Prompt Interval Temporal Logic

Dario Della Monica, Angelo Montanari, Aniello Murano, and Pietro Sala

JELIA 2016
Larnaca, Cyprus, 11/11
Outline

Introduction

The logic PROMPT–PNL
(Interval) Temporal Logic and PNL
PROMPT–PNL

Undecidability

Recovering decidability

Conclusions and future work
Outline

Introduction

The logic PROMPT–PNL

(Interval) Temporal Logic and PNL

PROMPT–PNL

Undecidability

Recovering decidability

Conclusions and future work
What is prompt

**Intuition**: to bound the delay with which a request is satisfied
What is prompt

Intuition: to bound the delay with which a request is satisfied

▶ the bound is constant ...

Delay sequence: 1, 2, 1, 2, 1, 2, ...
Bound: 2

Delay sequence: 1, N, 1, N, 1, N, ...
Bound: N (a constant)
What is prompt

**Intuition:** to bound the delay with which a request is satisfied

- the bound is constant ...

Delay sequence: 1, 2, 1, 2, 1, 2, ...
What is prompt

**Intuition:** to bound the delay with which a request is satisfied

- the bound is constant ...

Delay sequence: 1, 2, 1, 2, 1, 2, ...  
Bound: 2

\[ [G] (req \rightarrow Xsat \lor XXsat) \]
What is prompt

**Intuition:** to bound the delay with which a request is satisfied

- the bound is constant ...

![Diagram showing delay sequence: 1, 2, 1, 2, 1, 2, ... \(\text{Bound: } 2\)]

\([G](\text{req} \rightarrow \text{Xsat} \lor \text{XXsat})\)

- ... but unknown (or arbitrarily large)

![Diagram showing delay sequence: 1, N, 1, N, 1, N, ... \(\text{Bound: } N \text{ (a constant)}\)]
What is prompt

**Intuition:** to bound the delay with which a request is satisfied

- the bound is constant ...

```
D. Della Monica
```

```
Delay sequence: 1, 2, 1, 2, 1, 2, ...  
Bound: 2

[G](\text{req} \rightarrow \text{Xsat} \lor \text{XXsat})

... but unknown (or arbitrarily large)
```

```
Delay sequence: 1, N, 1, N, 1, N, ...
```

Prompt Interval Temporal Logic  
D. Della Monica
What is prompt

**Intuition:** to bound the delay with which a request is satisfied

- the bound is constant ...

Delay sequence: 1, 2, 1, 2, 1, 2, ...

Bound: 2

\[ G(req \rightarrow Xsat \lor XXsat) \]

- ... but unknown (or arbitrarily large)

Delay sequence: 1, N, 1, N, 1, N, ...

Bound: N (a constant)
What is not prompt

Delay sequence: 1, 2, 3, 4, 5, ...
What is not prompt

Delay sequence: 1, 2, 3, 4, 5, ...  
Bound: $\infty$ (unbounded)
Prompt extensions of temporal logic

- PLTL
  [Alur-Etessami-La Torre-Peled, 2001]

- PROMPT-LTL
  [Kupferman-Piterman-Vardi, 2009]
Outline

Introduction

The logic PROMPT–PNL
(Interval) Temporal Logic and PNL
PROMPT–PNL

Undecidability

Recovering decidability

Conclusions and future work
Outline

Introduction

The logic PROMPT-PNL

(Interval) Temporal Logic and PNL

PROMPT-PNL

Undecidability

Recovering decidability

Conclusions and future work
Temporal logics

- Temporal logics are (multi-)modal logics

set of worlds
primitive temporal entity
time points/instants

accessibility relations
- : next
- : finally

simplification
A different approach: from points to intervals

- worlds are intervals (time period — pairs of points)

set of worlds
primitive temporal entity
- time intervals/periods

accessibility relations
all binary relations between pairs of intervals
The logic PNL

**Syntax**

\[ \varphi ::= p \mid \neg \varphi \mid \varphi \land \varphi \mid \langle A \rangle \varphi \mid \langle \overline{A} \rangle \varphi \]

**Semantics**

Models: \( \mathbf{M} = \langle \mathbb{I}(\mathbb{D}), V \rangle \)

(intervals over a linear order + atomic propositions eval.)

\[ \langle A \rangle: \mathbf{M}, [d_0, d_1] \models \langle A \rangle \varphi \text{ iff } \exists d_2 \text{ s.t. } d_1 < d_2 \text{ and } \mathbf{M}, [d_1, d_2] \models \varphi \]

\[ \langle \overline{A} \rangle: \mathbf{M}, [d_0, d_1] \models \langle \overline{A} \rangle \varphi \text{ iff } \exists d_2 \text{ s.t. } d_2 < d_0 \text{ and } \mathbf{M}, [d_2, d_0] \models \varphi \]

current interval: \[ \begin{array}{c}
    d_0 \\
    \downarrow \\
    \downarrow \varphi \\
    d_1
\end{array} \]

\[ \langle A \rangle \varphi: \begin{array}{c}
    d_0 \\
    \downarrow \\
    d_1 \\
    \varphi \\
    d_2
\end{array} \]

\[ \langle \overline{A} \rangle \varphi: \begin{array}{c}
    d_2 \\
    \downarrow \varphi \\
    d_0
\end{array} \]
Outline

Introduction

The logic PROMPT–PNL

(Interval) Temporal Logic and PNL

PROMPT–PNL

Undecidability

Recovering decidability

Conclusions and future work
The logic PROMPT-PNL

Syntax

\[ \varphi ::= p \mid \neg \varphi \mid \varphi \land \varphi \mid \langle A \rangle \varphi \mid \langle \overline{A} \rangle \varphi \mid \langle A_x \rangle \varphi \mid \langle \overline{A}_x \rangle \varphi \]

Semantics

Models: \( M = \langle \mathbb{I}(\mathbb{D}), V, \delta, \sigma \rangle \)

(intervals over a linear order + atomic propositions eval. + metric + bounding variables eval.)

\[ \langle A_x \rangle: M, [d_0, d_1] \models \langle A_x \rangle \varphi \iff \exists d_2 \text{ s.t. } d_1 < d_2, \text{ length}_\delta([d_1, d_2]) \leq \sigma(x), \text{ and } M, [d_1, d_2] \models \varphi \]

\[ \langle \overline{A}_x \rangle: M, [d_0, d_1] \models \langle \overline{A}_x \rangle \varphi \iff \exists d_2 \text{ s.t. } d_2 < d_0, \text{ length}_\delta([d_2, d_0]) \leq \sigma(x), \text{ and } M, [d_2, d_0] \models \varphi \]
The satisfiability problem for PROMPT–PNL

**Input:** ▶ a PROMPT–PNL formula \( \varphi \)

**Question:** Are there

▶ a model \( M = \langle \mathbb{I}(D), V, \delta, \sigma \rangle \) and
▶ an interval \([a, b]\) \(\in \mathbb{I}(D)\)

that satisfy \( \varphi \) (i.e., \( M, [a, b] \models \varphi \))
Outline

Introduction

The logic PROMPT–PNL
(Interval) Temporal Logic and PNL
PROMPT–PNL

Undecidability

Recovering decidability

Conclusions and future work
Undecidability of PROMPT–PNL

Theorem

The satisfiability problem for PROMPT–PNL is undecidable

Proof

By reduction from the *Finite Coloring Problem* (aka. *Finite Tiling Problem*)
Finite Coloring Problem

Input:
\( C \): a set of colors
\( H \) and \( V \): two binary relations over colors
(horizontal and vertical color compatibilities)
\( c_i \) and \( c_f \): two distinguished colors in \( C \)
(initial and final color constraints)

Question: Are there
\( K \) and \( L \), and
a coloring function
\( C : \{1, \ldots, K\} \times \{1, \ldots, L\} \rightarrow C \)
such that that horizontal/vertical compatibilities and initial/final constraints are satisfied
Finite Coloring Problem

Input:
- \( C \): a set of colors
- \( H \) and \( V \): two binary relations over colors
  (horizontal and vertical color compatibilities)
- \( c_i \) and \( c_f \): two distinguished colors in \( C \)
  (initial and final color constraints)

Question: Are there
- naturals \( K \) and \( L \), and
- a coloring function
  \( C : \{1, \ldots, K\} \times \{1, \ldots, L\} \to C \)
such that that horizontal/vertical compatibilities and initial/final constraints are satisfied
Finite Coloring Problem

Input:  
► $C$: a set of colors
► $H$ and $V$: two binary relations over colors  
(horizontal and vertical color compatibilities)
► $c_i$ and $c_f$: two distinguished colors in $C$  
(initial and final color constraints)

Question: Are there  
► naturals $K$ and $L$, and  
► a coloring function  
$C : \{1, \ldots, K\} \times \{1, \ldots, L\} \to C$  
such that that horizontal/vertical compatibilities and initial/final constraints are satisfied
Finite Coloring Problem

**Input:**
- $C$: a set of colors
- $H$ and $V$: two binary relations over colors (horizontal and vertical color compatibilities)
- $c_i$ and $c_f$: two distinguished colors in $C$ (initial and final color constraints)

**Question:** Are there
- naturals $K$ and $L$, and
- a coloring function $C : \{1, \ldots, K\} \times \{1, \ldots, L\} \to C$ such that that horizontal/vertical compatibilities and initial/final constraints are satisfied
Overview of the proof
Overview of the proof

Every u-interval “meets” a small u-interval
\( u \rightarrow (A_x)u \)
Overview of the proof

\[
\begin{array}{cccc}
\ast & t_1^1 & t_2^1 & t_3^1 & t_4^1 \\
U & U & U & U & U \\
\ast & t_1^2 & t_2^2 & t_3^2 & t_4^2 \\
U & U & U & U & U \\
\ast & t_1^3 & t_2^3 & t_3^3 & t_4^3 \\
U & U & U & U & U \\
\end{array}
\]
Overview of the proof

$t_1^3 \quad t_2^3 \quad t_3^3 \quad t_4^3$
$t_1^2 \quad t_2^2 \quad t_3^2 \quad t_4^2$
$t_1^1 \quad t_2^1 \quad t_3^1 \quad t_4^1$

* $t_1^1$ $t_2^1$ $t_3^1$ $t_4^1$ * $t_1^2$ $t_2^2$ $t_3^2$ $t_4^2$ * $t_1^3$ $t_2^3$ $t_3^3$ $t_4^3$

$\times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \times \quad \time...
Overview of the proof

it is easy to give a length *upper bound*

\[ \langle A_x \rangle u \]
Overview of the proof

*lower bound* is trickier:

1 there is $u_{aux}$-interval starting at distance $x$ from beginning of $u$-interval

$$\langle A \rangle u \rightarrow [A_x] \langle A \rangle u_{aux}$$
Overview of the proof

*lower bound* is trickier:

2 no small interval “meets” a $u$-interval while starting with a $u_{aux}$-interval

$$[G_x] \neg (\langle A \rangle u \land \langle \overline{A} \rangle \langle A \rangle u_{aux})$$
Overview of the proof

*lower bound* is trickier:

2. no small interval “meets” a u-interval while starting with a \( u_{aux} \)-interval

\[
[G_x] \neg (\langle A \rangle u \land \langle \overline{A} \rangle \langle A \rangle u_{aux})
\]
SAT is undecidable for PROMPT-PNL

Theorem.

The satisfiability problem for the future fragment of PROMPT-PNL is undecidable
Outline

Introduction

The logic PROMPT–PNL
(Interval) Temporal Logic and PNL
PROMPT–PNL

Undecidability

Recovering decidability

Conclusions and future work
The culprit for undecidability

▶ using bound $\times$ both in existential and universal modalities

▶ this gives the ability of expressing lower and upper bound for the length of intervals

▶ thus we can define special chains of intervals

▶ ... and we can use such special chains as a ruler to suitably encode vertical color compatibility relation
The culprit for undecidability

- using bound $x$ both in existential and universal modalities
- this gives the ability of expressing lower and upper bound for the length of intervals
- thus we can define special chains of intervals
- ... and we can use such special chains as a ruler to suitably encode vertical color compatibility relation
Recipe for decidability

1. Remove the culprit for undecidability: get PROMPT<sup>d</sup>-PNL
   - split $X$ into two sets $X_{\diamond}$ (existential modalities) and $X_{\Box}$ (universal modalities)
Recipe for decidability

1. Remove the culprit for undecidability: get $\text{PROMPT}^d\text{-PNL}$
   - split $X$ into two sets $X\Diamond$ (existential modalities) and $X\Box$ (universal modalities)

2. Realize that now prompt modalities are monotone
   - if $\langle A_x \rangle \varphi$ is true when $x$ evaluates to $k$
   - then $\langle A_x \rangle \varphi$ is true when $x$ evaluates to $k' > k$
Recipe for decidability

1. Remove the culprit for undecidability: get PROMPT$^d$-PNL
   - split $X$ into two sets $X\Diamond$ (existential modalities) and $X\Box$ (universal modalities)

2. Realize that now prompt modalities are monotone
   - if $\langle A_x \rangle \varphi$ is true when $x$ evaluates to $k$
     then $\langle A_x \rangle \varphi$ is true when $x$ evaluates to $k' > k$

3. Realize that now one can reduces to the 2-variable case
   - $X\Diamond = x, X\Box = y$ thanks to monotonicity
Recipe for decidability

1. Remove the culprit for undecidability: get $\text{PROMPT}^d\text{PNL}$
   
   - split $X$ into two sets $X\Diamond$ (existential modalities) and $X\Box$ (universal modalities)

2. Realize that now prompt modalities are monotone
   
   - if $\langle A_x \rangle \varphi$ is true when $x$ evaluates to $k$
     
     then $\langle A_x \rangle \varphi$ is true when $x$ evaluates to $k' > k$

3. Realize that now one can reduces to the 2-variable case
   
   - $X\Diamond = x, X\Box = y$ thanks to monotonicity

4. Solve finite satisfiability (look for finite domains)
   
   - trivially reduce to satisfiability for PNL (non-prompt) thanks to monotonicity
Recipe for decidability

1. Remove the culprit for undecidability: get PROMPT\textsuperscript{d}-PNL
   ▶ split $X$ into two sets $X\Diamond$ (existential modalities) and $X\Box$ (universal modalities)

2. Realize that now prompt modalities are monotone
   ▶ if $\langle A_x \rangle \varphi$ is true when $x$ evaluates to $k$
     then $\langle A_x \rangle \varphi$ is true when $x$ evaluates to $k' > k$

3. Realize that now one can reduces to the 2-variable case
   ▶ $X\Diamond = x$, $X\Box = y$ thanks to monotonicity

4. Solve finite satisfiability (look for finite domains)
   ▶ trivially reduce to satisfiability for PNL (non-prompt) thanks to monotonicity

5. Solve infinite satisfiability
Infinite satisfiability

Proof via small model theorem

- infinite model

- finite witness (a periodic model)
Infinite satisfiability

Proof via small model theorem

infinite model

finite witness (a periodic model)

left period    middle land    right period
Infinite satisfiability

Proof via small model theorem

finite witness (a periodic model)

finite model
Infinite satisfiability

Proof via small model theorem

\[ \langle A_x \rangle \psi_1 \]

\[ \langle A \rangle \psi_2 \]

left period  middle land  right period
Infinite satisfiability

Proof via small model theorem

- infinite model
- finite witness (a periodic model)
- minimal periodic model

\[ \langle A_x \rangle \psi_1 \quad \langle A \rangle \psi_2 \quad \text{left period} \quad \text{middle land} \quad \text{right period} \]

“useless” points are removed without loss of information
Infinite satisfiability

Proof via small model theorem

- infinite model
- finite witness (a periodic model)
- minimal periodic model

\[ \langle A \rangle \psi_1 \]
\[ \langle A \rangle \psi_2 \]

left period \quad middle land \quad right period

“useless” points are removed without loss of information
SAT is decidable for PROMPT\textsuperscript{d}-PNL

Theorem.
The satisfiability problem for PROMPT\textsuperscript{d}-PNL is decidable (NEXPTIME-complete)
Outline

Introduction

The logic PROMPT–PNL
(Interval) Temporal Logic and PNL
PROMPT–PNL

Undecidability

Recovering decidability

Conclusions and future work
Conclusions and future work

Conclusions

two prompt extensions of Interval Temporal Logic PNL

- full logic PROMPT-PNL is undecidable
- its syntactic restriction PROMPT\(^d\)-PNL is decidable
  (NEXPTIME-complete)

Future work

- the unrestricted two variable fragment might be expressive and decidable
- parametric extensions of PNL
  - e.g., allowing both upper and lower bound
  - comparison between PROMPT-PNL and metric PNL
Conclusions and future work

Conclusions

two prompt extensions of Interval Temporal Logic PNL

▶ full logic PROMPT–PNL is undecidable

▶ its syntactic restriction PROMPT\textsuperscript{d}–PNL is decidable
  \((\text{NEXPTIME-complete})\)

Future work

▶ which is the minimum number of variables to make PROMPT–PNL undecidable
  
  ▶ the \textit{unrestricted} two variable fragment might be expressive and decidable

▶ parametric extensions of PNL
  
  ▶ e.g., allowing both upper and lower bound

▶ comparison between PROMPT–PNL and \textit{metric} PNL
Thank you!