An optimal tableau-based decision algorithm for Propositional Neighborhood Logic

Davide Bresolin   Angelo Montanari   Pietro Sala

Dipartimento di Matematica e Informatica
Università degli Studi di Udine
{bresolin,montana,sala}@dimi.uniud.it

STACS 2007

Outline

Interval temporal logics

- Interval temporal logics, such as HS, CDT, and PITL, are very expressive (compared to point-based temporal logics)
- Most interval temporal logics are (highly) undecidable

Problem

Find expressive, but decidable, fragments of existing interval temporal logics.

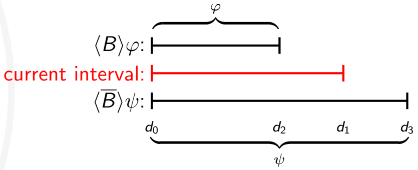
A simple path to decidability

Interval logics make it possible to express properties of pairs of time points rather than of single time points.

How has decidability been achieved? By imposing suitable syntactic and/or semantic restrictions that allow one to reduce interval logics to point-based ones:

- **Constraining interval modalities**
  - $\langle B\rangle\langle\overline{B}\rangle$ and $\langle E\rangle\langle\overline{E}\rangle$ fragments of HS.
- **Constraining temporal structures**
  - Split Logics: any interval can be chopped in at most one way (Split Structures).
- **Constraining semantic interpretations**
  - Local QPITL: a propositional variable is true over an interval if and only if it is true over its starting point (Locality Principle).

- Formulas of $\langle B \rangle \langle \overline{B} \rangle$ can be translated into formulas of LTL$[F, P]$ by replacing $\langle B \rangle$ with P and $\langle \overline{B} \rangle$ with F:



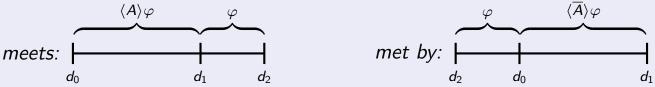
  - The case of $\langle E \rangle \langle \overline{E} \rangle$ is similar.

A major challenge

Identify expressive enough, yet decidable, fragments which are genuinely interval-based (an interval logic which cannot be directly translated into a point-based logic and does not invoke locality or any similar semantic restriction).

Propositional Neighborhood Logic (PNL) [Goranko et al. 2003]

PNL is based on the neighborhood operators $\langle A \rangle$ (*meets*) and $\langle \overline{A} \rangle$ (*met by*);



We restrict our attention to the integers (and their subsets).

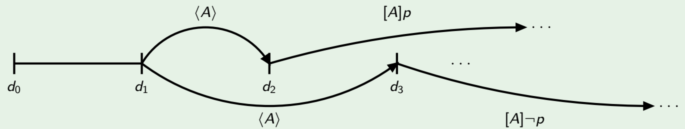
Outline

We cannot abstract way from intervals

Unlike the case of the $\langle B\rangle\langle\overline{B}\rangle$ and $\langle E\rangle\langle\overline{E}\rangle$ fragments, we cannot abstract way from the left endpoint of intervals:

- contradictory formulas can hold over intervals with the same right endpoint, but a different left one.

$\langle A\rangle[A]p \wedge \langle A\rangle[A]\neg p$ is satisfiable:



For any $d > d_3$ we have that p holds over $[d_2, d]$ and $\neg p$ holds over $[d_3, d]$.

Basic ingredients

Definition

An atom is a maximal, locally consistent set of subformulae of φ.

A relation connecting atoms

Connect every pair of atoms that can be associated with neighbor intervals:

$$A \; \mathrm{RL}_\varphi \; B \quad \text{iff} \quad \begin{cases} [A]\psi \in A \Rightarrow \psi \in B \\ [\overline{A}]\psi \in B \Rightarrow \psi \in A \end{cases}$$

Labelled Interval Structures

Definition

A (fulfilling) Labelled Interval Structure (LIS) is a pair $\langle \mathbb{I}(\mathbb{D}), \mathcal{L} \rangle$ where:

- $\mathbb{I}(\mathbb{D})$ is the set of intervals over $\mathbb{D}$;
- the labelling function $\mathcal{L}$ assigns an atom to every interval $[d_i, d_j]$;
- atoms assigned to neighbor intervals are related by RL_φ;
- for every $[d_i, d_j]$ and $\langle A \rangle \psi \in \mathcal{L}([d_i, d_j])$ there exists $d_k > d_j$ such that $\psi \in \mathcal{L}([d_j, d_k])$;
- for every $[d_i, d_j]$ and $\langle \overline{A} \rangle \psi \in \mathcal{L}([d_i, d_j])$ there exists $d_k < d_i$ such that $\psi \in \mathcal{L}([d_k, d_i])$;

Theorem

A formula φ is satisfiable if and only if there exists a (fulfilling) LIS $\langle \mathbb{I}(\mathbb{D}), \mathcal{L} \rangle$ and an interval $[d_i, d_j]$ such that $\varphi \in \mathcal{L}([d_i, d_j])$.

A small-model theorem for LIS

- We have reduced the satisfiability problem for PNL to the problem of finding a (fulfilling) LIS for φ.
- LIS can be of arbitrary size and even infinite!

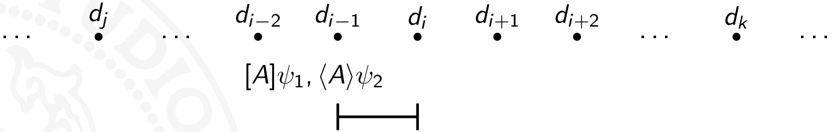
Problems

- How to bound the size of finite LIS?
- How to finitely represent infinite LIS?

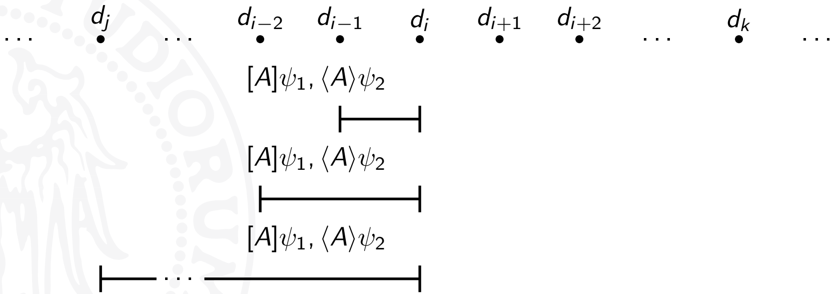
Solution

Any large (resp., infinite) model can be turned into a bounded (resp., bounded periodic) one by progressively removing exceeding points

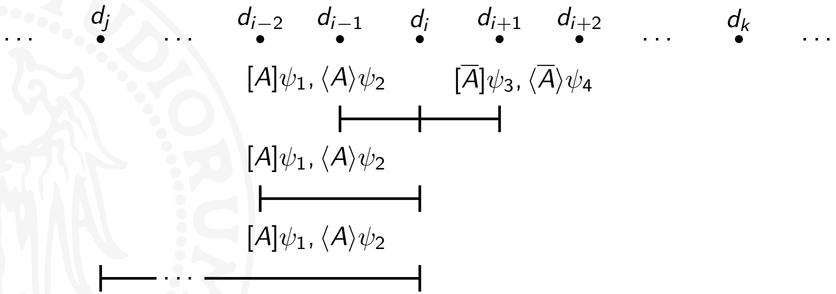
The set of requests of a point



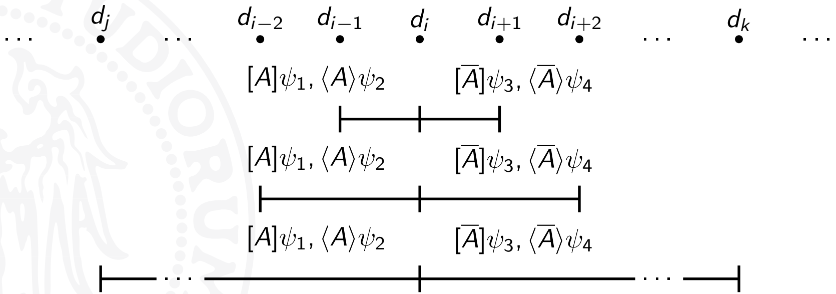
The set of requests of a point



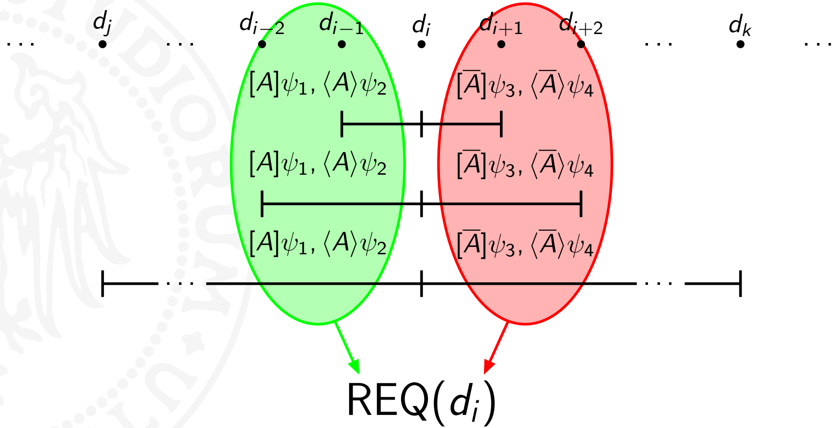
The set of requests of a point



The set of requests of a point



The set of requests of a point



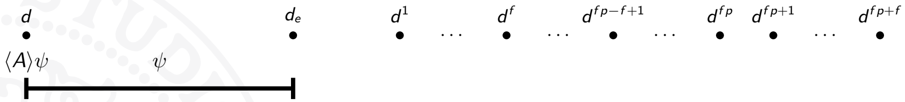
Removing points from a LIS

Lemma

- *Let p be the number of $\langle \overline{A} \rangle$-subformulae of φ and f the number of $\langle A \rangle$-subformulae of φ.*
- *Let $\langle \mathbb{I}(\mathbb{D}), \mathcal{L} \rangle$ be a (fulfilling) LIS for φ and $d_e \in \mathbb{D}$ be a point such that*
  - *there exists at least $p \cdot f + p$ points $d < d_e$ with $\mathrm{REQ}(d) = \mathrm{REQ}(d_e)$ and*
  - *there exists at least $p \cdot f + f$ points $d > d_e$ with $\mathrm{REQ}(d) = \mathrm{REQ}(d_e)$.*

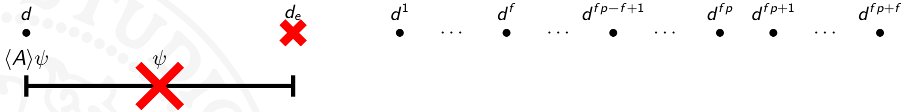
We can remove d_e from the LIS in such a way that the resulting LIS is still fulfilling.

The removal process



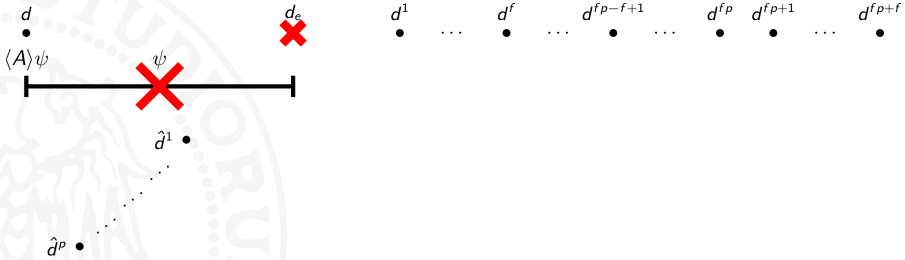
$p \cdot f + f$ points on the right of d_e
with the same set of requests of d_e

The removal process



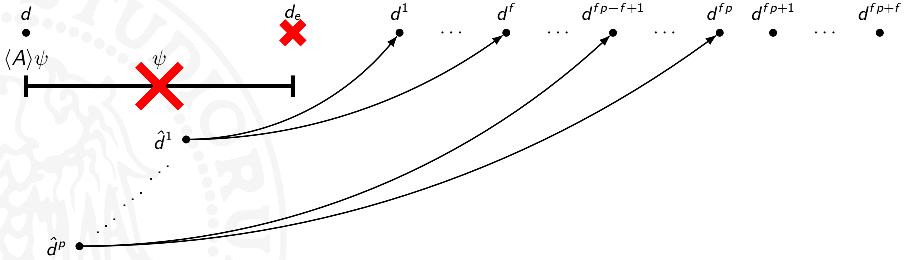
$p \cdot f + f$ points on the right of d_e
with the same set of requests of d_e

The removal process



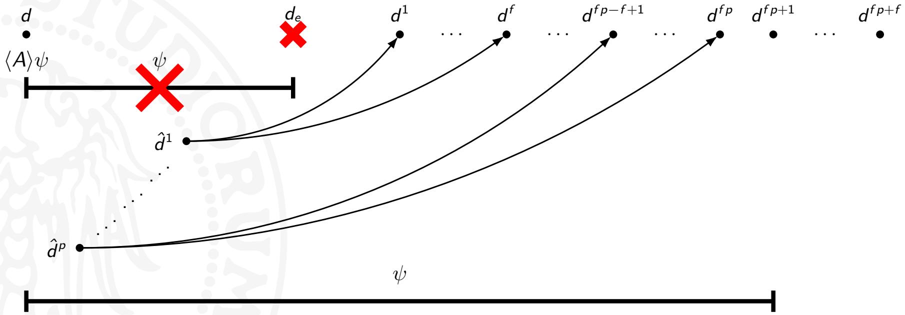
$p \cdot f + f$ points on the right of d_e
with the same set of requests of d_e

The removal process



$p \cdot f + f$ points on the right of d_e
with the same set of requests of d_e

The removal process



$p \cdot f + f$ points on the right of d_e

with the same set of requests of d_e

The small model theorem for PNL

By taking advantage of such a removal process, we can prove the following theorem:

Theorem

A formula φ is satisfiable if and only if there exists a LIS $\langle \mathbb{I}(\mathbb{D}), \mathcal{L} \rangle$ such that:

- *if $\mathbb{D}$ is finite, then every set of requests occurs at most $2pf + f + p$ times in $\mathbb{D}$;*
- *if $\mathbb{D}$ is infinite, then the LIS is ultimately periodic and such that every set of requests occurs at most $2pf + f + p$ times in the infix, in the right period, and in the left period.*

Outline

Nodes of the tableau

$$\mathbf{n} = \langle [d_i, d_j], A_{\mathbf{n}}, \mathrm{REQ}_{\mathbf{n}}, \mathbb{D}_{\mathbf{n}} \rangle$$

- $\mathbb{D}_{\mathbf{n}}$ is a finite linear order
- $[d_i, d_j]$ is an interval over $\mathbb{D}_{\mathbf{n}}$;
- $\mathrm{REQ}_{\mathbf{n}}$ assigns a set of requests to every point in $\mathbb{D}_{\mathbf{n}}$;
- $A_{\mathbf{n}}$ is the atom associated to $[d_i, d_j]$, where
  - for every $[A]\psi \in \mathrm{REQ}_{\mathbf{n}}(d_i)$, $\psi \in A_{\mathbf{n}}$;
  - for every $[\overline{A}]\psi \in \mathrm{REQ}_{\mathbf{n}}(d_j)$, $\psi \in A_{\mathbf{n}}$.

Expansion rules

Rules are applied to the last node $\mathbf{n} = \langle [d_i, d_j], A_{\mathbf{n}}, \text{REQ}_{\mathbf{n}}, \mathbb{D}_{\mathbf{n}} \rangle$ of a branch

Right-step rule: Add a new point d_r to the right of $\mathbb{D}_{\mathbf{n}}$ and label the interval $[d_j, d_r]$

Left-step rule: Add a new point d_l to the left of $\mathbb{D}_{\mathbf{n}}$ and label the interval $[d_l, d_i]$

Fill-in rule: Label emerging intervals

Remark

When the rules are applied, add a successor for every possible choice of the labelling.

Building the tableau for φ

- Start from the initial tableau

$$\emptyset$$

$$\langle [0,1], A_1, \mathrm{REQ}_1, \{0,1\} \rangle \quad \cdots \quad \langle [0,1], A_i, \mathrm{REQ}_i, \{0,1\} \rangle \quad \cdots \quad \langle [0,1], A_k, \mathrm{REQ}_k, \{0,1\} \rangle$$

  (all possible nodes with $\langle A \rangle \varphi \in \mathrm{REQ}_i(0)$).
- Apply the expansion rules.
- Stop the expansion of a branch when:
  - there are two points $d_i < d_j$ such that there are no nodes labelled with $[d_i, d_j]$, but the fill-in rule is not applicable.
  - all $\langle A \rangle$ and $\langle \overline{A} \rangle$ formulae are satisfied.
  - there exists a set of requests that is repeated more that $2pf + p + f$ times in $\mathbb{D}_\mathbf{n}$.

Fulfilling branches

Definition

A branch in a tableau is fulfilling if and only if:

- for every pair $d_i < d_j$ in $\mathbb{D}_{\mathbf{n}}$ there exists a node labelled with $[d_i, d_j]$;
- one of the following conditions hold:
  - all $\langle A \rangle$ and $\langle \overline{A} \rangle$ formulae are satisfied (finite model found);
  - we can exploit the information on the branch to build a periodic model where all $\langle A \rangle$ and $\langle \overline{A} \rangle$ formulae are satisfied (infinite model found).

Theorem

A formula φ is satisfiable if and only if there exists a fulfilling branch in every tableau for it.

Computational complexity

- The number of possible sets of requests is exponential in the length of φ.
- The number of $\langle A \rangle$ and $\langle \overline{A} \rangle$-subformulae of φ is linear in the length of φ.
- Every set of request is repeated at most $2pf + f + p$ times in $\mathbb{D}_{\mathbf{n}}$.
- Hence, the length of a branch is at most exponential in the length of φ.

The proposed tableau method for PNL is in NEXPTIME

$\Rightarrow$ Our tableau method is optimal: it is possible to reduce the exponential tiling problem to PNL satisfiability.

Outline

1. Introduction

2. A small model theorem for PNL

3. The tableau method for PNL

4. Future work

Future work

- **Adapt the tableau (for PNL) to other temporal structures:**
  - the class of all linear orderings
  - the class of dense orderings
  - other specific orderings ($\mathbb{Q}, \mathbb{R}, \ldots$)
  - branching time temporal structures
- **Extend PNL with other temporal operators:**
  - the sub-interval operator $\langle D \rangle$
  - CTL-like path quantifiers (A and E) over branching-time temporal structures