Now, the universally quantified part does not contribute either to $\mathcal{A}$ or $\mathcal{B}$.

$$\mathcal{A}_{\mathcal{C}} = \bigwedge_{\gamma\in\Gamma} \exists x A_\gamma(x) \wedge A_\theta \wedge \bigwedge_{c\in\mathrm{const}(C)} A_{\rho(c)}(c),$$
$$\mathcal{B}_{\mathcal{C}} = \bigwedge_{\gamma\in\Gamma} \exists x B_\gamma(x) \wedge B_\theta \wedge \bigwedge_{c\in\mathrm{const}(C)} B_{\rho(c)}(c).$$

This change affects the suitability of predicate colours.

LEMMA 10.3 ANALOGUE OF LEMMA 5.10. *Let H be the behaviour graph for the problem $\mathsf{P} = \langle \mathcal{U}, \mathcal{I}, \mathcal{S}, \mathcal{E}\rangle$ with an edge from a vertex $\mathcal{C} = \langle \Gamma, \theta, \rho\rangle$ to a vertex $\mathcal{C}' = \langle \Gamma', \theta', \rho'\rangle$. Then*

1. *for every $\gamma \in \Gamma$ there exists a $\gamma' \in \Gamma'$ such that the pair (γ, γ') is suitable;*
3. *the pair of propositional colours (θ, θ') is suitable;*
4. *the pair of constant distributions (ρ, ρ') is suitable.*

Note that the missing condition 2. of Lemma 5.10 does not hold in the expanding domain case. However, under the conditions of Lemma 10.3, if $\gamma' = \rho'(c)$, for some $c \in \mathrm{const}(\mathsf{P})$, there always exists a $\gamma \in \Gamma$ such that the pair (γ, γ') is suitable.

Since for a predicate colour γ there may not exist a colour γ' such that the pair (γ', γ) is suitable, the notion of a run is reformulated.

Definition 10.4 *Run*. Let π be a path through a behaviour graph H of a temporal problem P. By a *run* in π we mean a function $r(n)$ mapping its domain, $dom(r) = \{n \in \mathbb{N} \mid n \geq n_0\}$ for some $n_0 \in \mathbb{N}$, to $\bigcup_{i\in\mathbb{N}} \Gamma_i$ such that for every $n \in dom(r)$, $r(n) \in \Gamma_n$, $r(n)$ the pair $(r(n), r(n+1))$ is suitable.

Finally, the proof of Lemma 5.16 is modified as follows.

Proof [of Lemma 5.16 for the expanding domain case] We construct a path, π, through the behaviour graph, H, satisfying properties (a), (b), and (d) in exactly the same way as in the proof for constant domains. The only difference is in the way how we prove condition (c). We assume the denotation from that proof. So, let $\mathcal{C} = \pi(i)$ and $\gamma \in \Gamma_{\mathcal{C}}$.

Let $\mathcal{C} = \pi(i)$ and $\gamma \in \Gamma_{\mathcal{C}}$. Then there exists $\gamma'' \in \mathcal{C}_n$ such that $(\mathcal{C}, \gamma) \to^+ (\mathcal{C}_n, \gamma'')$. Since for every $\gamma'' \in \mathcal{C}_n$ there exists $\gamma''' \in \mathcal{C}_n^{(\gamma_{s_n}, L_k)}$ such that all eventualities are satisfied on the run-segment from γ'' to γ''' and there exists $\gamma^{(4)} \in \mathcal{C}_n$, $(\mathcal{C}_n^{(\gamma_{s_n}, L_k)}, \gamma''') \to^+ (\mathcal{C}_n, \gamma^{(4)})$, then there is an e-run, r, such that $r(i) = \gamma$, i.e., property (c) holds. (Note that we now do not assume that the e-run must start at $\mathcal{C}_0$.) ❏

This contributes to the following theorem.

THEOREM 10.5 EXPANDING DOMAINS: CORRECTNESS AND COMPLETNESS. *The rules of temporal resolution preserve satisfiability. Let an arbitrary monodic temporal problem P be unsatisfiable over expanding domain, then any* fair *derivation by temporal resolution from P^c successfully terminates.*

11. IMPLEMENTATION

Temporal resolution, as described in this paper, can be implemented in a straightforward way—one would enumerate all possible (full) merged step clauses and extend the universal part by all possible conclusions. While this approach benefits from the ability to employ *any* classical first-order method to test applicability of the rules, thus making the method widely applicable, it is obviously not very practical.

For the case when the universal part fits into a "nice" first-order fragment which can be decided by resolution [Fermüller et al. 2001], we have developed a more machine-oriented calculus which we call *fine-grained* temporal resolution [Konev et al. 2003a; 2003b]. We show that the set of conclusions by fine-grained resolution coincides with the set of conclusions by the inference rules given in this paper proving thus soundness and completeness of the new system. At the moment, we are performing some preliminary experiments combining the propositional resolution temporal prover TRP++ [Hustadt and Konev 2003] and a successful classical first-order theorem prover, Vampire [Riazanov and Voronkov 2001]; a report on the combination can be found in [Hustadt et al. 2004].

12. CONCLUSIONS

In this paper, we have modified and extended the clausal temporal resolution technique in order to enable its use in monodic FOTL. We have developed a specific normal form for FOTL and have provided a complete resolution calculus for formulae in this form. The use of this technique has provided us with increased understanding of the monodic fragment, allowing definitions of new decidable monodic classes, simplification of existing monodic classes by reductions, and completeness of clausal temporal resolution in the case of monodic logics with expanding domains.

However, not only is this approach useful in examining and extending the monodic fragment, but it is being used as the basis for a practical proof technique for certain monodic classes [Konev et al. 2003b]. Refining and analysing this implementation forms part of our future work, as does the application of this technique in practice. Since formulae such as $\Box\forall x, y(p(x,y) \Rightarrow \bigcirc p(x,y))$ are non-monodic, the monodic restriction disallows us from describing the evolution of relational structures in general. However, within the monodic class we still have the capability of describing interesting and useful systems, particularly where the evolution of properties of components is required (e.g. via formulae such as $\Box\forall x(p(x) \Rightarrow \bigcirc p(x)))$ which allows us to apply monodic reasoning to a range of areas including program verification [Fisher and Lisitsa 2003], temporal description logics [Artale and Franconi 2004], agent theories [Fisher and Ghidini 2002] and spatio-temporal logics [Wolter and Zakharyaschev 2002b].

We are developing an implementation of fine-grained temporal resolution; a detailed description of the system is a matter of forthcoming publications.

Acknowledgements We thank the anonymous referees for their helpful comments and suggestions for future work. The authors would like to acknowledge support from EPSRC via research grants GR/M46631 and GR/R45376. The third author is supported in part by Award No. RM1-2409-ST-02 of the U.S. Civilian Research & Development Foundation for the Independent States of the Former Soviet Union (CRDF).

REFERENCES

ARTALE, A. AND FRANCONI, E. 2004. Temporal description logics. In *Handbook of Temporal Reasoning in Artificial Intelligence*, M. Fisher, D. M. Gabbay, and L. Vila, Eds. Elsevier. (To appear).

ARTALE, A., FRANCONI, E., WOLTER, F., AND ZAKHARYASCHEV, M. 2002. A temporal description logic for reasoning over conceptual schemas and queries. In *Proceedings of JELIA'02*. LNCS, vol. 2424. Springer, 98–110.

BACHMAIR, L. AND GANZINGER, H. 2001. Resolution theorem proving. In *Handbook of Automated Reasoning*, A. Robinson and A. Voronkov, Eds. Elsevier, Chapter 2, 19–99.

BÖRGER, E., GRÄDEL, E., AND GUREVICH, Y. 1997. *The classical decision problem*. Springer.

DEGTYAREV, A. AND FISHER, M. 2001. Towards first-order temporal resolution. In *KI 2001, Proceedings*. LNCS, vol. 2174. Springer, 18–32.

DEGTYAREV, A., FISHER, M., AND KONEV, B. 2002. A simplified clausal resolution procedure for propositional linear-time temporal logic. In *Tableaux 2002, Proceedings*. Lecture Notes in Computer Science, vol. 2381. Springer, 85–99.

DEGTYAREV, A., FISHER, M., AND KONEV, B. 2003a. Handling equality in monodic temporal resolution. In *Proceedings of 10th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR)*. Lecture Notes in Computer Science, vol. 2850. Springer-Verlag, Almaty, Kazakhstan, 214–228.

DEGTYAREV, A., FISHER, M., AND KONEV, B. 2003b. Monodic temporal resolution. In *Proc. CADE-19*. LNAI, vol. 2741. Springer, 397–411.

DEGTYAREV, A., FISHER, M., AND LISITSA, A. 2002. Equality and monodic first-order temporal logic. *Studia Logica 72,* 2, 147–156.

DIXON, C. 1996. Search strategies for resolution in temporal logics. In *Proc. CADE-13*. LNAI, vol. 1104. Springer, 673–687.

FERMÜLLER, C., LEITSCH, A., HUSTADT, U., AND TAMMET, T. 2001. Resolution decision procedures. In *Handbook of Automated Reasoning*, A. Robinson and A. Voronkov, Eds. Vol. II. Elsevier, Chapter 25, 1791–1850.

FISHER, M. 1991. A resolution method for temporal logic. In *Proc. IJCAI'91*, J. Myopoulos and R. Reiter, Eds. Morgan Kaufman, 99–104.

FISHER, M. 1992. A normal form for first-order temporal formulae. In *11th International Conference on Automated Deduction*, D. Kapur, Ed. Lecture Notes in Artificial Intelligence, vol. 607. Springer Verlag, Saratoga Springs, NY, USA, 370–384.

FISHER, M. 1997. A normal form for temporal logics and its applications in theorem proving and execution. *Journal of Logic and Computation 7,* 4, 429–456.

FISHER, M., DIXON, C., AND PEIM, M. 2001. Clausal temporal resolution. *ACM Transactions on Computational Logic 2,* 1, 12–56.

FISHER, M. AND GHIDINI, C. 2002. The ABC of rational agent programming. In *Proc. First International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*. ACM Press, 849–856.

FISHER, M. AND LISITSA, A. 2003. Deductive verification of cache coherence protocols. In *Proceedings of 3rd International Workshop on Automated Verification of Critical Systems (AVoCS 2003)*. Southampton, UK, 177–186.

GABBAY, D. 1987. Declarative past and imperative future: executive temporal logic for interactive systems. In *Proceedings on Colloquium on Temporal Logic and Specification*, B. Banieqbal, H. Barringer, and A. Pnueli, Eds. LNCS, vol. 398. Springer Verlag, Altrincham, U.K., 402–450.

GABELAIA, D., KONTCHAKOV, R., KURUCZ, A., WOLTER, F., AND ZAKHARYASCHEV, M. 2003. On the computational complexity of spatio-temporal logics. In *Proc. 16th International FLAIRS Conference*. AAAI Press, 460–464.

GALLIER, H. 1986. *Logic for computer science*. Harper and Row, New York.

GANZINGER, H. AND DE NIVELLE, H. 1999. A superposition decision procedure for the guarded fragment with equality. In *Proc. 14th IEEE Symposium on Logic in Computer Science*. 295–305.

HODKINSON, I. 2002. Monodic packed fragment with equality is decidable. *Studia Logica 72,* 185–197.

HODKINSON, I., WOLTER, F., AND ZAKHARYASCHEV, M. 2000. Decidable fragments of first-order temporal logics. *Annals of Pure and Applied Logic 106,* 85–134.

HOLZMANN, G. J. 1997. The model checker Spin. *IEEE Trans. on Software Engineering 23,* 5, 279–295.

HUSTADT, U. AND KONEV, B. 2003. TRP++ 2.0: A temporal resolution prover. In *Proc. CADE-19*. LNAI, vol. 2741. Springer, 274–278.

HUSTADT, U., KONEV, B., VORONKOV, A., AND RIAZANOV, A. 2004. TeMP: A temporal monodic prover. Tech. Rep. ULCS-04-004, University of Liverpool, Department of Computer Science. Submitted.

HUSTADT, U. AND SCHMIDT, R. A. 1999. Maslov's class K revisited. In *Proceedings of the 16th International Conference on Automated Deduction (CADE-16)*, H. Ganzinger, Ed. LNAI, vol. 1632. Springer, 172–186.

KAIVOLA, R. 1995. Axiomatising linear-time mu-calculus. In *Proc. Concur'95*.

KESTEN, Y. AND PNUELI, A. 1995. A complete proof system for QPTL. In *Proceedings of 10th Annual IEEE Symposium on Logic in Computer Science (LICS)*. IEEE Computer Society Press, San Diego, California, USA, 2–12.

KONEV, B., DEGTYAREV, A., DIXON, C., FISHER, M., AND HUSTADT, U. 2003a. Mechanising first-order temporal resolution. Tech. Rep. ULCS-03-023, University of Liverpool, Department of Computer Science. Submitted.

KONEV, B., DEGTYAREV, A., DIXON, C., FISHER, M., AND HUSTADT, U. 2003b. Towards the implementation of first-order temporal resolution: the expanding domain case. In *Proceedings TIME-ICTL'03*. IEEE Computer Society Press, 72–82.

KONTCHAKOV, R., LUTZ, C., WOLTER, F., AND ZAKHARYASCHEV, M. 2004. Temporalising tableaux. *Studia Logica*, to appear.

MANNA, Z. AND PNUELI, A. 1992. *The temporal logic of reactive and concurrent systems: specification*. Springer.

MASLOV, S. 1968. The inverse method for establishing deducibility for logical calculi. *Trudy Math. Inst. Steklov XCVIII*, 22–25. Engl. transl. AMS 1971.

MERZ, S. 1992. Decidability and incompleteness results for first-order temporal logic of linear time. *Journal of Applied Non-Classical Logics 2*, 139–156.

NONNENGART, A. AND WEIDENBACH, C. 2001. Computing small clause normal forms. In *Handbook of Automated Reasoning*, A. Robinson and A. Voronkov, Eds. Elsevier, Chapter 6, 335–370.

PLAISTED, D. AND GREENBAUM, S. 1986. A structure-preserving clause form transformation. *Journal of Symbolic Computation 2*, 3 (Sept.), 293–304.

PNUELI, A. 1977. The Temporal Logic of Programs. In *Proceedings of the Eighteenth Symposium on the Foundations of Computer Science*. 46–57.

RIAZANOV, A. AND VORONKOV, A. 2001. Vampire 1.1 (System description). In *Proc. IJCAR 2001*. LNCS, vol. 2083. Springer.

SZALAS, A. AND HOLENDERSKI, L. 1988. Incompleteness of first-order temporal logic with until. *Theoretical Computer Science 57*, 317–325.

TSEITIN, G. 1983. On the complexity of derivations in propositional calculus. In *Automation of Reasoning (Classical papers on Computational Logic)*, J. Siekmann and G. Wrightson, Eds. Vol. 2. Springer Verlag, 466–483. Original paper (in Russian) appeared in 1968.

WOLPER, P. 1982. Synthesis of communicating processes from temporal logic specifications. Ph.D. thesis, Stanford University.

WOLTER, F. AND ZAKHARYASCHEV, M. 2001. Decidable fragments of first-order modal logics. *Journal of Symbolic Logic 66*, 1415–1438.

WOLTER, F. AND ZAKHARYASCHEV, M. 2002a. Axiomatizing the monodic fragment of first-order temporal logic. *Annals of Pure and Applied logic 118*, 133–145.

WOLTER, F. AND ZAKHARYASCHEV, M. 2002b. Qualitative spatio-temporal representation and reasoning: a computational perspective. In *Exploring Artificial Intelligence in the New Millenium*. Morgan Kaufmann, 175–216.

Received June 2003; revised February 2004; accepted April 2004