On First-Order Propositional Neighborhood Logics: a First Attempt

D. Della Monica[1], G. Sciavicco[2]

[1]University of Udine, Italy

[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

Outline

1 Introduction to Interval Temporal Logics

2 First-Order extension of Propositional Neighborhood Logics

3 Conclusions

Time and logics

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

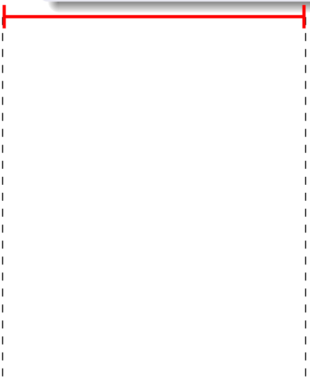
It would be nice to have expressive, yet decidable, ***temporal logics*** *that take time intervals as primary objects.*

Interval Temporal Logics

- 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 complicate than the point-based case

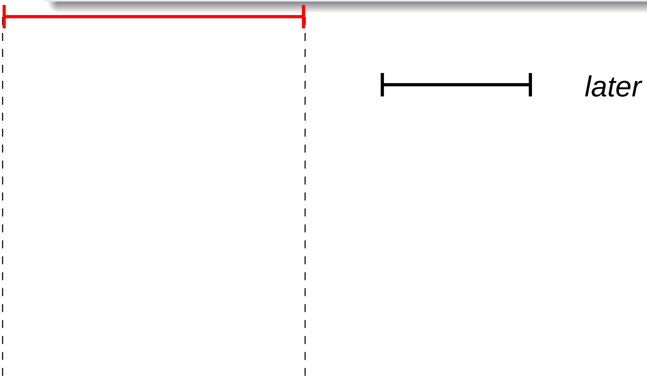
Allen's relations

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



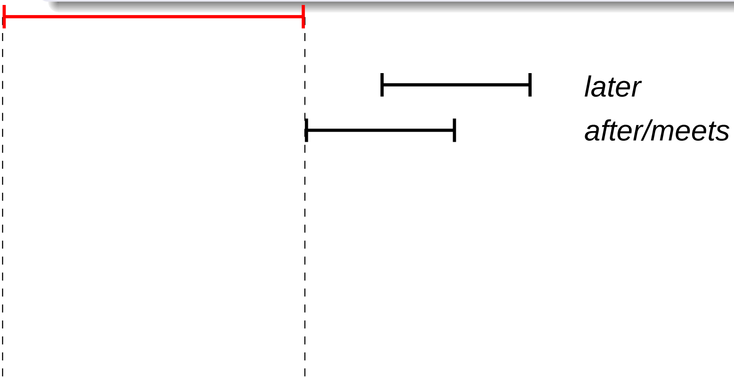
Allen's relations

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



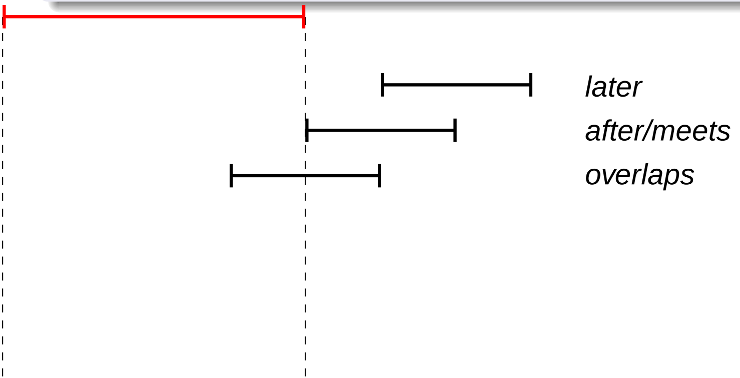
Allen's relations

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



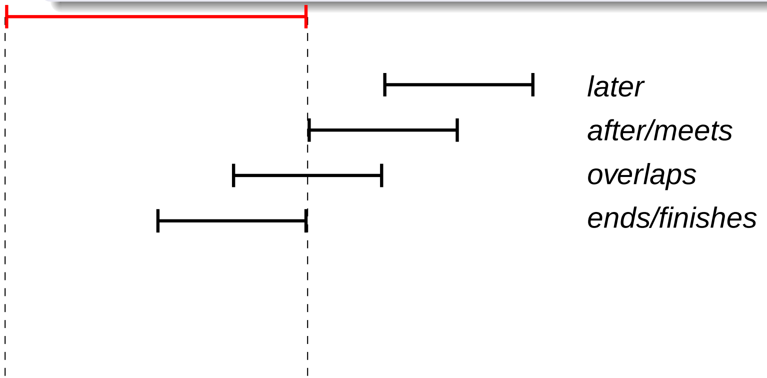
Allen's relations

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



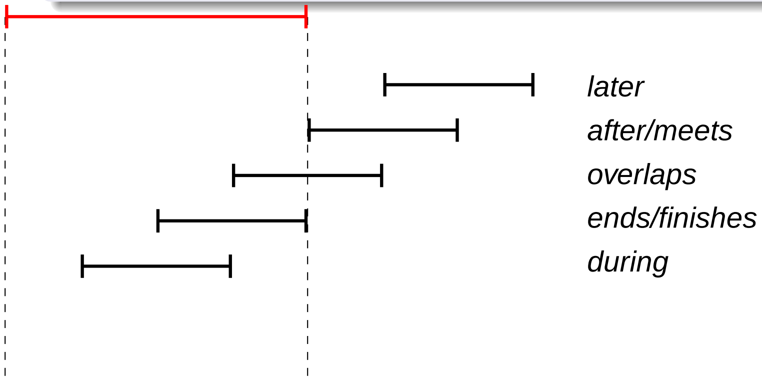
Allen's relations

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



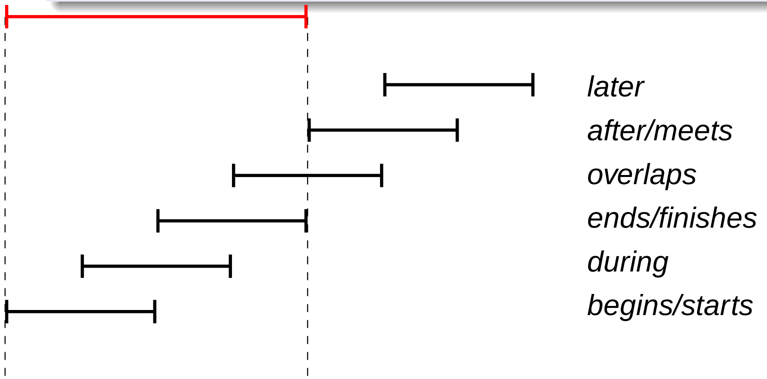
Allen's relations

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



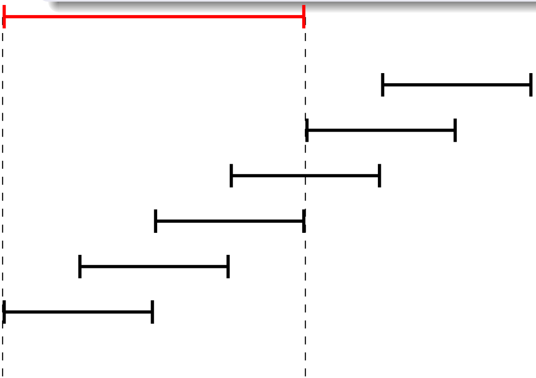
Allen's relations

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



Allen’s relations

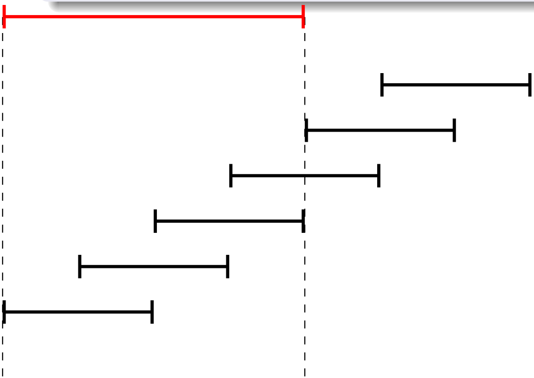
J. F. Allen
Maintaining knowledge about temporal intervals.
Communications of the ACM, 1983.



later	$\langle L\rangle$
after/meets	$\langle A\rangle$
overlaps	$\langle O\rangle$
ends/finishes	$\langle E\rangle$
during	$\langle D\rangle$
begins/starts	$\langle B\rangle$

Allen's relations

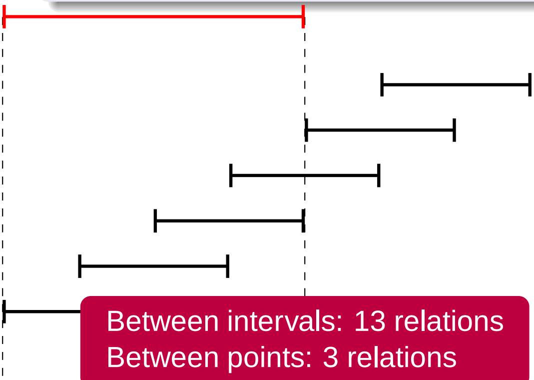
J. F. Allen
Maintaining knowledge about temporal intervals.
Communications of the ACM, 1983.



later	$\langle L \rangle$ $\langle \overline{L} \rangle$
after/meets	$\langle A \rangle$ $\langle \overline{A} \rangle$
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.



later	$\langle L \rangle$ $\langle \overline{L} \rangle$
after/meets	$\langle A \rangle$ $\langle \overline{A} \rangle$
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	$\langle B \rangle$ $\langle \overline{B} \rangle$

Between intervals: 13 relations
Between points: 3 relations

First discouraging undecidability results

HS is undecidable

J. Halpern and Y. Shoham
A propositional modal interval logic.
Journal of the ACM, 1991.

First discouraging undecidability results

HS is undecidable

J. Halpern and Y. Shoham
A propositional modal interval logic.
Journal of the ACM, 1991.

Undecidability of a small fragment of HS: BE

K. Lodaya
Sharpening the Undecidability of Interval Temporal Logic.
ASIAN 2000, volume 1961 of LNCS, pages 290-298. Springer, 2000.

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.
STACS 2007, volume 4393 of LNCS, pages 549-560. Springer, 2007.

Outline

1 Introduction to Interval Temporal Logics

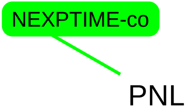
2 First-Order extension of Propositional Neighborhood Logics

3 Conclusions

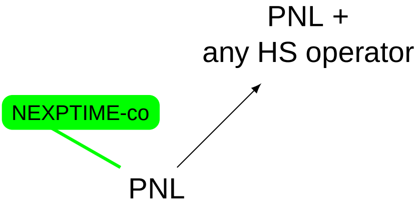
Extending PNL

PNL

Extending PNL



Extending PNL



Extending PNL

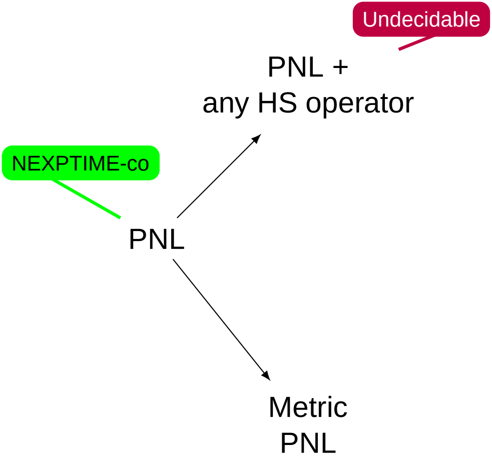
Undecidable

PNL +
any HS operator

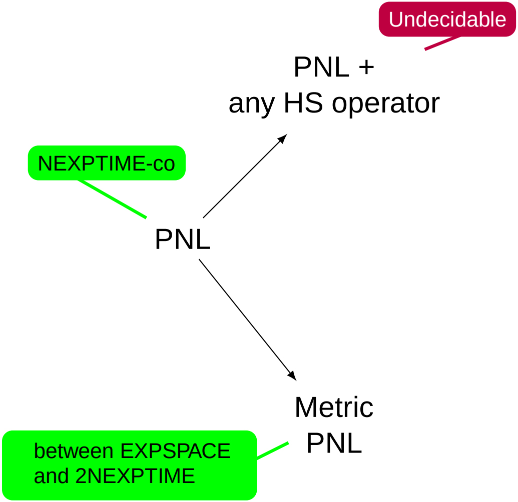
NEXPTIME-co

PNL

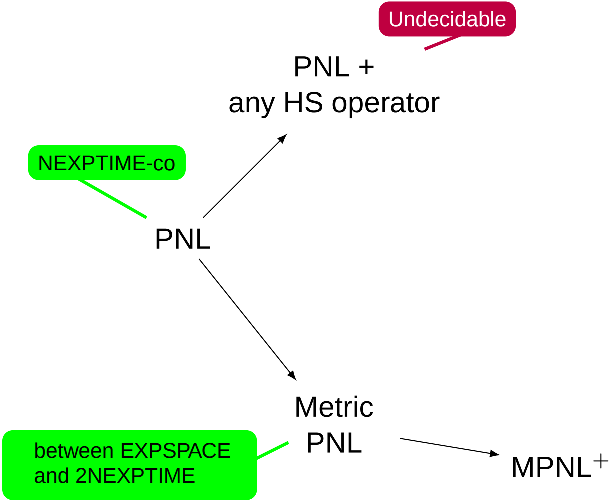
Extending PNL



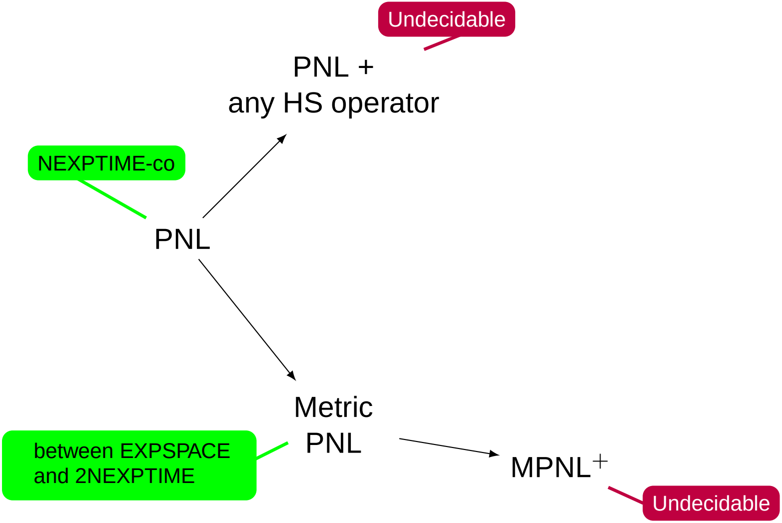
Extending PNL



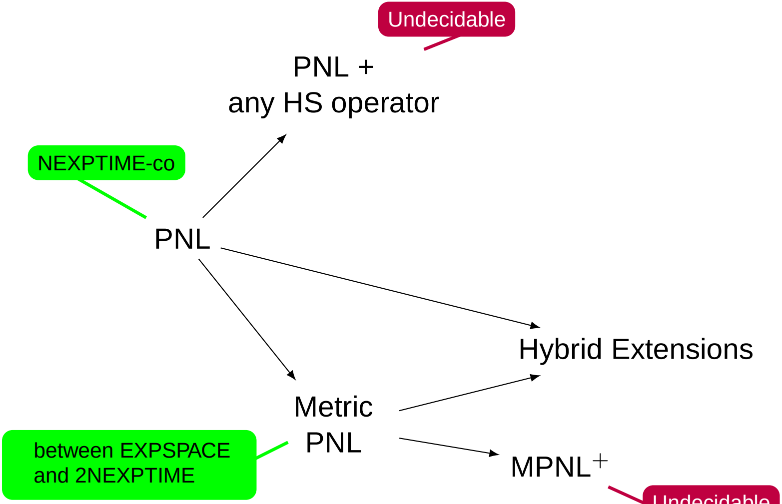
Extending PNL



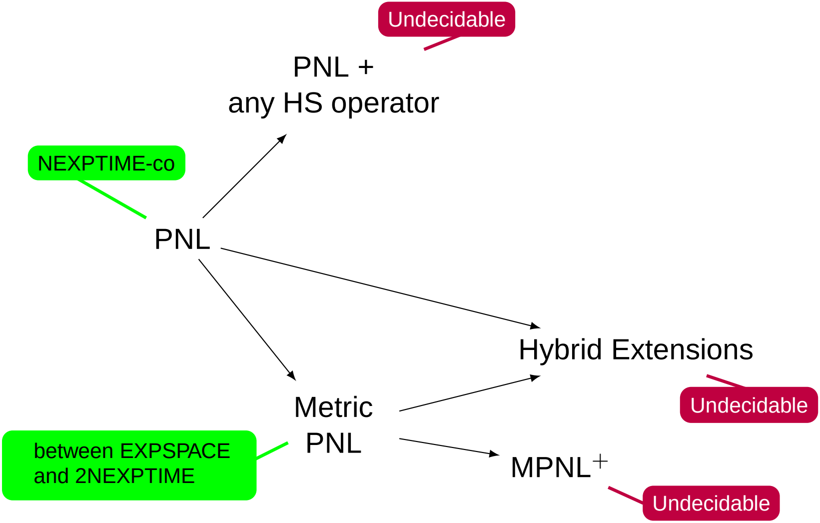
Extending PNL



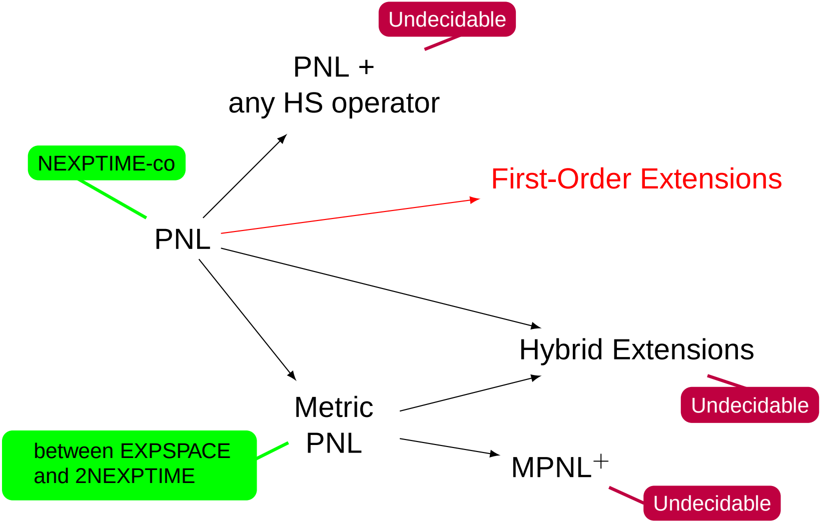
Extending PNL



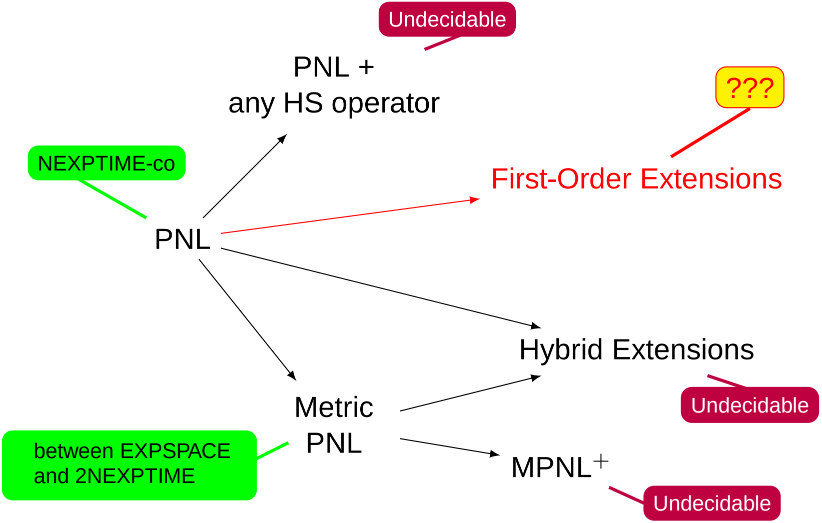
Extending PNL



Extending PNL



Extending PNL



First-Order together with Propositional

FORPNL

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

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

FOR~~P~~NL

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

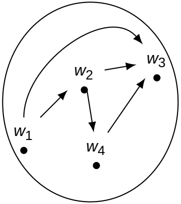
1. Propositional (modal) setting
2. First-Order setting
   - predicates over elements
   - existential and universal quantifications
3. Propositional (modal) + First-Order setting

First-Order together with Propositional

F O R P N L

First-Order Right Propositional Neighborhood Logic

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

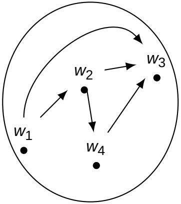


First-Order together with Propositional

FORPNL

First-Order 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

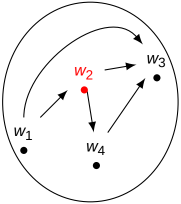


First-Order together with Propositional

F O R P N L

First-Order Right Propositional Neighborhood Logic

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

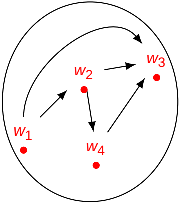


First-Order together with Propositional

F O R P N L

First-Order Right Propositional Neighborhood Logic

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

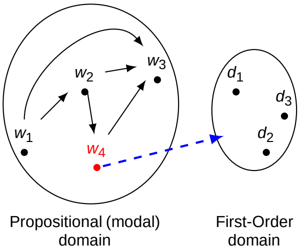


First-Order together with Propositional

F O R P N L

First-Order Right Propositional Neighborhood Logic

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 $P(\ldots), Q(\ldots), \ldots$
  - 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 $P(\ldots), Q(\ldots), \ldots$
  - 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 $P(\ldots), Q(\ldots), \ldots$
  - 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 $P(\ldots), Q(\ldots), \ldots$
  - 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 $P(\ldots), Q(\ldots), \ldots$
  - 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 $P(\ldots), Q(\ldots), \ldots$
  - 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)

    terms = variables

Parameters of the logic

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

    terms = variables

for tight undecidability only 1 variable (no free variables)

RPNL and FORPNL: syntax and semantics

Syntax

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

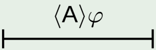
RPNL and FORPNL: syntax and semantics

Syntax

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

Semantics

- Operators *meets* ($\langle A\rangle$) :



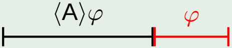
RPNL and FORPNL: syntax and semantics

Syntax

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

Semantics

- Operators *meets* ($\langle A \rangle$) :



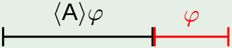
RPNL and FORPNL: syntax and semantics

Syntax

- RPNL: $\varphi ::= \pi \mid p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle A \rangle \varphi$
- FORPNL: $\mid P(x) \mid \forall x\, \varphi(x)$

Semantics

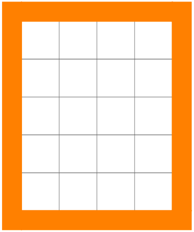
- Operators *meets* ($\langle A \rangle$) :



Undecidability of FORPNL

Reduction from the Finite Tiling Problem

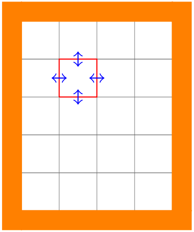
This is the problem of establishing whether, for a given finite set of tile types $\mathcal{T} = \{t_1, \ldots, t_k\}$, there exists a finite rectangle $\mathcal{R}$ having the border colored with a fixed color such that $\mathcal{T}$ can tile $\mathcal{R}$ respecting the color constraints.



Undecidability of FORPNL

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]

1. 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]

1. 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]

1. 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]

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

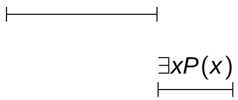
$$\Box\Box(\exists x \Diamond P(x) \wedge \forall x(\Diamond P(x) \rightarrow \Box(\neg\pi \rightarrow \Box\neg P(x))))$$

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

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

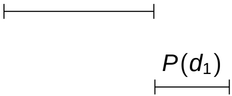


The core of the proof

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

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

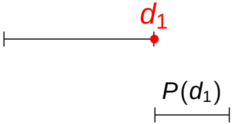


The core of the proof

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

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

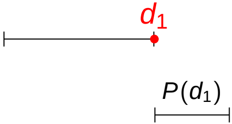


The core of the proof

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

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

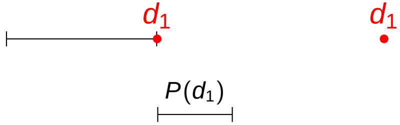


The core of the proof

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

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

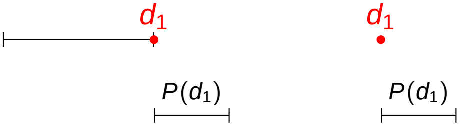


The core of the proof

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

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

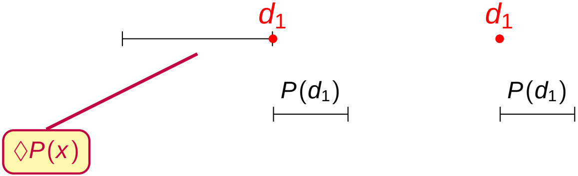


The core of the proof

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

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

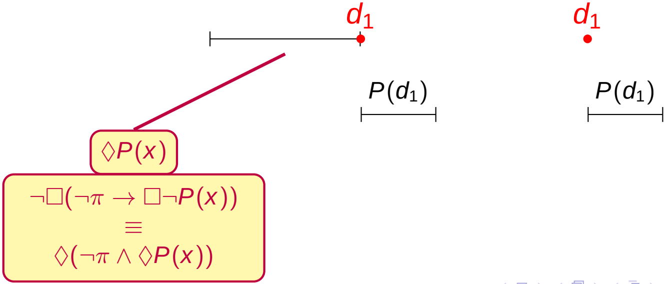


The core of the proof

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

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

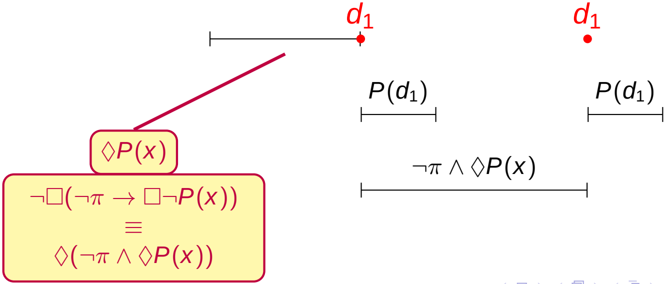


The core of the proof

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

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

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



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 φ holds then every starting interval satisfies ψ"* (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 φ holds then every starting interval satisfies ψ"* (i.e., $\Box\Box(\varphi \rightarrow [B]\psi)$)

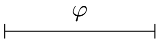
$$\Box\Box\forall x(\Diamond(\varphi \wedge \Diamond P(x)) \rightarrow \Box(\Diamond(\neg\pi \wedge \Diamond P(x)) \rightarrow \psi))$$

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 φ holds then every starting interval satisfies ψ"* (i.e., $\Box\Box(\varphi \rightarrow [B]\psi)$)

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

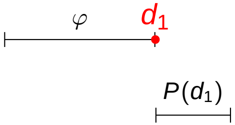


The core of the proof

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 φ holds then every starting interval satisfies ψ"* (i.e., $\Box\Box(\varphi \rightarrow [B]\psi)$)

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

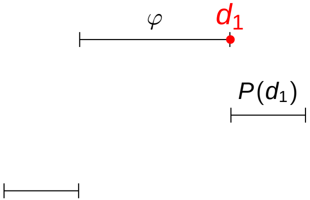


The core of the proof

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 φ holds then every starting interval satisfies ψ"* (i.e., $\Box\Box(\varphi \rightarrow [B]\psi)$)

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

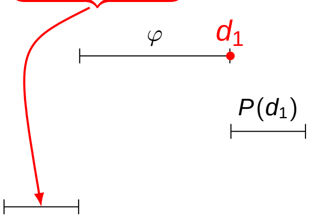


The core of the proof

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 φ holds then every starting interval satisfies ψ"* (i.e., $\Box\Box(\varphi \rightarrow [B]\psi)$)

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

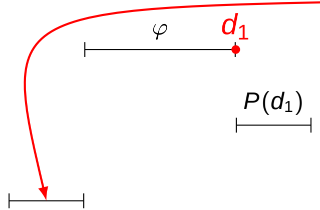


The core of the proof

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 φ holds then every starting interval satisfies ψ"* (i.e., $\Box\Box(\varphi \to [B]\psi)$)

$$\Box\Box\forall x(\Diamond(\varphi \wedge \Diamond P(x)) \to \underbrace{\Box(\Diamond(\neg\pi \wedge \Diamond P(x)) \to \psi)})$$

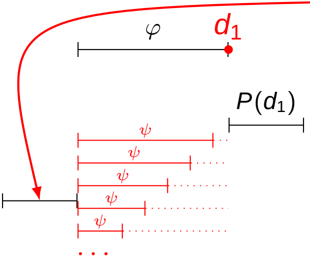


The core of the proof

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 φ holds then every starting interval satisfies ψ"* (i.e., $\Box\Box(\varphi \rightarrow [B]\psi)$)

$$\Box\Box\forall x(\Diamond(\varphi \wedge \Diamond P(x)) \rightarrow \underbrace{\Box(\Diamond(\neg\pi \wedge \Diamond P(x)) \rightarrow \psi)})$$



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 φ holds then every starting interval satisfies ψ"* (i.e., $\Box\Box(\varphi \rightarrow [B]\psi)$)
3. Say *"for every interval, if φ holds then every ending interval satisfies ψ"* (i.e., $\Box\Box(\varphi \rightarrow [E]\psi)$)
4. Say *"for every interval, if φ holds then every sub-interval satisfies ψ"* (i.e., $\Box\Box(\varphi \rightarrow [D]\psi)$)

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 φ holds then every starting interval satisfies ψ"* (i.e., $\Box\Box(\varphi \rightarrow [B]\psi)$)
3. Say *"for every interval, if φ holds then every ending interval satisfies ψ"* (i.e., $\Box\Box(\varphi \rightarrow [E]\psi)$)
4. Say *"for every interval, if φ holds then every sub-interval satisfies ψ"* (i.e., $\Box\Box(\varphi \rightarrow [D]\psi)$)

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 φ holds then every starting interval satisfies ψ"* (i.e., $\Box\Box(\varphi \to [B]\psi)$)
3. Say *"for every interval, if φ holds then every ending interval satisfies ψ"* (i.e., $\Box\Box(\varphi \to [E]\psi)$)
4. Say *"for every interval, if φ holds then every sub-interval satisfies ψ"* (i.e., $\Box\Box(\varphi \to [D]\psi)$)

$$[B^{\varphi}_{\psi}] \equiv \Box\Box(\varphi \to [B]\psi)$$
$$[E^{\varphi}_{\psi}] \equiv \Box\Box(\varphi \to [E]\psi)$$
$$[D^{\varphi}_{\psi}] \equiv \Box\Box(\varphi \to [D]\psi)$$

Proof overview

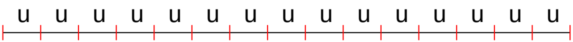
1. Encoding the rectangle
2. Encoding the neighbourhood relations

u	u	u	u
u	u	u	u
u	u	u	u
u	u	u	u
u	u	u	u

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

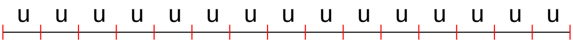


Proof overview

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

$\Diamond u$
$\Box\Box(u \rightarrow \neg\pi)$
$\Box\Box(u \rightarrow (\Diamond u \vee \Box\pi))$



Proof overview

1. Encoding the rectangle
2. Encoding the neighbourhood relations

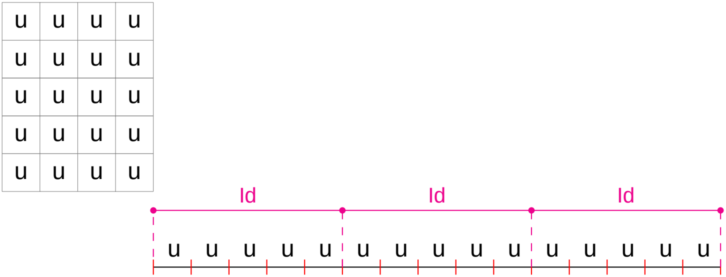
u	u	u	u
u	u	u	u
u	u	u	u
u	u	u	u
u	u	u	u

$\Diamond \mathrm{u}$
$\Box\Box(\mathrm{u} \rightarrow \neg\pi)$
$\Box\Box(\mathrm{u} \rightarrow (\Diamond \mathrm{u} \vee \Box\pi))$
$[B^{\mathrm{u}}_{\neg \mathrm{u}}] \wedge [B^{\mathrm{u}}_{\neg\pi \rightarrow \neg\Diamond \mathrm{u}}]$

u u u u u u u u u u u u u u u

Proof overview

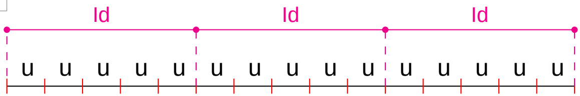
1. Encoding the rectangle
2. Encoding the neighbourhood relations



Proof overview

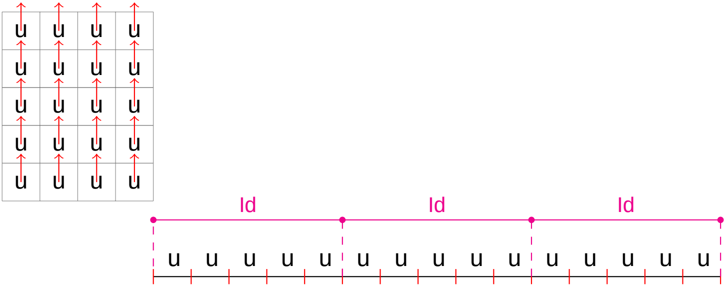
1. Encoding the rectangle
2. Encoding the neighbourhood relations

u	u	u	u
u	u	u	u
u	u	u	u
u	u	u	u
u	u	u	u



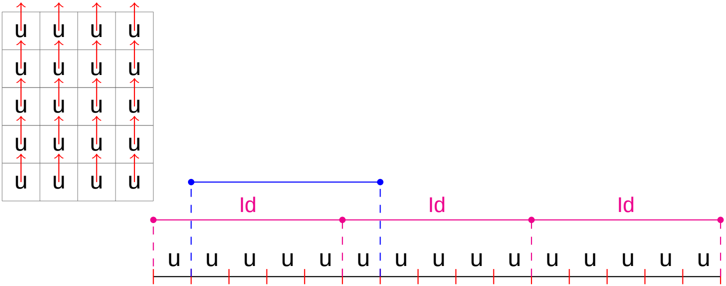
Proof overview

1. Encoding the rectangle
2. Encoding the neighbourhood relations



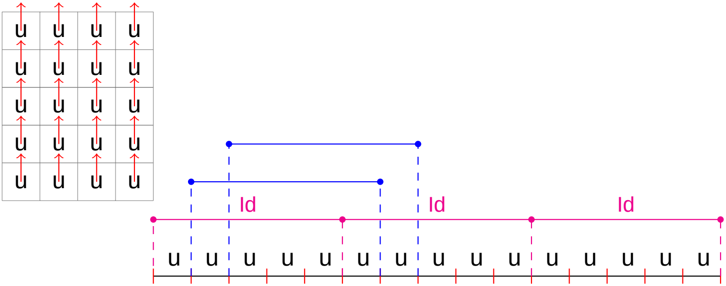
Proof overview

1. Encoding the rectangle
2. Encoding the neighbourhood relations



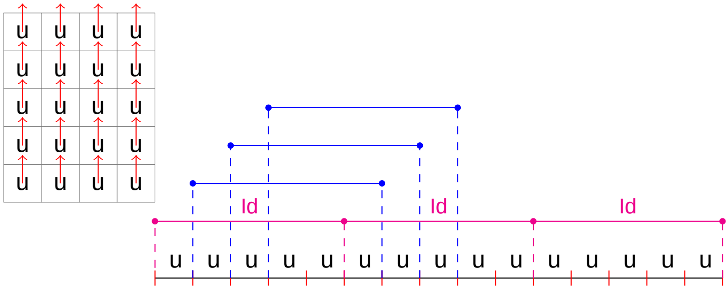
Proof overview

1. Encoding the rectangle
2. Encoding the neighbourhood relations



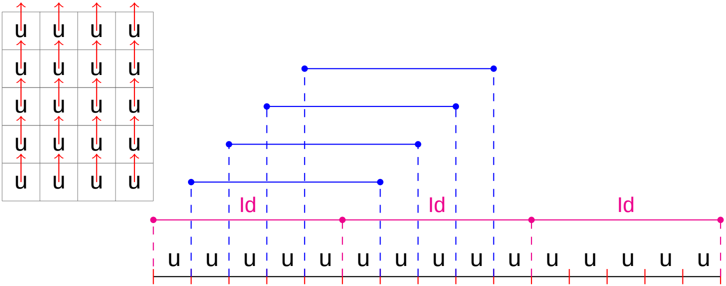
Proof overview

1. Encoding the rectangle
2. Encoding the neighbourhood relations



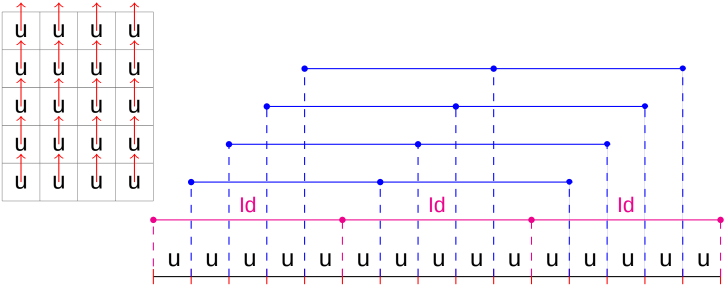
Proof overview

1. Encoding the rectangle
2. Encoding the neighbourhood relations



Proof overview

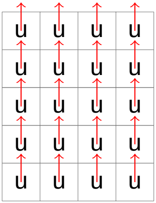
1. Encoding the rectangle
2. Encoding the neighbourhood relations

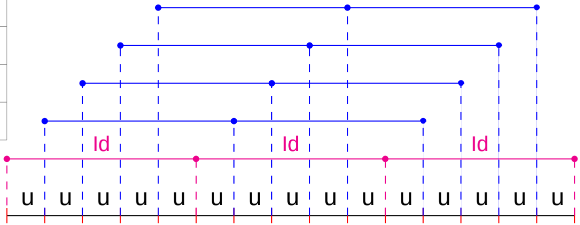


Proof overview

1. Encoding the rectangle
2. Encoding the neighbourhood relations

$[B^{up\_rel}_{\neg up\_rel}] \wedge [E^{up\_rel}_{\neg up\_rel}] \wedge [D^{up\_rel}_{\neg up\_rel}]$

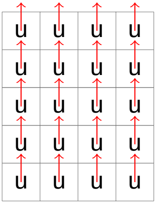


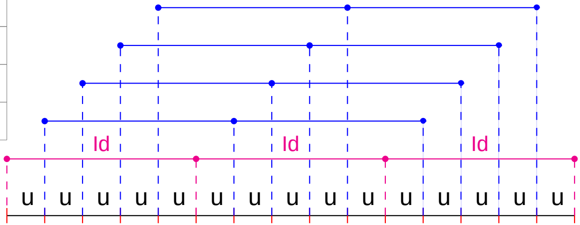


Proof overview

1. Encoding the rectangle
2. Encoding the neighbourhood relations

$$[B^{up\_rel}_{\neg up\_rel}] \wedge [E^{up\_rel}_{\neg up\_rel}] \wedge [D^{up\_rel}_{\neg up\_rel}]$$
$$[B^{up\_rel}_{\neg Id}] \wedge [E^{up\_rel}_{\neg Id}] \wedge [D^{up\_rel}_{\neg Id}]$$

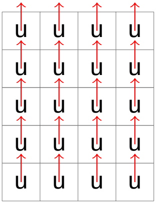


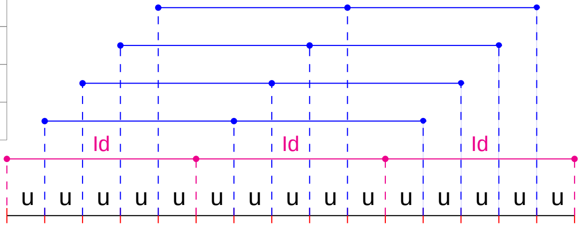


Proof overview

1. Encoding the rectangle
2. Encoding the neighbourhood relations

$$[B^{up\_rel}_{\neg up\_rel}] \wedge [E^{up\_rel}_{\neg up\_rel}] \wedge [D^{up\_rel}_{\neg up\_rel}]$$
$$[B^{up\_rel}_{\neg Id}] \wedge [E^{up\_rel}_{\neg Id}] \wedge [D^{up\_rel}_{\neg Id}]$$
$$[D^{Id}_{\neg up\_rel}]$$





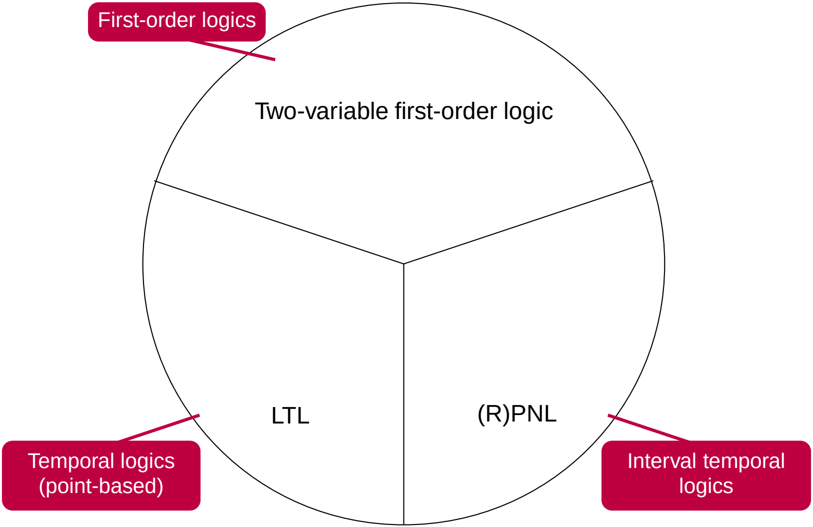
Outline

1 Introduction to Interval Temporal Logics

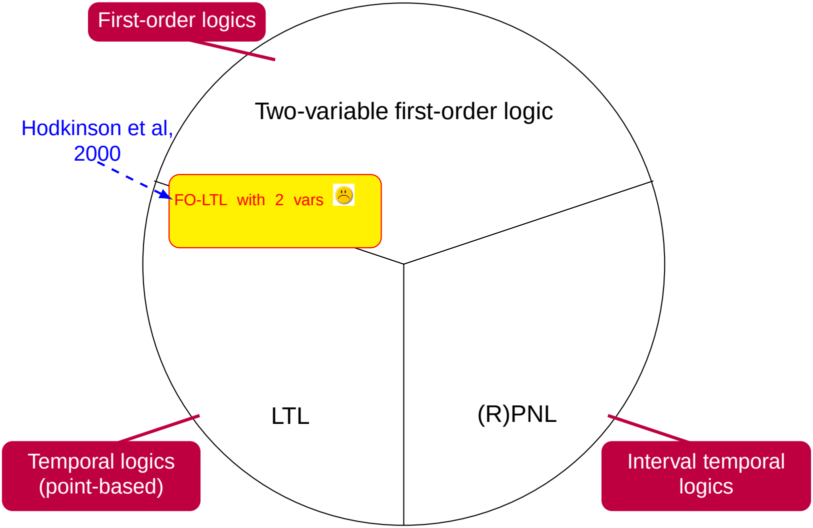
2 First-Order extension of Propositional Neighborhood Logics

3 Conclusions

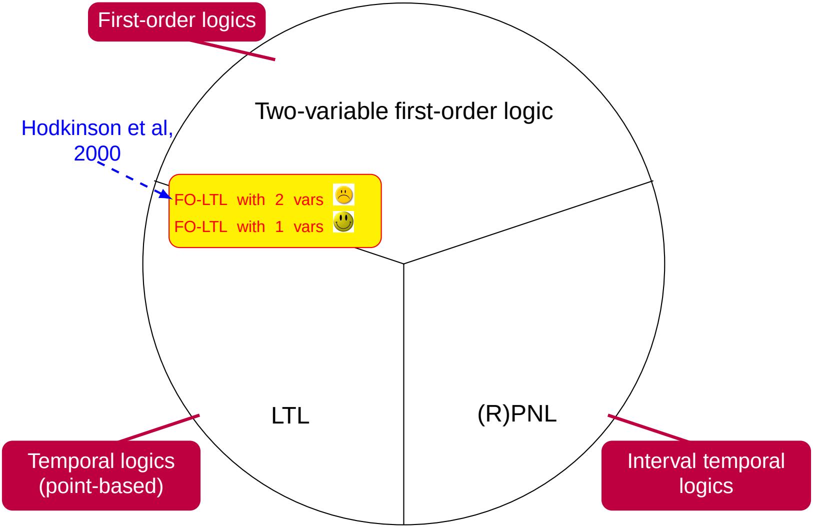
Conclusions and Final remarks



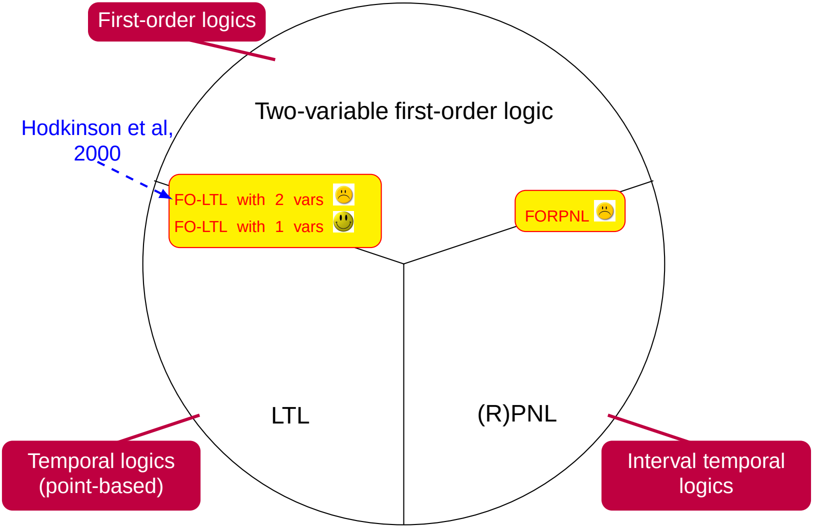
Conclusions and Final remarks



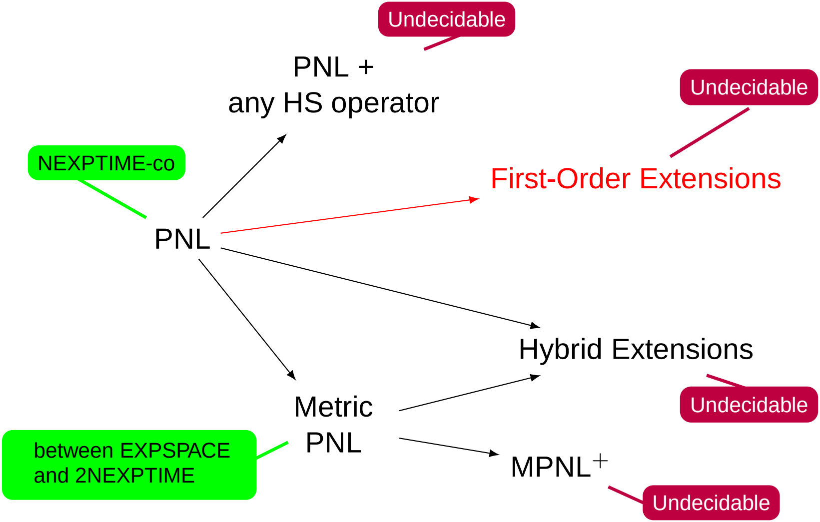
Conclusions and Final remarks



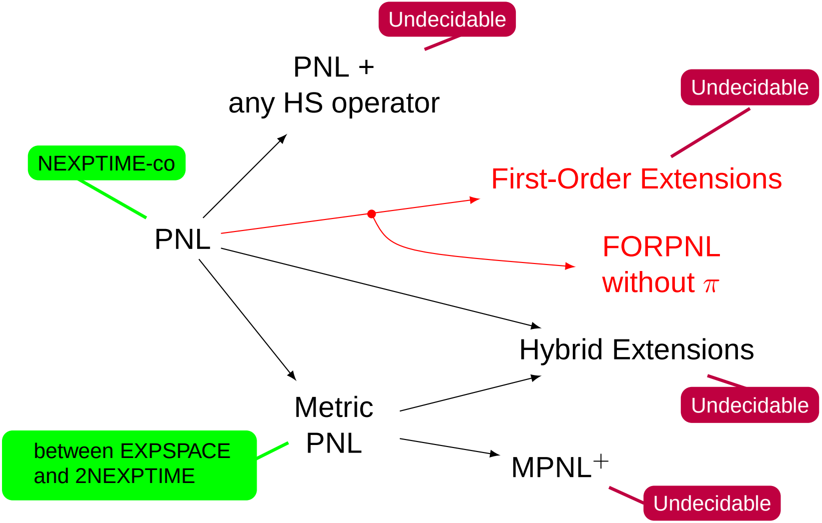
Conclusions and Final remarks



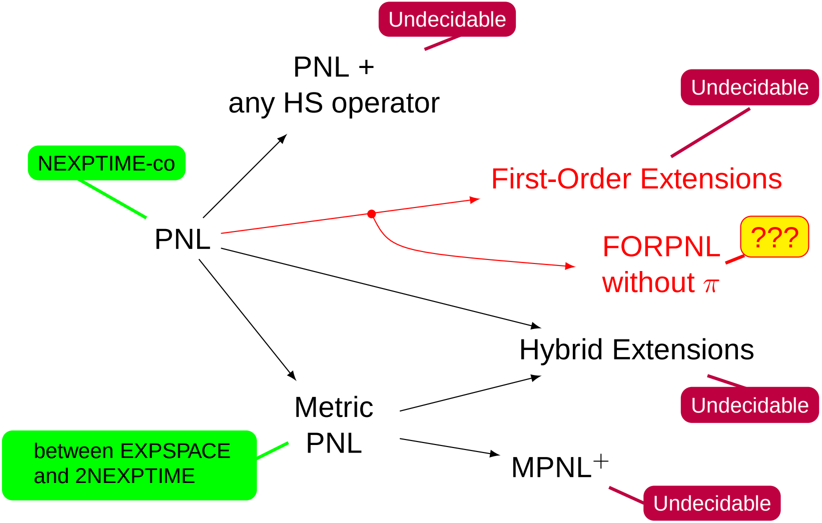
Future work


Undecidable
PNL +
any HS operator
Undecidable
NEXPTIME-co
First-Order Extensions
PNL
Hybrid Extensions
Undecidable
Metric
PNL
between EXPSPACE
and 2NEXPTIME
MPNL+
Undecidable

Future work


Undecidable
PNL +
any HS operator
Undecidable
NEXPTIME-co
First-Order Extensions
PNL
FORPNL
without π
Hybrid Extensions
Metric
PNL
Undecidable
between EXPSPACE
and 2NEXPTIME
MPNL$^+$
Undecidable

Future work


Undecidable
PNL +
any HS operator
Undecidable
NEXPTIME-co
First-Order Extensions
PNL
FORPNL
without π
???
Hybrid Extensions
Undecidable
Metric
PNL
between EXPSPACE
and 2NEXPTIME
MPNL$^+$
Undecidable