A Tableau System for Right Propositional Neighborhood Logic over Finite Linear Orders: an Implementation

Davide Bresolin, Dario Della Monica, Angelo Montanari, and Guido Sciavicco

ICE-TCS, School of Computer Science, Reykjavik University, Iceland

Please notice: these slides have been mostly produced by Guido Sciavicco (University of Murcia)

CILC 2013 - Catania, 27th September 2013

Outline

Outline

1. Introduction on Interval Temporal Logic
2. Halpern-Shoham's modal logic HS
3. The tableau method for the logic A
4. Conclusions

Presentation

- this is an implementation and experimental work
    - no new theoretical results here
- not free of unexpected problems and difficulties
- this implementation, although particularly simple, is the only one of its kind

Time and logics

In AI usually time is formalized with languages (logics) that are:

- point-based: formulas interpreted over points
- interval-based: formulas interpreted over intervals (our case)

Time and logics

In AI usually time is formalized with languages (logics) that are:

- point-based: formulas interpreted over points
- interval-based: formulas interpreted over intervals (our case)
  - the truth of a formula over an interval is independent from its truth over sub-intervals

Motivations

- properties intrinsically related to intervals (instead of points)
- points have no duration

Motivations

- properties intrinsically related to intervals (instead of points)
- points have no duration

Example: “traveling from Reykjavik to Catania”:

- true over a precise interval of time
- not true over all other intervals
  (starting/ending intervals, inner intervals, ecc.)

Motivations

- properties intrinsically related to intervals (instead of points)
- points have no duration

Example: “traveling from Reykjavik to Catania”:

- true over a precise interval of time
- not true over all other intervals
  (starting/ending intervals, inner intervals, ecc.)

Several philosophical and logical paradoxes disappear:

- Zeno’s flying arrow paradox (“if at each instant the flying arrow stands still, how is movement possible?”)

Motivations

- properties intrinsically related to intervals (instead of points)
- points have no duration

Example: “traveling from Reykjavik to Catania”:

- true over a precise interval of time
- not true over all other intervals
  (starting/ending intervals, inner intervals, ecc.)

Several philosophical and logical paradoxes disappear:

- Zeno’s flying arrow paradox (“if at each instant the flying arrow stands still, how is movement possible?”)
- The dividing instant dilemma (“if the light is on and it is turned off, what is its state at the instant between the two events?”)

Reasoning parameters

Common parameters (independent from point/interval setting):

- linear or branching?
- discrete or dense?
- with or without beginning/end?

Reasoning parameters

Common parameters (independent from point/interval setting):

- linear or branching?
- discrete or dense?
- with or without beginning/end?

Interval setting parameters:

- bounded/unbounded intervals?
- point intervals: yes/no?

Reasoning parameters

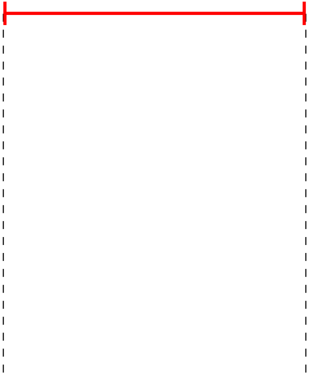
Common parameters (independent from point/interval setting):

- linear
- discrete
- with beginning/end

Interval setting parameters:

- bounded intervals
- point intervals: no

Binary interval relations on linear orders

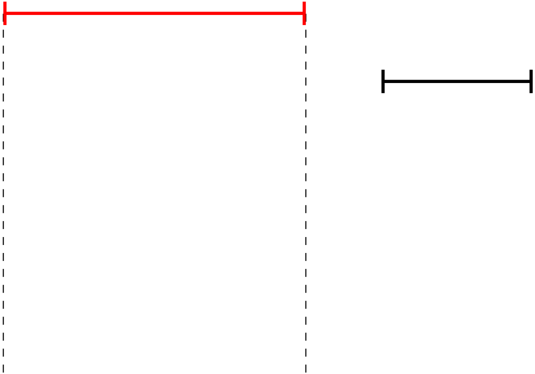


J. F. Allen

Maintaining knowledge about temporal intervals.

Communications of the ACM, volume 26(11), pages 832-843, 1983.

Binary interval relations on linear orders



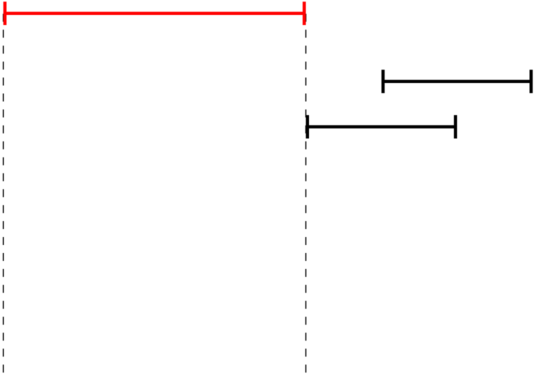
Later

J. F. Allen

Maintaining knowledge about temporal intervals.

Communications of the ACM, volume 26(11), pages 832-843, 1983.

Binary interval relations on linear orders



Later

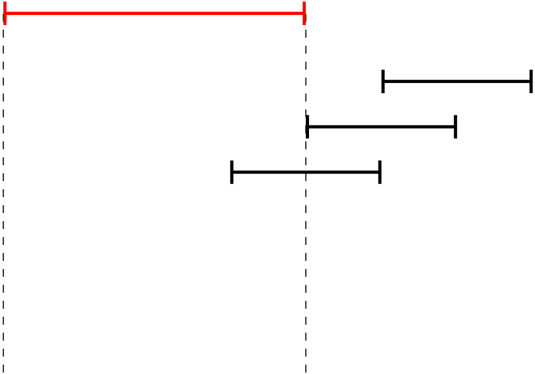
After (right neighbour)

J. F. Allen

Maintaining knowledge about temporal intervals.

Communications of the ACM, volume 26(11), pages 832-843, 1983.

Binary interval relations on linear orders



Later

After (right neighbour)

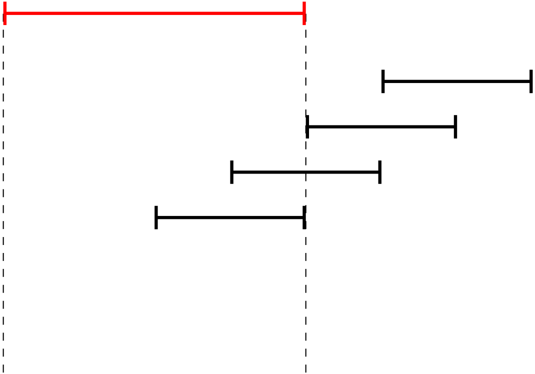
Overlaps (to right)

J. F. Allen

Maintaining knowledge about temporal intervals.

Communications of the ACM, volume 26(11), pages 832-843, 1983.

Binary interval relations on linear orders



Later

After (right neighbour)

Overlaps (to right)

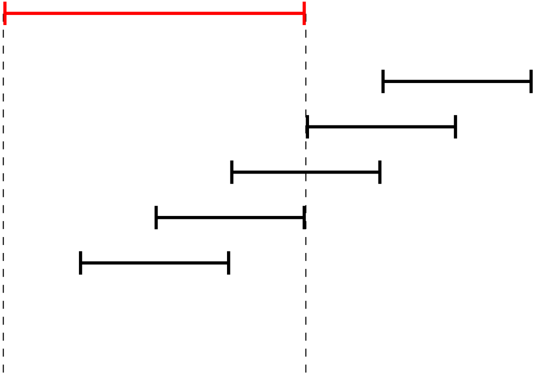
Ends

J. F. Allen

Maintaining knowledge about temporal intervals.

Communications of the ACM, volume 26(11), pages 832-843, 1983.

Binary interval relations on linear orders



Later

After (right neighbour)

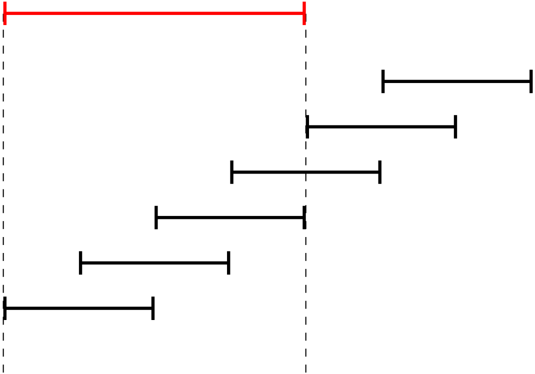
Overlaps (to right)

Ends

During (subinterval)

J. F. Allen
Maintaining knowledge about temporal intervals.
Communications of the ACM, volume 26(11), pages 832-843, 1983.

Binary interval relations on linear orders



Later

After (right neighbour)

Overlaps (to right)

Ends

During (subinterval)

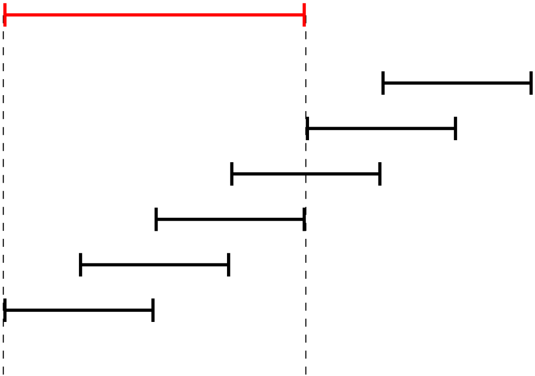
Begins

J. F. Allen

Maintaining knowledge about temporal intervals.

Communications of the ACM, volume 26(11), pages 832-843, 1983.

Binary interval relations on linear orders



Later

After (right neighbour)

Overlaps (to right)

Ends

During (subinterval)

Begins

6 relations + their inverses + equality = 13 Allen's relations.

J. F. Allen
Maintaining knowledge about temporal intervals.
Communications of the ACM, volume 26(11), pages 832-843, 1983.

Outline

1. Introduction on Interval Temporal Logic
2. Halpern-Shoham's modal logic HS
3. The tableau method for the logic A
4. Conclusions

Halpern-Shoham’s modal logic of interval relations

interval relations give rise to modal operators ➡ HS logic

Halpern-Shoham's modal logic of interval relations

interval relations give rise to modal operators → HS logic

HS is undecidable over all significant classes of linear orders

J. Halpern and Y. Shoham

A propositional modal logic of time intervals.

Journal of the ACM, volume 38(4), pages 935-962, 1991.

Halpern-Shoham's modal logic of interval relations

interval relations give rise to
modal operators

HS logic

HS is undecidable over all significant classes of linear orders

J. Halpern and Y. Shoham
A propositional modal logic of time intervals.
Journal of the ACM, volume 38(4), pages 935-962, 1991.

Syntax:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle X \rangle \varphi$$

$$\langle X \rangle \in \{\langle A \rangle, \langle L \rangle, \langle B \rangle, \langle E \rangle, \langle D \rangle, \langle O \rangle, \langle \overline{A} \rangle, \langle \overline{L} \rangle, \langle \overline{B} \rangle, \langle \overline{E} \rangle, \langle \overline{D} \rangle, \langle \overline{O} \rangle\}$$

Halpern-Shoham's modal logic of interval relations

interval relations give rise to modal operators ⇒ HS logic

HS is undecidable over all significant classes of linear orders

J. Halpern and Y. Shoham
A propositional modal logic of time intervals.
Journal of the ACM, volume 38(4), pages 935-962, 1991.

Syntax:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle X \rangle \varphi$$

$$\langle X \rangle \in \{\langle A \rangle, \langle L \rangle, \langle B \rangle, \langle E \rangle, \langle D \rangle, \langle O \rangle, \langle \overline{A} \rangle, \langle \overline{L} \rangle, \langle \overline{B} \rangle, \langle \overline{E} \rangle, \langle \overline{D} \rangle, \langle \overline{O} \rangle\}$$

Models:

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

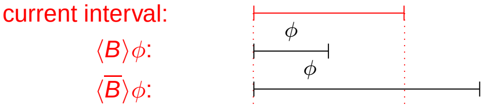
$$V : \mathbb{I}(\mathbb{D}) \mapsto 2^{\mathcal{AP}}$$

$\mathcal{AP}$ atomic propositions (over intervals)

Formal semantics of HS

$\langle B \rangle$: **M**, $[d_0, d_1] \Vdash \langle B \rangle \phi$ iff there exists d_2 such that $d_0 \leq d_2 < d_1$ and **M**, $[d_0, d_2] \Vdash \phi$.

$\langle \overline{B} \rangle$: **M**, $[d_0, d_1] \Vdash \langle \overline{B} \rangle \phi$ iff there exists d_2 such that $d_1 < d_2$ and **M**, $[d_0, d_2] \Vdash \phi$.



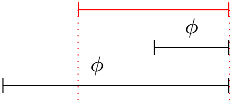
Formal semantics of HS

$\langle B \rangle$: $\mathbf{M}, [d_0, d_1] \Vdash \langle B \rangle \phi$ iff there exists d_2 such that $d_0 \leq d_2 < d_1$ and $\mathbf{M}, [d_0, d_2] \Vdash \phi$.

$\langle \overline{B} \rangle$: $\mathbf{M}, [d_0, d_1] \Vdash \langle \overline{B} \rangle \phi$ iff there exists d_2 such that $d_1 < d_2$ and $\mathbf{M}, [d_0, d_2] \Vdash \phi$.

$\langle E \rangle$: $\mathbf{M}, [d_0, d_1] \Vdash \langle E \rangle \phi$ iff there exists d_2 such that $d_0 < d_2 \leq d_1$ and $\mathbf{M}, [d_2, d_1] \Vdash \phi$.

$\langle \overline{E} \rangle$: $\mathbf{M}, [d_0, d_1] \Vdash \langle \overline{E} \rangle \phi$ iff there exists d_2 such that $d_2 < d_0$ and $\mathbf{M}, [d_2, d_1] \Vdash \phi$.



current interval:

$\langle E \rangle \phi$:

$\langle \overline{E} \rangle \phi$:

Formal semantics of HS

$\langle B \rangle$: $\mathbf{M}, [d_0, d_1] \Vdash \langle B \rangle \phi$ iff there exists d_2 such that $d_0 \leq d_2 < d_1$ and $\mathbf{M}, [d_0, d_2] \Vdash \phi$.

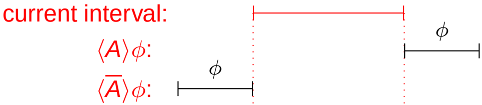
$\langle \overline{B} \rangle$: $\mathbf{M}, [d_0, d_1] \Vdash \langle \overline{B} \rangle \phi$ iff there exists d_2 such that $d_1 < d_2$ and $\mathbf{M}, [d_0, d_2] \Vdash \phi$.

$\langle E \rangle$: $\mathbf{M}, [d_0, d_1] \Vdash \langle E \rangle \phi$ iff there exists d_2 such that $d_0 < d_2 \leq d_1$ and $\mathbf{M}, [d_2, d_1] \Vdash \phi$.

$\langle \overline{E} \rangle$: $\mathbf{M}, [d_0, d_1] \Vdash \langle \overline{E} \rangle \phi$ iff there exists d_2 such that $d_2 < d_0$ and $\mathbf{M}, [d_2, d_1] \Vdash \phi$.

$\langle A \rangle$: $\mathbf{M}, [d_0, d_1] \Vdash \langle A \rangle \phi$ iff there exists d_2 such that $d_1 < d_2$ and $\mathbf{M}, [d_1, d_2] \Vdash \phi$.

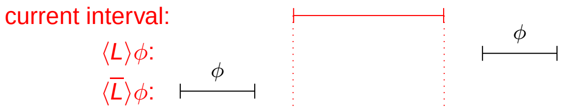
$\langle \overline{A} \rangle$: $\mathbf{M}, [d_0, d_1] \Vdash \langle \overline{A} \rangle \phi$ iff there exists d_2 such that $d_2 < d_0$ and $\mathbf{M}, [d_2, d_0] \Vdash \phi$.



Formal semantics of HS - contd'

$\langle L \rangle$: $\mathbf{M}, [d_0, d_1] \Vdash \langle L \rangle \phi$ iff there exists d_2, d_3 such that $d_1 < d_2 < d_3$ and $\mathbf{M}, [d_2, d_3] \Vdash \phi$.

$\langle \overline{L} \rangle$: $\mathbf{M}, [d_0, d_1] \Vdash \langle \overline{L} \rangle \phi$ iff there exists d_2, d_3 such that $d_2 < d_3 < d_0$ and $\mathbf{M}, [d_2, d_3] \Vdash \phi$.



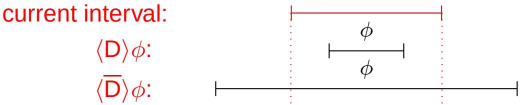
Formal semantics of HS - contd’

$\langle L \rangle$: $\mathbf{M}, [d_0, d_1] \Vdash \langle L \rangle \phi$ iff there exists d_2, d_3 such that $d_1 < d_2 < d_3$ and $\mathbf{M}, [d_2, d_3] \Vdash \phi$.

$\langle \overline{L} \rangle$: $\mathbf{M}, [d_0, d_1] \Vdash \langle \overline{L} \rangle \phi$ iff there exists d_2, d_3 such that $d_2 < d_3 < d_0$ and $\mathbf{M}, [d_2, d_3] \Vdash \phi$.

$\langle D \rangle$: $\mathbf{M}, [d_0, d_1] \Vdash \langle D \rangle \phi$ iff there exists d_2, d_3 such that $d_0 < d_2 < d_3 < d_1$ and $\mathbf{M}, [d_2, d_3] \Vdash \phi$.

$\langle \overline{D} \rangle$: $\mathbf{M}, [d_0, d_1] \Vdash \langle \overline{D} \rangle \phi$ iff there exists d_2, d_3 such that $d_2 < d_0 < d_1 < d_3$ and $\mathbf{M}, [d_2, d_3] \Vdash \phi$.



Formal semantics of HS - contd’

$\langle L\rangle$: $\mathbf{M}, [d_0, d_1] \Vdash \langle L\rangle\phi$ iff there exists d_2, d_3 such that $d_1 < d_2 < d_3$ and $\mathbf{M}, [d_2, d_3] \Vdash \phi$.

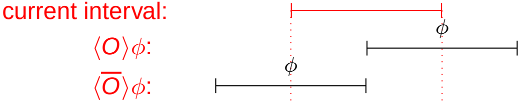
$\langle \overline{L}\rangle$: $\mathbf{M}, [d_0, d_1] \Vdash \langle \overline{L}\rangle\phi$ iff there exists d_2, d_3 such that $d_2 < d_3 < d_0$ and $\mathbf{M}, [d_2, d_3] \Vdash \phi$.

$\langle \mathrm{D}\rangle$: $\mathbf{M}, [d_0, d_1] \Vdash \langle \mathrm{D}\rangle\phi$ iff there exists d_2, d_3 such that $d_0 < d_2 < d_3 < d_1$ and $\mathbf{M}, [d_2, d_3] \Vdash \phi$.

$\langle \overline{\mathrm{D}}\rangle$: $\mathbf{M}, [d_0, d_1] \Vdash \langle \overline{\mathrm{D}}\rangle\phi$ iff there exists d_2, d_3 such that $d_2 < d_0 < d_1 < d_3$ and $\mathbf{M}, [d_2, d_3] \Vdash \phi$.

$\langle O\rangle$: $\mathbf{M}, [d_0, d_1] \Vdash \langle O\rangle\phi$ iff there exists d_2, d_3 such that $d_0 < d_2 < d_1 < d_3$ and $\mathbf{M}, [d_2, d_3] \Vdash \phi$.

$\langle \overline{O}\rangle$: $\mathbf{M}, [d_0, d_1] \Vdash \langle \overline{O}\rangle\phi$ iff there exists d_2, d_3 such that $d_2 < d_0 < d_3 < d_1$ and $\mathbf{M}, [d_2, d_3] \Vdash \phi$.



current interval:

$\langle O\rangle\phi$:

$\langle \overline{O}\rangle\phi$:

The zoo of fragments of HS

- $2^{12} = 4096$ fragments of HS
- several hundreds expressively different
- expressiveness and satisfiability issues wrt. class of interval structures (all, dense, discrete, finite, etc.)

Classifying HS fragments on finite linear orders

⏩ skip

Complexity class:

1: Non primitive recursive

2: EXPSPACE-complete

3: NEXPTIME-complete

4: NP-complete

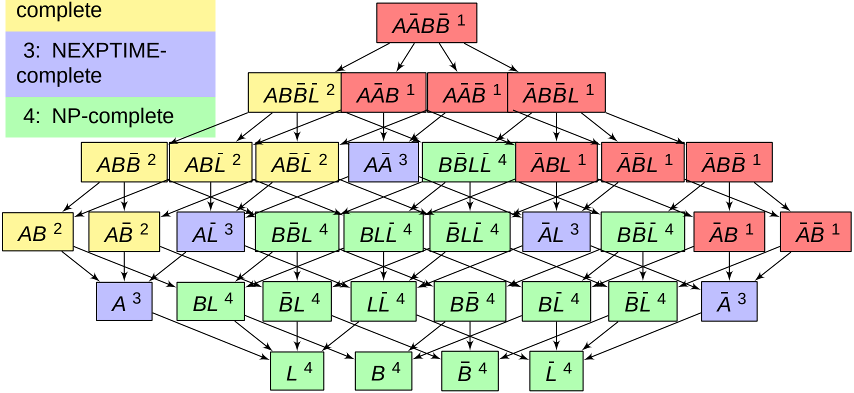


Figure: Hasse diagram of all and only decidable fragments of HS over finite linear orders.

Classifying HS fragments on finite linear orders

⏩ skip

Complexity class:

🟥 1: Non primitive recursive

🟨 2: EXPSPACE-complete

🟪 3: NEXPTIME-complete

🟩 4: NP-complete

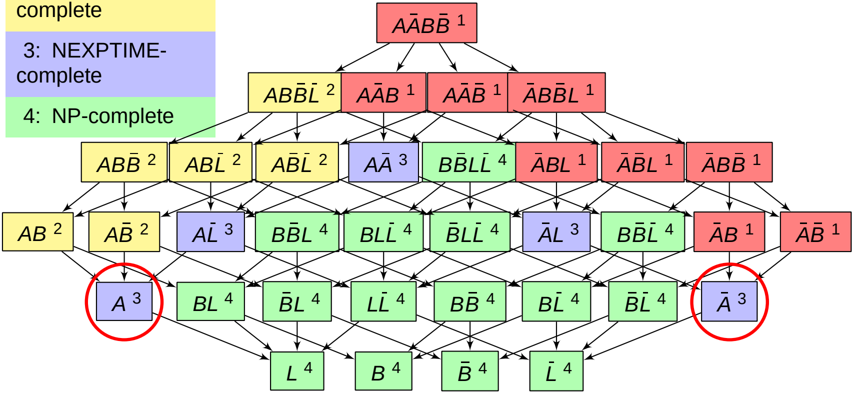


Figure: Hasse diagram of all and only decidable fragments of HS over finite linear orders.

Classifying HS fragments on finite linear orders - contd’

⏩ skip2

In the previous figure, only the decidable cases are displayed, along with their complexity.

Classifying HS fragments on finite linear orders - contd’

⏩ skip2

In the previous figure, only the decidable cases are displayed, along with their complexity.

They are 62, out of 1347 expressively different fragment in the finite case.

Classifying HS fragments on finite linear orders - contd’

⏩ skip2

In the previous figure, only the decidable cases are displayed, along with their complexity.

They are 62, out of 1347 expressively different fragment in the finite case.

F.Y.I., on this web address: *https://itl.dimi.uniud.it/content/logic-hs* you can find any information about HS-fragments and their fragments, updated to the latest advances, which are the results of over 10 years of research in this topic.

Classifying HS fragments on finite linear orders - contd’

⏩ skip2

In the previous figure, only the decidable cases are displayed, along with their complexity.

They are 62, out of 1347 expressively different fragment in the finite case.

F.Y.I., on this web address: *https://itl.dimi.uniud.it/content/logic-hs* you can find any information about HS-fragments and their fragments, updated to the latest advances, which are the results of over 10 years of research in this topic.

In this work, we use the results we have that concern the fragment A in the finite case, and we put them at work to build a usable satisfiability checker.

Outline

1. Introduction on Interval Temporal Logic
2. Halpern-Shoham's modal logic HS
3. The tableau method for the logic A
4. Conclusions

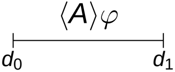
The logic A

Syntax

$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle A \rangle \varphi$

Semantics

$\mathbf{M}, [d_0, d_1] \Vdash \langle A \rangle \phi$



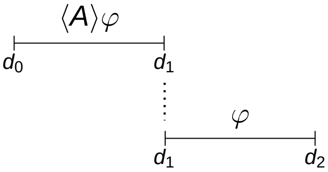
The logic A

Syntax

$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle A\rangle\varphi$

Semantics

$\mathbf{M}, [d_0, d_1] \Vdash \langle A\rangle\phi$ iff $\exists d_2$ such that $d_1 < d_2$ and $\mathbf{M}, [d_1, d_2] \Vdash \phi$



Theoretical basis

- 2003: the first semi-decidability procedure for A
    - Goranko-Montanari-Sciavicco
      Propositional interval neighborhood temporal logics
      JUCS 2003, 9(9):1137–1167

Theoretical basis

- 2003: the first semi-decidability procedure for A
  - Goranko-Montanari-Sciavicco
    Propositional interval neighborhood temporal logics
    JUCS 2003, 9(9):1137–1167
- 2007: A$\overline{\text{A}}$ (and thus A) is decidable in the finite and in the strongly discrete case (small-model property)
  - Bresolin-Montanari-Sciavicco
    An optimal decision procedure for Right Propositional Neighborhood Logic
    JAR 2007, 38(1-3):173–199

Theoretical basis

- 2003: the first semi-decidability procedure for A
  - Goranko-Montanari-Sciavicco
    Propositional interval neighborhood temporal logics
    JUCS 2003, 9(9):1137–1167
- 2007: $A\overline{A}$ (and thus A) is decidable in the finite and in the strongly discrete case (small-model property)
  - Bresolin-Montanari-Sciavicco
    An optimal decision procedure for Right Propositional Neighborhood Logic
    JAR 2007, 38(1-3):173–199
- In this work we adapted the JUCS 2003 result (a classical tableaux) with the closing condition from the JAR 2007 result

Theoretical basis - contd’

The main theorem (small-model property) [JAR 2007]

Let φ be a A-formula. Then, φ is finitely satisfiable if and only if it is satisfiable on a model whose cardinality is strictly less than $2^m \cdot m + 1$, where m is the number of diamonds and boxes in φ.

Theoretical basis - contd’

The main theorem (small-model property) [JAR 2007]

Let φ be a A-formula. Then, φ is finitely satisfiable if and only if it is satisfiable on a model whose cardinality is strictly less than $2^m \cdot m + 1$, where m is the number of diamonds and boxes in φ.

Corollary

A is in NEXPTIME

Theoretical basis - contd'

The main theorem (small-model property) [JAR 2007]

Let φ be a A-formula. Then, φ is finitely satisfiable if and only if it is satisfiable on a model whose cardinality is strictly less than $2^m \cdot m + 1$, where m is the number of diamonds and boxes in φ.

Corollary

A is in NEXPTIME

Theorem [JAR 2007]

A is NEXPTIME-hard, too

- A is NEXPTIME-complete

Satisfiability via tableaux

- satisfiability: given a A-formula φ, we want to establish if there exists a finite model **M** and an interval $[d_i, d_j]$ in it such that $\mathbf{M}, [d_i, d_j] \Vdash \varphi$

Satisfiability via tableaux

- satisfiability: given a A-formula φ, we want to establish if there exists a finite model **M** and an interval $[d_i, d_j]$ in it such that $\mathbf{M}, [d_i, d_j] \Vdash \varphi$
- initial satisfiability

Satisfiability via tableaux

- satisfiability: given a A-formula φ, we want to establish if there exists a finite model **M** and an interval $[d_i, d_j]$ in it such that $\mathbf{M}, [d_i, d_j] \Vdash \varphi$
- initial satisfiability
- tableaux rooted in φ
    - expansion rules
    - model production

Branch Managing

- a *branch* is a list of nodes
  - enriched with an integer for the size of the associated model

Branch Managing

- a *branch* is a list of nodes
  - enriched with an integer for the size of the associated model
- *nodes* are annotated formulas
  - (sub-)formula
  - interval
  - and a Boolean flag (node active or inactive)

Branch Managing

- a *branch* is a list of nodes
  - enriched with an integer for the size of the associated model
- *nodes* are annotated formulas
  - (sub-)formula
  - interval
  - and a Boolean flag (node active or inactive)
- a node (its labeled formula $\varphi, [d_i, d_j]$) is *active* if
  - not a box-formula: rule has not been already applied to it
  - box-formula: a new point added since the last rule application

Branch Managing

- a *branch* is a list of nodes
  - enriched with an integer for the size of the associated model
- *nodes* are annotated formulas
  - (sub-)formula
  - interval
  - and a Boolean flag (node active or inactive)
- a node (its labeled formula $\varphi, [d_i, d_j]$) is *active* if
  - not a box-formula: rule has not been already applied to it
  - box-formula: a new point added since the last rule application
- a branch is *closed* if
  - contradiction, or
  - model too large (small-model theorem)

Branch Managing

- a *branch* is a list of nodes
  - enriched with an integer for the size of the associated model
- *nodes* are annotated formulas
  - (sub-)formula
  - interval
  - and a Boolean flag (node active or inactive)
- a node (its labeled formula $\varphi, [d_i, d_j]$) is *active* if
  - not a box-formula: rule has not been already applied to it
  - box-formula: a new point added since the last rule application
- a branch is *closed* if
  - contradiction, or
  - model too large (small-model theorem)
- otherwise, the branch is *open*

Branch Managing

- a *branch* is a list of nodes
  - enriched with an integer for the size of the associated model
- *nodes* are annotated formulas
  - (sub-)formula
  - interval
  - and a Boolean flag (node active or inactive)
- a node (its labeled formula $\varphi, [d_i, d_j]$) is *active* if
  - not a box-formula: rule has not been already applied to it
  - box-formula: a new point added since the last rule application
- a branch is *closed* if
  - contradiction, or
  - model too large (small-model theorem)
- otherwise, the branch is *open*
- rules are applied only to active formulas in open branches

Branch Managing

- a *branch* is a list of nodes
  - enriched with an integer for the size of the associated model
- *nodes* are annotated formulas
  - (sub-)formula
  - interval
  - and a Boolean flag (node active or inactive)
- a node (its labeled formula $\varphi, [d_i, d_j]$) is *active* if
  - not a box-formula: rule has not been already applied to it
  - box-formula: a new point added since the last rule application
- a branch is *closed* if
  - contradiction, or
  - model too large (small-model theorem)
- otherwise, the branch is *open*
- rules are applied only to active formulas in open branches
- no more active formulas on a open branch: *formula satisfiable*
  - open branch represent models

Branch Managing

- a *branch* is a list of nodes
  - enriched with an integer for the size of the associated model
- *nodes* are annotated formulas
  - (sub-)formula
  - interval
  - and a Boolean flag (node active or inactive)
- a node (its labeled formula $\varphi, [d_i, d_j]$) is *active* if
  - not a box-formula: rule has not been already applied to it
  - box-formula: a new point added since the last rule application
- a branch is *closed* if
  - contradiction, or
  - model too large (small-model theorem)
- otherwise, the branch is *open*
- rules are applied only to active formulas in open branches
- no more active formulas on a open branch: *formula satisfiable*
  - open branch represent models
- all branches are closed: *formula unsatisfiable*

Expansion rules

- classical operators are treated in the standard way
- modal operators (diamond and box rules)

Expansion rules

- classical operators are treated in the standard way
- modal operators (diamond and box rules)

$$(box)\ \frac{[A]\psi : [d_i, d_j]}{\psi : [d_j, d_{j+1}], \ldots, \psi : [d_j, d_N]},$$

$$(dia)\ \frac{\langle A\rangle\psi : [d_i, d_j]}{\psi : [d_j, d_{j+1}] \mid \ldots \mid \psi : [d_j, d_N] \mid \psi : [d_j, d_j'] \mid \ldots \mid \psi : [d_j, d_N']},$$

Expansion rules

- classical operators are treated in the standard way
- modal operators (diamond and box rules)

$$(box)\ \frac{[A]\psi : [d_i, d_j]}{\psi : [d_j, d_{j+1}], \ldots, \psi : [d_j, d_N]},$$

$$(dia)\ \frac{\langle A\rangle\psi : [d_i, d_j]}{\psi : [d_j, d_{j+1}] \mid \ldots \mid \psi : [d_j, d_N] \mid \psi : [d_j, d'_j] \mid \ldots \mid \psi : [d_j, d'_N]},$$

- d_h ($j \leq h \leq N$) is a point in D
- d'_h ($j \leq h \leq N$) is a new point added to D ($d_h < d'_h d_{h+1}$)

Search Procedure

- initial tableau: 1 branch, 1 node, labeled with $\langle \varphi, [d_0, d_1] \rangle$

Search Procedure

- initial tableau: 1 branch, 1 node, labeled with $\langle \varphi, [d_0, d_1] \rangle$
- generic step

Search Procedure

- initial tableau: 1 branch, 1 node, labeled with $\langle \varphi, [d_0, d_1] \rangle$
- generic step
  - selects the open branch with highest priority
(several policies implemented)

Search Procedure

- initial tableau: 1 branch, 1 node, labeled with $\langle \varphi, [d_0, d_1] \rangle$
- generic step
  - selects the open branch with highest priority (several policies implemented)
  - if there are no active nodes, the formula is satisfiable

Search Procedure

- initial tableau: 1 branch, 1 node, labeled with $\langle \varphi, [d_0, d_1] \rangle$
- generic step
  - selects the open branch with highest priority (several policies implemented)
  - if there are no active nodes, the formula is satisfiable
  - otherwise applies the expansion rule to the closest-to-the-root active node

Search Procedure

- initial tableau: 1 branch, 1 node, labeled with $\langle \varphi, [d_0, d_1] \rangle$
- generic step
  - selects the open branch with highest priority (several policies implemented)
  - if there are no active nodes, the formula is satisfiable
  - otherwise applies the expansion rule to the closest-to-the-root active node
  - the application of expansions modifies the tableau

Search Procedure

- initial tableau: 1 branch, 1 node, labeled with $\langle \varphi, [d_0, d_1] \rangle$
- generic step
  - selects the open branch with highest priority (several policies implemented)
  - if there are no active nodes, the formula is satisfiable
  - otherwise applies the expansion rule to the closest-to-the-root active node
  - the application of expansions modifies the tableau
  - if all branches are deleted, the formula is unsatisfiable

Search Procedure

- initial tableau: 1 branch, 1 node, labeled with $\langle \varphi, [d_0, d_1] \rangle$
- generic step
  - selects the open branch with highest priority (several policies implemented)
  - if there are no active nodes, the formula is satisfiable
  - otherwise applies the expansion rule to the closest-to-the-root active node
  - the application of expansions modifies the tableau
  - if all branches are deleted, the formula is unsatisfiable
  - otherwise repeat the procedure

Priority Policies

To determine the priority of a branch in the queue, we implemented four different (and complete) policies:

Priority Policies

To determine the priority of a branch in the queue, we implemented four different (and complete) policies:

- FIFO: the standard first-in-first-out
- LDF: largest domain branches are expanded first
- SDF: smallest domain branches are expanded first
- GAN: branches with greatest number of active nodes are expanded first
- SDF: branches with smallest number of active nodes are expanded first

Formula representation

- input formula φ: conjunction of (sub-)formulas
  - input file: a line per each conjunct

- transformed into *negated normal form*

- stored into a *syntactic tree* (leaves are atomic propositions, and internal nodes are Boolean or modal operators).

Example

$$\boxed{\varphi = \langle A\rangle p \wedge [A](\neg p \vee \langle A\rangle p)}$$

$\varphi : [0, 1]$

Example

$$\boxed{\varphi = \langle A \rangle p \land [A](\neg p \lor \langle A \rangle p)}$$

→ $\varphi : [0,1]$

$\langle A \rangle p : [0,1]$

$[A](\neg p \lor \langle A \rangle p) : [0,1]$

Example

$$\boxed{\varphi = \langle A \rangle p \wedge [A](\neg p \vee \langle A \rangle p)}$$

$$\xcancel{\varphi : [0, 1]}$$

$$|$$

$$\rightarrow \langle A \rangle p : [0, 1]$$

$$|$$

$$[A](\neg p \vee \langle A \rangle p) : [0, 1]$$

$$|$$

$$p : [1, 2]$$

Example

$$\boxed{\varphi = \langle A \rangle p \wedge [A](\neg p \vee \langle A \rangle p)}$$

$\xcancel{\varphi : [0,1]}$

|

$\xcancel{\langle A \rangle p : [0,1]}$

|

→ $[A](\neg p \vee \langle A \rangle p) : [0,1]$

|

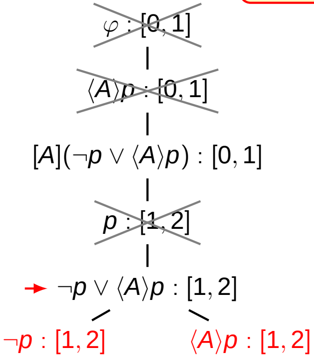
$\xcancel{p : [1,2]}$

|

$\neg p \vee \langle A \rangle p : [1,2]$

Example

$$\varphi = \langle A \rangle p \wedge [A](\neg p \vee \langle A \rangle p)$$



$\xcancel{\varphi : [0, 1]}$

$\xcancel{\langle A \rangle p : [0, 1]}$

$[A](\neg p \vee \langle A \rangle p) : [0, 1]$

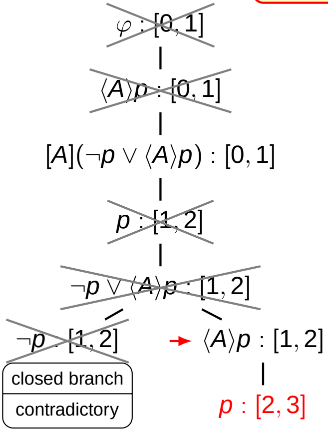
$\xcancel{p : [1, 2]}$

→ $\neg p \vee \langle A \rangle p : [1, 2]$

$\neg p : [1, 2]$ $\langle A \rangle p : [1, 2]$

Example

$\varphi = \langle A \rangle p \wedge [A](\neg p \vee \langle A \rangle p)$



~~$\varphi : [0, 1]$~~

~~$\langle A \rangle p : [0, 1]$~~

$[A](\neg p \vee \langle A \rangle p) : [0, 1]$

~~$p : [1, 2]$~~

~~$\neg p \vee \langle A \rangle p : [1, 2]$~~

~~$\neg p : [1, 2]$~~

closed branch

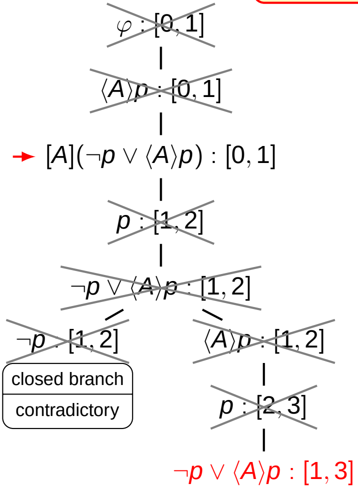
contradictory

→ $\langle A \rangle p : [1, 2]$

$p : [2, 3]$

Example

$\varphi = \langle A\rangle p \wedge [A](\neg p \vee \langle A\rangle p)$



~~$\varphi : [0,1]$~~

~~$\langle A\rangle p : [0,1]$~~

→ $[A](\neg p \vee \langle A\rangle p) : [0,1]$

~~$p : [1,2]$~~

~~$\neg p \vee \langle A\rangle p : [1,2]$~~

~~$\neg p : [1,2]$~~

closed branch

contradictory

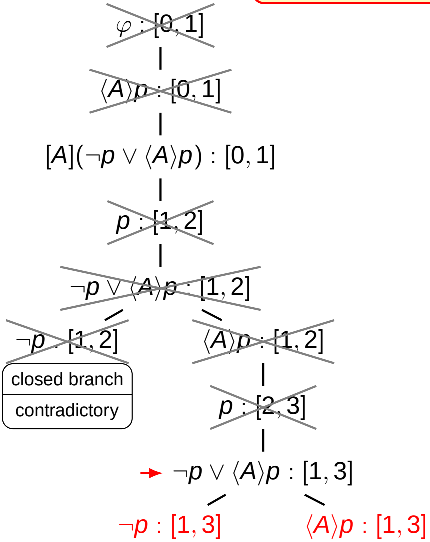
~~$\langle A\rangle p : [1,2]$~~

~~$p : [2,3]$~~

$\neg p \vee \langle A\rangle p : [1,3]$

Example

$\varphi = \langle A\rangle p \wedge [A](\neg p \vee \langle A\rangle p)$



~~$\varphi : [0, 1]$~~

~~$\langle A\rangle p : [0, 1]$~~

$[A](\neg p \vee \langle A\rangle p) : [0, 1]$

~~$p : [1, 2]$~~

~~$\neg p \vee \langle A\rangle p : [1, 2]$~~

~~$\neg p : [1, 2]$~~

closed branch

contradictory

~~$\langle A\rangle p : [1, 2]$~~

~~$p : [2, 3]$~~

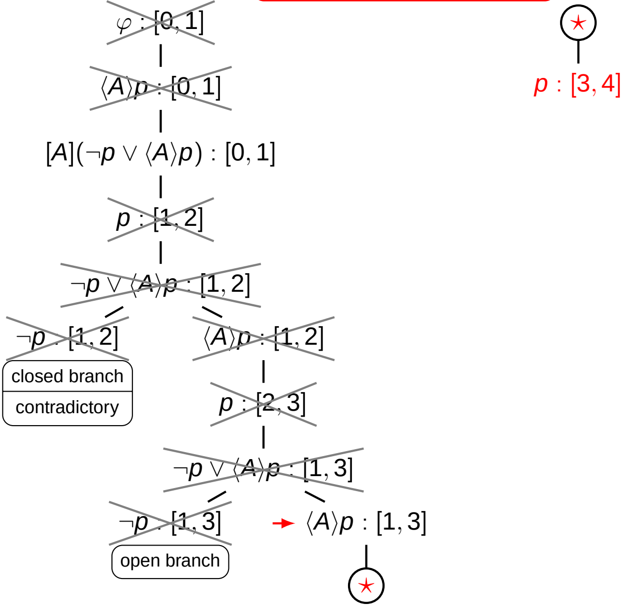
$\neg p \vee \langle A\rangle p : [1, 3]$

$\neg p : [1, 3]$

$\langle A\rangle p : [1, 3]$

Example

$\varphi = \langle A \rangle p \wedge [A](\neg p \vee \langle A \rangle p)$



~~$\varphi : [0,1]$~~

~~$\langle A \rangle p : [0,1]$~~

$[A](\neg p \vee \langle A \rangle p) : [0,1]$

~~$p : [1,2]$~~

~~$\neg p \vee \langle A \rangle p : [1,2]$~~

~~$\neg p : [1,2]$~~

closed branch

contradictory

~~$\langle A \rangle p : [1,2]$~~

~~$p : [2,3]$~~

~~$\neg p \vee \langle A \rangle p : [1,3]$~~

~~$\neg p : [1,3]$~~

open branch

→ $\langle A \rangle p : [1,3]$

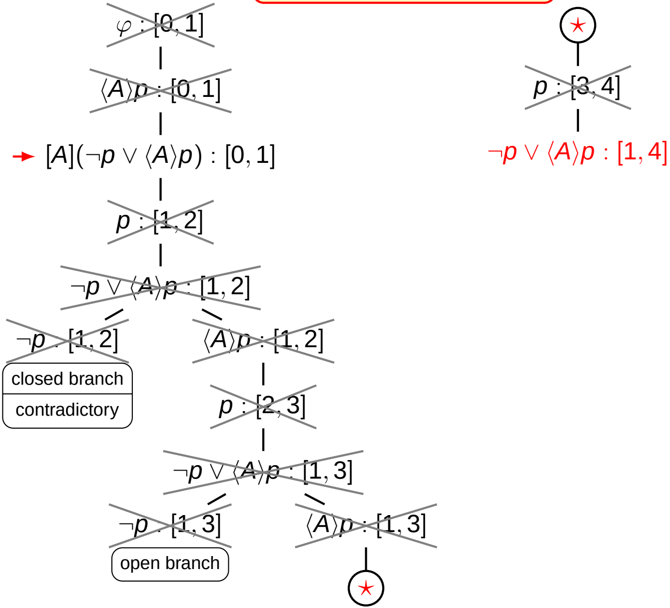
⋆

⋆

$p : [3,4]$

Example

$\varphi = \langle A \rangle p \wedge [A](\neg p \vee \langle A \rangle p)$



~~$\varphi : [0,1]$~~

~~$\langle A \rangle p : [0,1]$~~

$[A](\neg p \vee \langle A \rangle p) : [0,1]$

~~$p : [1,2]$~~

~~$\neg p \vee \langle A \rangle p : [1,2]$~~

~~$\neg p : [1,2]$~~

closed branch

contradictory

~~$\langle A \rangle p : [1,2]$~~

~~$p : [2,3]$~~

~~$\neg p \vee \langle A \rangle p : [1,3]$~~

~~$\neg p : [1,3]$~~

open branch

~~$\langle A \rangle p : [1,3]$~~

$\star$

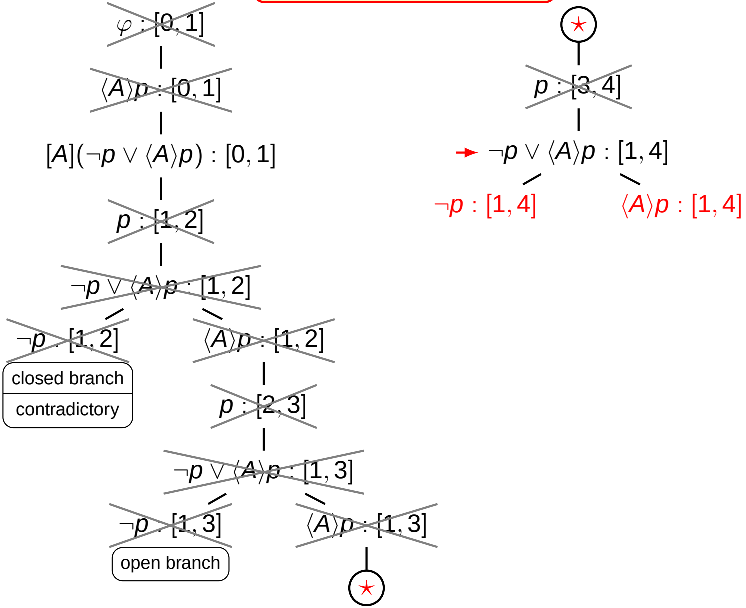
$\star$

~~$p : [3,4]$~~

$\neg p \vee \langle A \rangle p : [1,4]$

Example

$\varphi = \langle A\rangle p \wedge [A](\neg p \vee \langle A\rangle p)$



~~$\varphi : [0,1]$~~

~~$\langle A\rangle p : [0,1]$~~

$[A](\neg p \vee \langle A\rangle p) : [0,1]$

~~$p : [1,2]$~~

~~$\neg p \vee \langle A\rangle p : [1,2]$~~

~~$\neg p : [1,2]$~~ ~~$\langle A\rangle p : [1,2]$~~

closed branch

contradictory

~~$p : [2,3]$~~

~~$\neg p \vee \langle A\rangle p : [1,3]$~~

~~$\neg p : [1,3]$~~ ~~$\langle A\rangle p : [1,3]$~~

open branch

⋆

⋆

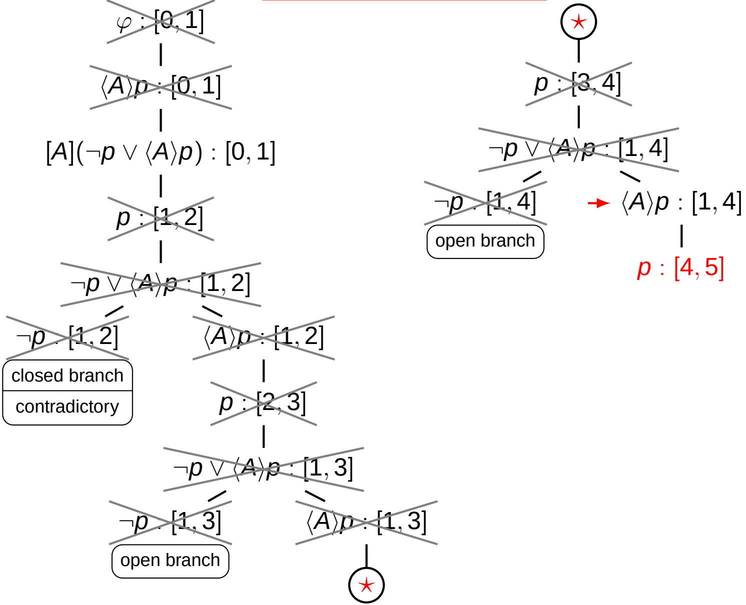
~~$p : [3,4]$~~

→ $\neg p \vee \langle A\rangle p : [1,4]$

$\neg p : [1,4]$ $\langle A\rangle p : [1,4]$

Example

$\varphi = \langle A\rangle p \wedge [A](\neg p \vee \langle A\rangle p)$



~~$\varphi : [0,1]$~~

~~$\langle A\rangle p : [0,1]$~~

$[A](\neg p \vee \langle A\rangle p) : [0,1]$

~~$p : [1,2]$~~

~~$\neg p \vee \langle A\rangle p : [1,2]$~~

~~$\neg p : [1,2]$~~

closed branch

contradictory

~~$\langle A\rangle p : [1,2]$~~

~~$p : [2,3]$~~

~~$\neg p \vee \langle A\rangle p : [1,3]$~~

~~$\neg p : [1,3]$~~

open branch

~~$\langle A\rangle p : [1,3]$~~

$\star$

$\star$

~~$p : [3,4]$~~

~~$\neg p \vee \langle A\rangle p : [1,4]$~~

~~$\neg p : [1,4]$~~

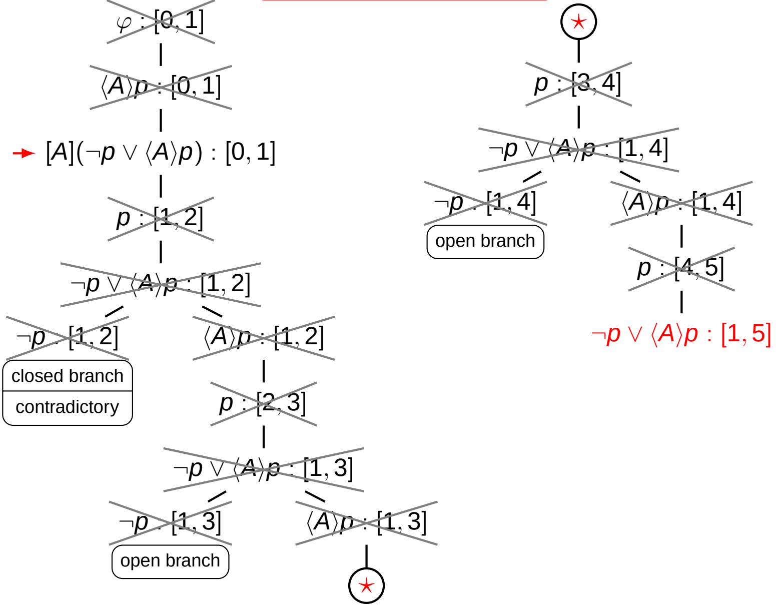
open branch

→ $\langle A\rangle p : [1,4]$

$p : [4,5]$

Example

$\varphi = \langle A \rangle p \wedge [A](\neg p \vee \langle A \rangle p)$



$\xcancel{\varphi : [0, 1]}$

$\xcancel{\langle A \rangle p : [0, 1]}$

→ $[A](\neg p \vee \langle A \rangle p) : [0, 1]$

$\xcancel{p : [1, 2]}$

$\xcancel{\neg p \vee \langle A \rangle p : [1, 2]}$

$\xcancel{\neg p : [1, 2]}$

closed branch

contradictory

$\xcancel{\langle A \rangle p : [1, 2]}$

$\xcancel{p : [2, 3]}$

$\xcancel{\neg p \vee \langle A \rangle p : [1, 3]}$

$\xcancel{\neg p : [1, 3]}$

open branch

$\xcancel{\langle A \rangle p : [1, 3]}$

$\star$

$\star$

$\xcancel{p : [3, 4]}$

$\xcancel{\neg p \vee \langle A \rangle p : [1, 4]}$

$\xcancel{\neg p : [1, 4]}$

open branch

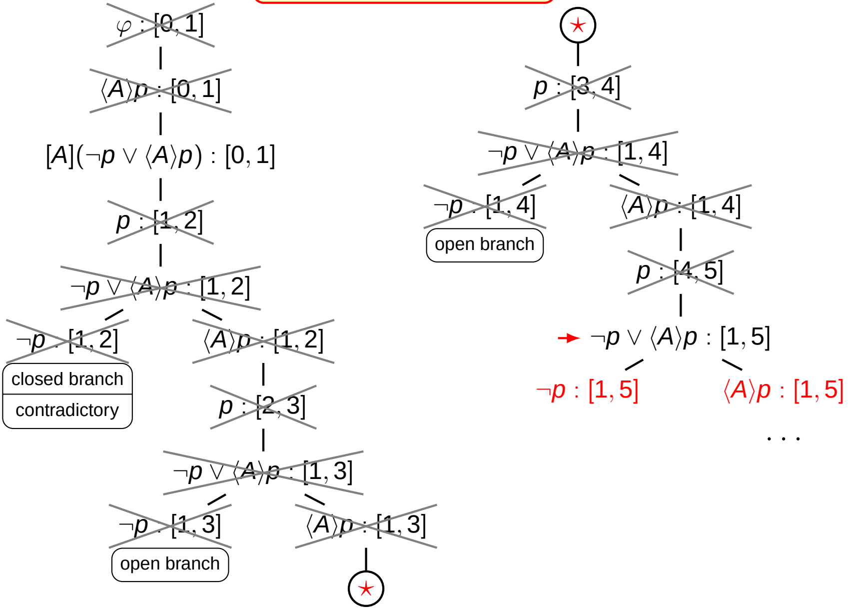
$\xcancel{\langle A \rangle p : [1, 4]}$

$\xcancel{p : [4, 5]}$

$\neg p \vee \langle A \rangle p : [1, 5]$

Example

$\varphi = \langle A \rangle p \wedge [A](\neg p \vee \langle A \rangle p)$



$\xcancel{\varphi : [0,1]}$

$\xcancel{\langle A \rangle p : [0,1]}$

$[A](\neg p \vee \langle A \rangle p) : [0,1]$

$\xcancel{p : [1,2]}$

$\xcancel{\neg p \vee \langle A \rangle p : [1,2]}$

$\xcancel{\neg p : [1,2]}$

closed branch

contradictory

$\xcancel{\langle A \rangle p : [1,2]}$

$\xcancel{p : [2,3]}$

$\xcancel{\neg p \vee \langle A \rangle p : [1,3]}$

$\xcancel{\neg p : [1,3]}$

open branch

$\xcancel{\langle A \rangle p : [1,3]}$

$\star$

$\star$

$\xcancel{p : [3,4]}$

$\xcancel{\neg p \vee \langle A \rangle p : [1,4]}$

$\xcancel{\neg p : [1,4]}$

open branch

$\xcancel{\langle A \rangle p : [1,4]}$

$\xcancel{p : [4,5]}$

$\rightarrow \neg p \vee \langle A \rangle p : [1,5]$

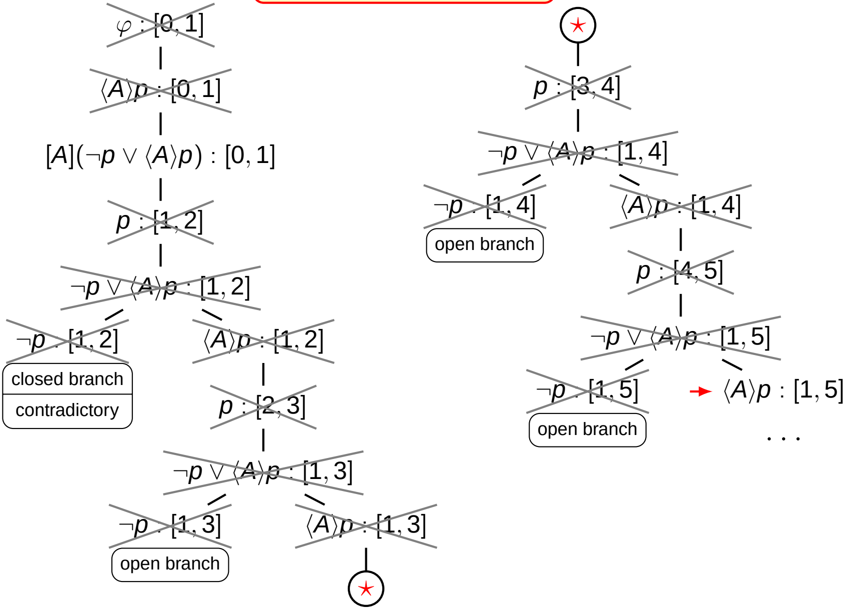
$\neg p : [1,5]$

$\langle A \rangle p : [1,5]$

$\dots$

Example

$\varphi = \langle A\rangle p \wedge [A](\neg p \vee \langle A\rangle p)$



~~$\varphi : [0,1]$~~

~~$\langle A\rangle p : [0,1]$~~

$[A](\neg p \vee \langle A\rangle p) : [0,1]$

~~$p : [1,2]$~~

~~$\neg p \vee \langle A\rangle p : [1,2]$~~

~~$\neg p : [1,2]$~~

closed branch

contradictory

~~$\langle A\rangle p : [1,2]$~~

~~$p : [2,3]$~~

~~$\neg p \vee \langle A\rangle p : [1,3]$~~

~~$\neg p : [1,3]$~~

open branch

~~$\langle A\rangle p : [1,3]$~~

$\star$

$\star$

~~$p : [3,4]$~~

~~$\neg p \vee \langle A\rangle p : [1,4]$~~

~~$\neg p : [1,4]$~~

open branch

~~$\langle A\rangle p : [1,4]$~~

~~$p : [4,5]$~~

~~$\neg p \vee \langle A\rangle p : [1,5]$~~

~~$\neg p : [1,5]$~~

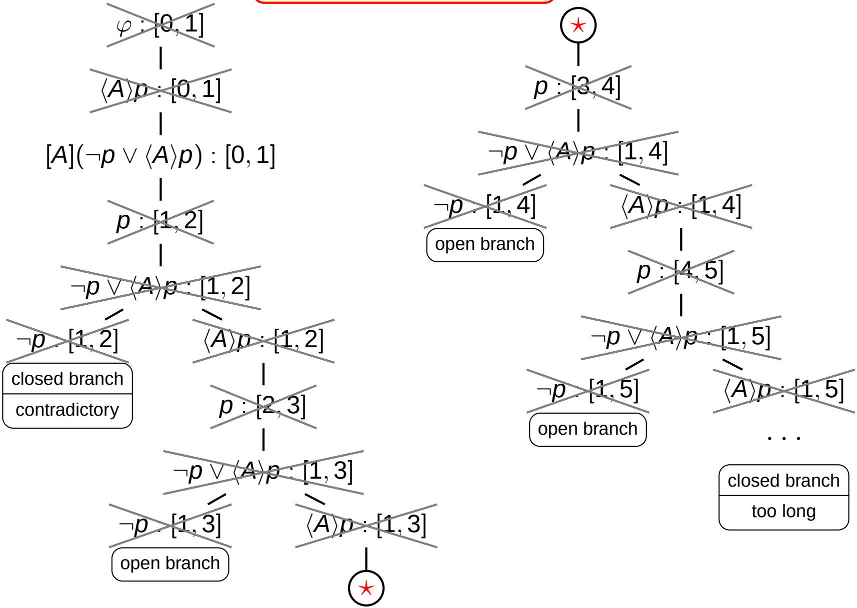
open branch

→ $\langle A\rangle p : [1,5]$

. . .

Example

$\varphi = \langle A \rangle p \wedge [A](\neg p \vee \langle A \rangle p)$



~~$\varphi : [0,1]$~~

~~$\langle A \rangle p : [0,1]$~~

$[A](\neg p \vee \langle A \rangle p) : [0,1]$

~~$p : [1,2]$~~

~~$\neg p \vee \langle A \rangle p : [1,2]$~~

~~$\neg p : [1,2]$~~

closed branch

contradictory

~~$\langle A \rangle p : [1,2]$~~

~~$p : [2,3]$~~

~~$\neg p \vee \langle A \rangle p : [1,3]$~~

~~$\neg p : [1,3]$~~

open branch

~~$\langle A \rangle p : [1,3]$~~

$\star$

$\star$

~~$p : [3,4]$~~

~~$\neg p \vee \langle A \rangle p : [1,4]$~~

~~$\neg p : [1,4]$~~

open branch

~~$\langle A \rangle p : [1,4]$~~

~~$p : [4,5]$~~

~~$\neg p \vee \langle A \rangle p : [1,5]$~~

~~$\neg p : [1,5]$~~

open branch

~~$\langle A \rangle p : [1,5]$~~

$\ldots$

closed branch

too long

Benchmark

Tests on benchmarks:

- *combinatorics* (to test scalability)
  - the n-th combinatorial problem
    - n conjuncts $\langle A \rangle p_i$ $(0 \leq i \leq n)$
    - $\frac{n(n+1)}{2}$ formulas $[A]\neg(p_i \wedge p_j)$ $(i \neq j)$
- *randomized* (to simulate the behaviour in real cases)
  - 36 completely random problems

Experimental Results - combinatorics

COMBINATORICS

	Policy (sec)					Outcome
n	FIFO	SDF	LDF	SAN	GAN	(size)
1	0.004	0.004	0.004	0.004	0.004	4
2	0.004	0.008	0.004	0.004	0.008	5
3	0.008	0.15	0.03	0.008	0.03	6
4	0.01	–	30.07	0.01	30.29	7
5	0.012	–	–	0.012	–	8
6	0.02	–	–	0.03	–	9
7	0.07	–	–	0.07	–	10
8	0.15	–	–	0.16	–	11
9	0.3	–	–	0.32	–	12
10	0.56	–	–	0.59	–	13
11	0.99	–	–	1.06	–	14
12	1.67	–	–	1.79	–	15
13	2.73	–	–	2.94	–	16
14	4.25	–	–	4.55	–	17
15	6.56	–	–	7.08	–	18
16	9.77	–	–	10.82	–	19
17	14.42	–	–	15.40	–	20
18	20.79	–	–	22.20	–	21
19	29.28	–	–	32.11	–	22
20	40.91	–	–	44.09	–	23
21	–	–	–	–	–	–
22	–	–	–	–	–	–

Experimental Results - randomized

RANDOMIZED

	Policy (sec)					Outcome
n	FIFO	SDF	LDF	SAN	GAN	(size)
1	0.004	0.004	0.004	0.004	0.004	4
2	0.004	0.004	0.004	0.004	0.004	4
3	0.004	0.004	0.004	0.004	0.004	4
4	0.004	0.004	0.004	0.004	0.004	4
5	0.004	0.004	0.004	0.004	0.004	4
6	0.004	0.004	0.004	0.004	0.004	4
7	0.07	0.23	0.004	0.18	0.004	3 / 4
8	0.004	0.004	0.004	0.004	0.004	4
9	0.004	0.004	0.004	0.004	0.004	4
10	0.004	0.004	0.004	0.004	0.004	4
11	0.004	0.004	0.004	0.004	0.004	4
12	0.004	0.004	0.004	0.004	0.004	4
13	0.01	0.04	0.004	0.02	0.004	4
14	0.004	0.004	0.004	0.004	0.004	4
15	0.004	0.004	0.004	0.004	0.004	4
16	0.004	1.37	0.004	0.01	0.004	4
17	0.004	0.004	0.004	0.004	0.004	4
18	0.004	0.004	0.004	0.004	0.004	3

	Policy (sec)					Outcome
n	FIFO	SDF	LDF	SAN	GAN	(size)
19	1.66	45.43	0.68	1.91	0.02	3 / 4
20	0.02	0.004	0.03	0.03	0.004	2 / 4
21	0.004	0.004	0.004	0.004	0.004	4
22	0.74	14.08	0.004	1.04	0.004	4
23	0.004	0.004	0.004	0.004	0.004	4
24	0.004	0.004	0.004	0.004	0.004	4
25	–	–	–	–	–	–
26	0.004	0.004	0.004	0.004	0.004	4
27	0.004	–	0.004	0.01	–	3 / 4
28	0.004	0.004	0.004	0.004	0.004	4
29	0.004	–	0.004	0.004	0.004	4
30	0.14	0.08	0.04	0.19	0.01	2 / 4
31	0.004	0.004	0.004	0.004	0.004	unsat
32	0.25	–	0.02	0.31	0.004	2 / 4
33	0.004	0.004	0.004	0.004	0.004	4
34	–	–	0.02	0.004	0.02	2 / 4
35	0.004	–	0.004	–	0.004	2 / 4
36	–	–	–	–	1.2	3

Outline

1. Introduction on Interval Temporal Logic
2. Halpern-Shoham’s modal logic HS
3. The tableau method for the logic A
4. Conclusions

Conclusions

Interval temporal logics

- a lot of theoretical results in the last 10 years
    - decidability, undecidability, expressiveness issues
- a first step towards an implementation
- the system is available at the page

`http://www.di.unisa.it/dottorandi/dario.dellamonica/tableaux/`

Future directions

- more expressive languages
- different classes of linear orders
- generation of a proper benchmark
- use of generic theorem prover (Mettel?)

The end

Thank you