A Tableau-Based Decision Procedure for Right Propositional Neighborhood Logic ($RPNL^-$)

Davide Bresolin   Angelo Montanari

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

TABLEAUX 2005

Outline

Outline

Interval temporal logics

Interval temporal logics (HS, CDT, PITL) are very expressive

- simple syntax and semantics;
- can naturally express statements that refer to time intervals and continuous processes;
- the most expressive ones (HS and CDT) are strictly more expressive than every point-based temporal logic.

Interval temporal logics are (highly) undecidable

The validity problem for HS is not recursively axiomatizable.

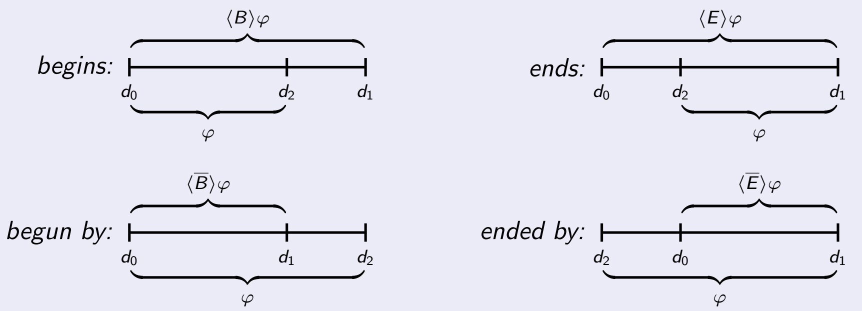
Problem

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

Halpern and Shoam's HS

HS features four basic unary operators

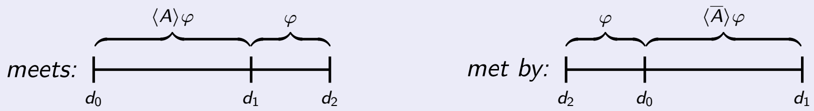
- $\langle B \rangle$ (*begins*) and $\langle E \rangle$ (*ends*), and their transposes $\langle \overline{B} \rangle$ (*begun by*) and $\langle \overline{E} \rangle$ (*ended by*).
- Given a formula φ and an interval $[d_0, d_1]$, $\langle B \rangle \varphi$ holds over $[d_0, d_1]$ if φ holds over $[d_0, d_2]$, for some $d_0 \leq d_2 < d_1$, and $\langle E \rangle \varphi$ holds over $[d_0, d_1]$ if φ holds over $[d_2, d_1]$, for some $d_0 < d_2 \leq d_1$.



Some interesting fragments of HS

- **The $\langle B \rangle \langle E \rangle$ fragment** (*undecidable*);
- **Goranko, Montanari, and Sciavicco's PNL:**
  - based on the derived operators $\langle A \rangle$ (*meets*) and $\langle \overline{A} \rangle$ (*met by*);

Neighborhood Operators



  - decidable (by reduction to 2FO[<]), but no tableau methods.

A simple path to decidability

- In propositional interval temporal logics undecidability is the rule and decidability the exception.
- Interval logics make it possible to express properties of pairs of time points:
  - In most cases, this feature prevents one from the possibility of reducing interval-based temporal logics to point-based ones.
- There are a few exceptions where suitable syntactic and/or semantic restrictions allows one to reduce interval logics to point-based ones.

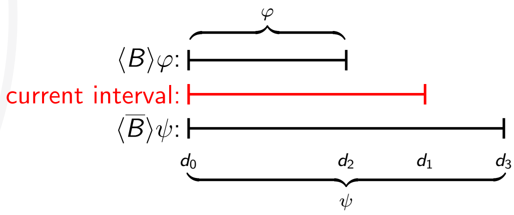
P.S. truth is not monotonic with respect to inclusion: if φ is true over an interval, it is not necessarily true over its subintervals.

Three different strategies

- **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 Structures (any interval can be chopped in at most one way) and Split Logics.
- **Constraining semantic interpretations:**
  - Locality principle (a propositional variable is true over an interval if and only if it is true over its starting point) and Local QPITL.

The $\langle B\rangle\langle \overline{B}\rangle$ and $\langle E\rangle\langle \overline{E}\rangle$ fragments

- Decidability of $\langle B\rangle\langle \overline{B}\rangle$ and $\langle E\rangle\langle \overline{E}\rangle$ can be obtained by embedding them into the propositional temporal logic of linear time LTL$[F, P]$ with temporal modalities F (sometime in the future) and P (sometime in the past).
- Formulas of $\langle B\rangle\langle \overline{B}\rangle$ are simply 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.
- LTL$[F, P]$ has the finite model property and is decidable.

An alternative path to decidability

A major challenge

Identify expressive enough, yet decidable, fragments and/or logics which are genuinely interval-based.

What is a genuinely interval-based logic?

A logic is genuinely interval-based if it is an interval logic which cannot be directly translated into a point-based logic and does not invoke locality, or any other semantic restriction reducing the interval-based semantics to the point-based one.

Outline

A Tableau for RPNL$^-$ (1)

We exploited:

- **Syntactic restrictions:**
  - no past operators ($\langle A \rangle$ operator only)
- **Semantic restrictions:**
  - natural numbers

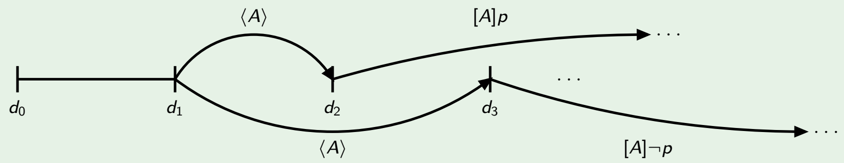
to devise a tableau based decision procedure for the future fragment of (strict) PNL (RPNL$^-$ for short).

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 $[d_3, d]$.

A Tableau for RPNL$^-$ (2)

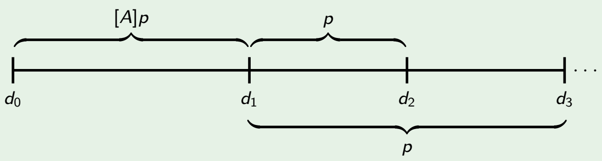
The proposed tableau method partly resembles the tableau-based decision procedure for LTL,

but the tableau for LTL takes advantage of a straightforward “fix-point definition” of temporal operators:

- every formula is split in a part related to the current state and a part related to the next state.

We cannot ignore the past

We must keep track of universal and (pending) existential requests coming from the past.



Warning:
p can be $\langle A \rangle q$!

Atoms

Definition

An atom is a pair (A, C) such that:

- C is a maximal, locally consistent set of subformulas of φ;
- A is a consistent (but not necessarily complete) set of temporal formulas ($\langle A \rangle \psi$ and $[A]\psi$);
- A and C must be coherent: if $[A]\psi \in A$, then $\psi \in C$.

Atoms and Intervals

- Associate with every interval $[d_i, d_j]$ an atom (A, C):
  - C contains the formulas that (should) hold over $[d_i, d_j]$;
  - A contains temporal requests coming from the past.
- Connect every pair of atoms that are associated with neighbor intervals.

A Layered Structure for Satisfiability

For all j, the sets $C_{[d_i,d_j]}$, with $i < j$, must agree on their temporal formulas.

Universal formulas belonging to the sets $C_{[d_i,d_j]}$ must be satisfied by all sets $C_{[d_j,d_k]}$.

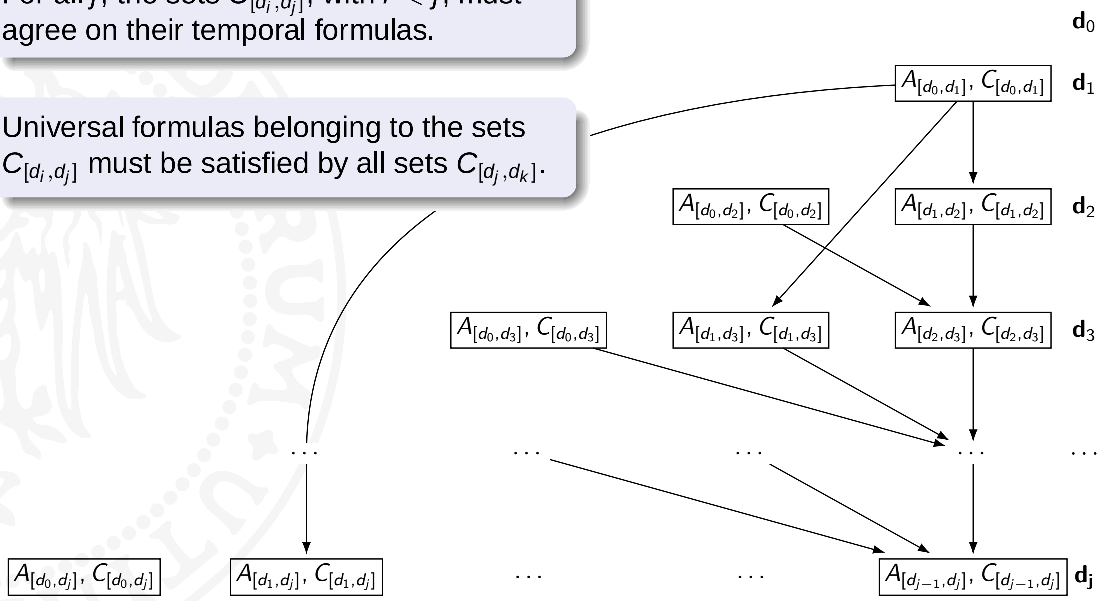


Tableau Construction

Tableau Construction (Idea)

- Layers of the picture becomes nodes of the tableau.
- Connects two nodes if they are associated with successive layers.
- A path in the tableau is a quasi-model of the formula:
  - formulas without temporal operators are satisfied;
  - $[A]\psi$ formulas are satisfied;
  - it is not guaranteed that $\langle A\rangle\psi$ formulas are satisfied.

Problem

To find a model for φ, we must guarantee that $\langle A\rangle\psi$ formulas get satisfied.

The X_φ relation

Definition

X_φ is a relation over atoms such that $(A, C)X_\varphi(A', C')$ iff:

- $A' \subseteq A$;
- if $[A]\psi \in A$, then $[A]\psi \in A'$;
- if $\langle A\rangle\psi \in A$, then $\langle A\rangle\psi \in A'$ iff $\psi \notin C$.

Connecting intervals

X_φ connects an atom (A, C) associated to an interval $[d_i, d_j]$ with the atom (A', C') associated to $[d_i, d_{j+1}]$:

- universal requests coming from the past are preserved;
- existential requests are discarded when fulfilled.

Tableau nodes

Definition

A node N of the tableau is a set of atoms such that, for every temporal formula $\langle A\rangle\psi$ ($[A]\psi$) and every pair of atoms $(A, C), (A', C') \in N$, $\langle A\rangle\psi \in C$ iff $\langle A\rangle\psi \in C'$ ($[A]\psi \in C$ iff $[A]\psi \in C'$).

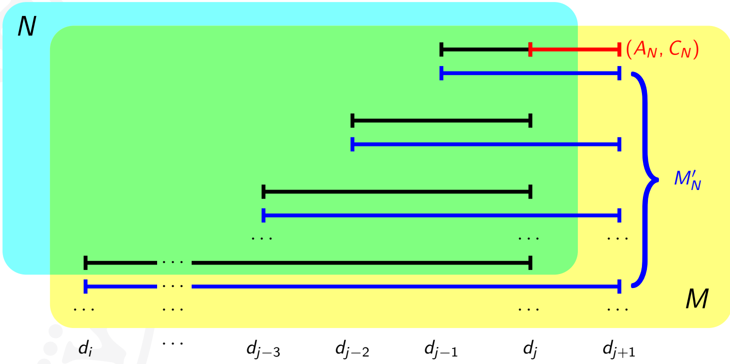
Nodes and points

A node N represents a point d_j of the temporal domain:

- every atom in N represents an interval $[d_i, d_j]$ ending in d_j.

Connecting nodes

We put an edge between two nodes if they represent successive time points:



- (A_N, C_N) is an atom such that A_N contains all requests (temporal formulas) of N;
- for every $(A, C) \in N$ there is $(A', C') \in M'_N$ such that $(A, C)\ X_\varphi\ (A', C')$;
- for every $(A', C') \in M'_N$ there is $(A, C) \in N$ such that $(A, C)\ X_\varphi\ (A', C')$.

Fulfilling paths and satisfiability

Definition (Fulfilling path)

A path π is fulfilling iff every $\langle A \rangle \psi$ active formula that belongs to a node in π gets satisfied by a descendant node in π.

Theorem

φ is satisfiable iff there exists a fulfilling path in the tableau for φ.

Problem

How to check for the presence of fulfilling paths in the tableau?

Strongly Connected Components

Definition

A strongly connected component $\mathcal{S}$ is a subgraph of the tableau such that there exists a path between every two nodes in $\mathcal{S}$.

Definition

An SCC $\mathcal{S}$ is self-fulfilling iff every $\langle A \rangle \psi$ active formula that belongs to a node in $\mathcal{S}$ gets satisfied by a node in $\mathcal{S}$.

Remarks

- Fulfilling paths can be reduced to self-fulfilling SCCs.
- We can restrict ourselves to maximal SCCs, MSCCs for short (monotonicity).

The Decision Procedure

Decision Procedure (Idea)

- Eliminate those MSCCs that cannot participate in a fulfilling path.
- The formula is satisfiable iff the final tableau is non-empty.

Computational Complexity

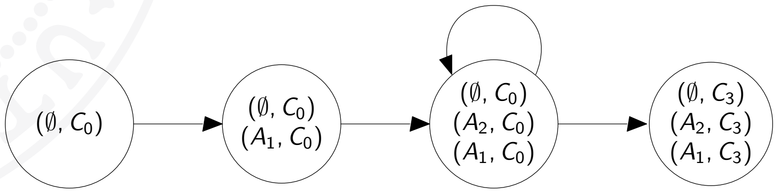
- Checking for self-fulfilling MSCCs can be done in time linear in the size of the tableau.
- The size of the tableau is doubly exponential in the size of φ.
- The decision procedure takes time doubly exponential in the size of φ.

An Example

Tableau for the satisfiable formula $\varphi = \langle A \rangle [A] \perp$.

Atoms obtained by combining the following sets of formulas:

$$
\begin{array}{ll}
A_0 = \emptyset; & C_0 = \{\langle A \rangle [A] \perp, \langle A \rangle \top, \top\}; \\
A_1 = \{\langle A \rangle [A] \perp, \langle A \rangle \top\}; & C_1 = \{\langle A \rangle [A] \perp, [A] \perp, \top\}; \\
A_2 = \{\langle A \rangle [A] \perp\}; & C_2 = \{\neg(\langle A \rangle [A] \perp), \langle A \rangle \top, \top\}; \\
A_3 = \{[A]\langle A \rangle \top, \langle A \rangle \top\}; & C_3 = \{\neg(\langle A \rangle [A] \perp), [A] \perp, \top\}. \\
A_4 = \{[A]\langle A \rangle \top\}; & \\
A_5 = \{\langle A \rangle \top\}; &
\end{array}
$$



Improving the Complexity

Lemma

If there exists a fulfilling path, then (either it is finite or) there exists an ultimately periodic fulfilling path of prefix and period length bounded by $|\varphi|$.

By exploiting nondeterminism, such an ultimately periodic path can be built one node at a time:

1. the algorithm nondeterministically guess the next node of the path;
2. it is necessary to store only two nodes at a time: the current one and the next one.

Theorem

The decidability problem for $RPNL^-$, interpreted over the naturals, is in EXPSPACE.

Outline

1. Introduction
2. A Tableau for RPNL$^-$
3. Future work

Future work

- **Extend our tableau to other temporal structures:**
  - branching-time temporal structures over $\mathbb{N}$ (infinite trees) **Done!**
  - dense domains ($\mathbb{Q}, \mathbb{R}, \ldots$)
- **Extend our logic with other operators:**
  - CTL-like path quantifiers (A and E) over branching-time temporal structures **Done!**
- **A Tableau for full PNL:**
  - over the naturals,
  - over the integers,
  - over the reals,
  - ...