Redundancy Elimination in Monodic Temporal Reasoning

Michel Ludwig[1] and Ullrich Hustadt[1]

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

Abstract. The elimination of redundant clauses is an essential part of resolution-based theorem proving in order to reduce the size of the search space. In this paper we focus on ordered fine-grained resolution with selection, a sound and complete resolution-based calculus for monodic first-order temporal logic. We define a subsumption relation on temporal clauses, show how the calculus can be extended with reduction rules that eliminate redundant clauses, and we illustrate the effectiveness of redundancy elimination with some experiments.

1 Introduction

Monodic first-order temporal logic [9] is a fragment of first-order temporal logic (without equality) which, in contrast to first-order temporal logic itself, has a semi-decidable validity and satisfiability problem. Besides semi-decidability, monodic first-order temporal logic enjoys a number of other beneficial properties, e.g. the existence of non-trivial decidable subclasses, complete reasoning procedures, etc.

In addition to a tableaux-based calculus [13] for monodic first-order temporal logic, several resolution-based calculi have been proposed for the logic, starting with (monodic) temporal resolution [3]. A more machine-oriented version, the fine-grained first-order temporal resolution calculus, was described in [12]. Subsequently, a refinement of fine-grained temporal resolution, the ordered fine-grained temporal resolution with selection calculus, was presented in [10].

Essentially, the inference rules of ordered fine-grained resolution with selection can be classified into two different categories. The majority of the rules are based on standard first-order resolution between different types of temporal clauses. The remaining inference rules, the so-called eventuality resolution rules, reflect the induction principle that holds for monodic temporal logic over a flow of time isomorphic to the natural numbers. The applicability of the rules in this second category is only semi-decidable, making the construction of fair derivations, that is, derivations in which every non-redundant clause that is derivable from a given clause set is eventually derived, a non-trivial problem. A new inference procedure solving this problem has been recently been described in [15] and is implemented in the theorem prover TSPASS [14].

In this paper we focus on a different aspect of ordered fine-grained temporal resolution with selection, namely, redundancy elimination. The use of an ordering and a selection function which restricts inferences to literals which are selected or, in the absence of selected literals, to (strictly) maximal literals, already reduces the possible inferences considerably. However, it cannot prevent the derivation of redundant clauses, e.g. tautological clauses or clauses which are subsumed by other, simpler, clauses. Redundancy elimination is therefore an important ingredient for practical resolution calculi and for theorem provers based on such calculi.

The paper is organised as follows. In Section 2 we briefly recall the syntax and semantics of monodic first-order temporal logic. The ordered fine-grained resolution with selection calculus is presented in Section 3. In Sections 4 and 5 we show how redundancy elimination can be added to the calculus. Finally, in Section 6 we briefly discuss how redundancy elimination fits with our implementation of the calculus and present some experimental results which show the effectiveness of redundancy elimination.

2 First-Order Temporal Logic

We assume the reader to be familiar with first-order logic and associated notions, including, for example, terms and substitutions.

Then, the language of First-Order (Linear Time) Temporal Logic, FOTL, is an extension of classical first-order logic by temporal operators for a discrete linear model of time (i.e. isomorphic to $\mathbb{N}$). The vocabulary of FOTL (without equality and function symbols) is composed of a countably infinite set X of *variables* x_0, x_1, ..., a countably infinite set of *constants* c_0, c_1, ..., a non-empty set of *predicate symbols* P, P_0, ..., each with a fixed arity ≥ 0, the *propositional operators* $\top$ (**true**), $\neg$, $\vee$, the *quantifiers* $\exists x_i$ and $\forall x_i$, and the *temporal operators* $\Box$ ('always in the future'), $\Diamond$ ('eventually in the future'), $\bigcirc$ ('at the next moment'), U ('until') and W ('weak until') (see e.g. [7]). We also use $\perp$ (**false**), $\wedge$, and $\Rightarrow$ as additional operators, defined using $\top$, $\neg$, and $\vee$ in the usual way. The set of FOTL formulae is defined as follows: $\top$ is a FOTL formula; if P is an n-ary predicate symbol and t_1, ..., t_n are variables or constants, then $P(t_1, \ldots, t_n)$ is an *atomic* FOTL formula; if φ and ψ are FOTL formulae, then so are $\neg\varphi$, $\varphi \vee \psi$, $\exists x\varphi$, $\forall x\varphi$, $\Box\varphi$, $\Diamond\varphi$, $\bigcirc\varphi$, $\varphi \,\mathsf{U}\, \psi$, and $\varphi \,\mathsf{W}\, \psi$. Free and bound variables of a formula are defined in the standard way, as well as the notions of open and closed formulae. For a given formula φ, we write $\varphi(x_1, \ldots, x_n)$ to indicate that all the free variables of φ are among x_1, ..., x_n. As usual, a *literal* is either an atomic formula or its negation, and a *proposition* is a predicate of arity 0.

Formulae of this logic are interpreted over structures $\mathfrak{M} = (D_n, I_n)_{n\in\mathbb{N}}$ that associate with each element n of $\mathbb{N}$, representing a moment in time, a first-order structure $\mathfrak{M}_n = (D_n, I_n)$ with its own non-empty domain D_n and interpretation I_n. An *assignment* $\mathfrak{a}$ is a function from the set of variables to $\bigcup_{n\in\mathbb{N}} D_n$. The application of an assignment to formulae, predicates, constants and variables is defined in the standard way, in particular, $\mathfrak{a}(c) = c$ for every constant c. The

$\mathfrak{M}_n \models^{\mathfrak{a}} \top$	
$\mathfrak{M}_n \models^{\mathfrak{a}} P(t_1, \ldots, t_n)$	iff $(I_n(\mathfrak{a}(t_1)), \ldots, I_n(\mathfrak{a}(t_n))) \in I_n(P)$
$\mathfrak{M}_n \models^{\mathfrak{a}} \neg\varphi$	iff not $\mathfrak{M}_n \models^{\mathfrak{a}} \varphi$
$\mathfrak{M}_n \models^{\mathfrak{a}} \varphi \lor \psi$	iff $\mathfrak{M}_n \models^{\mathfrak{a}} \varphi$ or $\mathfrak{M}_n \models^{\mathfrak{a}} \psi$
$\mathfrak{M}_n \models^{\mathfrak{a}} \exists x\varphi$	iff $\mathfrak{M}_n \models^{\mathfrak{b}} \varphi$ for some assignment $\mathfrak{b}$ that may differ from $\mathfrak{a}$ only in x and such that $\mathfrak{b}(x) \in D_n$
$\mathfrak{M}_n \models^{\mathfrak{a}} \forall x\varphi$	iff $\mathfrak{M}_n \models^{\mathfrak{b}} \varphi$ for every assignment $\mathfrak{b}$ that may differ from $\mathfrak{a}$ only in x and such that $\mathfrak{b}(x) \in D_n$
$\mathfrak{M}_n \models^{\mathfrak{a}} \bigcirc\varphi$	iff $\mathfrak{M}_{n+1} \models^{\mathfrak{a}} \varphi$
$\mathfrak{M}_n \models^{\mathfrak{a}} \Diamond\varphi$	iff there exists $m \geq n$ such that $\mathfrak{M}_m \models^{\mathfrak{a}} \varphi$
$\mathfrak{M}_n \models^{\mathfrak{a}} \Box\varphi$	iff for all $m \geq n$, $\mathfrak{M}_m \models^{\mathfrak{a}} \varphi$
$\mathfrak{M}_n \models^{\mathfrak{a}} \varphi \,\mathsf{U}\, \psi$	iff there exists $m \geq n$ such that $\mathfrak{M}_m \models^{\mathfrak{a}} \psi$ and $\mathfrak{M}_i \models^{\mathfrak{a}} \varphi$ for every $i, n \leq i < m$
$\mathfrak{M}_n \models^{\mathfrak{a}} \varphi \,\mathsf{W}\, \psi$	iff $\mathfrak{M}_n \models^{\mathfrak{a}} \varphi \,\mathsf{U}\, \psi$ or $\mathfrak{M}_n \models^{\mathfrak{a}} \Box\varphi$

Fig. 1. Truth-Relation for First-Order Temporal Logic

definition of the *truth relation* $\mathfrak{M}_n \models^{\mathfrak{a}} \varphi$ (only for those $\mathfrak{a}$ such that $\mathfrak{a}(x) \in D_n$ for every variable x) is given in Fig. 1.

In this paper we make the *expanding domain assumption*, that is, $D_n \subseteq D_m$ if $n < m$, and we assume that the interpretation of constants is *rigid*, that is, $I_n(c) = I_m(c)$ for all $n, m \in \mathbb{N}$.

A structure $\mathfrak{M} = (D_n, I_n)_{n\in\mathbb{N}}$ is said to be a *model* for a formula φ if and only if for every assignment $\mathfrak{a}$ with $\mathfrak{a}(x) \in D_0$ for every variable x it holds that $\mathfrak{M}_0 \models^{\mathfrak{a}} \varphi$. A formula is *satisfiable* if and only there exists a model for φ. A formula φ is *valid* if and only if every temporal structure $\mathfrak{M} = (D_n, I_n)_{n\in\mathbb{N}}$ is a model for φ.

The set of valid formulae of this logic is not recursively enumerable. However, the set of valid *monodic* formulae is known to be finitely axiomatisable [19]. A formula φ of FOTL is called *monodic* if any subformula of φ of the form $\bigcirc\psi$, $\Box\psi$, $\Diamond\psi$, $\psi_1 \,\mathsf{U}\, \psi_2$, or $\psi_1 \,\mathsf{W}\, \psi_2$ contains at most one free variable. For example, the formulae $\exists x\Box\forall y P(x, y)$ and $\forall x\Box P(c, x)$ are monodic, whereas the formula $\forall x\exists y(Q(x, y) \Rightarrow \Box Q(x, y))$ is not monodic.

Every monodic temporal formula can be transformed into an equi-satisfiable normal form, called *divided separated normal form (DSNF)* [12].

Definition 1. *A* monodic temporal problem P in divided separated normal form (DSNF) *is a quadruple $\langle \mathcal{U}, \mathcal{I}, \mathcal{S}, \mathcal{E} \rangle$, where the universal part $\mathcal{U}$ and the initial part $\mathcal{I}$ are finite sets of first-order formulae; the step part $\mathcal{S}$ is a finite set of* step clauses *of the form $p \Rightarrow \bigcirc q$, where p and q are propositions, and $P(x) \Rightarrow \bigcirc Q(x)$, where P and Q are unary predicate symbols and x is a variable; and the eventuality part $\mathcal{E}$ is a finite set of formulae of the form $\Diamond L(x)$ (a* non-ground *eventuality clause) and $\Diamond l$ (a* ground eventuality *clause), where l is a propositional literal and $L(x)$ is a unary non-ground literal with the variable x as its only argument.*

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

The transformation to DSNF is based on a renaming and unwinding technique which substitutes non-atomic subformulae by atomic formulae with new predicate symbols and replaces temporal operators by their fixed point definitions as described, for example, in [8].

Theorem 1 (see [4], Theorem 3.4). *Any monodic formula in first-order temporal logic can be transformed into an equi-satisfiable monodic temporal problem in DSNF with at most a linear increase in the size of the problem.*

The main purpose of the divided separated normal form is to cleanly separate different temporal aspects of a FOTL formula from each other. For the resolution calculus in this paper we will need to go one step further by transforming the universal and initial part of a monodic temporal problem into clause normal form.

Definition 2. *Let* $\mathsf{P} = \langle \mathcal{U}, \mathcal{I}, \mathcal{S}, \mathcal{E} \rangle$ *be a monodic temporal problem. With every eventuality* $\Diamond L(x) \in \mathcal{E}$ *and constant* c *occurring in* P *we uniquely associate a propositional symbol* p_c^L*. Then the* classification $\mathrm{Cls}(\mathsf{P})$ of P *is a quadruple* $\langle \mathcal{U}', \mathcal{I}', \mathcal{S}', \mathcal{E}' \rangle$ *such that* $\mathcal{U}'$ *is a set of clauses*[1]*, called* universal clauses*, consisting of the clausification of* $\mathcal{U}$ *and clauses* $\neg p_c^L \vee L(c)$ *for every* $\Diamond L(x) \in \mathcal{E}$ *and constant* c *occurring in* P*;* $\mathcal{I}'$ *is a set of clauses, called* initial clauses*, obtained by clausification of* $\mathcal{I}$*;* $\mathcal{S}'$ *is the smallest set of step clauses such that all step clauses from* $\mathcal{S}$ *are in* $\mathcal{S}'$ *and for every non-ground step clause* $P(x) \Rightarrow \bigcirc L(x)$ *in* $\mathcal{S}$ *and every constant* c *occurring in* P*, the clause* $P(c) \Rightarrow \bigcirc L(c)$ *is in* $\mathcal{S}'$*;* $\mathcal{E}'$ *is the smallest set of eventuality clauses such that all eventuality clauses from* $\mathcal{E}$ *are in* $\mathcal{E}'$ *and for every non-ground eventuality clause* $\Diamond L(x)$ *in* $\mathcal{E}$ *and every constant* c *occurring in* P*, the eventuality clause* $\Diamond p_c^L$ *is in* $\mathcal{E}'$*.*

One has to note that new constants and, especially, function symbols of an arbitrary arity can be introduced during the Skolemization process. As a consequence it is not possible in general to instantiate every variable that occurs in the original problem with all the constants and function symbols. On the other hand, the variables occurring in the step and eventuality clauses have to be instantiated with the constants that are present in the *original* problem (before Skolemization) in order to ensure the completeness of the calculus presented in Section 3.

Note further that more general step clauses with more than one atom on the left-hand side and more than one literal on the right-hand side can be derived by the calculus introduced in Section 3. In what follows $\mathcal{U}$ denotes the (current) universal part of a monodic temporal problem P.

3 Ordered Fine-Grained Resolution with Selection

We assume that we are given an *atom ordering* $\succ$, that is, a strict partial ordering on ground atoms which is well-founded and total, and a *selection function* S

[1] Clauses, as well as disjunctions and conjunctions, will be considered as multisets.

which maps any first-order clause C to a (possibly empty) subset of its negative literals and which is instance compatible:

Definition 3. *We say that a selection function S is* instance compatible *if and only if for every clause C, for every substitution σ and for every literal $l \in C\sigma$ it holds that $\in S(C\sigma)$ iff there exists a literal $l' \in S(C)$ such that $l'\sigma = l$.*

The atom ordering $\succ$ is extended to ground literals by $\neg A \succ A$ and $(\neg)A \succ (\neg)B$ if and only if $A \succ B$. The ordering is extended on the non-ground level as follows: for two arbitrary literals L and L', $L \succ L'$ if and only if $L\sigma \succ L'\sigma$ for every grounding substitution σ. A literal L is called (strictly) maximal w.r.t. a clause C if and only if there is no literal $L' \in C$ with $L' \succ L$ ($L' \succeq L$). A literal L is *eligible* in a clause $L \vee C$ for a substitution σ if either it is selected in $L \vee C$, or otherwise no literal is selected in C and $L\sigma$ is maximal w.r.t. $C\sigma$.

The atom ordering $\succ$ and the selection function S are used to restrict the applicability of the deduction rules of fine-grained resolution as follows.

(1) *First-order ordered resolution with selection between two universal clauses*

$$\frac{C_1 \vee A \quad \neg B \vee C_2}{(C_1 \vee C_2)\sigma}$$

if σ is a most general unifier of the atoms A and B, A *is eligible in* $(C_1 \vee A)$ *for σ, and $\neg B$ is eligible in* $(\neg B \vee C_2)$ *for σ*. The result is a universal clause.

(2) *First-order ordered positive factoring with selection*

$$\frac{C_1 \vee A \vee B}{(C_1 \vee A)\sigma}$$

if σ is a most general unifier of the atoms A and B, and A *is eligible in* $(C_1 \vee A \vee B)$ *for σ*. The result is again a universal clause.

(3) *First-order ordered resolution with selection between an initial and a universal clause, between two initial clauses, and ordered positive factoring with selection on an initial clause.* These are defined in analogy to the two deduction rules above with the only difference that the result is an initial clause.

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

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

where $C_1 \Rightarrow \bigcirc(D_1 \vee A)$ and $C_2 \Rightarrow \bigcirc(D_2 \vee \neg B)$ are step clauses, σ is a most general unifier of the atoms A and B such that σ does not map variables from C_1 or C_2 into a constant or a functional term, A *is eligible in* $(D_1 \vee A)$ *for σ, and $\neg B$ is eligible in* $(D_2 \vee \neg B)$ *for σ.*

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

where $C_1 \Rightarrow \bigcirc(D_1 \vee A)$ is a step clause, $D_2 \vee \neg B$ is a universal clause, and σ is a most general unifier of the atoms A and B such that σ does not map variables from C_1 into a constant or a functional term, A *is eligible in* $(D_1 \vee A)$ *for σ, and $\neg B$ is eligible in* $(D_2 \vee \neg B)$ *for σ*. There also exists an analogous rule where the positive literal A is contained in a universal clause and the negative literal $\neg B$ in a step clause.

(5) *Ordered fine-grained positive step factoring with selection.*

$$\frac{C \Rightarrow \bigcirc(D \vee A \vee B)}{C\sigma \Rightarrow \bigcirc(D \vee A)\sigma}$$

where σ is a most general unifier of the atoms A and B such that σ does not map variables from C into a constant or a functional term, and A *is eligible in* $(D \vee A \vee B)$ *for* σ.

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

Step clauses of the form $C \Rightarrow \bigcirc\bot$ will also be called *terminating* or *final* step clauses.

(7) *Duplicate literal elimination in left-hand sides of terminating step clauses.* A clause of the form $(C \wedge A \wedge A) \Rightarrow \bigcirc\bot$ yields the clause $(C \wedge A) \Rightarrow \bigcirc\bot$.

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

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

where $\forall x(\mathcal{A}_i(x) \Rightarrow \bigcirc\mathcal{B}_i(x))$ are formulae computed from the set of step clauses such that for every i, $1 \leq i \leq n$, the *loop* side conditions $\forall x(\mathcal{U} \wedge \mathcal{B}_i(x) \Rightarrow \neg L(x))$ and $\forall x(\mathcal{U} \wedge \mathcal{B}_i(x) \Rightarrow \bigvee_{j=1}^{n}(\mathcal{A}_j(x)))$ are valid.[2]

The set of full merged step clauses, satisfying the loop side conditions, is called a *loop in* $\Diamond L(x)$ and the formula $\bigvee_{j=1}^{n} \mathcal{A}_j(x)$ is called a *loop formula.* More details can be found in [10].

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

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

where $\mathcal{A}_i \Rightarrow \bigcirc\mathcal{B}_i$ are ground formulae computed from the set of step clauses such that for every i, $1 \leq i \leq n$, the *loop* side conditions $\mathcal{U} \wedge \mathcal{B}_i \models \neg l$ and $\mathcal{U} \wedge \mathcal{B}_i \models \bigvee_{j=1}^{n} \mathcal{A}_j$ are valid. The notions of *ground loop* and *ground loop formula* are defined similarly to the case above.

Rules (1) to (7), also called rules of *fine-grained step resolution*, are either identical or closely related to the deduction rules of ordered first-order resolution with selection; a fact that we exploit in our implementation of the calculus. The condition in rules (4) and (5) that a unifier σ may not map variables from the antecedent into a constant or a functional term is a consequence of the expanding domain assumption. Without this restriction, the calculus would be unsound [12, Example 5].

Loop formulae, which are required for applications of the rules (8) and (9), can be computed by the fine-grained breadth-first search algorithm (FG-BFS), depicted in Fig. 2. In this algorithm, $\mathrm{LT}(\mathcal{S})$ is the minimal set of clauses containing $\mathcal{S}$ such that for every non-ground step clause $(P(x) \Rightarrow \bigcirc M(x)) \in \mathcal{S}$, the set $\mathrm{LT}(\mathcal{S})$ contains the clause $P(c^l) \Rightarrow \bigcirc M(c^l)$ (c^l is a constant used only for loop search). The process of running the FG-BFS algorithm is called *loop search.* A variant of the FG-BFS algorithm for handling ground eventualities also exists.

[2] In the case $\mathcal{U} \models \forall x \neg L(x)$, the *degenerate clause*, $\top \Rightarrow \bigcirc\top$, can be considered as a premise of this rule; the conclusion of the rule is then $\neg\top$.

Function FG-BFS

Input: A set of universal clauses $\mathcal{U}$ and a set of step clauses $\mathcal{S}$, saturated by ordered fine-grained resolution with selection, and an eventuality clause $\Diamond L(x) \in \mathcal{E}$.

Output: A loop formula $H(x)$ with at most one free variable.

Method: (1) Let $H_0(x) = \mathbf{true}$; $\mathcal{M}_0 = \emptyset$; $i = 0$

(2) Let $\mathcal{N}_{i+1} = \mathcal{U} \cup \mathrm{LT}(\mathcal{S}) \cup \{\mathbf{true} \Rightarrow \bigcirc(\neg H_i(c^l) \vee L(c^l))\}$. Apply the rules of ordered fine-grained resolution with selection *except the clause conversion rule* to $\mathcal{N}_{i+1}$. If we obtain a contradiction, then return the loop **true** (in this case $\forall x \neg L(x)$ is implied by the universal part).

Otherwise let $\mathcal{M}_{i+1} = \{C_j \Rightarrow \bigcirc\bot\}_{j=1}^{n}$ be the set of all new *terminating* step clauses in the saturation of $\mathcal{N}_{i+1}$.

(3) If $\mathcal{M}_{i+1} = \emptyset$, return **false**; else let $H_{i+1}(x) = \bigvee_{j=1}^{n}(\tilde{\exists} C_j)\{c^l \rightarrow x\}$

(4) If $\forall x(H_i(x) \Rightarrow H_{i+1}(x))$, return $H_{i+1}(x)$.

(5) i = i + 1; goto 2.

Note: The constant c^l is a fresh constant used for loop search only

Fig. 2. Breadth-First Search Algorithm Using Fine-Grained Step Resolution.

Let *ordered fine-grained resolution with selection* be the calculus consisting of the rules (1) to (7) above, together with the ground and non-ground eventuality resolution rules described above, i.e. rules (8) and (9). We denote this calculus by $\mathfrak{I}_{FG}^{S,\succ}$.

Definition 4 (Derivation). *A (linear)* derivation Δ *(in $\mathfrak{I}_{FG}^{S,\succ}$) from the clausification* $\mathrm{Cls}(P) = \langle \mathcal{U}_1, \mathcal{I}_1, \mathcal{S}_1, \mathcal{E} \rangle$ *of a monodic temporal problem P is a sequence of tuples $\Delta = \langle \mathcal{U}_1, \mathcal{I}_1, \mathcal{S}_1, \mathcal{E} \rangle, \langle \mathcal{U}_2, \mathcal{I}_2, \mathcal{S}_2, \mathcal{E} \rangle, \ldots$ such that each tuple at an index $i+1$ is obtained from the tuple at the index i by adding the conclusion of an application of one of the inference rules of $\mathfrak{I}_{FG}^{S,\succ}$ to premises from one of the sets $\mathcal{U}_i$, $\mathcal{I}_i$, $\mathcal{S}_i$ to that set, with the other sets as well as $\mathcal{E}$ remaining unchanged*[3].

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

A derivation Δ is fair *if and only if for each clause C which can be derived from premises in $\langle \bigcup_{i \geq 1} \mathcal{U}_i, \bigcup_{i \geq 1} \mathcal{I}_i, \bigcup_{i \geq 1} \mathcal{S}_i, \mathcal{E} \rangle$ there exists an index j such that C occurs in $\langle \mathcal{U}_j, \mathcal{I}_j, \mathcal{S}_j, \mathcal{E} \rangle$.*

Ordered fine-grained resolution with selection is sound and complete for constant flooded monodic temporal problems over expanding domains as stated in the following theorem.

Theorem 2 (see [10], Theorem 5). *Let P be a monodic temporal problem. Let $\succ$ be an atom ordering and S an instance compatible selection function. Then P is unsatisfiable iff there exists a $\mathfrak{I}_{FG}^{S,\succ}$-refutation of* $\mathrm{Cls}(P)$*. Moreover, P is unsatisfiable iff any fair $\mathfrak{I}_{FG}^{S,\succ}$-derivation is a refutation of* $\mathrm{Cls}(P)$*.*

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

To prove Theorem 2, we show that any refutation of a temporal problem by the $\mathfrak{I}_e$ calculus of [12] there is a corresponding refutation by $\mathfrak{I}_{FG}^{S,\succ}$. Rule (7), which allows the elimination of duplicate literals in the left-hand sides of step clauses, is required to establish this correspondence.

4 Adding Redundancy Elimination

Given that our calculus uses an ordering refinement, it seems natural to establish that the calculus admits redundancy elimination by using the approach in [1, Section 4.2]. To do so, we would first need to define a model functor I that maps any (not necessarily satisfiable) temporal problem P not containing the empty clause to an interpretation $\mathfrak{M}_{\mathsf{P}}$ and then show that $\mathfrak{I}_{FG}^{S,\succ}$ has the reduction property for counterexamples with respect to the model functor I and the ordering $\succ$, that is, for every temporal problem P and minimal clause $\mathcal{C}$ in P which is false in $\mathfrak{M}_{\mathsf{P}}$, there exists an inference with (main) premise $\mathcal{C}$ and conclusion $\mathcal{D}$ that is also false in P but smaller than $\mathcal{C}$ wrt. $\succ$. We could then define a clause $\mathcal{C}$ to be redundant wrt. P if there exists clauses $\mathcal{C}_1, \ldots, \mathcal{C}_k$ in P such that $\mathcal{C}_1, \ldots, \mathcal{C}_k \models \mathcal{C}$ and $\mathcal{C} \succ \mathcal{C}_i$ for all i, $1 \leq i \leq k$, and it would be straightforward to show that $\mathfrak{I}_{FG}^{S,\succ}$ remains complete if redundant clauses are eliminated from derivations.

However, due to the presence of eventualities in temporal problems, defining an appropriate model functor is a non-trivial and open problem. For example, consider the satisfiable propositional temporal problem $\mathsf{P} = \langle \{p \vee q\}, \emptyset, \{p \Rightarrow \bigcirc \neg l\}, \{\Diamond l\} \rangle$ and an ordering $\succ$ such that $p \succ q$. Applying the standard model functor defined in [1] to the clause $p \vee q$ results in a model in which p is true. Given that $p \vee q$ is a universal clause, it would be natural to define $\mathfrak{M}_{\mathsf{P}}$ in such way that p is true at every moment of time. However, due to the step clause $p \Rightarrow \bigcirc \neg l$, $\Box \Diamond l$ is not true in $\mathfrak{M}_{\mathsf{P}}$ which means that $\mathfrak{M}_{\mathsf{P}}$ is not a model of P. Thus, this simplistic approach to defining a model functor is not correct for temporal problems containing eventualities.

We have recently introduced a model functor I for propositional temporal problems [16], which is able to associate a model $\mathfrak{M}$ of P with every satisfiable temporal problem P, but $\mathfrak{I}_{FG}^{S,\succ}$ is not reductive wrt. I. Thus, this model functor is not suitable for establishing that $\mathfrak{I}_{FG}^{S,\succ}$ admits redundancy elimination.

Thus, we have to follow a different approach in order to show how ordered fine-grained resolution with selection can be extended with redundancy elimination rules. In the following we will define the notions of a tautological clause and of a subsumed clause. In order to show that $\mathfrak{I}_{FG}^{S,\succ}$ is still complete if such clauses are eliminated during a derivation, we need to show that for every refutation without redundancy elimination there exists a refutation with redundancy elimination. It turns out that in order to be able to do so, we need to add two inference rules to our calculus and impose a restriction on the selection function.

First of all, we consider tautological clauses. As a tautological clause is defined to be a clause that is true in every structure $\mathfrak{M} = (D_n, I_n)_{n \in \mathbb{N}}$, we obtain the following lemma:

Lemma 1. *Let $\mathcal{C}$ be a initial, universal or step clause. Then:*

(i) If $\mathcal{C}$ is an initial or universal clause, then $\mathcal{C}$ is a tautology iff $\mathcal{C} = \neg L \vee L \vee C'$, for some possibly empty disjunction of literals C'.

(ii) If $\mathcal{C} = C_1 \Rightarrow \bigcirc C_2$ is a step clause, then $\mathcal{C}$ is a tautology iff $C_2 = \neg L \vee L \vee C_2'$, for some possibly empty disjunction of literals C_2'.

It has be noted that for point (*ii*) of the lemma above C_1 is assumed to be **true** or a non-empty conjunction of atoms.

Thus, just as in the non-temporal first-order case, there is again a syntactic criterion for characterising tautologies, namely the presence of complementary literals. For a set of clauses $\mathcal{N}$ (or a temporal problem) we denote by $\mathrm{taut}(\mathcal{N})$ the set of all the tautological clauses contained in the set $\mathcal{N}$.

The subsumption relation on initial, universal and step clauses is now defined as follows.

Definition 5. *We define a subsumption relation $\leq_s$ on initial, universal and step clauses as follows:*

(i) For two initial clauses $\mathcal{C}$ and $\mathcal{D}$, two universal clauses $\mathcal{C}$ and $\mathcal{D}$, or a universal clause $\mathcal{C}$ and an initial clause $\mathcal{D}$ we define

$$\mathcal{C} \leq_s \mathcal{D} \text{ iff there exists a substitution } \sigma \text{ with } \mathcal{C}\sigma \subseteq \mathcal{D}.$$

(ii) For two step clauses $\mathcal{C} = C_1 \Rightarrow \bigcirc C_2$ and $\mathcal{D} = D_1 \Rightarrow \bigcirc D_2$ we define

$$\mathcal{C} \leq_s \mathcal{D} \text{ iff there exists a substitution } \sigma \text{ with } C_1\sigma \subseteq D_1, C_2\sigma \subseteq D_2 \text{ and for every } x \in var(C_1) \cap var(C_2)\colon \sigma(x) \in X.$$

(iii) For a universal clause $\mathcal{C}$ and a step clause $\mathcal{D} = D_1 \Rightarrow \bigcirc D_2$ we define

$$\mathcal{C} \leq_s \mathcal{D} \text{ iff there exists a substitution } \sigma \text{ with } \mathcal{C}\sigma \subseteq \neg D_1 \text{ or } \mathcal{C}\sigma \subseteq D_2.$$

By $\mathcal{N} \leq_s \mathcal{N}'$ we denote that all clauses in $\mathcal{N}'$ are subsumed by clauses in $\mathcal{N}$.

Thus, subsumption between two initial, two universal or an initial and a universal clause is defined analogously to the subsumption on regular first-order clauses. However, we can only allow a universal clause to subsume a initial clause, but not conversely, as an initial clause only holds in the initial moment of time while a universal clause is true at every moment of time. We also allow subsumption between a universal and a step clause if and only if the universal either subsumes the negated left-hand side or the right-hand side of the step clause.

For subsumption between two step clauses $C_1 \Rightarrow \bigcirc C_2$ and $D_1 \Rightarrow \bigcirc D_2$, we have to impose an additional constraint on the substitution that is used for multiset inclusion: in analogy to inference rules (4) and (5), it has to be ensured that variables occurring in the left-hand sides C_1 and C_2 are only mapped to variables. While for the two inference rules this restriction is imposed to ensure soundness, here the motivation is completeness.

To see that, consider a temporal problem P with universal clauses $P(x)$ and $\neg Q(c)$ and a step clause $P(x) \Rightarrow \bigcirc Q(x)$. The clausification of P will then also contain a step clause $P(c) \Rightarrow \bigcirc Q(c)$. This additional step clause can be resolved with $\neg Q(c)$ using rule (4) with the identity substitution as unifier to obtain $P(c) \Rightarrow \bigcirc\bot$ which, using the conversion rules, gives us a new universal clause $\neg P(c)$. Another inference step with $P(x)$ results in a contradiction. Now, without a restriction on the substitution that can be used in subsumption, $P(x) \Rightarrow \bigcirc Q(x)$ would subsume $P(c) \Rightarrow \bigcirc Q(c)$. We could then try to derive a contradiction by resolving $P(x) \Rightarrow \bigcirc Q(x)$ with $\neg Q(c)$. However, the unifier of $Q(x)$ and $Q(c)$ maps the variable x, which also occurs on the left-hand side of the step clause to the constant c. Thus, an inference by rule (4) using these two premises is not possible and a contradiction can no longer be derived.

Definition 6. *Let $\mathcal{C}$ and $\mathcal{D}$ be initial, step or universal clauses. Then we say that $\mathcal{C}$* properly *subsumes $\mathcal{D}$, written $\mathcal{C} <_s \mathcal{D}$, if and only if $\mathcal{C}$ subsumes $\mathcal{D}$ but not vice-versa, i.e. $\mathcal{C} <_s \mathcal{D}$ iff $\mathcal{C} \leq_s \mathcal{D}$ and $\mathcal{D} \not\leq_s \mathcal{C}$.*

Lemma 2. *Let $\mathcal{C}$ and $\mathcal{D}$ be initial, step or universal clauses such that $\mathcal{C} \leq_s \mathcal{D}$. Then it holds for an initial clause $\mathcal{D}$ that the formula $[(\Box)\tilde{\forall}\mathcal{C}] \Rightarrow [\tilde{\forall}\mathcal{D}]$ is valid, and for a step or universal clause $\mathcal{D}$ that the formula $[\Box\tilde{\forall}\mathcal{C}] \Rightarrow [\Box\tilde{\forall}\mathcal{D}]$ is valid, where $\tilde{\forall}\mathcal{C}$ denotes the universal closure of $\mathcal{C}$.*

Having defined criteria for identifying tautological and subsumed clauses, we could now try to prove that for every refutation without redundancy elimination there exists a refutation with redundancy elimination. However, it turns out that such a correspondence is difficult to establish if the refutation contains applications of the duplicate literal elimination rule whose premise is subsumed.

For example, consider the step clause $\mathcal{D}_1 = P(x) \wedge P(x) \Rightarrow \bigcirc\bot$ which is subsumed by $\mathcal{C}_1 = P(x) \wedge P(y) \Rightarrow \bigcirc\bot$. From $\mathcal{D}_1$ we can derive $\mathcal{D}_2 = P(x) \Rightarrow \bigcirc\bot$ using the duplicate literal elimination rule. But our calculus does not contain a rule which allows us to derive a clause $\mathcal{C}_2$ from $\mathcal{C}_1$ that subsumes $\mathcal{D}_2$ nor does $\mathcal{C}_1$ itself subsume $\mathcal{D}_2$. Similarly, the universal clause $\mathcal{C}_3 = \neg P(x) \vee \neg P(y)$ would also subsume $\mathcal{C}_1$. But again, $\mathcal{C}_3$ does not subsume $\mathcal{D}_2$ nor can we derive a clause from $\mathcal{C}_3$ which subsumes $\mathcal{D}_2$ using the rules of our calculus.

In order to deal with these two cases we need additional factoring rules, in particular, we need to extend our calculus by the following two rules:

- (Arbitrary) Factoring in left-hand sides of terminating step clauses:

$$\frac{C \wedge A \wedge B \Rightarrow \bigcirc\bot}{(C \wedge A)\sigma \Rightarrow \bigcirc\bot} \ ,$$

  where σ is a most general unifier of the atoms A and B.
- (Arbitrary) Factoring in (at most) monadic negative universal clauses:

$$\frac{\neg A_1 \vee \cdots \vee \neg A_n \vee \neg A_{n+1}}{(\neg A_1 \vee \cdots \vee \neg A_n)\sigma} \ ,$$

  where every atom $A_1, \ldots, A_{n+1}$ contains at most one free variable and σ is a most general unifier of the atoms A_n and A_{n+1}.

The calculus ordered fine-grained resolution with selection extended by the two rules introduced above will be called *subsumption complete* ordered fine-grained resolution with selection and will be denoted by $\mathfrak{I}^{S,\succ}_{FG,Sub}$.

Finally, for our completeness proof we also need to require that the selection function is *subsumption compatible*, as defined below.

Definition 7. *We say that a selection function S is* subsumption compatible *if and only if for every substitution σ and for every two clauses C, D with $C\sigma \subseteq D$ it holds for every literal $l \in D$ that $l \in S(D)$ iff $l\sigma \in S(C)$*

We now have everything in place to show that subsumption complete ordered fine-grained resolution with selection allows the elimination of tautological and subsumed clauses.

Lemma 3. *Let $\mathcal{C}_1$, $\mathcal{C}_2$ be initial, universal or step clauses such that $\mathcal{C}_1$ is a tautology. Then it holds that every resolvent $\mathcal{C}$ of $\mathcal{C}_1$ and $\mathcal{C}_2$ is either a tautology or subsumed by $\mathcal{C}_2$.*

Lemma 4. *Let $\mathcal{C}_1$ be a tautology. Then it holds that every factor $\mathcal{C}$ of $\mathcal{C}_1$ is a tautology.*

Lemma 5. *Let $\mathcal{U}$ be a set of universal clauses and let $\mathcal{N}, \tilde{\mathcal{N}}$ be sets of step clauses such that $\mathcal{N} \leq_s \tilde{\mathcal{N}}$. Additionally, let $\tilde{\Delta}$ be a derivation of a step clause $\tilde{\mathcal{C}}$ by subsumption complete ordered fine-grained resolution with selection without the clause conversion rule from clauses in $\mathcal{U} \cup \tilde{\mathcal{N}}$.*

Then there exists a derivation Δ of a step clause $\mathcal{C}$ by subsumption complete ordered fine-grained resolution with selection from clauses in $\mathcal{U} \cup \mathcal{N}$ such that $\mathcal{C} \leq_s \tilde{\mathcal{C}}$.

Proof. Lemma 5 is shown by induction on the length of the derivation $\tilde{\Delta}$. For the base case we assume that $\tilde{\mathcal{C}}$ is a step clause in $\tilde{\mathcal{N}}$. Then there is a clause $\mathcal{C}$ in $\mathcal{N}$ with $\mathcal{C} \leq_s \tilde{\mathcal{C}}$. For the induction step we consider a step clause $\tilde{\mathcal{C}}$ that is derived by one of the rules of fine-grained step resolution excluding the clause conversion rule, that is, rules (1) to (5) and (7), from premises $\tilde{\mathcal{C}_1}$ and $\tilde{\mathcal{C}_2}$ which are either elements of $\mathcal{U} \cup \tilde{\mathcal{N}}$ or previously derived clauses. By the induction hypothesis there are clauses $\mathcal{C}_1$ and $\mathcal{C}_2$ with $\mathcal{C}_1 \leq_s \tilde{\mathcal{C}_1}$ and $\mathcal{C}_2 \leq_s \tilde{\mathcal{C}_2}$ which are either elements of $\mathcal{U} \cup \mathcal{N}$ or previously derived, and either $\mathcal{C}_1 \leq_s \tilde{\mathcal{C}}$, $\mathcal{C}_2 \leq_s \tilde{\mathcal{C}}$ or we can derive a clause $\mathcal{C}$ with $\mathcal{C} \leq_s \tilde{\mathcal{C}}$ from $\mathcal{C}_1$ and $\mathcal{C}_2$.

Lemma 6. *Let $\mathcal{N}$ and $\tilde{\mathcal{N}}$ be sets of initial, universal clauses or step clauses such that $\mathcal{N} \leq_s \tilde{\mathcal{N}}$. Additionally, let $\tilde{\Delta}$ be a derivation of a clause $\tilde{\mathcal{C}}$ by subsumption complete ordered fine-grained resolution with selection from clauses in $\tilde{\mathcal{N}}$.*

Then there exists a derivation Δ of a clause $\mathcal{C}$ by subsumption complete ordered fine-grained resolution with selection from clauses in $\mathcal{N}$ such that $\mathcal{C} \leq_s \tilde{\mathcal{C}}$.

The previous statement still holds if $\mathcal{N} \leq_s \tilde{\mathcal{N}} \setminus \mathrm{taut}(\tilde{\mathcal{N}})$ and $\tilde{\mathcal{C}}$ is not a tautology.

Function Subsumption-Restricted-FG-BFS

Input: A set of universal clauses $\mathcal{U}$ and a set of step clauses $\mathcal{S}$, saturated by ordered fine-grained resolution with selection, and an eventuality clause $\Diamond L(x) \in \mathcal{E}$.

Output: A formula $R(x)$ with at most one free variable.

Method: (1) Let $R_0(x) = \mathbf{true}$; $M_0 = \emptyset$; $i = 0$

(2) Let $\mathcal{N}'_{i+1} = \mathcal{U} \cup \mathrm{LT}(\mathcal{S}) \cup \{\mathbf{true} \Rightarrow \bigcirc(\neg R_i(c^l) \vee L(c^l))\}$. Apply the rules of ordered fine-grained resolution with selection *except the clause conversion rule* to $\mathcal{N}'_{i+1}$, together with the *removal of tautological and subsumed clauses*. If we obtain a contradiction, then return the loop **true** (in this case $\forall x \neg L(x)$ is implied by the universal part).
Otherwise let $\mathcal{M}'_{i+1} = \{C_j \Rightarrow \bigcirc\bot\}_{j=1}^{n}$ be the set of all new *terminating* step clauses in the saturation of $\mathcal{N}'_{i+1}$.

(3) If $\mathcal{M}'_{i+1} = \emptyset$, return **false**; else let $R_{i+1}(x) = \bigvee_{j=1}^{n}(\tilde{\exists} D_j)\{c^l \rightarrow x\}$

(4) If $\forall x(R_i(x) \Rightarrow R_{i+1}(x))$, return $R_{i+1}(x)$.

(5) i = i + 1; goto 2.

Fig. 3. Restricted Breadth-First Search Using Ordered Fine-Grained Step Resolution with Selection

Proof. Let $\tilde{\Delta} = \tilde{\mathcal{D}}_1, \ldots, \tilde{\mathcal{D}}_{n-1}, \tilde{\mathcal{C}}(= \tilde{\mathcal{D}}_n)$. If $\mathcal{N} \leq_s \tilde{\mathcal{N}}$, then one can show the existence of the derivation Δ by induction on the length of the derivation $\tilde{\Delta}$ in analogy to Lemma 5.

In the case where $\mathcal{N} \leq_s \tilde{\mathcal{N}} \setminus \mathrm{taut}(\tilde{\mathcal{N}})$ holds, it can be shown inductively for every clause $\tilde{\mathcal{D}}_i$ $(1 \leq i \leq n)$ which is not a tautology that there exists a derivation Δ of a clause $\mathcal{D}_i$ with $\mathcal{D}_i \leq_s \tilde{\mathcal{D}}_i$ by subsumption complete ordered fine-grained resolution with selection from clauses in $\mathcal{N}$.

Theorem 3. *Let* P *be the clausification of a monodic temporal problem. Let* $\succ$ *be an atom ordering and* S *a subsumption compatible selection function. Then* P *is unsatisfiable iff there exists a* $\mathfrak{I}^{S,\succ}_{FG,Sub}$*-refutation of* $\mathrm{Cls}(\mathsf{P})$*. Moreover,* P *is unsatisfiable iff any fair* $\mathfrak{I}^{S,\succ}_{FG,Sub}$*-derivation is a refutation of* $\mathrm{Cls}(\mathsf{P})$*.*

Proof. Given a $\mathfrak{I}^{S,\succ}_{FG}$-refutation Δ of P we establish by induction on Δ that there also exists a $\mathfrak{I}^{S,\succ}_{FG,Sub}$-refutation of P using Lemmata 3 to 6.

5 Subsumption and Loop Search

Theorem 3 shows that we can eliminate tautologies and subsumed clauses during the construction of a derivation at the level of inference rule applications of the $\mathfrak{I}^{S,\succ}_{FG,Sub}$ calculus. However, the rules of the calculus are also applied within the fine-grained breadth-first search algorithm FG-BFS which is used to find loop formulae for the application of the eventuality resolution rules. Naturally, the question arises whether tautological and subsumed clauses can also be eliminated within FG-BFS.

The answer to that is positive. Figure 3 shows the so-called *subsumption restricted* breadth-first search algorithm using ordered fine-grained step resolution with selection, a modification of FG-BFS which removes tautological and subsumed clauses during the saturation process by ordered fine-grained resolution with selection in step (2) of the algorithm. In the way in which the algorithm shown in Figure 3 is defined the constructed sets $\mathcal{M}'_i$ will not contain terminating step clauses $C \Rightarrow \bigcirc\bot$ and $D \Rightarrow \bigcirc\bot$ such that $C \leq_s D$.

We are then able to prove the following result:

Theorem 4. *Let P be a monodic temporal problem. Let $\succ$ be an atom ordering and S a subsumption compatible selection function. Then P is unsatisfiable iff there exists a $\mathfrak{I}^{S,\succ}_{FG,Sub}$-refutation of* $\mathrm{Cls}(\mathsf{P})$ *with applications of the eventuality resolution rule restricted to loop formulae found by the function Subsumption-Restricted-FG-BFS. Moreover, P is unsatisfiable iff any fair $\mathfrak{I}^{S,\succ}_{FG,Sub}$-derivation with applications of the eventuality resolution rule restricted to loop formulae found by the function Subsumption-Restricted-FG-BFS is a refutation of* $\mathrm{Cls}(\mathsf{P})$.

6 Implementation and Experimental Results

The subsumption complete ordered fine-grained resolution calculus including the restricted breath-first loop search procedure have been implemented in the theorem prover TSPASS [4], which is based on the first-order resolution prover SPASS 3.0. In order to be able to construct fair derivations, the loop search procedure has been integrated with the remainder of the calculus as described in [14, 15].

The main procedure of TSPASS uses a given-clause algorithm [18] in which a clause selected from the set $\mathcal{US}$ of usable clauses, which initially contains all clauses from a given temporal problem, is used to derive all consequences $\mathcal{NEW}$ by resolving the selected clause with all the clauses in the set $\mathcal{WO}$ of worked-off clauses, which is initially empty, after the selected clause has been moved from $\mathcal{US}$ to $\mathcal{WO}$. Based on the results in the previous two sections, we can apply tautology elimination (to $\mathcal{NEW}$), forward subsumption and backward subsumption [18]. With forward subsumption all clauses in $\mathcal{NEW}$ that are subsumed by a clause in $\mathcal{WO}$ or $\mathcal{US}$ are deleted from $\mathcal{NEW}$. With backward subsumption clauses in $\mathcal{WO}$ and $\mathcal{US}$ subsumed by a clause in $\mathcal{NEW}$ are deleted from these sets. After redundancy elimination the remaining clauses in $\mathcal{NEW}$ are added to $\mathcal{US}$. This process continues until either a contradiction is derived or $\mathcal{US}$ becomes empty.

To show the effectiveness of tautology elimination, forward subsumption, and backward subsumption for the subsumption complete ordered fine-grained resolution calculus, we have applied TSPASS 0.92-0.16, for example, on the specification of the game Cluedo [2]. Due to lack of space we cannot present other examples, but we obtained similar results. The Cluedo problem specifications

[4] Available at `http://www.csc.liv.ac.uk/~michel/software/tspass/`

	-F/-B/-T		-F/+B/-T		-F/-B/+T		-F/+B/+T	
	Clauses	Time	Clauses	Time	Clauses	Time	Clauses	Time
1	—	TO	—	TO	239202	328.938s	17664	0.193s
2	—	TO	—	TO	—	TO	694574	10.959s
3	—	TO	—	TO	—	TO	—	TO
4	—	TO	—	TO	—	TO	529244	12.566s
5	—	TO	—	TO	—	TO	—	TO
6	—	TO	—	TO	—	TO	—	TO
	+F/-B/-T		+F/+B/-T		+F/-B/+T		+F/+B/+T	
	Clauses	Time	Clauses	Time	Clauses	Time	Clauses	Time
1	481	0.035s	480	0.039s	445	0.033s	444	0.038s
2	2354	0.130s	2262	0.129s	1926	0.115s	1892	0.124s
3	11065	1.375s	9912	1.310s	10102	1.350s	9170	1.278s
4	1460	0.087s	1559	0.097s	1125	0.074s	1343	0.093s
5	594	0.051s	594	0.052s	488	0.044s	488	0.049s
6	765	0.059s	765	0.055s	645	0.050s	645	0.054s

Table 1. Results Obtained for the Cluedo Examples

consist of six valid, i.e. unsatisfiable, assertions that can be made in an example Cluedo game. The full details can be found in [5, 6]. Problem 4 is the only specification that contains eventuality formulae.

The experiments were run on a PC with an Intel Core 2 Duo E6400 CPU and 2 GB of main memory with a timeout (TO) of 1 CPU hour for each problem. The results are shown in Table 1. Here, '+B', '+F', and '+T' indicate that backward subsumption, forward subsumption, and tautology elimination, respectively, have been enabled while '-B', '-F', and '-T' indicate that they have been disabled. Given that all six assertions are valid, proofs can theoretically be found by a complete reasoner without the need for redundancy elimination. As the experiments indicate this is clearly not the case within a reasonable amount of time. On the other hand with all options for redundancy elimination enabled even the most difficult problem can be solved in little more than one second. As one might expect, forward subsumption is the most effective of the three options, followed by tautology elimination, while backward subsumption can on occasion slow down the process of finding a proof rather than speeding it up. Overall, the experiments confirm that redundancy elimination is crucial for effective resolution-based theorem proving in monodic first-order temporal logic.

7 Conclusion

In this paper we have considered redundancy elimination in the context of ordered fine-grained resolution with selection, a sound and complete resolution-based calculus for monodic first-order temporal logic. We have shown that a slight modification of the calculus is compatible with the elimination of tautologies and subsumed clauses.

Our results can be used to show that the calculus can also be extended with additional rules, for example, condensation and matching replacement resolution [18] with suitable restrictions on the substitutions and on the orderings of the literals that are to be removed, which reduce to a sequence of inference and redundancy elimination steps. Such rules can be useful to further increase the effectiveness of the calculus and for the construction of decision procedures for decidable fragments of monodic first-order temporal logic [10].

In addition, we have presented experimental results which confirm that the elimination of redundant clauses is essential for effective resolution-based theorem proving in monodic first-order temporal logic.

References

1. L. Bachmair and H. Ganzinger. Resolution theorem proving. In Robinson and Voronkov [17], chapter 2, pages 19–99.
2. Cluedo. http://www.hasbro.com.
3. A. Degtyarev, M. Fisher, and B. Konev. Monodic temporal resolution. In *Proc. CADE-19*, volume 2741 of *LNCS*, pages 397–411. Springer, 2003.
4. A. Degtyarev, M. Fisher, and B. Konev. Monodic temporal resolution. *ACM Transactions On Computational Logic*, 7(1):108–150, 2006.
5. C. Dixon. Specifying and verifying the game Cluedo using temporal logics of knowledge. Technical Report ULCS-04-003, University of Liverpool, 2004.
6. C. Dixon, M. Fisher, B. Konev, and A. Lisitsa. Practical first-order temporal reasoning. In *Proc. TIME'08*, pages 156–163. IEEE Comp. Soc., 2008.
7. E. A. Emerson. Temporal and modal logic. In *Handbook of Theoretical Computer Science, Volume B: Formal Models and Semantics*, pages 995–1072. Elsevier, 1990.
8. M. Fisher, C. Dixon, and M. Peim. Clausal temporal resolution. *ACM Transactions on Computational Logic*, 2(1):12–56, 2001.
9. I. Hodkinson, F. Wolter, and M. Zakharyaschev. Decidable fragments of first-order temporal logics. *Annals of Pure and Applied Logic*, 106:85–134, 2000.
10. U. Hustadt, B. Konev, and R. A. Schmidt. Deciding monodic fragments by temporal resolution. In *Proc. CADE-20*, volume 3632 of *LNCS*, pages 204–218. Springer, 2005.
11. B. Konev, A. Degtyarev, C. Dixon, M. Fisher, and U. Hustadt. Towards the implementation of first-order temporal resolution: the expanding domain case. In *Proc. TIME-ICTL 2003*, pages 72–82. IEEE Comp. Soc., 2003.
12. B. Konev, A. Degtyarev, C. Dixon, M. Fisher, and U. Hustadt. Mechanising first-order temporal resolution. *Information and Computation*, 199(1-2):55–86, 2005.
13. R. Kontchakov, C. Lutz, F. Wolter, and M. Zakharyaschev. Temporalising tableaux. *Studia Logica*, 76(1):91–134, 2004.
14. M. Ludwig and U. Hustadt. Implementing a fair monodic temporal logic prover. *AI Communications*. To appear.
15. M. Ludwig and U. Hustadt. Fair monodic temporal reasoning. In *Proc. CADE-22*, 2009. To appear.
16. M. Ludwig and U. Hustadt. Resolution-based model construction for PLTL. In *Proc. of TIME'09*, 2009. To appear.
17. J. A. Robinson and A. Voronkov, editors. *Handbook of Automated Reasoning (in 2 volumes)*. Elsevier and MIT Press, 2001.
18. C. Weidenbach. Combining superposition, sorts and splitting. In Robinson and Voronkov [17], pages 1965–2013.
19. F. Wolter and M. Zakharyaschev. Axiomatizing the monodic fragment of first-order temporal logic. *Annals of Pure and Applied Logic*, 118:133–145, 2002.