DECIDABLE FRAGMENTS OF LTL$_f$ MODULO THEORIES

Luca Geatti
University of Udine, Italy

Alessandro Gianola
Instituto Superior Técnico de Lisboa, Portugal

Nicola Gigante
Sarah Winkler
Free University of Bozen-Bolzano, Italy

26th European Conference on Artificial Intelligence
2 October 2023, Krakow

The Big Picture

Satisfiability checking of $\mathrm{LTL}_f\mathrm{MT}$

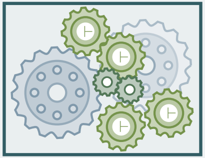
$$
\begin{aligned}
& \mathit{dist} = 100 \wedge \\
& \mathsf{G}\,(\neg \mathit{near\_goal}(\mathit{dist}) \longrightarrow (\bigcirc\, \mathit{dist} < \mathit{dist})) \wedge \\
& \mathsf{F}\,(\mathit{goal\_reached})
\end{aligned}
$$

- LTL over finite traces with arbitrary decidable logical theories

The Big Picture

Satisfiability checking of LTL$_f$MT

$$
\begin{aligned}
&\mathit{dist} = 100 \wedge \\
&\mathsf{G}\,(\neg \mathit{near\_goal}(\mathit{dist}) \longrightarrow (\bigcirc\, \mathit{dist} < \mathit{dist})) \wedge \\
&\mathsf{F}\,(\mathit{goal\_reached})
\end{aligned}
\longrightarrow
$$



→ satisfiable

⇢ timeout

→ unsatisfiable

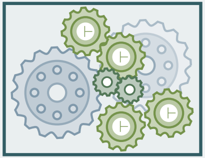
- LTL over finite traces with arbitrary decidable logical theories
- satisfiability can be checked by tableau method

The Big Picture

Satisfiability checking of LTL$_f$MT

$$\mathit{dist} = 100 \wedge$$
$$\mathsf{G}\,(\neg \mathit{near\_goal}(\mathit{dist}) \longrightarrow (\bigcirc\, \mathit{dist} < \mathit{dist})) \wedge$$
$$\mathsf{F}\,(\mathit{goal\_reached})$$

$\longrightarrow$



$\longrightarrow$ satisfiable

$\cdots\cdots\rightarrow$ timeout

$\longrightarrow$ unsatisfiable

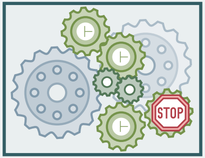
- LTL over finite traces with arbitrary decidable logical theories
- satisfiability can be checked by tableau method
- unsatisfiability can be rarely detected

The Big Picture

Satisfiability checking of LTL$_f$MT

$$dist = 100 \land$$
$$\mathsf{G}\,(\neg near\_goal(dist) \longrightarrow (\bigcirc\, dist < dist)) \land$$
$$\mathsf{F}\,(goal\_reached)$$

$\longrightarrow$



$\longrightarrow$ satisfiable

$\cdots\cdots\rightarrow$ timeout

$\longrightarrow$ unsatisfiable

- LTL over finite traces with arbitrary decidable logical theories
- satisfiability can be checked by tableau method
- unsatisfiability can be rarely detected

This talk

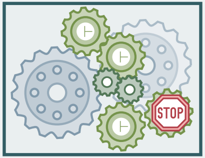
- sound and complete PRUNE rule for tableau method

The Big Picture

Satisfiability checking of LTL$_f$MT

$$
\begin{aligned}
& dist = 100 \wedge \\
& \mathsf{G}\,(\neg near\_goal(dist) \longrightarrow (\bigcirc\, dist < dist)) \wedge \\
& \mathsf{F}\,(goal\_reached)
\end{aligned}
$$

⟶



⟶ satisfiable

⋯⟶ timeout

⟶ unsatisfiable

- LTL over finite traces with arbitrary decidable logical theories
- satisfiability can be checked by tableau method
- unsatisfiability can be rarely detected

This talk

- sound and complete PRUNE rule for tableau method
- new, very general decidable fragments: generalizes results from literature

LTL_f modulo theories ($\mathsf{LTL}_f\mathsf{MT}$)

$$\varphi ::= \psi \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \wedge \varphi_2 \mid \mathsf{X}\,\varphi \mid \widetilde{\mathsf{X}}\,\varphi \mid \mathsf{F}\,\varphi \mid \mathsf{G}\,\varphi \mid \varphi_1\,\mathsf{U}\,\varphi_2$$

LTL_f modulo theories ($\mathsf{LTL}_f\mathsf{MT}$)

$$\varphi ::= \psi \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \wedge \varphi_2 \mid \mathsf{X}\,\varphi \mid \widetilde{\mathsf{X}}\,\varphi \mid \mathsf{F}\,\varphi \mid \mathsf{G}\,\varphi \mid \varphi_1\,\mathsf{U}\,\varphi_2$$

$$\psi ::= p(t_1, \ldots, t_n) \mid \neg\psi \mid \psi_1 \vee \psi_2 \mid \psi_1 \wedge \psi_2 \mid \exists z\,\psi \mid \forall z\,\psi$$

LTL_f modulo theories (LTL_fMT)

$$\varphi ::= \psi \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \wedge \varphi_2 \mid \mathsf{X}\,\varphi \mid \widetilde{\mathsf{X}}\,\varphi \mid \mathsf{F}\,\varphi \mid \mathsf{G}\,\varphi \mid \varphi_1\,\mathsf{U}\,\varphi_2$$

$$\psi ::= p(t_1, \ldots, t_n) \mid \neg\psi \mid \psi_1 \vee \psi_2 \mid \psi_1 \wedge \psi_2 \mid \exists z\,\psi \mid \forall z\,\psi$$

- first-order atoms over arbitrary theories

LTL_f modulo theories ($\mathsf{LTL}_f\mathsf{MT}$)

$$\varphi ::= \psi \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \wedge \varphi_2 \mid \mathsf{X}\,\varphi \mid \widetilde{\mathsf{X}}\,\varphi \mid \mathsf{F}\,\varphi \mid \mathsf{G}\,\varphi \mid \varphi_1\,\mathsf{U}\,\varphi_2$$

$$\psi ::= p(t_1, \dots, t_n) \mid \neg\psi \mid \psi_1 \vee \psi_2 \mid \psi_1 \wedge \psi_2 \mid \exists z\,\psi \mid \forall z\,\psi$$

- existential and universal quantification

LTL$_f$ modulo theories (LTL$_f$MT)

$$\varphi ::= \psi \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \wedge \varphi_2 \mid \mathsf{X}\,\varphi \mid \widetilde{\mathsf{X}}\,\varphi \mid \mathsf{F}\,\varphi \mid \mathsf{G}\,\varphi \mid \varphi_1\,\mathsf{U}\,\varphi_2$$

$$\psi ::= p(t_1, \dots, t_n) \mid \neg\psi \mid \psi_1 \vee \psi_2 \mid \psi_1 \wedge \psi_2 \mid \exists z\,\psi \mid \forall z\,\psi$$

$$t ::= f(t_1, \dots, t_n) \mid c \mid z \mid x \mid \bigcirc x \mid \circledsim x$$

- function application

LTL_f modulo theories ($\mathsf{LTL}_f\mathsf{MT}$)

$$\varphi ::= \psi \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \wedge \varphi_2 \mid \mathsf{X}\,\varphi \mid \widetilde{\mathsf{X}}\,\varphi \mid \mathsf{F}\,\varphi \mid \mathsf{G}\,\varphi \mid \varphi_1\,\mathsf{U}\,\varphi_2$$

$$\psi ::= p(t_1, \dots, t_n) \mid \neg\psi \mid \psi_1 \vee \psi_2 \mid \psi_1 \wedge \psi_2 \mid \exists z\,\psi \mid \forall z\,\psi$$

$$t ::= f(t_1, \dots, t_n) \mid c \mid z \mid x \mid \bigcirc x \mid \circledsim x$$

- first-order constants

LTL_f modulo theories ($\mathsf{LTL}_f\mathsf{MT}$)

$$\varphi ::= \psi \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \wedge \varphi_2 \mid \mathsf{X}\,\varphi \mid \widetilde{\mathsf{X}}\,\varphi \mid \mathsf{F}\,\varphi \mid \mathsf{G}\,\varphi \mid \varphi_1\,\mathsf{U}\,\varphi_2$$

$$\psi ::= p(t_1, \dots, t_n) \mid \neg\psi \mid \psi_1 \vee \psi_2 \mid \psi_1 \wedge \psi_2 \mid \exists z\,\psi \mid \forall z\,\psi$$

$$t ::= f(t_1, \dots, t_n) \mid c \mid z \mid x \mid \bigcirc x \mid \circledS x$$

- quantified first-order variables z

LTL_f modulo theories (LTL_fMT)

$$\varphi ::= \psi \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \wedge \varphi_2 \mid \mathsf{X}\,\varphi \mid \widetilde{\mathsf{X}}\,\varphi \mid \mathsf{F}\,\varphi \mid \mathsf{G}\,\varphi \mid \varphi_1\,\mathsf{U}\,\varphi_2$$

$$\psi ::= p(t_1, \dots, t_n) \mid \neg\psi \mid \psi_1 \vee \psi_2 \mid \psi_1 \wedge \psi_2 \mid \exists z\,\psi \mid \forall z\,\psi$$

$$t ::= f(t_1, \dots, t_n) \mid c \mid z \mid x \mid \bigcirc x \mid \circledtilde x$$

- first-order variables x — $x \in V$ for fixed, finite set of variables V

LTL_f modulo theories (LTL_fMT)

$$\varphi ::= \psi \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \wedge \varphi_2 \mid \mathsf{X}\,\varphi \mid \widetilde{\mathsf{X}}\,\varphi \mid \mathsf{F}\,\varphi \mid \mathsf{G}\,\varphi \mid \varphi_1 \,\mathsf{U}\, \varphi_2$$

$$\psi ::= p(t_1, \dots, t_n) \mid \neg\psi \mid \psi_1 \vee \psi_2 \mid \psi_1 \wedge \psi_2 \mid \exists z\, \psi \mid \forall z\, \psi$$

$$t ::= f(t_1, \dots, t_n) \mid c \mid z \mid x \mid \bigcirc x \mid \circledS x$$

- next-value variables $\bigcirc x$ — $x \in V$ for V fixed, finite set of data variables

  i.e. the value of x at the next state
- weak next-value variables

  i.e. the value of x at the next state, if it exists

LTL_f modulo theories ($\mathsf{LTL}_f\mathsf{MT}$)

$$\varphi ::= \psi \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \wedge \varphi_2 \mid \mathsf{X}\,\varphi \mid \widetilde{\mathsf{X}}\,\varphi \mid \mathsf{F}\,\varphi \mid \mathsf{G}\,\varphi \mid \varphi_1 \,\mathsf{U}\, \varphi_2$$

$$\psi ::= p(t_1, \dots, t_n) \mid \neg\psi \mid \psi_1 \vee \psi_2 \mid \psi_1 \wedge \psi_2 \mid \exists z\, \psi \mid \forall z\, \psi$$

$$t ::= f(t_1, \dots, t_n) \mid c \mid z \mid x \mid \bigcirc x \mid \widetilde{\bigcirc}\, x$$

Examples

- robot motion planning
  - variables V: *dist* (type *real*)   *goal-reached* (type *bool*)

LTL_f modulo theories (LTL_fMT)

$$\varphi ::= \psi \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \wedge \varphi_2 \mid \mathsf{X}\,\varphi \mid \widetilde{\mathsf{X}}\,\varphi \mid \mathsf{F}\,\varphi \mid \mathsf{G}\,\varphi \mid \varphi_1\,\mathsf{U}\,\varphi_2$$

$$\psi ::= p(t_1, \dots, t_n) \mid \neg\psi \mid \psi_1 \vee \psi_2 \mid \psi_1 \wedge \psi_2 \mid \exists z\,\psi \mid \forall z\,\psi$$

$$t ::= f(t_1, \dots, t_n) \mid c \mid z \mid x \mid \bigcirc x \mid \widetilde{\bigcirc}\, x$$

Examples

- robot motion planning
  - variables V: *dist* (type *real*) *goal-reached* (type *bool*)
  - $\mathit{dist} = 100$

initially, *dist* has value 100

LTL_f modulo theories (LTL_fMT)

$$\varphi ::= \psi \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \wedge \varphi_2 \mid \mathsf{X}\,\varphi \mid \widetilde{\mathsf{X}}\,\varphi \mid \mathsf{F}\,\varphi \mid \mathsf{G}\,\varphi \mid \varphi_1 \,\mathsf{U}\, \varphi_2$$
$$\psi ::= p(t_1, \dots, t_n) \mid \neg\psi \mid \psi_1 \vee \psi_2 \mid \psi_1 \wedge \psi_2 \mid \exists z\, \psi \mid \forall z\, \psi$$
$$t ::= f(t_1, \dots, t_n) \mid c \mid z \mid x \mid \bigcirc x \mid \widetilde{\bigcirc}\, x$$

Examples

- robot motion planning
  - variables V: *dist* (type *real*) *goal-reached* (type *bool*)
  - $dist = 100 \wedge \mathsf{F}\,(goal\text{-}reached)$

at some point, *goal-reached* is true

LTL$_f$ modulo theories (LTL$_f$MT)

$$\varphi ::= \psi \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \wedge \varphi_2 \mid \mathsf{X}\,\varphi \mid \widetilde{\mathsf{X}}\,\varphi \mid \mathsf{F}\,\varphi \mid \mathsf{G}\,\varphi \mid \varphi_1\,\mathsf{U}\,\varphi_2$$

$$\psi ::= p(t_1,\dots,t_n) \mid \neg\psi \mid \psi_1 \vee \psi_2 \mid \psi_1 \wedge \psi_2 \mid \exists z\,\psi \mid \forall z\,\psi$$

$$t ::= f(t_1,\dots,t_n) \mid c \mid z \mid x \mid \bigcirc x \mid \widetilde{\bigcirc}\, x$$

Examples

- robot motion planning
  - variables V: *dist* (type *real*)   *goal-reached* (type *bool*)
  - $dist = 100 \wedge \mathsf{F}\,(goal\text{-}reached) \wedge \mathsf{G}\,(\neg near\text{-}goal(dist) \longrightarrow (\bigcirc\, dist < dist))$

while *dist* is not near goal, *dist* decreases monotonically

LTL_f modulo theories ($\mathsf{LTL}_f\mathsf{MT}$)

$$\varphi ::= \psi \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \wedge \varphi_2 \mid \mathsf{X}\,\varphi \mid \widetilde{\mathsf{X}}\,\varphi \mid \mathsf{F}\,\varphi \mid \mathsf{G}\,\varphi \mid \varphi_1\,\mathsf{U}\,\varphi_2$$

$$\psi ::= p(t_1, \dots, t_n) \mid \neg\psi \mid \psi_1 \vee \psi_2 \mid \psi_1 \wedge \psi_2 \mid \exists z\, \psi \mid \forall z\, \psi$$

$$t ::= f(t_1, \dots, t_n) \mid c \mid z \mid x \mid \bigcirc x \mid \widetilde{\bigcirc}\, x$$

Definitions

- trace is finite sequence of valuations of V

Examples

- robot motion planning
  - variables V: *dist* (type *real*) *goal-reached* (type *bool*)
  - $\mathit{dist} = 100 \wedge \mathsf{F}\,(\mathit{goal\text{-}reached}) \wedge \mathsf{G}\,(\neg \mathit{near\text{-}goal}(\mathit{dist}) \longrightarrow (\bigcirc\, \mathit{dist} < \mathit{dist}))$

LTL_f modulo theories ($\mathrm{LTL}_f\mathrm{MT}$)

$$\varphi ::= \psi \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \wedge \varphi_2 \mid \mathsf{X}\,\varphi \mid \widetilde{\mathsf{X}}\,\varphi \mid \mathsf{F}\,\varphi \mid \mathsf{G}\,\varphi \mid \varphi_1\,\mathsf{U}\,\varphi_2$$

$$\psi ::= p(t_1, \dots, t_n) \mid \neg\psi \mid \psi_1 \vee \psi_2 \mid \psi_1 \wedge \psi_2 \mid \exists z\,\psi \mid \forall z\,\psi$$

$$t ::= f(t_1, \dots, t_n) \mid c \mid z \mid x \mid \bigcirc x \mid \circledsim x$$

Definitions

- trace is finite sequence of valuations of V

Examples

- robot motion planning
  - variables V: dist (type *real*) $\mathit{goal\text{-}reached}$ (type *bool*)
  - $\mathit{dist} = 100 \wedge \mathsf{F}\,(\mathit{goal\text{-}reached}) \wedge \mathsf{G}\,(\neg \mathit{near\text{-}goal}(\mathit{dist}) \longrightarrow (\bigcirc\,\mathit{dist} < \mathit{dist}))$
  - trace

$\mathit{goal\text{-}reached}=\mathit{false}$, $\mathit{dist}=10$	$\mathit{goal\text{-}reached}=\mathit{false}$, $\mathit{dist}=7.1$	$\mathit{goal\text{-}reached}=\mathit{false}$, $\mathit{dist}=5.6$	$\mathit{goal\text{-}reached}=\mathit{false}$, $\mathit{dist}=2.1$	$\mathit{goal\text{-}reached}=\mathit{false}$, $\mathit{dist}=0.01$

LTL_f modulo theories (LTL_fMT)

$$\varphi ::= \psi \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \wedge \varphi_2 \mid \mathsf{X}\,\varphi \mid \widetilde{\mathsf{X}}\,\varphi \mid \mathsf{F}\,\varphi \mid \mathsf{G}\,\varphi \mid \varphi_1\,\mathsf{U}\,\varphi_2$$

$$\psi ::= p(t_1,\dots,t_n) \mid \neg\psi \mid \psi_1 \vee \psi_2 \mid \psi_1 \wedge \psi_2 \mid \exists z\,\psi \mid \forall z\,\psi$$

$$t ::= f(t_1,\dots,t_n) \mid c \mid z \mid x \mid \bigcirc x \mid \circledsim x$$

Definitions

- trace is finite sequence of valuations of V
- LTL_fMT formula is satisfiable if $\exists$ first-order model and trace that satisfy it

Examples

- robot motion planning
  - variables V: *dist* (type *real*)   *goal-reached* (type *bool*)
  - $dist = 100 \wedge \mathsf{F}\,(\mathit{goal\text{-}reached}) \wedge \mathsf{G}\,(\neg \mathit{near\text{-}goal}(dist) \longrightarrow (\bigcirc\, dist < dist))$
  - trace

goal-reached=*false* *dist*=10	*goal-reached*=*false* *dist*=7.1	*goal-reached*=*false* *dist*=5.6	*goal-reached*=*false* *dist*=2.1	*goal-reached*=*false* *dist*=0.01

LTL_f modulo theories ($\mathrm{LTL}_f\mathrm{MT}$)

$$\varphi ::= \psi \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \wedge \varphi_2 \mid \mathsf{X}\,\varphi \mid \widetilde{\mathsf{X}}\,\varphi \mid \mathsf{F}\,\varphi \mid \mathsf{G}\,\varphi \mid \varphi_1\,\mathsf{U}\,\varphi_2$$

$$\psi ::= p(t_1,\dots,t_n) \mid \neg\psi \mid \psi_1 \vee \psi_2 \mid \psi_1 \wedge \psi_2 \mid \exists z\,\psi \mid \forall z\,\psi$$

$$t ::= f(t_1,\dots,t_n) \mid c \mid z \mid x \mid \bigcirc x \mid \circledS\, x$$

Definitions

- trace is finite sequence of valuations of V
- $\mathrm{LTL}_f\mathrm{MT}$ formula is satisfiable if $\exists$ first-order model and trace that satisfy it

Examples

- robot motion planning
  - variables V: *dist* (type *real*) *goal-reached* (type *bool*)
  - $dist = 100 \wedge \mathsf{F}\,(goal\text{-}reached) \wedge \mathsf{G}\,(\neg near\text{-}goal(dist) \longrightarrow (\bigcirc\, dist < dist))$ satisfiable
  - trace

LRA + $near\text{-}goal_{\mathcal{M}} = \begin{cases} true & \text{if } dist \leqslant 0.01 \\ false & \text{otherwise} \end{cases}$

goal-reached=false *dist=10*	*goal-reached=false* *dist=7.1*	*goal-reached=false* *dist=5.6*	*goal-reached=false* *dist=2.1*	*goal-reached=false* *dist=0.01*

LTL_f modulo theories (LTL_fMT)

$$\varphi ::= \psi \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \wedge \varphi_2 \mid \mathsf{X}\,\varphi \mid \widetilde{\mathsf{X}}\,\varphi \mid \mathsf{F}\,\varphi \mid \mathsf{G}\,\varphi \mid \varphi_1\,\mathsf{U}\,\varphi_2$$

$$\psi ::= p(t_1,\dots,t_n) \mid \neg\psi \mid \psi_1 \vee \psi_2 \mid \psi_1 \wedge \psi_2 \mid \exists z\,\psi \mid \forall z\,\psi$$

$$t ::= f(t_1,\dots,t_n) \mid c \mid z \mid x \mid \bigcirc x \mid \widetilde{\bigcirc}\, x$$

Definitions

undecidable

- trace is finite sequence of valuations of V
- LTL_fMT formula is satisfiable if $\exists$ first-order model and trace that satisfy it

Examples

- robot motion planning
  - variables V: *dist* (type *real*) *goal-reached* (type *bool*)
  - $dist = 100 \wedge \mathsf{F}\,(goal\text{-}reached) \wedge \mathsf{G}\,(\neg near\text{-}goal(dist) \longrightarrow (\bigcirc\, dist < dist))$ satisfiable
  - trace

goal-reached=*false* *dist*=10	*goal-reached*=*false* *dist*=7.1	*goal-reached*=*false* *dist*=5.6	*goal-reached*=*false* *dist*=2.1	*goal-reached*=*false* *dist*=0.01

LTL_f modulo theories ($\mathsf{LTL}_f\mathsf{MT}$)

$$\varphi ::= \psi \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \wedge \varphi_2 \mid \mathsf{X}\,\varphi \mid \widetilde{\mathsf{X}}\,\varphi \mid \mathsf{F}\,\varphi \mid \mathsf{G}\,\varphi \mid \varphi_1\,\mathsf{U}\,\varphi_2$$

$$\psi ::= p(t_1,\dots,t_n) \mid \neg\psi \mid \psi_1 \vee \psi_2 \mid \psi_1 \wedge \psi_2 \mid \exists z\,\psi \mid \forall z\,\psi$$

$$t ::= f(t_1,\dots,t_n) \mid c \mid z \mid x \mid \bigcirc x \mid \circledsim x$$

Definitions

- trace is finite sequence of valuations of V
- $\mathsf{LTL}_f\mathsf{MT}$ formula is satisfiable if $\exists$ first-order model and trace that satisfy it

Examples

- robot motion planning
  - variables V: *dist* (type *real*) *goal-reached* (type *bool*)
  - $\mathit{dist} = 100 \wedge \mathsf{F}\,(\mathit{goal\text{-}reached}) \wedge \mathsf{G}\,(\neg\mathit{near\text{-}goal}(\mathit{dist}) \longrightarrow (\bigcirc\,\mathit{dist} < \mathit{dist}))$ satisfiable
  - trace

goal-reached=*false* *dist*=10	*goal-reached*=*false* *dist*=7.1	*goal-reached*=*false* *dist*=5.6	*goal-reached*=*false* *dist*=2.1	*goal-reached*=*false* *dist*=0.01

- can describe business processes that operate over data and access read-only database

Tableau method for LTL_fMT

- semi-decision procedure to check satisfiability
- can construct model if it exists
- implemented in LTL/LTL_f satisfiability checker BLACK, using SMT solver as backend

[Geatti, Gianola & Gigante, IJCAI 2022]

Tableau method for $\mathsf{LTL}_f\mathsf{MT}$

- semi-decision procedure to check satisfiability
- can construct model if it exists
- implemented in $\mathsf{LTL}/\mathsf{LTL}_f$ satisfiability checker BLACK, using SMT solver as backend

[Geatti, Gianola & Gigante, IJCAI 2022]

Example (Tableau)

- $\psi := (x = 1) \wedge ((\bigcirc x > x) \mathsf{U} (x = 5))$

Tableau method for LTL_fMT

- semi-decision procedure to check satisfiability
- can construct model if it exists
- implemented in LTL/LTL_f satisfiability checker BLACK, using SMT solver as backend

[Geatti, Gianola & Gigante, IJCAI 2022]

Example (Tableau)

- $\psi := (x = 1) \wedge ((\bigcirc x > x) \,\mathsf{U}\, (x = 5))$

$\{\psi\}$

root node labeled with ψ

...

Tableau method for $\mathsf{LTL}_f\mathsf{MT}$

- semi-decision procedure to check satisfiability
- can construct model if it exists
- implemented in LTL/LTL_f satisfiability checker BLACK, using SMT solver as backend

[Geatti, Gianola & Gigante, IJCAI 2022]

Example (Tableau)

- $\psi := (x = 1) \wedge ((\bigcirc x > x) \;\mathsf{U}\; (x = 5))$

$$\{\psi\}$$

rule to split conjunction

$$\{x = 1, (\bigcirc x > x) \;\mathsf{U}\; (x = 5)\}$$

...

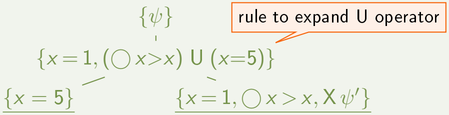
Tableau method for $\mathsf{LTL}_f\mathsf{MT}$

- semi-decision procedure to check satisfiability
- can construct model if it exists
- implemented in LTL/LTL_f satisfiability checker BLACK, using SMT solver as backend

[Geatti, Gianola & Gigante, IJCAI 2022]

Example (Tableau)

- $\psi := (x = 1) \wedge ((\bigcirc x > x) \,\mathsf{U}\, (x = 5))$
  $\psi' := (\bigcirc x > x) \,\mathsf{U}\, (x = 5)$



...

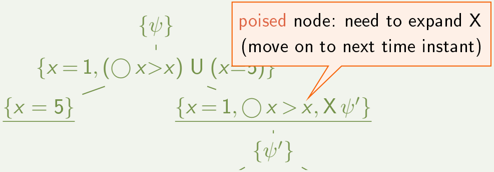
Tableau method for $\mathrm{LTL}_f\mathrm{MT}$

- semi-decision procedure to check satisfiability
- can construct model if it exists
- implemented in LTL/LTL_f satisfiability checker BLACK, using SMT solver as backend

[Geatti, Gianola & Gigante, IJCAI 2022]

Example (Tableau)

- $\psi := (x = 1) \wedge ((\bigcirc x > x) \mathsf{U} (x = 5))$
  $\psi' := (\bigcirc x > x) \mathsf{U} (x = 5)$



...

Tableau method for LTL$_f$MT

- semi-decision procedure to check satisfiability
- can construct model if it exists
- implemented in LTL/LTL$_f$ satisfiability checker BLACK, using SMT solver as backend

[Geatti, Gianola & Gigante, IJCAI 2022]

Example (Tableau)

- $\psi := (x = 1) \wedge ((\bigcirc x > x) \,\mathsf{U}\, (x = 5))$
  $\psi' := (\bigcirc x > x) \,\mathsf{U}\, (x = 5)$

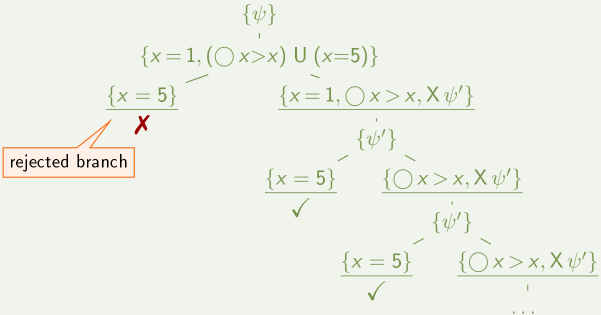


Tableau method for LTL$_f$MT

- semi-decision procedure to check satisfiability
- can construct model if it exists
- implemented in LTL/LTL$_f$ satisfiability checker BLACK, using SMT solver as backend

[Geatti, Gianola & Gigante, IJCAI 2022]

Example (Tableau)

- $\psi := (x = 1) \wedge ((\bigcirc x > x) \,\mathsf{U}\, (x = 5))$
  $\psi' := (\bigcirc x > x) \,\mathsf{U}\, (x = 5)$

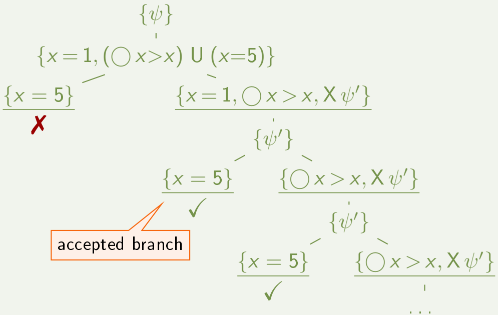


Tableau method for LTL$_f$MT

- semi-decision procedure to check satisfiability
- can construct model if it exists
- implemented in LTL/LTL$_f$ satisfiability checker BLACK, using SMT solver as backend

[Geatti, Gianola & Gigante, IJCAI 2022]

Example (Tableau)

- $\psi := (x = 1) \wedge ((\bigcirc x > x) \, \mathsf{U} \, (x = 5))$
  $\psi' := (\bigcirc x > x) \, \mathsf{U} \, (x = 5)$
- can extract model from accepted branches:
  - $\langle \{x = 1\}, \{x = 5\} \rangle$

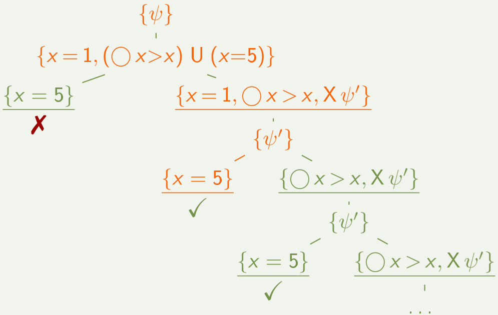


Tableau method for LTL$_f$MT

- semi-decision procedure to check satisfiability
- can construct model if it exists
- implemented in LTL/LTL$_f$ satisfiability checker BLACK, using SMT solver as backend

[Geatti, Gianola & Gigante, IJCAI 2022]

Example (Tableau) satisfiable

- $\psi := (x = 1) \wedge ((\bigcirc x > x) \; \mathsf{U} \; (x = 5))$
  $\psi' := (\bigcirc x > x) \; \mathsf{U} \; (x = 5)$
- can extract model from accepted branches:
  - $\langle \{x = 1\}, \{x = 5\} \rangle$
  - $\langle \{x = 1\}, \{x = 3\}, \{x = 5\} \rangle$

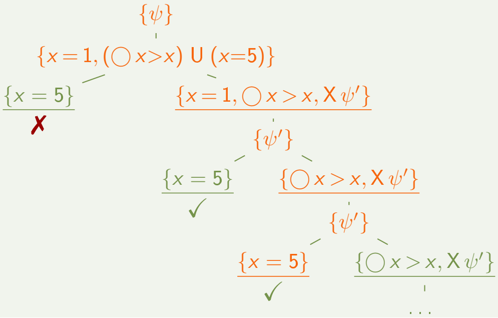


Tableau method for $\mathsf{LTL}_f\mathsf{MT}$

- semi-decision procedure to check satisfiability
- can construct model if it exists
- implemented in LTL/LTL_f satisfiability checker BLACK, using SMT solver as backend

[Geatti, Gianola & Gigante, IJCAI 2022]

Example (Tableau)

- $\psi := (x = 1) \wedge ((\bigcirc x > x) \mathsf{U} (x = 0))$

Tableau method for LTL$_f$MT

- semi-decision procedure to check satisfiability
- can construct model if it exists
- implemented in LTL/LTL$_f$ satisfiability checker BLACK, using SMT solver as backend

[Geatti, Gianola & Gigante, IJCAI 2022]

Example (Tableau) — unsatisfiable

- $\psi := (x=1) \wedge ((\bigcirc x > x) \, \mathsf{U} \, (x=0))$
  $\psi' := (\bigcirc x > x) \, \mathsf{U} \, (x=0)$

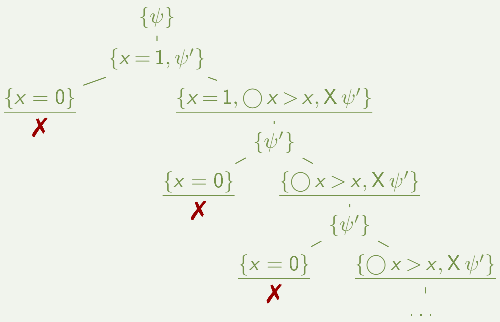


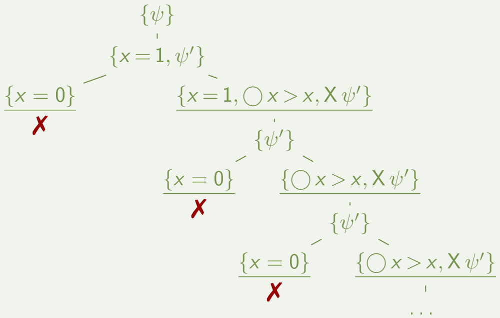
Tableau method for LTL$_f$MT

- semi-decision procedure to check satisfiability
- can construct model if it exists
- implemented in LTL/LTL$_f$ satisfiability checker BLACK, using SMT solver as backend

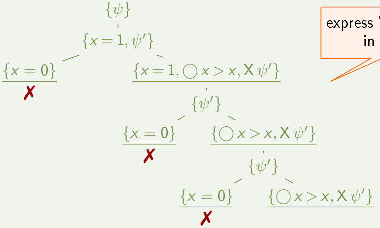
[Geatti, Gianola & Gigante, IJCAI 2022]

Example (Tableau) unsatisfiable

- $\psi := (x = 1) \wedge ((\bigcirc x > x) \, \mathsf{U} \, (x = 0))$
  $\psi' := (\bigcirc x > x) \, \mathsf{U} \, (x = 0)$
- tableau is infinite,
  no accepted branch



Key idea

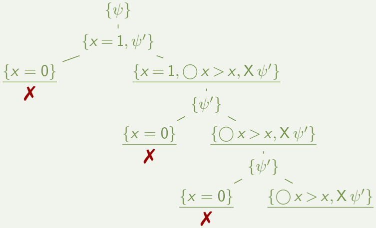


express "situation of variables" in node, as formula

Definition (History constraints)

for tableau branch $\overline{\pi}$ where poised nodes have first-order formulas $F_0, \dots, F_{m-1}$,

Key idea



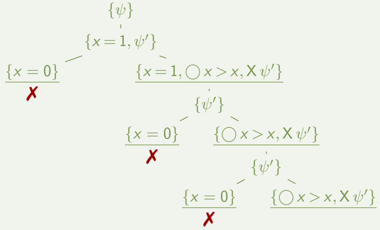
Definition (History constraints)

for tableau branch $\overline{\pi}$ where poised nodes have first-order formulas $F_0, \ldots, F_{m-1}$,

$$F_0(\overline{V}_0, \overline{V}_1) \wedge \cdots \wedge F_{m-1}(\overline{V}_{m-1}, \overline{V}_m)$$

for fresh copies $V_0, \ldots V_m$ of V

Key idea



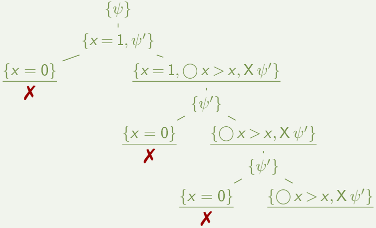
Definition (History constraints)

for tableau branch $\overline{\pi}$ where poised nodes have first-order formulas $F_0, \dots, F_{m-1}$,

$$(\exists V_0 \dots V_{m-1}. F_0(\overline{V}_0, \overline{V}_1) \wedge \dots \wedge F_{m-1}(\overline{V}_{m-1}, \overline{V}_m))$$

for fresh copies $V_0, \dots V_m$ of V

Key idea



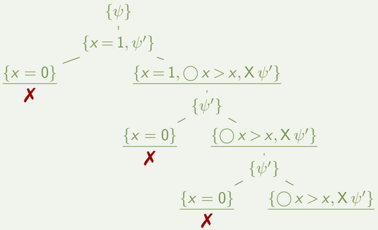
Definition (History constraints)

for tableau branch $\overline{\pi}$ where poised nodes have first-order formulas $F_0, \dots, F_{m-1}$,

$$h(\overline{\pi}) = (\exists V_0 \dots V_{m-1}.F_0(\overline{V}_0, \overline{V}_1) \wedge \dots \wedge F_{m-1}(\overline{V}_{m-1}, \overline{V}_m))[V_m/V]$$

for fresh copies $V_0, \dots V_m$ of V

Key idea



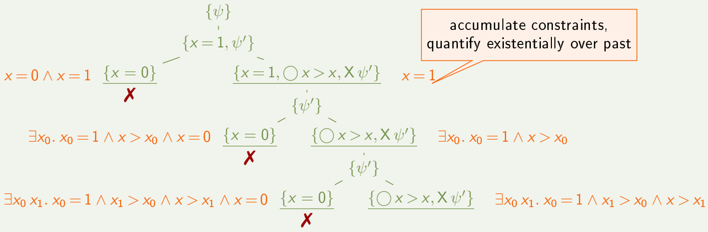
Definition (History constraints)

for tableau branch $\overline{\pi}$ where poised nodes have first-order formulas $F_0, \dots, F_{m-1}$,

$$h(\overline{\pi}) = (\exists V_0 \dots V_{m-1}. F_0(\overline{V}_0, \overline{V}_1) \wedge \dots \wedge F_{m-1}(\overline{V}_{m-1}, \overline{V}_m))[V_m/V]$$

for fresh copies $V_0, \dots V_m$ of V

Key idea



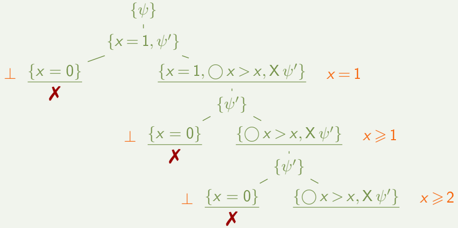
Definition (History constraints)

for tableau branch $\overline{\pi}$ where poised nodes have first-order formulas $F_0, \dots, F_{m-1}$,

$$h(\overline{\pi}) = (\exists V_0 \dots V_{m-1}.F_0(\overline{V}_0, \overline{V}_1) \wedge \dots \wedge F_{m-1}(\overline{V}_{m-1}, \overline{V}_m))[V_m/V]$$

for fresh copies $V_0, \dots V_m$ of V

Key idea



Definition (History constraints)

for tableau branch $\overline{\pi}$ where poised nodes have first-order formulas $F_0, \dots, F_{m-1}$,

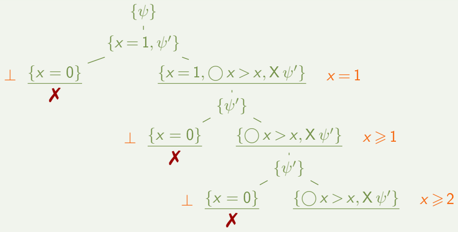
$$h(\overline{\pi}) = (\exists V_0 \dots V_{m-1}. F_0(\overline{V}_0, \overline{V}_1) \wedge \dots \wedge F_{m-1}(\overline{V}_{m-1}, \overline{V}_m))[V_m/V]$$

for fresh copies $V_0, \dots V_m$ of V

Definition (PRUNE rule)

for branch with poised nodes $\overline{\pi} = \langle \pi_0, \dots, \pi_i, \dots, \pi_{m-1} \rangle$, if π_i and π_{m-1} have same labels and $h(\overline{\pi}) \models_{\mathcal{T}} h(\overline{\pi}_{\leqslant i})$ then branch is rejected

Key idea



Definition (History constraints)

for tableau branch $\overline{\pi}$ where poised nodes have first-order formulas $F_0, \ldots, F_{m-1}$,

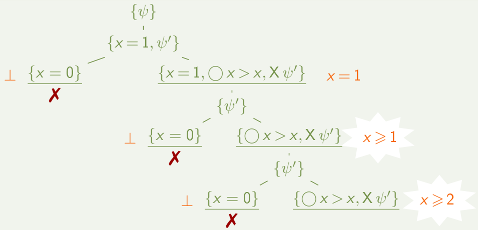
$$h(\overline{\pi}) = (\exists V_0 \ldots V_{m-1}. F_0(\overline{V}_0, \overline{V}_1) \wedge \cdots \wedge F_{m-1}(\overline{V}_{m-1}, \overline{V}_m))[V_m/V]$$

for fresh copies $V_0, \ldots V_m$ of V

Definition (PRUNE rule)

for branch with poised nodes $\overline{\pi} = \langle \pi_0, \ldots, \pi_i, \ldots, \pi_{m-1} \rangle$, if π_i and π_{m-1} have same labels and $h(\overline{\pi}) \models_{\mathcal{T}} h(\overline{\pi}_{\leqslant i})$ then branch is rejected

Key idea



Definition (History constraints)

for tableau branch $\overline{\pi}$ where poised nodes have first-order formulas $F_0, \dots, F_{m-1}$,

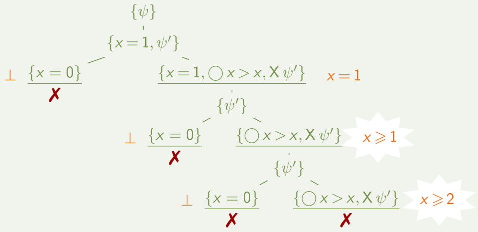
$$h(\overline{\pi}) = (\exists V_0 \dots V_{m-1}. F_0(\overline{V}_0, \overline{V}_1) \wedge \dots \wedge F_{m-1}(\overline{V}_{m-1}, \overline{V}_m))[V_m/V]$$

for fresh copies $V_0, \dots V_m$ of V

Definition (PRUNE rule)

for branch with poised nodes $\overline{\pi} = \langle \pi_0, \dots, \pi_i, \dots, \pi_{m-1} \rangle$, if π_i and π_{m-1} have same labels and $h(\overline{\pi}) \models_{\mathcal{T}} h(\overline{\pi}_{\leqslant i})$ then branch is rejected

Key idea



Definition (History constraints)

for tableau branch $\overline{\pi}$ where poised nodes have first-order formulas $F_0, \dots, F_{m-1}$,

$$h(\overline{\pi}) = (\exists V_0 \dots V_{m-1}. F_0(\overline{V}_0, \overline{V}_1) \wedge \dots \wedge F_{m-1}(\overline{V}_{m-1}, \overline{V}_m))[V_m/V]$$

for fresh copies $V_0, \dots V_m$ of V

Definition (PRUNE rule)

for branch with poised nodes $\overline{\pi} = \langle \pi_0, \dots, \pi_i, \dots, \pi_{m-1} \rangle$, if π_i and π_{m-1} have same labels and $h(\overline{\pi}) \models_{\mathcal{T}} h(\overline{\pi}_{\leqslant i})$ then branch is rejected

Theorem (Soundness and completeness)

for LTL$_f$MT formula ψ, tableau with PRUNE rule has accepted branch iff ψ is satisfiable

Decidability Results

Observation: Satisfiability can be decidable

tableau with PRUNE halts on classes of formulas where set of history constraints is finite up to $\equiv$

Decidability Results

Observation: Satisfiability can be decidable

tableau with PRUNE halts on classes of formulas where set of history constraints is finite up to $\equiv$

Decidable fragments of $\mathsf{LTL}_f\mathsf{MT}$

1. formulas without $\bigcirc$ and $\widetilde{\bigcirc}$ — $(x>y \;\mathsf{U}\; x+y=2z) \wedge \mathsf{G}\,(x+y>0)$

Decidability Results

Observation: Satisfiability can be decidable

tableau with PRUNE halts on classes of formulas where set of history constraints is finite up to $\equiv$

Decidable fragments of $\mathrm{LTL}_f\mathrm{MT}$

1. formulas without $\bigcirc$ and $\circledsim$ — $(x > y \mathbin{\mathsf{U}} x+y = 2z) \wedge \mathsf{G}\,(x+y > 0)$
2. formulas without G and U — $\mathsf{F}\,(p(\bigcirc x) \wedge \mathsf{X}\,(\neg p(x))) \wedge \mathsf{X}\,\mathsf{F}\,(r(x,y) \vee r(\bigcirc x, y))$

Decidability Results

Observation: Satisfiability can be decidable

tableau with PRUNE halts on classes of formulas where set of history constraints is finite up to $\equiv$

Decidable fragments of LTL$_f$MT

1. formulas without $\bigcirc$ and $\circledsim$ $\quad (x>y \mathsf{U}\, x+y=2z) \wedge \mathsf{G}\,(x+y>0)$
2. formulas without G and U $\quad \mathsf{F}\,(p(\bigcirc x) \wedge \mathsf{X}\,(\neg p(x))) \wedge \mathsf{X}\,\mathsf{F}\,(r(x,y) \vee r(\bigcirc x, y))$
3. bounded lookback formulas, that restrict variable dependencies via $\bigcirc$ and $\circledsim$ to boundedly many configurations $\quad p(x, \bigcirc y)\, \mathsf{U}\, (\bigcirc x = x + y)$

Decidability Results

Observation: Satisfiability can be decidable

tableau with PRUNE halts on classes of formulas where set of history constraints is finite up to $\equiv$

Decidable fragments of LTL$_f$MT

1. formulas without $\bigcirc$ and $\circledS$ $\qquad (x>y \mathbin{\mathsf{U}} x+y=2z) \land \mathsf{G}\,(x+y>0)$
2. formulas without G and U $\qquad \mathsf{F}\,(p(\bigcirc x) \land \mathsf{X}\,(\neg p(x))) \land \mathsf{X}\,\mathsf{F}\,(r(x,y) \lor r(\bigcirc x, y))$
3. bounded lookback formulas, that restrict variable dependencies via $\bigcirc$ and $\circledS$ to boundedly many configurations $\qquad p(x, \bigcirc y) \mathbin{\mathsf{U}} (\bigcirc x = x + y)$
4. formulas over theory of linear real arithmetic, where first arguments of G and U are variable-to-variable/constant comparisons $\qquad (x=1) \land \mathsf{G}\,(\circledS x > x) \land \mathsf{F}\,(x+y=\frac{1}{2})$

Decidability Results

Observation: Satisfiability can be decidable

tableau with PRUNE halts on classes of formulas where set of history constraints is finite up to $\equiv$

Decidable fragments of LTL$_f$MT

1. formulas without $\bigcirc$ and $\circleddash$ $\quad (x > y \mathbin{\mathsf{U}} x + y = 2z) \wedge \mathsf{G}\,(x + y > 0)$
2. formulas without G and U $\quad \mathsf{F}\,(p(\bigcirc x) \wedge \mathsf{X}\,(\neg p(x))) \wedge \mathsf{X}\,\mathsf{F}\,(r(x, y) \vee r(\bigcirc x, y))$
3. bounded lookback formulas, that restrict variable dependencies via $\bigcirc$ and $\circleddash$ to boundedly many configurations $\quad p(x, \bigcirc y) \mathbin{\mathsf{U}} (\bigcirc x = x + y)$
4. formulas over theory of linear real arithmetic, where first arguments of G and U are variable-to-variable/constant comparisons $\quad (x = 1) \wedge \mathsf{G}\,(\circleddash x > x) \wedge \mathsf{F}\,(x + y = \frac{1}{2})$
5. formulas over theory of linear integer arithmetic, where first arguments of G and U are integer periodicity constraints $\quad (z = 1) \wedge \mathsf{G}\,(z \approx_5 \circleddash z) \wedge \mathsf{F}\,(z = 42)$

Decidability Results

Observation: Satisfiability can be decidable

tableau with PRUNE halts on classes of formulas where set of history constraints is finite up to $\equiv$

Decidable fragments of $\mathsf{LTL}_f\mathsf{MT}$

1. formulas without $\bigcirc$ and $\circledsim$ $\quad (x > y \,\mathsf{U}\, x + y = 2z) \wedge \mathsf{G}\,(x + y > 0)$
2. formulas without G and U $\quad \mathsf{F}\,(p(\bigcirc x) \wedge \mathsf{X}\,(\neg p(x))) \wedge \mathsf{X}\,\mathsf{F}\,(r(x, y) \vee r(\bigcirc x, y))$
3. bounded lookback formulas, that restrict variable dependencies via $\bigcirc$ and $\circledsim$ to boundedly many configurations
4. formulas over theory of linear real arithmetic, wh
variable-to-variable/constant comparisons
5. formulas over theory of linear integer arithmetic, where firs
integer periodicity constraints $\quad (z = 1) \wedge \mathsf{G}\,(z \approx_5 \circledsim z) \wedge \mathsf{F}\,(z = 42)$

generalizes Demri & D'Souza (2007):
An automata-theoretic approach to constraint LTL

generalizes Demri (2006):
LTL over integer periodicity constraints

Conclusion

Summary

- tableau method to check satisfiability of LTL_f modulo theories: propose sound and complete PRUNE rule
- new decidable fragments for very general logic
- generalizes results from literature

Conclusion

Summary

- tableau method to check satisfiability of LTL_f modulo theories: propose sound and complete PRUNE rule
- new decidable fragments for very general logic
- generalizes results from literature

Future work

- implementation in BLACK
- in traces, allow interpretations of relations and functions to change over time
- extension to branching time logics
- LTL_fMT monitoring