On First-Order Propositional Neighborhood Logics: a First Attempt

D. Della Monica\textsuperscript{1}, G. Sciavicco\textsuperscript{2}

\textsuperscript{1}University of Udine, Italy
\textsuperscript{2}Universidad de Murcia, Spain

Lisbon, 16th August - STeDY 2010
Outline

1. Introduction to Interval Temporal Logics
2. First-Order extension of Propositional Neighborhood Logics
3. Conclusions
Introduction to Interval Temporal Logics

First-Order extension of Propositional Neighborhood Logics

Conclusions
Studying time and its structure is of great importance in computer science:

- **Artificial Intelligence.**
  Planning, Natural Language Recognition, . . .

- **Databases.**
  Temporal Databases.

- **Formal methods.**
  Specification and Verification of Systems and Protocols, Model Checking, . . .
Points vs. intervals

Usually, time is formalized as a set of time points without duration.

But... this concept is extremely abstract:

- time is actually viewed as a set of intervals (periods) with a duration.

Problem

It would be nice to have expressive, yet decidable, temporal logics that take time intervals as primary objects.
The time period, instead of the time instant, is the primitive temporal entity.

Propositional letters are evaluated over pairs of points (instead of individual points).

Relations between worlds are more complicated than the point-based case.
Allen’s relations

J. F. Allen
Maintaining knowledge about temporal intervals.
*Communications of the ACM, 1983.*
Allen’s relations

J. F. Allen
Maintaining knowledge about temporal intervals.
*Communications of the ACM, 1983.*
Allen’s relations

J. F. Allen
Maintaining knowledge about temporal intervals.
*Communications of the ACM, 1983.*
Allen’s relations

J. F. Allen

Maintaining knowledge about temporal intervals.

*Communications of the ACM, 1983.*

- later
- after/meets
- overlaps
Allen’s relations

J. F. Allen
Maintaining knowledge about temporal intervals.
*Communications of the ACM, 1983.*
Allen’s relations

J. F. Allen
Maintaining knowledge about temporal intervals.

*Communications of the ACM, 1983.*

- later
- after/meets
- overlaps
- ends/finishes
- during
Allen’s relations

J. F. Allen
Maintaining knowledge about temporal intervals.
*Communications of the ACM, 1983.*

later
after/meets
overlaps
ends/finishes
during
begins/starts
Allen’s relations

J. F. Allen
Maintaining knowledge about temporal intervals.
*Communications of the ACM, 1983.*
Allen’s relations

J. F. Allen
Maintaining knowledge about temporal intervals.
*Communications of the ACM, 1983.*

- *later*  \(\langle L \rangle \langle \overline{L} \rangle\)
- *after/meets*  \(\langle A \rangle \langle A \rangle\)
- *overlaps*  \(\langle O \rangle \langle \overline{O} \rangle\)
- *ends/finishes*  \(\langle E \rangle \langle \overline{E} \rangle\)
- *during*  \(\langle D \rangle \langle \overline{D} \rangle\)
- *begins/starts*  \(\langle B \rangle \langle \overline{B} \rangle\)
Allen’s relations

J. F. Allen
Maintaining knowledge about temporal intervals.
*Communications of the ACM, 1983.*

Between intervals: 13 relations
Between points: 3 relations

later: ⟨L⟩ ⟨L⟩

after/meets: ⟨A⟩ ⟨A⟩

overlaps: ⟨O⟩ ⟨O⟩

ends/finishes: ⟨E⟩ ⟨E⟩

during: ⟨D⟩ ⟨D⟩

begins/starts: ⟨B⟩ ⟨B⟩
First discouraging undecidability results

**HS is undecidable**

J. Halpern and Y. Shoham

A propositional modal interval logic.

First discouraging undecidability results

HS is undecidable

J. Halpern and Y. Shoham
A propositional modal interval logic.

Undecidability of a small fragment of HS: BE

K. Lodaya
Sharpening the Undecidability of Interval Temporal Logic.
Some decidable fragments

- **RPNL (A)**

D. Bresolin, A. Montanari, and G. Sciavicco
An optimal decision procedure for Right Propositional Neighborhood Logic.
*Journal of Automated Reasoning, 2007.*
Some decidable fragments

- **RPNL** \((A)\)
- **PNL** \((A\overline{A})\)

D. Bresolin, A. Montanari, and P. Sala

An optimal tableau-based decision algorithm for Propositional Neighborhood Logic.

Outline

1. Introduction to Interval Temporal Logics
2. First-Order extension of Propositional Neighborhood Logics
3. Conclusions
Extending PNL
Extending PNL

NEXPTIME-co

PNL
Extending PNL

PNL + any HS operator

NEXPTIME-co

PNL
Extending PNL

PNL + any HS operator

PNL

NEXPTIME-co

Undecidable

On First-Order PNL: a First Attempt
Extending PNL

PNL

PNL +
any HS operator

PNL +
any HS operator

Undecidable

NEXPTIME-co

PNL

Metric

PNL

D. Della Monica and G. Sciavicco

On First-Order PNL: a First Attempt
Extending PNL

PNL + any HS operator

- NEXPTIME-co
- Undecidable

between EXPSPACE and 2NEXPTIME
Extending PNL

PNL + any HS operator

PNL

NEXPTIME–co

Undecidable

Metric

PNL

between EXPSPACE and 2NEXPTIME

MPNL⁺
Extending PNL

PNL + any HS operator

Undecidable

NEXPTIME-co

between EXPSPACE and 2NEXPTIME

Undecidable

D. Della Monica and G. Sciavicco

On First-Order PNL: a First Attempt
Extending PNL

PNL + any HS operator

PNL +

NEXPTIME-co

between EXPSPACE and 2NEXPTIME

MPNL+

Undecidable

Hybrid Extensions

Metric

PNL

Undecidable

D. Della Monica and G. Sciavicco
On First-Order PNL: a First Attempt
Extending PNL

PNL + any HS operator

PNL

PNL +

NEXPTIME-co

between EXPSPACE and 2NEXPTIME

Hybrid Extensions

Metric PNL

MPNL^+

Undecidable

Undecidable

Undecidable
Extending PNL

- PNL + any HS operator
- NEXPTIME-co
- First-Order Extensions
  - Undecidable
- Hybrid Extensions
  - Undecidable
  - MPNL+
  - Undecidable

between EXPSPACE and 2NEXPTIME

D. Della Monica and G. Sciavicco
Extending PNL

PNL + any HS operator

Undecidable

First-Order Extensions

PNL

Undecidable

Hybrid Extensions

Metric PNL

between EXPSPACE and 2NEXPTIME

Undecidable

MPNL^+

Undecidable

D. Della Monica and G. Sciavicco

On First-Order PNL: a First Attempt
First-Order together with Propositional

F O R P N L

First-Order Right Propositional Neighborhood Logic

1. Propositional (modal) setting
2. First-Order setting
   - predicates over elements
   - existential and universal quantifications
3. Propositional (modal) + First-Order setting
First-Order together with Propositional

FORPNL

First-Order

Propositional

Right

First-Order setting

Propositional (modal) setting

First-Order setting

Predicates over elements

Existential and universal quantifications

Propositional (modal) +

First-Order setting
First-Order together with Propositional

1. Propositional (modal) setting
2. First-Order setting
   - predicates over elements
   - existential and universal quantifications
3. Propositional (modal) + First-Order setting

D. Della Monica and G. Sciavicco
On First-Order PNL: a First Attempt
First-Order together with Propositional

**F O R P N L**

1. Propositional (modal) setting
2. First-Order setting
   - predicates over elements
   - existential and universal quantifications
3. Propositional (modal) + First-Order setting
Introduction to ITLs
First-Order extension of Propositional Neighborhood Logics
Conclusions

First-Order together with Propositional

**F O R P N L**

1. **Propositional (modal) setting**
2. **First-Order setting**
   - predicates over elements
   - existential and universal quantifications
3. **Propositional (modal) + First-Order setting**

D. Della Monica and G. Sciavicco
On First-Order PNL: a First Attempt
First-Order together with Propositional

F O R P N L

First-Order Right Propositional Neighborhood Logic

1. Propositional (modal) setting
2. First-Order setting
   - predicates over elements
   - existential and universal quantifications
3. Propositional (modal) + First-Order setting
First-Order together with Propositional

**F O R P N L**

- **First-Order Right Propositional Neighborhood Logic**
  - Propositional (modal) setting
  - First-Order setting
    - predicates over elements
    - existential and universal quantifications
  - Propositional (modal) + First-Order setting

D. Della Monica and G. Sciavicco
On First-Order PNL: a First Attempt
First-Order together with Propositional

F O R P N L

First-Order Right Propositional Neighborhood Logic

1. Propositional (modal) setting
2. First-Order setting
   - predicates over elements
   - existential and universal quantifications
3. Propositional (modal) + First-Order setting
First-Order together with Propositional

F O R P N L

1. Propositional (modal) setting
2. First-Order setting
   - predicates over elements
   - existential and universal quantifications
3. Propositional (modal) + First-Order setting
Parameters of the logic

- Temporal domain: discrete, dense, finite, bounded, unbounded, ...
- First-order domain: finite, infinite, expanding, ...
- First-order constructs:
  - predicates $P(...), Q(...), ...$
  - individual variables $x, y, ...$
  - individual constants $a, b, ...$
  - function $f(...), g(...), ...$
  - quantifiers
  - terms $t_1, t_2, ...$ (variables, constants, and functions)
Introduction to ITLs
First-Order extension of Propositional Neighborhood Logics
Conclusions

Parameters of the logic

- Temporal domain: discrete, dense, finite, bounded, unbounded, …
- First-order domain: finite, infinite, expanding, …
- First-order constructs:
  - predicates \( P(\ldots), Q(\ldots), \ldots \)
  - individual variables \( x, y, \ldots \)
  - individual constants \( a, b, \ldots \)
  - function \( f(\ldots), g(\ldots), \ldots \)
  - quantifiers
  - terms \( t_1, t_2, \ldots \) (variables, constants, and functions)
Parameters of the logic

- Temporal domain: discrete, dense, finite, bounded, unbounded, ...
- First-order domain: finite, infinite, expanding, ...
- First-order constructs:
  - predicates $P(...), Q(...), ...$
  - individual variables $x, y, ...$
  - individual constants $a, b, ...$
  - function $f(...), g(...), ...$
  - quantifiers
  - terms $t_1, t_2, ...$ (variables, constants, and functions)
Parameters of the logic

- Temporal domain: discrete, dense, finite, bounded, unbounded, ... 
- First-order domain: finite, infinite, expanding, ... 
- First-order constructs:
  - predicates $P(...), Q(...), ...$
  - individual variables $x, y, ...$
  - individual constants $a, b, ...$
  - function $f(...), g(...), ...$
  - quantifiers
  - terms $t_1, t_2, ...$ (variables, constants, and functions)
Parameters of the logic

- Temporal domain: discrete, dense, **finite**, bounded, unbounded, . . .
- First-order domain: **finite**, infinite, expanding, . . .
- First-order constructs:
  - predicates $P(...), Q(...), . . .$
  - individual variables $x, y, . . .$
  - individual constants $a, b, . . .$
  - function $f(...), g(...), . . .$
  - quantifiers
  - terms $t_1, t_2, . . .$ (variables, constants, and functions)
Parameters of the logic

- Temporal domain: discrete, dense, finite, bounded, unbounded, ...
- First-order domain: finite, infinite, expanding, ...
- First-order constructs:
  - predicates $P(...), Q(...), ...$
  - individual variables $x, y, ...$
  - individual constants $a, b, ...$
  - function $f(...), g(...), ...$
  - quantifiers
  - terms $t_1, t_2, ...$ (variables, constants, and functions)
    terms = variables
Parameters of the logic

- Temporal domain: discrete, dense, \textit{finite}, bounded, unbounded, \ldots
- First-order domain: \textit{finite}, infinite, expanding, \ldots
- First-order constructs:
  - predicates $P(\ldots), Q(\ldots), \ldots$
  - individual variables $x, y, \ldots$
  - individual constants $a, b, \ldots$
  - function $f(\ldots), g(\ldots), \ldots$
  - quantifiers
  - terms $t_1, t_2, \ldots$ (variables, constants, and functions)

**for tight undecidability only 1 variable (no free variables)**
RPNL and FORPNL: syntax and semantics

Syntax

RPNL: \( \varphi ::= \pi \mid p \mid \neg \varphi \mid \varphi \lor \varphi \mid \langle A \rangle \varphi \)
RPNL and FORPNL: syntax and semantics

**Syntax**

RPNL: $\varphi ::= \pi \mid p \mid \neg \varphi \mid \varphi \lor \varphi \mid \langle A \rangle \varphi$

**Semantics**

Operators \textit{meets} ($\langle A \rangle$):

$\langle A \rangle \varphi$
RPNL and FORPNL: syntax and semantics

Syntax

- RPNL: \( \varphi ::= \pi \mid p \mid \neg \varphi \mid \varphi \lor \varphi \mid \langle A \rangle \varphi \)
RPNL and FORPNL: syntax and semantics

Syntax

RPNL: \[ \varphi ::= \pi \mid p \mid \neg \varphi \mid \varphi \lor \varphi \mid \langle A \rangle \varphi \]

FORPNL: \[ \mid P(x) \mid \forall x \varphi(x) \]

Semantics

Operators *meets* \( \langle A \rangle \):
Reduction from the Finite Tiling Problem

This is the problem of establishing whether, for a given finite set of tile types $T = \{t_1, \ldots, t_k\}$, there exists a finite rectangle $R$ having the border colored with a fixed color such that $T$ can tile $R$ respecting the color constraints.
Reduction from the Finite Tiling Problem

This is the problem of establishing whether, for a given finite set of tile types $\mathcal{T} = \{t_1, \ldots, t_k\}$, there exists a finite rectangle $\mathcal{R}$ having the border colored with a fixed color such that $\mathcal{T}$ can tile $\mathcal{R}$ respecting the color constraints.
The core of the proof

It is possible to *simulate* HS operators \([B] [E] [D]\)

Put a label over the first-order domain for each point of the temporal domain.
The core of the proof

It is possible to *simulate* HS operators \([B] [E] [D]\)

Put a label over the first-order domain for each point of the temporal domain
The core of the proof

It is possible to simulate HS operators [B] [E] [D]

Put a label over the first-order domain for each point of the temporal domain

\[
\Box \Box (\exists x \Diamond P(x) \land \forall x (\Diamond P(x) \to \Box (\neg \pi \to \Box \neg P(x))))
\]
The core of the proof

It is possible to simulate HS operators [B] [E] [D]

Put a label over the first-order domain for each point of the temporal domain

\[
\Box\Box(\exists x \Diamond P(x) \land \forall x (\Diamond P(x) \rightarrow \Box (\neg \pi \rightarrow \Box \neg P(x))))
\]
The core of the proof

It is possible to simulate HS operators \([B] [E] [D]\)

Put a label over the first-order domain for each point of the temporal domain

\[
\square \square (\exists x \Box P(x) \land \forall x(\Box P(x) \rightarrow \square (\neg \pi \rightarrow \square \neg P(x))))
\]

\[
\exists x P(x)
\]
The core of the proof

It is possible to *simulate* HS operators \([B] [E] [D]\)

Put a label over the first-order domain for each point of the temporal domain

\[
□□(∃x ◊ P(x) \land ∀x(◊ P(x) \rightarrow □(¬π \rightarrow □¬P(x))))
\]

\[
P(d_1)
\]
The core of the proof

It is possible to simulate HS operators $[B] [E] [D]$.

Put a label over the first-order domain for each point of the temporal domain

$$\Box \Box (\exists x \Diamond P(x) \land \forall x (\Diamond P(x) \rightarrow \Box (\neg \pi \rightarrow \Box \neg P(x))))$$

$$d_1$$

$$P(d_1)$$
The core of the proof

It is possible to *simulate* HS operators [B] [E] [D]

Put a label over the first-order domain for each point of the temporal domain

\[
\Box\Box (\exists x \Diamond P(x) \land \forall x (\Diamond P(x) \rightarrow \Box (\neg \pi \rightarrow \Box \neg P(x))))
\]

\[d_1\]

\[P(d_1)\]
The core of the proof

It is possible to *simulate* HS operators \([B] [E] [D]\)

Put a label over the first-order domain for each point of the temporal domain

\[
\square\square(\exists x \lozenge P(x) \land \forall x (\lozenge P(x) \rightarrow \square(\neg \pi \rightarrow \square \neg P(x))))
\]

\[\begin{array}{c}
\hline
P(d_1) \\
\end{array}\]

\[
d_1 \quad d_1
\]
The core of the proof

Put a label over the first-order domain for each point of the temporal domain

\[
\Box \Box (\exists x \Diamond P(x) \land \forall x (\Diamond P(x) \rightarrow \Box (\neg \pi \rightarrow \Box \neg P(x))))
\]
The core of the proof

It is possible to simulate HS operators [B] [E] [D]

Put a label over the first-order domain for each point of the temporal domain

\[
\Box\Box (\exists x \Diamond P(x) \land \forall x (\Diamond P(x) \rightarrow \Box (\neg \pi \rightarrow \Box \neg P(x))))
\]

D. Della Monica and G. Sciavicco
On First-Order PNL: a First Attempt
The core of the proof

It is possible to simulate HS operators \([B] [E] [D]\)

Put a label over the first-order domain for each point of the temporal domain

\[\square \square (\exists x \Diamond P(x) \land \forall x (\Diamond P(x) \rightarrow \square (\neg \pi \rightarrow \square \neg P(x))))\]
It is possible to simulate HS operators [B] [E] [D]

Put a label over the first-order domain for each point of the temporal domain

$$\square\Box(\exists x \Diamond P(x) \land \forall x (\Diamond P(x) \rightarrow \Box (\neg \pi \rightarrow \Box \neg P(x))))$$
The core of the proof

It is possible to simulate HS operators [B] [E] [D]

1. Put a label over the first-order domain for each point of the temporal domain
2. Say “for every interval, if $\varphi$ holds then every starting interval satisfies $\psi$” (i.e., $\Box\Box(\varphi \rightarrow [B]\psi)$)
The core of the proof

It is possible to simulate HS operators [B] [E] [D]

1. Put a label over the first-order domain for each point of the temporal domain

2. Say “for every interval, if $\varphi$ holds then every starting interval satisfies $\psi$” (i.e., $\Box\Box(\varphi \rightarrow [B]\psi)$)

$$\Box\Box\forall x(\Diamond(\varphi \land \Diamond P(x)) \rightarrow \Box(\Diamond(\neg \pi \land \Diamond P(x)) \rightarrow \psi))$$
The core of the proof

It is possible to simulate HS operators \([B] [E] [D]\)

1. Put a label over the first-order domain for each point of the temporal domain
2. Say “for every interval, if \(\varphi\) holds then every starting interval satisfies \(\psi\)” (i.e., \(\Box\Box (\varphi \rightarrow [B]\psi))\)
   
   \[
   \Box\Box \forall x (\Diamond (\varphi \land \Diamond P(x)) \rightarrow \Box (\Diamond (\neg \pi \land \Diamond P(x)) \rightarrow \psi))
   \]

\(\varphi\)
The core of the proof

It is possible to simulate HS operators $[B] [E] [D]$

1. Put a label over the first-order domain for each point of the temporal domain

2. Say “for every interval, if $\varphi$ holds then every starting interval satisfies $\psi$” (i.e., $\Box \Box (\varphi \rightarrow [B] \psi)$)

$$\Box \Box \forall x (\Diamond (\varphi \land \Diamond P(x)) \rightarrow \Box (\Diamond (\neg \pi \land \Diamond P(x)) \rightarrow \psi))$$

\[
\begin{align*}
\varphi & \quad \downarrow d_1 \\
P(d_1) & \quad \downarrow
\end{align*}
\]
The core of the proof

It is possible to simulate HS operators \([B] [E] [D]\)

1. Put a label over the first-order domain for each point of the temporal domain
2. Say “for every interval, if \(\varphi\) holds then every starting interval satisfies \(\psi\)” (i.e., \(\Box\Box(\varphi \rightarrow [B]\psi)\))

\[\Box\Box\forall x(\Diamond(\varphi \land \Diamond P(x)) \rightarrow \Box(\Diamond(\neg \pi \land \Diamond P(x)) \rightarrow \psi))\]

\[\varphi \quad d_1\]

\[P(d_1)\]
The core of the proof

It is possible to simulate HS operators [B] [E] [D]

1. Put a label over the first-order domain for each point of the temporal domain
2. Say “for every interval, if \( \varphi \) holds then every starting interval satisfies \( \psi \)” (i.e., \( \Box \Box (\varphi \rightarrow [B] \psi) \))

\[
\Box \Box \forall x (\Diamond (\varphi \land \Diamond P(x)) \rightarrow \Box (\Diamond (\neg \pi \land \Diamond P(x)) \rightarrow \psi))
\]
The core of the proof

It is possible to simulate HS operators [B] [E] [D]

1. Put a label over the first-order domain for each point of the temporal domain

2. Say "for every interval, if \( \varphi \) holds then every starting interval satisfies \( \psi \)" (i.e., \( \Box \Box (\varphi \rightarrow [B] \psi) \))

\[ \Box \Box \forall x(\Diamond (\varphi \land \Diamond P(x)) \rightarrow \Box (\Diamond (\neg \pi \land \Diamond P(x)) \rightarrow \psi)) \]

D. Della Monica and G. Sciavicco

On First-Order PNL: a First Attempt
The core of the proof

It is possible to *simulate* HS operators [B] [E] [D]

1. Put a label over the first-order domain for each point of the temporal domain
2. Say “for every interval, if \( \varphi \) holds then every starting interval satisfies \( \psi \)” (i.e., \( \Box \Box (\varphi \rightarrow [B] \psi) \))

\[
\Box \Box \forall x (\Diamond (\varphi \land \Diamond P(x)) \rightarrow \Box (\Diamond (\neg \pi \land \Diamond P(x)) \rightarrow \psi))
\]
The core of the proof

It is possible to simulate HS operators \([B] \ [E] \ [D]\)

1. Put a label over the first-order domain for each point of the temporal domain
2. Say “for every interval, if \(\varphi\) holds then every starting interval satisfies \(\psi\)” (i.e., \(\square\square (\varphi \rightarrow [B]\psi)\))
3. Say “for every interval, if \(\varphi\) holds then every ending interval satisfies \(\psi\)” (i.e., \(\square\square (\varphi \rightarrow [E]\psi)\))
4. Say “for every interval, if \(\varphi\) holds then every sub-interval satisfies \(\psi\)” (i.e., \(\square\square (\varphi \rightarrow [D]\psi)\))
The core of the proof

It is possible to simulate HS operators $[B]$ $[E]$ $[D]$

1. Put a label over the first-order domain for each point of the temporal domain
2. Say “for every interval, if $\varphi$ holds then every starting interval satisfies $\psi$” (i.e., $\square \square (\varphi \rightarrow [B] \psi)$)
3. Say “for every interval, if $\varphi$ holds then every ending interval satisfies $\psi$” (i.e., $\square \square (\varphi \rightarrow [E] \psi)$)
4. Say “for every interval, if $\varphi$ holds then every sub-interval satisfies $\psi$” (i.e., $\square \square (\varphi \rightarrow [D] \psi)$)
The core of the proof

It is possible to *simulate* HS operators \([B] [E] [D]\)

1. Put a label over the first-order domain for each point of the temporal domain
2. Say “for every interval, if \(\varphi\) holds then every starting interval satisfies \(\psi\)” (i.e., \(\Box\Box (\varphi \rightarrow [B] \psi)\))
3. Say “for every interval, if \(\varphi\) holds then every ending interval satisfies \(\psi\)” (i.e., \(\Box\Box (\varphi \rightarrow [E] \psi)\))
4. Say “for every interval, if \(\varphi\) holds then every sub-interval satisfies \(\psi\)” (i.e., \(\Box\Box (\varphi \rightarrow [D] \psi)\))

\[
[B\psi] \equiv \Box\Box (\varphi \rightarrow [B] \psi)
\]

\[
[E\psi] \equiv \Box\Box (\varphi \rightarrow [E] \psi)
\]

\[
[D\psi] \equiv \Box\Box (\varphi \rightarrow [D] \psi)
\]
Proof overview

1. Encoding the rectangle
2. Encoding the neighbourhood relations

u u u u
u u u u
u u u u
u u u u
u u u u
u u u u
u u u u
u u u u
u u u u
Proof overview

1. Encoding the rectangle
2. Encoding the neighbourhood relations
Proof overview

1. Encoding the rectangle
2. Encoding the neighbourhood relations

\[ \diamondsuit u \]
\[ \Box \Box (u \rightarrow \neg \pi) \]
\[ \Box \Box (u \rightarrow (\diamondsuit u \lor \Box \pi)) \]
Proof overview

1. Encoding the rectangle
2. Encoding the neighbourhood relations

<table>
<thead>
<tr>
<th>u</th>
<th>u</th>
<th>u</th>
<th>u</th>
</tr>
</thead>
<tbody>
<tr>
<td>u</td>
<td>u</td>
<td>u</td>
<td>u</td>
</tr>
<tr>
<td>u</td>
<td>u</td>
<td>u</td>
<td>u</td>
</tr>
<tr>
<td>u</td>
<td>u</td>
<td>u</td>
<td>u</td>
</tr>
<tr>
<td>u</td>
<td>u</td>
<td>u</td>
<td>u</td>
</tr>
</tbody>
</table>

\[ \text{\#u} \]
\[ \Box\Box (u \to \neg \pi) \]
\[ \Box\Box (u \to (\#u \lor \Box\pi)) \]
\[ [B^u_{\neg u}] \land [B^u_{\neg \pi \to \neg \#u}] \]
Proof overview

1. Encoding the rectangle
2. Encoding the neighbourhood relations
Proof overview

1. Encoding the rectangle
2. Encoding the neighbourhood relations
Proof overview

1. Encoding the rectangle
2. Encoding the neighbourhood relations
Proof overview

1. Encoding the rectangle
2. Encoding the neighbourhood relations
Proof overview

1. Encoding the rectangle
2. Encoding the neighbourhood relations
Proof overview

1. Encoding the rectangle
2. Encoding the neighbourhood relations
Proof overview

1. Encoding the rectangle
2. Encoding the neighbourhood relations
Proof overview

1. Encoding the rectangle
2. Encoding the neighbourhood relations
Proof overview

1. Encoding the rectangle
2. Encoding the neighbourhood relations

\[ B_{\text{up\_rel}} \land E_{\text{up\_rel}} \land D_{\text{up\_rel}} \]
Proof overview

1. Encoding the rectangle
2. Encoding the neighbourhood relations

\[ [B_{\text{up\_rel}}] \land [E_{\text{up\_rel}}] \land [D_{\text{up\_rel}}] \land [B_{\text{Id}}] \land [E_{\text{Id}}] \land [D_{\text{Id}}] \]
Proof overview

1. Encoding the rectangle
2. Encoding the neighbourhood relations

\[ B_{up\_rel} \wedge E_{up\_rel} \wedge D_{up\_rel} \]
\[ B_{up\_rel} \wedge E_{up\_rel} \wedge D_{up\_rel} \]
\[ B_{Id} \wedge E_{Id} \wedge D_{Id} \]
\[ D_{up\_rel} \]
Outline

1. Introduction to Interval Temporal Logics
2. First-Order extension of Propositional Neighborhood Logics
3. Conclusions
Conclusions and Final remarks

- First-order logics
- Interval temporal logics
- Two-variable first-order logic
- Temporal logics (point-based)
- LTL
- (R)PNL

D. Della Monica and G. Sciavicco

On First-Order PNL: a First Attempt
Conclusions and Final remarks

- First-order logics
  - Two-variable first-order logic
    - Hodkinson et al, 2000
    - FO-LTL with 2 vars

- Temporal logics (point-based)
- Interval temporal logics
- LTL
- (R)PNL

D. Della Monica and G. Sciavicco
On First-Order PNL: a First Attempt
Conclusions and Final remarks

- First-order logics
- Two-variable first-order logic
- LTL
- (R)PNL
- Temporal logics (point-based)
- Interval temporal logics

Hodkinson et al, 2000

FO-LTL with 2 vars
FO-LTL with 1 vars
Conclusions and Final remarks

First-order logics

Hodkinson et al, 2000

Two-variable first-order logic

FO-LTL with 2 vars
FO-LTL with 1 vars

FORPNL

LTL

(R)PNL

Temporal logics (point-based)

Interval temporal logics

D. Della Monica and G. Sciavicco

On First-Order PNL: a First Attempt
Future work

PNL

First-Order Extensions

PNL + any HS operator

Hybrid Extensions

Metric PNL

Undecidable

Undecidable

Undecidable

Undecidable

between EXPSpace and 2NEXPTIME

D. Della Monica and G. Sciavicco

On First-Order PNL: a First Attempt
Future work

PNL + any HS operator

PNL

PNL +

First-Order Extensions

MPNL+

Hybrid Extensions

Metric

Undecidable

Undecidable

Undecidable

Undecidable

Undecidable

between EXPSPACE and 2NEXPTIME

NEXPTIME-co
Future work

PNL + any HS operator

PNL

First-Order Extensions

FORPNL without π

Hybrid Extensions

Metric PNL

Undecidable

Undecidable

Undecidable

Undecidable

Undecidable

between EXPSPACE and 2NEXPTIME

PNL

NEXPTIME-co

D. Della Monica and G. Sciavicco

On First-Order PNL: a First Attempt