Towards the Implementation of First-Order Temporal Resolution: the Expanding Domain Case

Boris Konev[†], **Anatoli Degtyarev**[‡], **Clare Dixon**[†], **Michael Fisher**[†], **Ullrich Hustadt**[†]

† Department of Computer Science, University of Liverpool, UK
{konev,clare,michael,ullrich}@csc.liv.ac.uk

‡ Department of Computer Science, Kings College, London, UK
anatoli@dcs.kcl.ac.uk

Overview

- First-order temporal logic is very expressive but has no finite axiom system.
- The *monodic* fragment of FOTL has been shown to have completeness and sometimes even decidability properties (Hodkinson et al.).
- Tableau and resolution calculi have been defined for the monodic fragment (Kontchakov et al., Degtyarev et al.), however neither are very practical.
- Here we focus on expanding domains i.e. the domain which first-order terms can range over can increase at each temporal step.
- We present a fine grained calculus with steps amenable to implementation.

Plan

- First-Order Temporal Logic (FOTL)
- A normal form for FOTL
- Step Resolution
- Eventuality Resolution
- Example
- Conclusions

Motivation

The monodic fragment of first-order temporal logic is useful:

- for spatio-temporal reasoning;
- for reasoning about temporal entity relation models;
- as a unifying framework for:
  - temporal logics of knowledge/belief;
  - spatio-temporal logics;
  - temporal description logics.

First-Order Temporal Logic

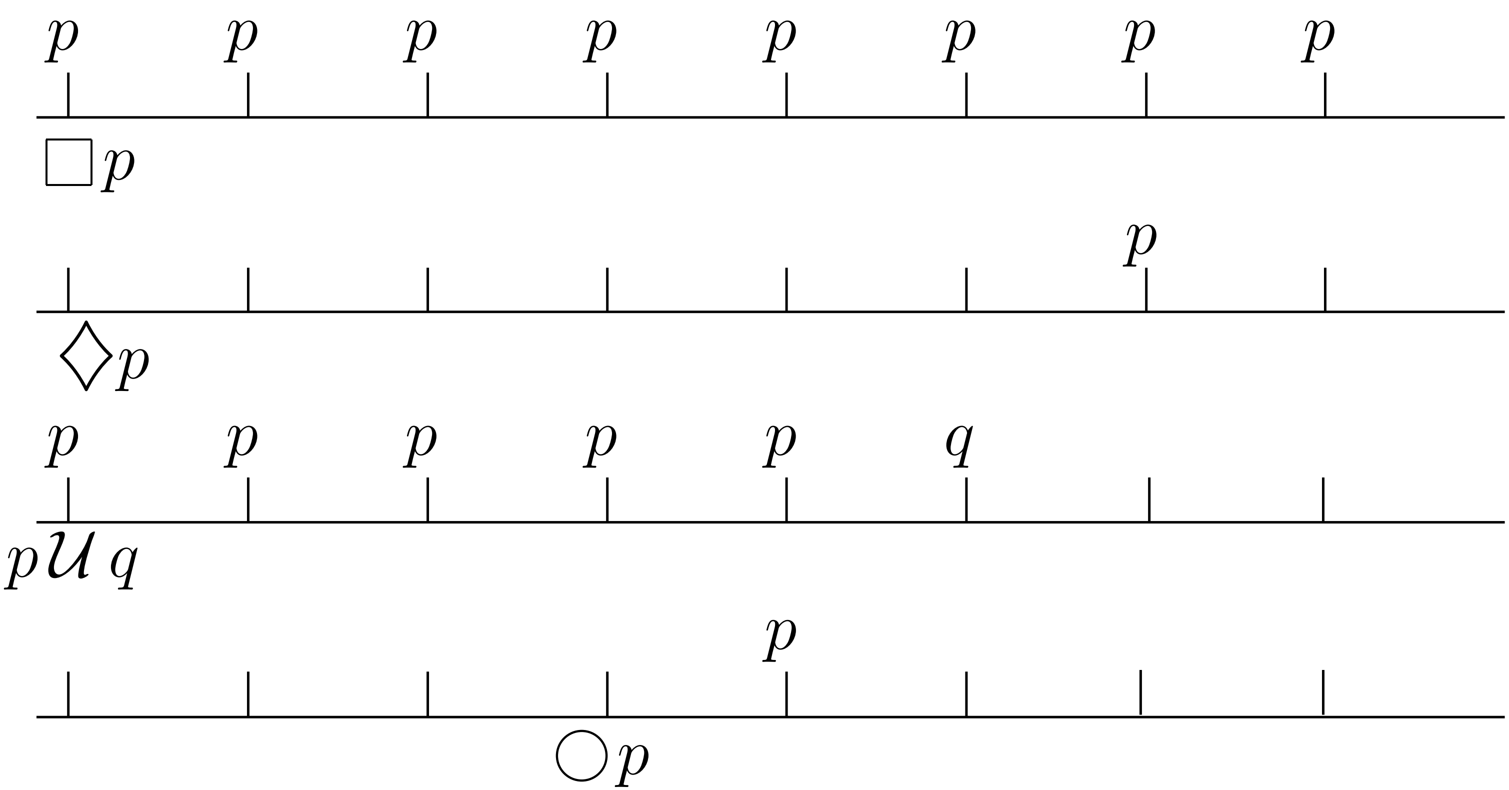
- First-order language without functional symbols extended with temporal connectives: $\bigcirc$, $\Diamond$, $\Box$, $\mathcal{U}$, $\mathcal{W}$.
- Formulae are interpreted in a sequence of first-order models, $\mathfrak{M}_n = \langle D_n, I_n \rangle$, where $D_n \subseteq D_m$ (expanding domain) for $n < m$ and I_n is an interpretation of constant and predicate symbols over D_n.
- The interpretation of constants and variables is rigid.

Definition 1 *An* FOTL-*formula* ϕ *is called* monodic *if any subformulae of the form* $\mathcal{T}\psi$*, where* $\mathcal{T}$ *is* $\bigcirc$*,* $\Box$*, or* $\Diamond$ *(or* $\psi_1 \mathcal{T} \psi_2$*, where* $\mathcal{T}$ *is* $\mathcal{U}$ *or* $\mathcal{W}$*), contains at most one free variable.*

The set of valid *monodic* formulae is finitely axiomatisable.

Semantics of Temporal Operators

Usual semantics for the temporal operators.



where

$$p \mathcal{W} q \equiv p \mathcal{U} q \vee \Box p$$

Overview of the Resolution Method

The resolution method consists of three main steps:-

- translation to normal form;
- classical style resolution between clauses involving the next-time operator, or between purely classical logic clauses or between these two sets;
- eventuality resolution between eventuality clauses ($\Diamond L(x)$) and complex combinations of clauses involving the next-time operator $\mathcal{A}_i(x) \Rightarrow \bigcirc \mathcal{B}_i(x)$ which have the effect $\forall x(\bigvee_i \mathcal{A}_i(x) \Rightarrow \bigcirc \Box \neg L(x))$ (and ground versions of these).

For simplicity, in the talk, we consider the case where there are no constants in the original problem.

Translation to Normal Form

- We translate formulae into a satisfiability preserving normal form known as *Divided Separated Normal Form*.
- A *temporal step clause* is a formula either of the form $p \Rightarrow \bigcirc l$, where p is a proposition and l is a propositional literal, or $(P(x) \Rightarrow \bigcirc M(x))$, where $P(x)$ is a unary predicate and $M(x)$ is a unary literal.
- 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.

Divided Separated Normal Form

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

- the universal part, $\mathcal{U}$, is a set of first-order formulae;
- the initial part, $\mathcal{I}$, is a set of first-order formulae;
- the step part, $\mathcal{S}$, is a set of original (ground and non-ground) temporal step clauses; and
- the eventuality part, $\mathcal{E}$, is a set of eventuality clauses 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.

$\mathcal{U}$, $\mathcal{I}$, $\mathcal{S}$, and $\mathcal{S}$ are finite. With each monodic temporal problem, we associate $\mathcal{I} \wedge \Box \mathcal{U} \wedge \Box \forall x \mathcal{S} \wedge \Box \forall x \mathcal{E}$.

The Original Calculus (Step)

The original calculus required complex combinations of the the ground and non-ground step clauses to apply such rules as the following.

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

  where $\mathcal{U} \cup \{\mathcal{B}\} \vdash \perp$.

Here $\mathcal{A} \Rightarrow \bigcirc \mathcal{B}$ denote complex combinations of step clauses (see the paper for their definition).

A Fine-Grained Calculus

- The idea of the new (fine-grained) calculus is to define small resolution steps, more like classical binary resolution, which are easier to implement.
- Several of the smaller steps may be required to mimic the original (complex) step resolution rule.
- Thus we may generate additional step clauses of the form

$$C \Rightarrow \bigcirc D.$$

  Here, C is a *conjunction* of propositions and unary predicates of the form $P(x)$ and D is a *disjunction* of literals.
- First, rewrite the universal and initial parts into clausal form.

Deduction rules

1. *Fine-grained (restricted) step resolution*

$$\frac{C_1 \Rightarrow \bigcirc(D_1 \vee L) \quad C_2 \Rightarrow \bigcirc(D_2 \vee \neg M)}{(C_1 \wedge C_2)\sigma \Rightarrow \bigcirc(D_1 \vee D_2)\sigma} \; ,$$

$$\frac{C_1 \Rightarrow \bigcirc(D_1 \vee L) \quad D_2 \vee \neg M}{C_1\sigma \Rightarrow \bigcirc(D_1 \vee D_2)\sigma} \; ,$$

where $C_1 \Rightarrow \bigcirc(D_1 \vee L)$ and $C_2 \Rightarrow \bigcirc(D_2 \vee \neg M)$ are step clauses, $D_2 \vee \neg M$ is a universal clause, and and σ is an mgu of the literals L and M *such that* σ *does not map variables from* C_1 *or* C_2 *into a constant or a functional term.*

Deduction rules (Cont.)

2. *Resolution (first-order) between universal clauses*, resulting in a universal clause.
3. *Resolution (first-order) between initial and universal clauses (or initial clauses)*, resulting in an initial clause.
4. *Clause conversion.* A step clause of the form $C \Rightarrow \bigcirc$**false** is rewritten into the *universal clause* $\neg C$.
5. *Right/left factor*

$$\frac{C \Rightarrow \bigcirc(D \vee L \vee M)}{C\sigma \Rightarrow \bigcirc(D \vee L)\sigma}\ , \quad \frac{(C \wedge L \wedge M) \Rightarrow \bigcirc D}{(C \wedge L)\sigma \Rightarrow \bigcirc D\sigma}\ ,$$

where σ is an mgu of L and M *s.t.* σ *does not map variables from* C *into a constant or a functional term.*

The Original Calculus (Eventualities)

Eventuality resolution rule w.r.t. $\mathcal{U}$:

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

where $\forall x(\mathcal{A}_i(x) \Rightarrow \bigcirc \mathcal{B}_i(x))$ are complex combinations of step clauses s.t. for all $i \in \{1, \ldots, n\}$, the 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. $\forall x(\mathcal{A}_i(x) \Rightarrow \bigcirc(\mathcal{B}_i(x)))$ are known as a *loop in* $\neg L(x)$.

Loop Search and BFS

- Using fine-grained resolution for eventuality resolution is based on an extension to FOTL of the BFS algorithm for loop search in PTL.
- To find a loop in $\neg L(x)$ BFS finds a sequence of first-order formulae $H_0, \ldots H_n$ (where $H_0 = \textbf{true}$) s.t. for each $i \geqslant 0$ there are sets of complex combinations of step clause, $\forall x(\mathcal{A}_{i+1}(x) \Rightarrow \bigcirc \mathcal{B}_{i+1}(x))$, s.t.

$$\forall x((\mathcal{B}_{i+1}(x) \wedge \mathcal{U}) \Rightarrow (H_i(x) \wedge \neg L(x))).$$

- The algorithm terminates when either $H_n(x) = \textbf{false}$ (no loop) or $\forall x(H_{n-1}(x) \Rightarrow H_n(x))$.
- Note $\forall x((\mathcal{B}_{i+1}(x) \wedge \mathcal{U}) \Rightarrow (H_i(x) \wedge \neg L(x)))$ is valid when $\exists x((\mathcal{B}_{i+1}(x) \wedge \mathcal{U}) \wedge \neg(H_i(x) \wedge \neg L(x)))$ is unsatisfiable.

Fine-Grained Resolution and BFS

- Introduce a new constant c^l called the *loop constant*.
- To detect each $H_{i+1}(x)$ we add the clause

$$\textbf{true} \Rightarrow \bigcirc(\neg H_i(c^l) \vee L(c^l))$$

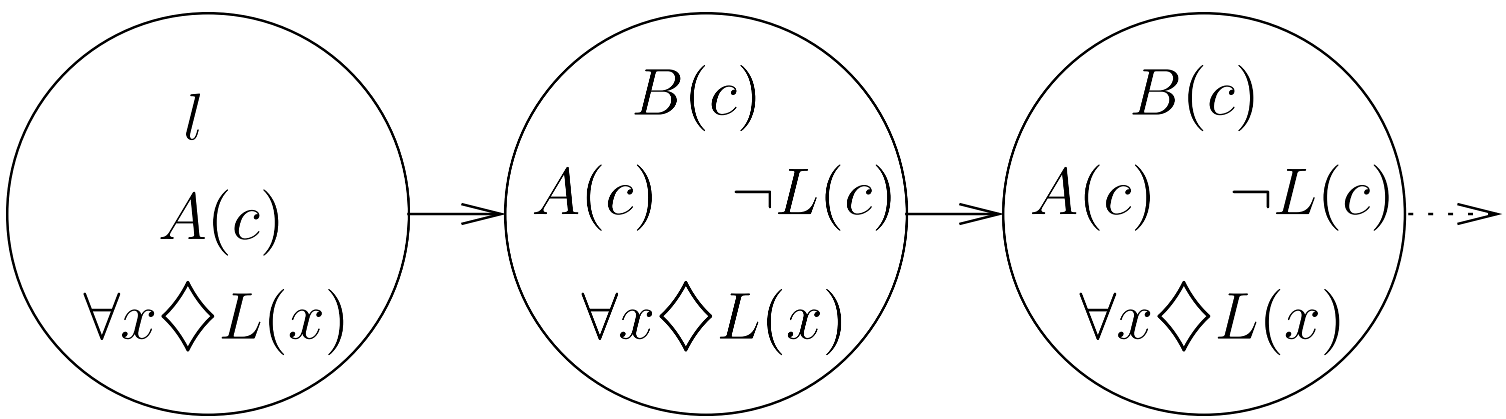
  and resolve.
- For all new clauses $C_j \Rightarrow \bigcirc$ **false** generated let $H_{i+1}(x) = \bigvee_{j=1}^{k} C_j\{c^l \rightarrow x\}$.
- We use all the rules for fine-grained resolution except the clause conversion rule.
- We relax the restriction on substitutions and allow the loop constant to be substituted into the left hand side of a step clause.

Example

Let us consider a monodic temporal problem P given by

$$\mathcal{I} = \{i1 : l\},\ \mathcal{U} = \{\forall x(B(x) \Rightarrow A(x) \wedge \neg L(x)),\ l \Rightarrow \exists x A(x)\},$$
$$\mathcal{S} = \{s1 : A(x) \Rightarrow \bigcirc B(x)\},\ \mathcal{E} = \{e1 : \Diamond L(x)\}.$$

Recall, this is short for $\mathcal{I} \wedge \Box\mathcal{U} \wedge \Box\forall x\mathcal{S} \wedge \Box\forall x\mathcal{E}$.



This is unsatisfiable which we will show by resolution.

Example (Cont.)

$$\mathcal{I} = \{i1 : l\},\ \mathcal{U} = \{\forall x(B(x) \Rightarrow A(x) \wedge \neg L(x)),\ l \Rightarrow \exists x A(x)\},$$
$$\mathcal{S} = \{s1 : A(x) \Rightarrow \bigcirc B(x)\},\ \mathcal{E} = \{e1 : \Diamond L(x)\}$$

We clausify $\mathcal{U}$ resulting in

$$\mathcal{U}^s = \{u1 : (\neg B(x) \vee A(x)),\ u2 : (\neg B(x) \vee \neg L(x)),\ u3 : \neg l \vee A(c)\}.$$

- Step resolution

$$s2 : A(x) \Rightarrow \bigcirc A(x) \quad (s1, u1)$$
$$s3 : A(x) \Rightarrow \bigcirc \neg L(x) \quad (s1, u2)$$

Now we try finding a loop in $\Diamond L(x)$.

Example (Cont.)

$$\mathcal{S} = \{s1 : A(x) \Rightarrow \bigcirc B(x),\ s2 : A(x) \Rightarrow \bigcirc A(x),$$
$$s3 : A(x) \Rightarrow \bigcirc \neg L(x)\}$$

- Loop resolution: resolve $\{l1 : \textbf{true} \Rightarrow \bigcirc L(c^l)\}$ with $\mathcal{U}$, $\mathcal{S}$.

$$l2 : A(c^l) \Rightarrow \bigcirc \textbf{false} \quad (s3, l1)$$

$H_1(x) = A(x)$. At the second iteration resolve $\{l3 : \textbf{true} \Rightarrow \bigcirc(\neg A(c^l) \vee L(c^l))\}$ with $\mathcal{U}$ and $\mathcal{S}$.

$$l4 : A(c^l) \Rightarrow \bigcirc L(c^l) \quad (s2, l3)$$
$$l5 : A(c^l) \Rightarrow \bigcirc \textbf{false} \quad (s3, l4)$$

$H_2(x) = A(x)$ and $\forall x(H_1(x) \Rightarrow H_2(x))$ so we terminate with the loop $A(x)$.

Example (Cont.)

$\mathcal{I} = \{i1 : l\},\ \mathcal{U}^s = \{u1 : (\neg B(x) \vee A(x)),\ u2 : (\neg B(x) \vee \neg L(x)),$
$u3 : \neg l \vee A(c)\},\ \mathcal{S} = \{s1 : A(x) \Rightarrow \bigcirc B(x),\ s2 : A(x) \Rightarrow \bigcirc A(x),$
$s3 : A(x) \Rightarrow \bigcirc \neg L(x)\},\ \mathcal{E} = \{e1 : \Diamond L(x)\}$

- Eventuality resolution: we can apply now the eventuality resolution rule whose conclusion is

$$u4 : \neg A(x).$$

- Universal/initial resolution

$$u5 : \neg l \quad (u3, u4) \quad i2 : \textbf{false} \quad (i1, u5)$$

The problem is unsatisfiable.

Theoretical Issues

- The translation to the normal form is satisfiability preserving.
- **Theorem 1** *The calculus consisting of the rules of fine-grained step resolution, together with the (both ground and non-ground) eventuality resolution rule, is sound and complete for the monodic fragment over expanding domains.*
- **Theorem 2** *The calculus consisting of the rules of fine-grained step resolution, together with the (both ground and non-ground) eventuality resolution rule, is complete for the monodic fragment over expanding domains even if we restrict ourselves to loops found by the BFS algorithm.*

Conclusions

- We have described a fine-grained resolution calculus for monodic first order temporal logics over expanding domains.
- Soundness of the fine-grained inference steps is easy to prove.
- Completeness is shown relative to the completeness proof for the expanding domain for the non-fine grained version.
- The fine-grained resolution inference rules are more amenable to efficient implementation and could be implemented directly using any appropriate first-order theorem prover for classical logics.