A Guided Tour through Interval Temporal Logics
Lecture 6: Interval logics: decidability

Angelo Montanari

Department of Mathematics and Computer Science,
University of Udine, Italy
angelo.montanari@uniud.it

Gargnano, August 20-25, 2012

Interval logics: are they all undecidable?

Most fragments of HS (and CDT) are undecidable on most of the interesting classes of interval structures.

Still, some pockets of decidability remain.

Various decidability results are based on some semantic restrictions reducing the interval-based semantics to point-based.

But there are some quite non-trivial decidable cases of interval logics with genuinely interval-based semantics.

This lecture will give an overview of decidable interval logics.

Interval logics: are they all undecidable?

Most fragments of HS (and CDT) are undecidable on most of the interesting classes of interval structures.

Still, some pockets of decidability remain.

Various decidability results are based on some semantic restrictions reducing the interval-based semantics to point-based.

But there are some quite non-trivial decidable cases of interval logics with genuinely interval-based semantics.

This lecture will give an overview of decidable interval logics.

Interval logics: are they all undecidable?

Most fragments of HS (and CDT) are undecidable on most of the interesting classes of interval structures.

Still, some pockets of decidability remain.

Various decidability results are based on some semantic restrictions reducing the interval-based semantics to point-based.

But there are some quite non-trivial decidable cases of interval logics with genuinely interval-based semantics.

This lecture will give an overview of decidable interval logics.

Interval logics: are they all undecidable?

Most fragments of HS (and CDT) are undecidable on most of the interesting classes of interval structures.

Still, some pockets of decidability remain.

Various decidability results are based on some semantic restrictions reducing the interval-based semantics to point-based.

But there are some quite non-trivial decidable cases of interval logics with genuinely interval-based semantics.

This lecture will give an overview of decidable interval logics.

Interval logics: are they all undecidable?

Most fragments of HS (and CDT) are undecidable on most of the interesting classes of interval structures.

Still, some pockets of decidability remain.

Various decidability results are based on some semantic restrictions reducing the interval-based semantics to point-based.

But there are some quite non-trivial decidable cases of interval logics with genuinely interval-based semantics.

This lecture will give an overview of decidable interval logics.

Interval logics: are they all undecidable?

Most fragments of HS (and CDT) are undecidable on most of the interesting classes of interval structures.

Still, some pockets of decidability remain.

Various decidability results are based on some semantic restrictions reducing the interval-based semantics to point-based.

But there are some quite non-trivial decidable cases of interval logics with genuinely interval-based semantics.

This lecture will give an overview of decidable interval logics.

Decidability of interval logics by semantic restrictions

Some strong semantic restrictions can restore decidability:

- Locality: an atomic proposition is true over an interval if and only if it is true at its starting point.
- Homogeneity: an atomic proposition is true over an interval if and only if it is true at every subinterval / every point in that interval.
- Convexity: if a formula is true on two overlapping intervals, then it is true on their union.
- Considering incomplete interval structures, e.g., split structures, where every interval can be chopped in a unique way (thus creating a tree-like subinterval structure).

Decidability of interval logics by semantic restrictions

Some strong semantic restrictions can restore decidability:

- Locality: an atomic proposition is true over an interval if and only if it is true at its starting point.
- Homogeneity: an atomic proposition is true over an interval if and only if it is true at every subinterval / every point in that interval.
- Convexity: if a formula is true on two overlapping intervals, then it is true on their union.
- Considering incomplete interval structures, e.g., split structures, where every interval can be chopped in a unique way (thus creating a tree-like subinterval structure).

Decidability of interval logics by semantic restrictions

Some strong semantic restrictions can restore decidability:

- Locality: an atomic proposition is true over an interval if and only if it is true at its starting point.
- Homogeneity: an atomic proposition is true over an interval if and only if it is true at every subinterval / every point in that interval.
- Convexity: if a formula is true on two overlapping intervals, then it is true on their union.
- Considering incomplete interval structures, e.g., split structures, where every interval can be chopped in a unique way (thus creating a tree-like subinterval structure).

Decidability of interval logics by semantic restrictions

Some strong semantic restrictions can restore decidability:

- Locality: an atomic proposition is true over an interval if and only if it is true at its starting point.
- Homogeneity: an atomic proposition is true over an interval if and only if it is true at every subinterval / every point in that interval.
- Convexity: if a formula is true on two overlapping intervals, then it is true on their union.
- Considering incomplete interval structures, e.g., split structures, where every interval can be chopped in a unique way (thus creating a tree-like subinterval structure).

Decidability of interval logics by semantic restrictions

Some strong semantic restrictions can restore decidability:

- Locality: an atomic proposition is true over an interval if and only if it is true at its starting point.
- Homogeneity: an atomic proposition is true over an interval if and only if it is true at every subinterval / every point in that interval.
- Convexity: if a formula is true on two overlapping intervals, then it is true on their union.
- Considering incomplete interval structures, e.g., split structures, where every interval can be chopped in a unique way (thus creating a tree-like subinterval structure).

Some strong semantic restrictions can restore decidability:

- Locality: an atomic proposition is true over an interval if and only if it is true at its starting point.
- Homogeneity: an atomic proposition is true over an interval if and only if it is true at every subinterval / every point in that interval.
- Convexity: if a formula is true on two overlapping intervals, then it is true on their union.
- Considering incomplete interval structures, e.g., split structures, where every interval can be chopped in a unique way (thus creating a tree-like subinterval structure).

Decidability of interval logics: a few easy cases

The above-mentioned decidability results have been obtained by imposing drastic syntactic restrictions. The resulting logics can be hardly considered genuine interval logics.

There exist a few cases of (genuine) interval logics which can be easily checked:

- the fragment $B\overline{B}$ of HS is essentially point-based, because the left endpoint of the current interval remains fixed;
- similarly, the fragment $E\overline{E}$ of HS is essentially point-based, because the left endpoint of the current interval remains fixed.

Decidability of interval logics: a few easy cases

The above-mentioned decidability results have been obtained by imposing drastic syntactic restrictions. The resulting logics can be hardly considered genuine interval logics.

There exist a few cases of (genuine) interval logics which can be easily checked:

- the fragment $B\overline{B}$ of HS is essentially point-based, because the left endpoint of the current interval remains fixed;
- similarly, the fragment $E\overline{E}$ of HS is essentially point-based, because the left endpoint of the current interval remains fixed.

Decidability of interval logics: a few easy cases

The above-mentioned decidability results have been obtained by imposing drastic syntactic restrictions. The resulting logics can be hardly considered genuine interval logics.

There exist a few cases of (genuine) interval logics which can be easily checked:

- the fragment $B\overline{B}$ of HS is essentially point-based, because the left endpoint of the current interval remains fixed;
- similarly, the fragment $E\overline{E}$ of HS is essentially point-based, because the left endpoint of the current interval remains fixed.

Decidability of interval logics: a few easy cases

The above-mentioned decidability results have been obtained by imposing drastic syntactic restrictions. The resulting logics can be hardly considered genuine interval logics.

There exist a few cases of (genuine) interval logics which can be easily checked:

- the fragment $B\overline{B}$ of HS is essentially point-based, because the left endpoint of the current interval remains fixed;
- similarly, the fragment $E\overline{E}$ of HS is essentially point-based, because the left endpoint of the current interval remains fixed.

Decidability of interval neighborhood logics via reduction to $FO^2[<]$ - 1

Recall that Propositional Neighborhood Logic $A\overline{A}^{\pi+}$ is expressively complete for $FO^2[<]$ (the first-order language with 2 variables, any set of uninterpreted binary relations, =, and a linear order $<$).

D. Bresolin, V. Goranko, A. Montanari, and G. Sciavicco, *Propositional interval neighborhood logics: Expressiveness, decidability, and undecidable extensions*, Annals of Pure and Applied Logic, 161(3):289–304, 2009

Satisfiability of formulae in $FO^2[<]$ was first proved decidable (NEXPTIME-complete) by Martin Otto on various classes of linear orders.

M. Otto, *Two Variable First-order Logic Over Ordered Domains*, Journal of Symbolic Logic, 66(2):685–702, 2001

Decidability of interval neighborhood logics via reduction to $FO^2[<]$ - 1

Recall that Propositional Neighborhood Logic $A\overline{A}^{\pi+}$ is expressively complete for $FO^2[<]$ (the first-order language with 2 variables, any set of uninterpreted binary relations, =, and a linear order $<$).

D. Bresolin, V. Goranko, A. Montanari, and G. Sciavicco, *Propositional interval neighborhood logics: Expressiveness, decidability, and undecidable extensions*, Annals of Pure and Applied Logic, 161(3):289–304, 2009

Satisfiability of formulae in $FO^2[<]$ was first proved decidable (NEXPTIME-complete) by Martin Otto on various classes of linear orders.

M. Otto, *Two Variable First-order Logic Over Ordered Domains*, Journal of Symbolic Logic, 66(2):685–702, 2001

Decidability of interval neighborhood logics via reduction to FO$^2[<]$ - 1

Recall that Propositional Neighborhood Logic $A\overline{A}^{\pi+}$ is expressively complete for FO$^2[<]$ (the first-order language with 2 variables, any set of uninterpreted binary relations, =, and a linear order $<$).

D. Bresolin, V. Goranko, A. Montanari, and G. Sciavicco, *Propositional interval neighborhood logics: Expressiveness, decidability, and undecidable extensions*, Annals of Pure and Applied Logic, 161(3):289–304, 2009

Satisfiability of formulae in FO$^2[<]$ was first proved decidable (NEXPTIME-complete) by Martin Otto on various classes of linear orders.

M. Otto, *Two Variable First-order Logic Over Ordered Domains*, Journal of Symbolic Logic, 66(2):685–702, 2001

Decidability of interval neighborhood logics via reduction to $FO^2[<]$ - 2

More precisely, Otto proved the decidability of the satisfiability problem for $FO^2[<]$ on the classes of all linear orders, of well-orders, and of finite linear orders, as well as on the linear order on the natural numbers.

His proofs are based on an elaborated model-theoretic argument, analyzing the types of elements and pairs in models of $FO^2[<]$.

Decidability of $A\overline{A}^{\pi+}$ on each of these classes immediately follows.

Moreover, since $A\overline{A}^{+}$ and $A\overline{A}^{-}$ are strictly less expressive than $A\overline{A}^{\pi+}$, their decidability on the same classes of structures immediately follows ($A\overline{A}^{-}$ can be naturally embedded into $A\overline{A}^{\pi+}$ via the translation clauses $\tau(\langle A\rangle\varphi) = \Diamond_r(\neg\pi \wedge \tau(\varphi))$ and $\tau(\langle\overline{A}\rangle\varphi) = \Diamond_l(\neg\pi \wedge \tau(\varphi))$).

Hereafter, we denote interval logic(s) of temporal neighborhood by PNL.

Decidability of interval neighborhood logics via reduction to $FO^2[<]$ - 2

More precisely, Otto proved the decidability of the satisfiability problem for $FO^2[<]$ on the classes of all linear orders, of well-orders, and of finite linear orders, as well as on the linear order on the natural numbers.

His proofs are based on an elaborated model-theoretic argument, analyzing the types of elements and pairs in models of $FO^2[<]$.

Decidability of $A\overline{A}^{\pi+}$ on each of these classes immediately follows.

Moreover, since $A\overline{A}^{+}$ and $A\overline{A}^{-}$ are strictly less expressive than $A\overline{A}^{\pi+}$, their decidability on the same classes of structures immediately follows ($A\overline{A}^{-}$ can be naturally embedded into $A\overline{A}^{\pi+}$ via the translation clauses $\tau(\langle A\rangle\varphi) = \Diamond_r(\neg\pi \wedge \tau(\varphi))$ and $\tau(\langle \overline{A}\rangle\varphi) = \Diamond_l(\neg\pi \wedge \tau(\varphi))$).

Hereafter, we denote interval logic(s) of temporal neighborhood by PNL.

Decidability of interval neighborhood logics via reduction to $FO^2[<]$ - 2

More precisely, Otto proved the decidability of the satisfiability problem for $FO^2[<]$ on the classes of all linear orders, of well-orders, and of finite linear orders, as well as on the linear order on the natural numbers.

His proofs are based on an elaborated model-theoretic argument, analyzing the types of elements and pairs in models of $FO^2[<]$.

Decidability of $A\overline{A}^{\pi+}$ on each of these classes immediately follows.

Moreover, since $A\overline{A}^{+}$ and $A\overline{A}^{-}$ are strictly less expressive than $A\overline{A}^{\pi+}$, their decidability on the same classes of structures immediately follows ($A\overline{A}^{-}$ can be naturally embedded into $A\overline{A}^{\pi+}$ via the translation clauses $\tau(\langle A\rangle\varphi) = \Diamond_r(\neg\pi \wedge \tau(\varphi))$ and $\tau(\langle\overline{A}\rangle\varphi) = \Diamond_l(\neg\pi \wedge \tau(\varphi))$).

Hereafter, we denote interval logic(s) of temporal neighborhood by PNL.

Decidability of interval neighborhood logics via reduction to $FO^2[<]$ - 2

More precisely, Otto proved the decidability of the satisfiability problem for $FO^2[<]$ on the classes of all linear orders, of well-orders, and of finite linear orders, as well as on the linear order on the natural numbers.

His proofs are based on an elaborated model-theoretic argument, analyzing the types of elements and pairs in models of $FO^2[<]$.

Decidability of $\mathsf{A\overline{A}}^{\pi+}$ on each of these classes immediately follows.

Moreover, since $\mathsf{A\overline{A}}^{+}$ and $\mathsf{A\overline{A}}^{-}$ are strictly less expressive than $\mathsf{A\overline{A}}^{\pi+}$, their decidability on the same classes of structures immediately follows ($\mathsf{A\overline{A}}^{-}$ can be naturally embedded into $\mathsf{A\overline{A}}^{\pi+}$ via the translation clauses $\tau(\langle A\rangle\varphi) = \Diamond_r(\neg\pi \wedge \tau(\varphi))$ and $\tau(\langle \overline{A}\rangle\varphi) = \Diamond_l(\neg\pi \wedge \tau(\varphi))$).

Hereafter, we denote interval logic(s) of temporal neighborhood by PNL.

Decidability of interval neighborhood logics via reduction to $FO^2[<]$ - 2

More precisely, Otto proved the decidability of the satisfiability problem for $FO^2[<]$ on the classes of all linear orders, of well-orders, and of finite linear orders, as well as on the linear order on the natural numbers.

His proofs are based on an elaborated model-theoretic argument, analyzing the types of elements and pairs in models of $FO^2[<]$.

Decidability of $A\overline{A}^{\pi+}$ on each of these classes immediately follows.

Moreover, since $A\overline{A}^{+}$ and $A\overline{A}^{-}$ are strictly less expressive than $A\overline{A}^{\pi+}$, their decidability on the same classes of structures immediately follows ($A\overline{A}^{-}$ can be naturally embedded into $A\overline{A}^{\pi+}$ via the translation clauses $\tau(\langle A\rangle\varphi) = \Diamond_r(\neg\pi \wedge \tau(\varphi))$ and $\tau(\langle\overline{A}\rangle\varphi) = \Diamond_l(\neg\pi \wedge \tau(\varphi))$).

Hereafter, we denote interval logic(s) of temporal neighborhood by PNL.

Tableau-based decision procedures for PNL - 1

This is not the end of the story as (i) it is far from being trivial to extract a decision procedure from Otto's proof, and (ii) some meaningful cases are missing (dense linear orders, weakly discrete linear orders)

Tableau-based decision procedures have been developed for various propositional interval logics of temporal neighborhood, including:

- the future fragment of PNL (Right PNL, RPNL for short) interpreted over $\langle \mathbb{N}, < \rangle$ (or over a prefix of it)

D. Bresolin, A. Montanari, and G. Sciavicco, *An Optimal Decision Procedure for Right Propositional Neighborhood Logic*, Journal of Automated Reasoning, 38(1-3):173–199, 2007

Tableau-based decision procedures for PNL - 1

This is not the end of the story as (i) it is far from being trivial to extract a decision procedure from Otto's proof, and (ii) some meaningful cases are missing (dense linear orders, weakly discrete linear orders)

Tableau-based decision procedures have been developed for various propositional interval logics of temporal neighborhood, including:

- the future fragment of PNL (Right PNL, RPNL for short) interpreted over $\langle \mathbb{N}, < \rangle$ (or over a prefix of it)

D. Bresolin, A. Montanari, and G. Sciavicco, *An Optimal Decision Procedure for Right Propositional Neighborhood Logic*, Journal of Automated Reasoning, 38(1-3):173–199, 2007

Tableau-based decision procedures for PNL - 1

This is not the end of the story as (i) it is far from being trivial to extract a decision procedure from Otto's proof, and (ii) some meaningful cases are missing (dense linear orders, weakly discrete linear orders)

Tableau-based decision procedures have been developed for various propositional interval logics of temporal neighborhood, including:

- the future fragment of PNL (Right PNL, RPNL for short) interpreted over $\langle \mathbb{N}, < \rangle$ (or over a prefix of it)

D. Bresolin, A. Montanari, and G. Sciavicco, *An Optimal Decision Procedure for Right Propositional Neighborhood Logic*, Journal of Automated Reasoning, 38(1-3):173–199, 2007

Tableau-based decision procedures for PNL - 2

- Two variants of RPNL:
  - RPNL interpreted over trees, where every path is either $\langle \mathbb{N}, < \rangle$ (or a prefix of it)
  - the logic BTNL (= RPNL + A / E), interpreted over trees, that combines neighborhood modalities of RPNL with path quantifiers of branching time temporal logics

D. Bresolin, A. Montanari, and P. Sala, *An optimal tableau for Right Propositional Neighborhood Logic over trees*, TIME 2008

- RPNL interpreted over all linear orders

D. Bresolin, A. Montanari, P. Sala, and G. Sciavicco G., *Optimal tableaux for Right Propositional Neighborhood Logic over Linear Orders*, JELIA 2008

Tableau-based decision procedures for PNL - 2

- Two variants of RPNL:
  - RPNL interpreted over trees, where every path is either $\langle \mathbb{N}, < \rangle$ (or a prefix of it)
  - the logic BTNL (= RPNL + A / E), interpreted over trees, that combines neighborhood modalities of RPNL with path quantifiers of branching time temporal logics

D. Bresolin, A. Montanari, and P. Sala, *An optimal tableau for Right Propositional Neighborhood Logic over trees*, TIME 2008

- RPNL interpreted over all linear orders

D. Bresolin, A. Montanari, P. Sala, and G. Sciavicco G., *Optimal tableaux for Right Propositional Neighborhood Logic over Linear Orders*, JELIA 2008

Tableau-based decision procedures for PNL - 3

- full PNL interpreted over dense linear orders, (weakly) discrete linear orders, and all linear orders, as well as over $\mathbb{Z}$ (or over a subset of it) and over $\mathbb{R}$

D. Bresolin, A. Montanari, and P. Sala, *An Optimal Tableau-based Decision Algorithm for Propositional Neighborhood Logic*, STACS 2007

D. Bresolin, A. Montanari, P. Sala, and G. Sciavicco, *Optimal Tableau Systems for Propositional Neighborhood Logic over All, Dense, and Discrete Linear Orders*, TABLEAUX 2011

A. Montanari and P. Sala, *An Optimal Tableau System for the Logic of Temporal Neighborhood over the Reals*, TIME 2012

A decision procedure for RPNL over $\langle \mathbb{N}, < \rangle$

The decision procedure for RPNL partly resembles that for (future) LTL

The decision procedure for (future) LTL takes advantage of the fix-point definition of temporal operators (expansion rules), which splits every temporal formula into a (possibly empty) part related to the current state and a part related to the next state, and completely forgets the past

Expansion rules

$$Gp = p \wedge XGp; \quad Fp = p \vee XFp; \quad pUq = q \vee (p \wedge X(pUq))$$

The decision procedure for RPNL must keep track of universal requests as well as pending existential requests coming from the past

A decision procedure for RPNL over $\langle \mathbb{N}, < \rangle$

The decision procedure for RPNL partly resembles that for (future) LTL

The decision procedure for (future) LTL takes advantage of the fix-point definition of temporal operators (expansion rules), which splits every temporal formula into a (possibly empty) part related to the current state and a part related to the next state, and completely forgets the past

Expansion rules

$$Gp = p \wedge XGp; \quad Fp = p \vee XFp; \quad pUq = q \vee (p \wedge X(pUq))$$

The decision procedure for RPNL must keep track of universal requests as well as pending existential requests coming from the past

A decision procedure for RPNL over $\langle \mathbb{N}, < \rangle$

The decision procedure for RPNL partly resembles that for (future) LTL

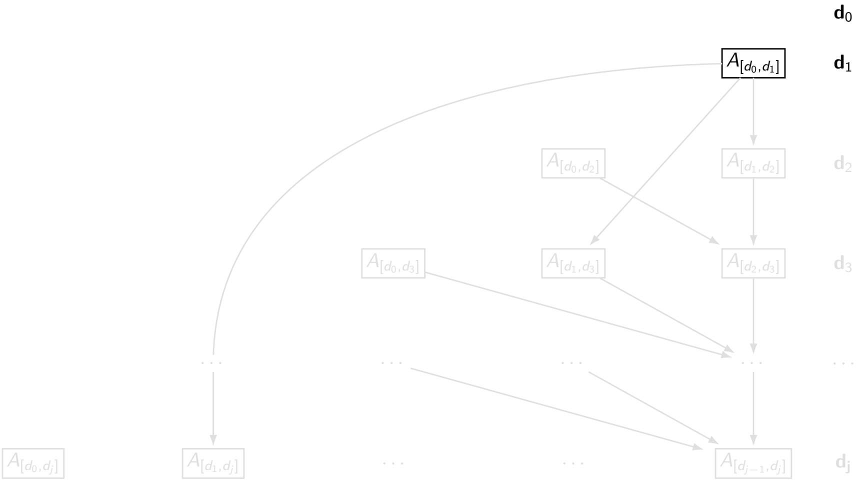
The decision procedure for (future) LTL takes advantage of the fix-point definition of temporal operators (expansion rules), which splits every temporal formula into a (possibly empty) part related to the current state and a part related to the next state, and completely forgets the past

Expansion rules

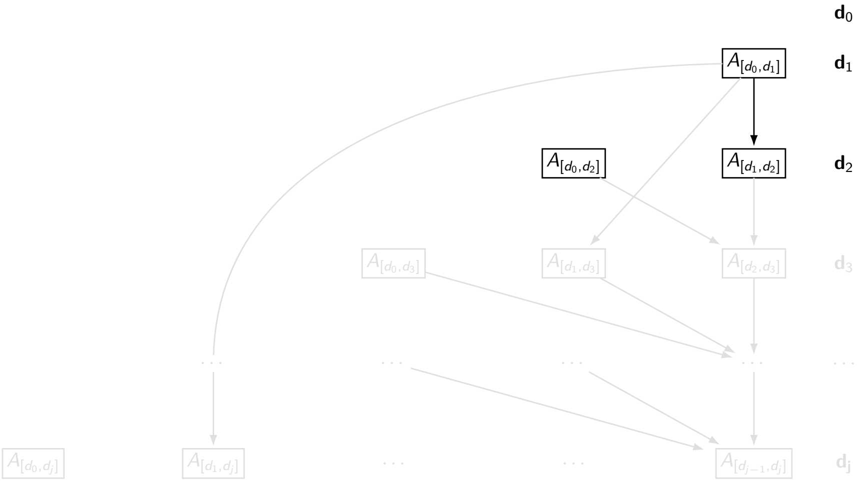
$$Gp = p \wedge XGp; \quad Fp = p \vee XFp; \quad pUq = q \vee (p \wedge X(pUq))$$

The decision procedure for RPNL must keep track of universal requests as well as pending existential requests coming from the past

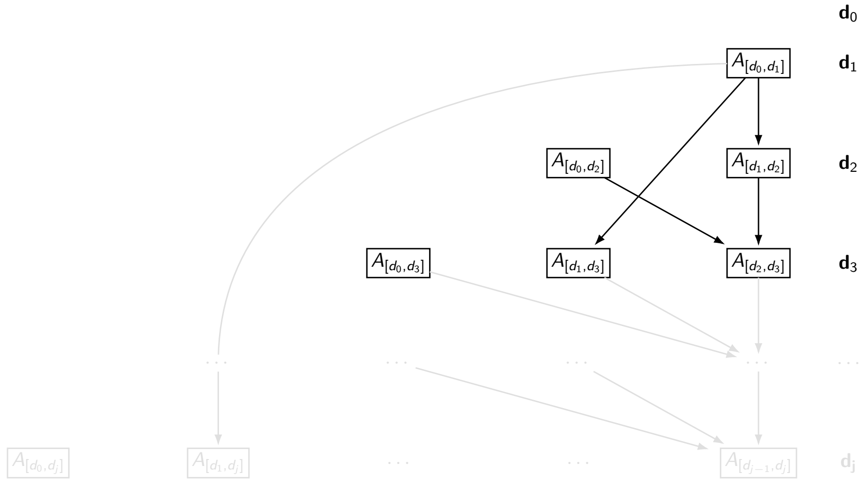
The idea: a step-by-step model building process



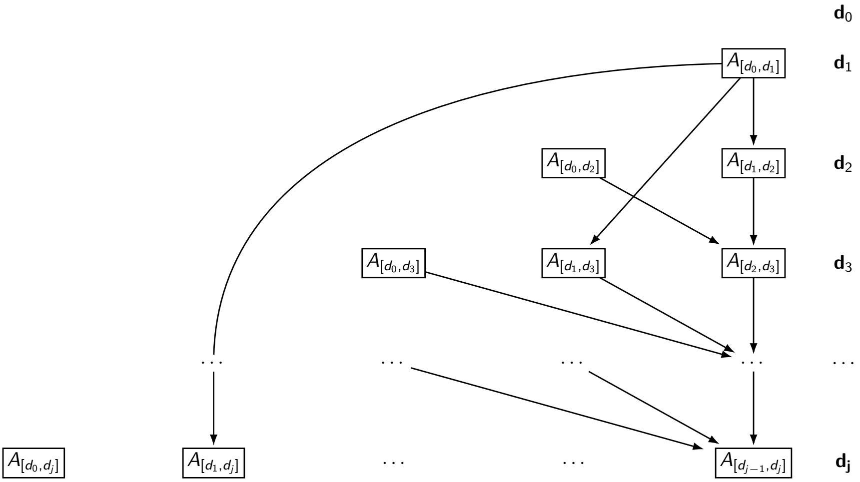
The idea: a step-by-step model building process



The idea: a step-by-step model building process



The idea: a step-by-step model building process



Graph components

The nodes of the graph

For any interval $[d_i, d_j]$, $A_{[d_i,d_j]}$ is the set of all and only the formulas that hold over it (the formula to be tested must belong to $A_{[d_0,d_1]}$)

The edges of the graph

For any pair of adjacent intervals $[d_i, d_j]$ and $[d_j, d_k]$, there exists an edge that links $A_{[d_i,d_j]}$ to $A_{[d_j,d_k]}$.

Graph construction

The construction starts from the node $A_{[d_0,d_1]}$ corresponding to the initial interval $[d_0, d_1]$.

At the j-th step, it adds all nodes $A_{[d_i,d_j]}$, with $i < j$ (and the relevant edges).

Graph components

The nodes of the graph

For any interval $[d_i, d_j]$, $A_{[d_i,d_j]}$ is the set of all and only the formulas that hold over it (the formula to be tested must belong to $A_{[d_0,d_1]}$)

The edges of the graph

For any pair of adjacent intervals $[d_i, d_j]$ and $[d_j, d_k]$, there exists an edge that links $A_{[d_i,d_j]}$ to $A_{[d_j,d_k]}$.

Graph construction

The construction starts from the node $A_{[d_0,d_1]}$ corresponding to the initial interval $[d_0, d_1]$.

At the j-th step, it adds all nodes $A_{[d_i,d_j]}$, with $i < j$ (and the relevant edges).

Graph components

The nodes of the graph

For any interval $[d_i, d_j]$, $A_{[d_i,d_j]}$ is the set of all and only the formulas that hold over it (the formula to be tested must belong to $A_{[d_0,d_1]}$)

The edges of the graph

For any pair of adjacent intervals $[d_i, d_j]$ and $[d_j, d_k]$, there exists an edge that links $A_{[d_i,d_j]}$ to $A_{[d_j,d_k]}$.

Graph construction

The construction starts from the node $A_{[d_0,d_1]}$ corresponding to the initial interval $[d_0, d_1]$.

At the j-th step, it adds all nodes $A_{[d_i,d_j]}$, with $i < j$ (and the relevant edges).

Consistency constraints

Constraint 1

If $[A]\psi$ belongs to $A_{[d_i,d_j]}$, then ψ must belong to $A_{[d_j,d_k]}$.

Constraint 2

Since every right neighbor of $[d_i, d_k]$ is a right neighbor of $[d_j, d_k]$, $A_{[d_i,d_k]}$ and $A_{[d_j,d_k]}$ must agree on their (universal and existential) temporal formulas.

Consistency constraints

Constraint 1

If $[A]\psi$ belongs to $A_{[d_i,d_j]}$, then ψ must belong to $A_{[d_j,d_k]}$.

Constraint 2

Since every right neighbor of $[d_i, d_k]$ is a right neighbor of $[d_j, d_k]$, $A_{[d_i,d_k]}$ and $A_{[d_j,d_k]}$ must agree on their (universal and existential) temporal formulas.

How do we guarantee that existential temporal requests are eventually satisfied?

If, after the execution of the i-th step, there still exists a pending existential temporal request, that is, a $\langle A \rangle \psi$ formula belonging to a node $A_{[d_h,d_j]}$ such that there exists no node $A_{[d_j,d_k]}$ including ψ, we (try to) satisfy it by adding a new node $A_{[d_j,d_{i+1}]}$ (in fact, a new bunch of nodes $A_{[d_0,d_{i+1}]}, \ldots, A_{[d_i,d_{i+1}]}$) and the relevant edges.

How do we guarantee termination?

New nodes may introduce new existential temporal requests.

Such a model building process can be turned into an effective procedure.

Critical aspects

How do we guarantee that existential temporal requests are eventually satisfied?

If, after the execution of the *i*-th step, there still exists a pending existential temporal request, that is, a $\langle A \rangle \psi$ formula belonging to a node $A_{[d_h, d_j]}$ such that there exists no node $A_{[d_j, d_k]}$ including ψ, we (try to) satisfy it by adding a new node $A_{[d_j, d_{i+1}]}$ (in fact, a new bunch of nodes $A_{[d_0, d_{i+1}]}, \ldots, A_{[d_i, d_{i+1}]}$) and the relevant edges.

How do we guarantee termination?

New nodes may introduce new existential temporal requests.

Such a model building process can be turned into an effective procedure.

Critical aspects

How do we guarantee that existential temporal requests are eventually satisfied?

If, after the execution of the *i*-th step, there still exists a pending existential temporal request, that is, a $\langle A \rangle \psi$ formula belonging to a node $A_{[d_h, d_j]}$ such that there exists no node $A_{[d_j, d_k]}$ including ψ, we (try to) satisfy it by adding a new node $A_{[d_j, d_{i+1}]}$ (in fact, a new bunch of nodes $A_{[d_0, d_{i+1}]}, \ldots, A_{[d_i, d_{i+1}]}$) and the relevant edges.

How do we guarantee termination?

New nodes may introduce new existential temporal requests.

Such a model building process can be turned into an effective procedure.

The main technical ingredient

Problem

Interval structures satisfying φ may be arbitrarily large or even infinite.

A small (pseudo)model theorem for RPNL over $\langle \mathbb{N}, < \rangle$

Result 1. We give a bound on the size of finite interval structures that must be checked for satisfiability, when searching for finite φ-models

Result 2. We show that we can restrict ourselves to infinite interval structures with a finite bounded representation, when searching for infinite φ-models.

The main technical ingredient

Problem

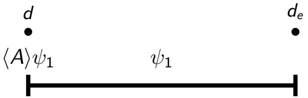
Interval structures satisfying φ may be arbitrarily large or even infinite.

A small (pseudo)model theorem for RPNL over $\langle \mathbb{N}, < \rangle$

Result 1. We give a bound on the size of finite interval structures that must be checked for satisfiability, when searching for finite φ-models

Result 2. We show that we can restrict ourselves to infinite interval structures with a finite bounded representation, when searching for infinite φ-models.

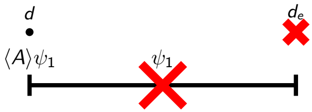
The key step: removing points from a model



f points to the right of d_e

with the same set of requests of d_e

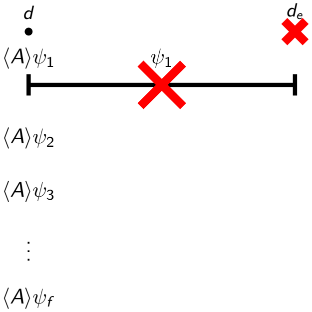
The key step: removing points from a model



f points to the right of d_e

with the same set of requests of d_e

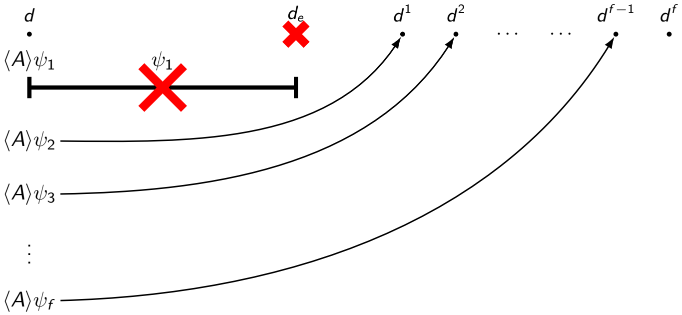
The key step: removing points from a model



f points to the right of d_e

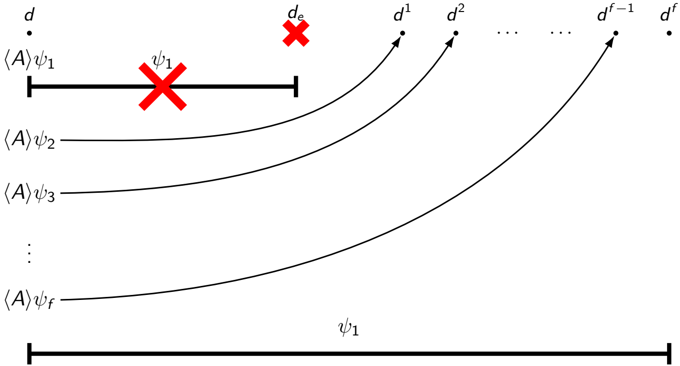
with the same set of requests of d_e

The key step: removing points from a model



f points to the right of d_e
with the same set of requests of d_e

The key step: removing points from a model



f points to the right of d_e

with the same set of requests of d_e

The distinctive features

- The removal of a point d_e causes the removal of all intervals either beginning or ending at it.
- Since RPNL features only future time modalities, the removal of intervals beginning at d_e is not critical.
- The removal of intervals ending at d_e may introduce “defects”.
- By properly choosing the point d_e to remove, we can guarantee that there exist sufficiently many points in the future of d which allows us to fix such defects.

The distinctive features

- The removal of a point d_e causes the removal of all intervals either beginning or ending at it.
- Since RPNL features only future time modalities, the removal of intervals beginning at d_e is not critical.
- The removal of intervals ending at d_e may introduce “defects”.
- By properly choosing the point d_e to remove, we can guarantee that there exist sufficiently many points in the future of d which allows us to fix such defects.

The distinctive features

- The removal of a point d_e causes the removal of all intervals either beginning or ending at it.
- Since RPNL features only future time modalities, the removal of intervals beginning at d_e is not critical.
- The removal of intervals ending at d_e may introduce “defects”.
- By properly choosing the point d_e to remove, we can guarantee that there exist sufficiently many points in the future of d which allows us to fix such defects.

The distinctive features

- The removal of a point d_e causes the removal of all intervals either beginning or ending at it.
- Since RPNL features only future time modalities, the removal of intervals beginning at d_e is not critical.
- The removal of intervals ending at d_e may introduce “defects”.
- By properly choosing the point d_e to remove, we can guarantee that there exist sufficiently many points in the future of d which allows us to fix such defects.

The distinctive features

- The removal of a point d_e causes the removal of all intervals either beginning or ending at it.
- Since RPNL features only future time modalities, the removal of intervals beginning at d_e is not critical.
- The removal of intervals ending at d_e may introduce “defects”.
- By properly choosing the point d_e to remove, we can guarantee that there exist sufficiently many points in the future of d which allows us to fix such defects.

The tableau system for RPNL over $\langle \mathbb{N}, < \rangle$

Tableau construction is based on two expansion rules:

- step rule
- fill-in rule

and a blocking condition, that guarantees termination.

It does not need to differentiate the search for a finite model from that for an infinite one!

NEXPTIME-completeness

The procedure has a nondeterministic time complexity which is exponential in the size of φ. Moreover, a NEXPTIME lower bound can be obtained by a reduction from the NEXPTIME-complete exponential tiling problem.

The tableau system for RPNL over $\langle \mathbb{N}, < \rangle$

Tableau construction is based on two expansion rules:

- step rule
- fill-in rule

and a blocking condition, that guarantees termination.

It does not need to differentiate the search for a finite model from that for an infinite one!

NEXPTIME-completeness

The procedure has a nondeterministic time complexity which is exponential in the size of φ. Moreover, a NEXPTIME lower bound can be obtained by a reduction from the NEXPTIME-complete exponential tiling problem.

The tableau system for RPNL over $\langle \mathbb{N}, < \rangle$

Tableau construction is based on two expansion rules:

- step rule
- fill-in rule

and a blocking condition, that guarantees termination.

It does not need to differentiate the search for a finite model from that for an infinite one!

NEXPTIME-completeness

The procedure has a nondeterministic time complexity which is exponential in the size of φ. Moreover, a NEXPTIME lower bound can be obtained by a reduction from the NEXPTIME-complete exponential tiling problem.

From RPNL over $\langle \mathbb{N}, < \rangle$ to full PNL over $\langle \mathbb{Z}, < \rangle$

In the case of full PNL, the removal process is still possible, but it turns out to be much more involved.

Complications

- The removal of a point d from a PNL model may affect the satisfiability of formulae over intervals in the past as well as in the future of d.
- To fix the defects possibly caused by the removal of d, one must guarantee that there exist sufficiently many points with the same characteristics as d both in the future and in the past of d.
- Moreover, one must guarantee that changing the valuation of intervals that either end or start at these points does not generate new defects.

Theorem

The decision problem for PNL over $\langle \mathbb{Z}, < \rangle$ is NEXPTIME-complete.

From RPNL over $\langle \mathbb{N}, < \rangle$ to full PNL over $\langle \mathbb{Z}, < \rangle$

In the case of full PNL, the removal process is still possible, but it turns out to be much more involved.

Complications

- The removal of a point *d* from a PNL model may affect the satisfiability of formulae over intervals in the past as well as in the future of *d*.
- To fix the defects possibly caused by the removal of *d*, one must guarantee that there exist sufficiently many points with the same characteristics as *d* both in the future and in the past of *d*.
- Moreover, one must guarantee that changing the valuation of intervals that either end or start at these points does not generate new defects.

Theorem

The decision problem for PNL over $\langle \mathbb{Z}, < \rangle$ is NEXPTIME-complete.

From RPNL over $\langle \mathbb{N}, < \rangle$ to full PNL over $\langle \mathbb{Z}, < \rangle$

In the case of full PNL, the removal process is still possible, but it turns out to be much more involved.

Complications

- The removal of a point *d* from a PNL model may affect the satisfiability of formulae over intervals in the past as well as in the future of *d*.
- To fix the defects possibly caused by the removal of *d*, one must guarantee that there exist sufficiently many points with the same characteristics as *d* both in the future and in the past of *d*.
- Moreover, one must guarantee that changing the valuation of intervals that either end or start at these points does not generate new defects.

Theorem

The decision problem for PNL over $\langle \mathbb{Z}, < \rangle$ is NEXPTIME-complete.

From RPNL over $\langle \mathbb{N}, < \rangle$ to full PNL over $\langle \mathbb{Z}, < \rangle$

In the case of full PNL, the removal process is still possible, but it turns out to be much more involved.

Complications

- The removal of a point d from a PNL model may affect the satisfiability of formulae over intervals in the past as well as in the future of d.
- To fix the defects possibly caused by the removal of d, one must guarantee that there exist sufficiently many points with the same characteristics as d both in the future and in the past of d.
- Moreover, one must guarantee that changing the valuation of intervals that either end or start at these points does not generate new defects.

Theorem

The decision problem for PNL over $\langle \mathbb{Z}, < \rangle$ is NEXPTIME-complete.

From RPNL over $\langle \mathbb{N}, < \rangle$ to full PNL over $\langle \mathbb{Z}, < \rangle$

In the case of full PNL, the removal process is still possible, but it turns out to be much more involved.

Complications

- The removal of a point *d* from a PNL model may affect the satisfiability of formulae over intervals in the past as well as in the future of *d*.
- To fix the defects possibly caused by the removal of *d*, one must guarantee that there exist sufficiently many points with the same characteristics as *d* both in the future and in the past of *d*.
- Moreover, one must guarantee that changing the valuation of intervals that either end or start at these points does not generate new defects.

Theorem

The decision problem for PNL over $\langle \mathbb{Z}, < \rangle$ is NEXPTIME-complete.

From RPNL over $\langle \mathbb{N}, < \rangle$ to RPNL over all linear orders

There exist RPNL formulae that cannot be satisfied over $\langle \mathbb{N}, < \rangle$, but turn out to be satisfiable over other linearly-ordered domains.

The formula *AccPoints*

$$
\begin{aligned}
[G]\psi &= \psi \wedge [A]\psi \wedge [A][A]\psi \\
seq_p &= p \rightarrow \langle A \rangle p \\
AccPoints &= \langle A \rangle p \wedge [G]seq_p \wedge \langle A \rangle [G]\neg p
\end{aligned}
$$

The formula *AccPoints* is unsatisfiable over $\langle \mathbb{N}, < \rangle$, but it is satisfiable over the class of all linear orders.

From RPNL over $\langle \mathbb{N}, < \rangle$ to RPNL over all linear orders

There exist RPNL formulae that cannot be satisfied over $\langle \mathbb{N}, < \rangle$, but turn out to be satisfiable over other linearly-ordered domains.

The formula *AccPoints*

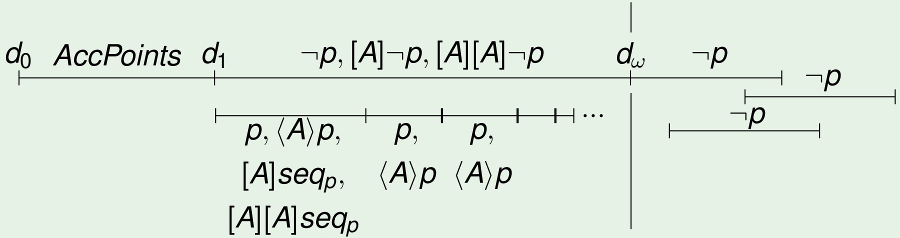
$$\begin{array}{rcl} [G]\psi & = & \psi \wedge [A]\psi \wedge [A][A]\psi \\ seq_p & = & p \rightarrow \langle A \rangle p \\ AccPoints & = & \langle A \rangle p \wedge [G]seq_p \wedge \langle A \rangle [G]\neg p \end{array}$$

The formula *AccPoints* is unsatisfiable over $\langle \mathbb{N}, < \rangle$, but it is satisfiable over the class of all linear orders.

Satisfiablity of *AccPoints*

$$AccPoints = \langle A \rangle p \wedge [G] seq_p \wedge \langle A \rangle [G] \neg p$$

Every model of *AccPoints* contains at least one accumulation point d_ω placed after an infinite sequence of points:



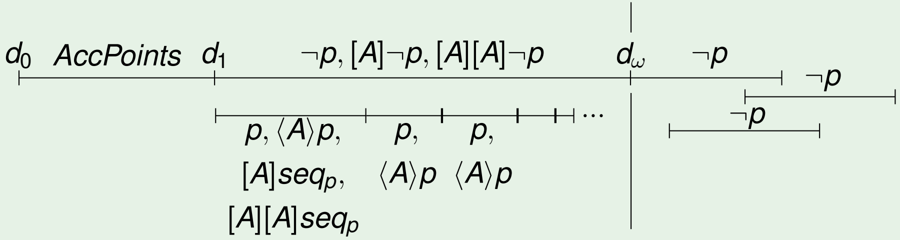
Theorem

The decision problem for RPNL over all linear orders is NEXPTIME-complete.

Satisfiablity of *AccPoints*

$$AccPoints = \langle A \rangle p \wedge [G]seq_p \wedge \langle A \rangle [G] \neg p$$

Every model of *AccPoints* contains at least one accumulation point d_ω placed after an infinite sequence of points:



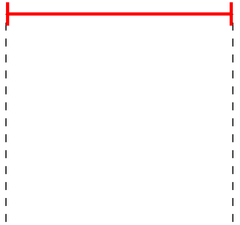
Theorem

The decision problem for RPNL over all linear orders is NEXPTIME-complete.

Propositional Logics of Subinterval Structures

Three different subinterval relations are possible.

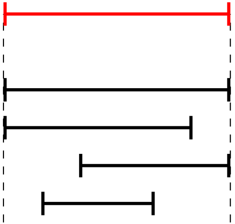
- **The reflexive subinterval relation $\sqsubseteq$:**
  $[c,d] \sqsubseteq [a,b]$ iff $a \leq c$ and $d \leq b$;
- **The irreflexive subinterval relation $\sqsubset$:**
  $[c,d] \sqsubset [a,b]$ iff $a \leq c$, $d \leq b$ and $[a,b] \neq [c,d]$;
- **The strict subinterval relation $\dot{\sqsubset}$:**
  $[c,d] \dot{\sqsubset} [a,b]$ iff $a < c$ and $d < b$.



Propositional Logics of Subinterval Structures

Three different subinterval relations are possible.

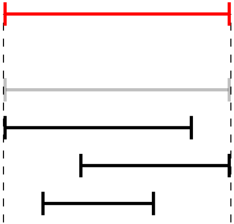
- **The reflexive subinterval relation $\sqsubseteq$:** $[c, d] \sqsubseteq [a, b]$ iff $a \leq c$ and $d \leq b$;
- **The irreflexive subinterval relation $\sqsubset$:** $[c, d] \sqsubset [a, b]$ iff $a \leq c$, $d \leq b$ and $[a, b] \neq [c, d]$;
- **The strict subinterval relation $\dot\sqsubset$:** $[c, d] \dot\sqsubset [a, b]$ iff $a < c$ and $d < b$.



Propositional Logics of Subinterval Structures

Three different subinterval relations are possible.

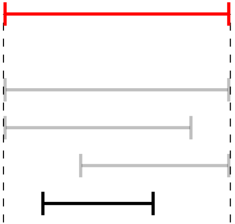
- **The reflexive subinterval relation $\sqsubseteq$:** $[c, d] \sqsubseteq [a, b]$ iff $a \leq c$ and $d \leq b$;
- **The irreflexive subinterval relation $\sqsubset$:** $[c, d] \sqsubset [a, b]$ iff $a \leq c$, $d \leq b$ and $[a, b] \neq [c, d]$;
- **The strict subinterval relation $\sqsubset\!\!\cdot$:** $[c, d] \sqsubset\!\!\cdot [a, b]$ iff $a < c$ and $d < b$.



Propositional Logics of Subinterval Structures

Three different subinterval relations are possible.

- **The reflexive subinterval relation $\sqsubseteq$:**
  $[c, d] \sqsubseteq [a, b]$ iff $a \leq c$ and $d \leq b$;
- **The irreflexive subinterval relation $\sqsubset$:**
  $[c, d] \sqsubset [a, b]$ iff $a \leq c$, $d \leq b$ and
  $[a, b] \neq [c, d]$;
- **The strict subinterval relation $\dot\sqsubset$:**
  $[c, d] \dot\sqsubset [a, b]$ iff $a < c$ and $d < b$.



We focus our attention on *dense subinterval structures* $\mathbb{I}(\mathbf{D})^{-}$, devoid of point intervals, for all three versions of the subinterval relation.

- ▷ The reflexive case: this is the logic $S4$ (van Benthem'91).
- ▷ The strict (irreflexive) case.
- ▷ The proper (irreflexive) case.

The logics of dense subinterval structures

We focus our attention on *dense subinterval structures* $\mathbb{I}(\mathbf{D})^{-}$, devoid of point intervals, for all three versions of the subinterval relation.

- ▷ The reflexive case: this is the logic $S4$ (van Benthem'91).
- ▷ The strict (irreflexive) case.
- ▷ The proper (irreflexive) case.

The logics of dense subinterval structures

We focus our attention on *dense subinterval structures* $\mathbb{I}(\mathbf{D})^{-}$, devoid of point intervals, for all three versions of the subinterval relation.

- ▷ The reflexive case: this is the logic $S4$ (van Benthem'91).
- ▷ The strict (irreflexive) case.
- ▷ The proper (irreflexive) case.

The logics of dense subinterval structures

We focus our attention on *dense subinterval structures* $\mathbb{I}(\mathbf{D})^-$, devoid of point intervals, for all three versions of the subinterval relation.

- ▷ The reflexive case: this is the logic $S4$ (van Benthem'91).
- ▷ The strict (irreflexive) case.
- ▷ The proper (irreflexive) case.

General features of tableau systems for dense subinterval structures

▷ All (standard) models are infinite.

▷ Every model can be assumed to be built on the rational interval [0,1].

▷ The tableaux require special looping control mechanism to guarantee termination.

▷ Open terminating tableaux do not produce models, but only finite pseudo-models, which then are expanded to infinite standard models.

▷ Thus, there are three stages in proving satisfiability of a formula in a dense subinterval structure:
open saturated tableau $\Rightarrow$ pseudo-model $\Rightarrow$ standard model.

▷ Both transitions are provably successful and constructive, thus reducing the task to the construction of an open saturated tableau for the formula.

General features of tableau systems for dense subinterval structures

▷ All (standard) models are infinite.

▷ Every model can be assumed to be built on the rational interval [0,1].

▷ The tableaux require special looping control mechanism to guarantee termination.

▷ Open terminating tableaux do not produce models, but only finite pseudo-models, which then are expanded to infinite standard models.

▷ Thus, there are three stages in proving satisfiability of a formula in a dense subinterval structure:
open saturated tableau ⇒ pseudo-model ⇒ standard model.

▷ Both transitions are provably successful and constructive, thus reducing the task to the construction of an open saturated tableau for the formula.

General features of tableau systems for dense subinterval structures

- ▷ All (standard) models are infinite.
- ▷ Every model can be assumed to be built on the rational interval [0,1].
- ▷ The tableaux require special looping control mechanism to guarantee termination.
- ▷ Open terminating tableaux do not produce models, but only finite pseudo-models, which then are expanded to infinite standard models.
- ▷ Thus, there are three stages in proving satisfiability of a formula in a dense subinterval structure:
  open saturated tableau $\Rightarrow$ pseudo-model $\Rightarrow$ standard model.
- ▷ Both transitions are provably successful and constructive, thus reducing the task to the construction of an open saturated tableau for the formula.

General features of tableau systems for dense subinterval structures

▷ All (standard) models are infinite.

▷ Every model can be assumed to be built on the rational interval [0,1].

▷ The tableaux require special looping control mechanism to guarantee termination.

▷ Open terminating tableaux do not produce models, but only finite pseudo-models, which then are expanded to infinite standard models.

▷ Thus, there are three stages in proving satisfiability of a formula in a dense subinterval structure:
open saturated tableau ⇒ pseudo-model ⇒ standard model.

▷ Both transitions are provably successful and constructive, thus reducing the task to the construction of an open saturated tableau for the formula.

General features of tableau systems for dense subinterval structures

▷ All (standard) models are infinite.

▷ Every model can be assumed to be built on the rational interval [0,1].

▷ The tableaux require special looping control mechanism to guarantee termination.

▷ Open terminating tableaux do not produce models, but only finite pseudo-models, which then are expanded to infinite standard models.

▷ Thus, there are three stages in proving satisfiability of a formula in a dense subinterval structure:
open saturated tableau ⇒ pseudo-model ⇒ standard model.

▷ Both transitions are provably successful and constructive, thus reducing the task to the construction of an open saturated tableau for the formula.

General features of tableau systems for dense subinterval structures

▷ All (standard) models are infinite.

▷ Every model can be assumed to be built on the rational interval [0,1].

▷ The tableaux require special looping control mechanism to guarantee termination.

▷ Open terminating tableaux do not produce models, but only finite pseudo-models, which then are expanded to infinite standard models.

▷ Thus, there are three stages in proving satisfiability of a formula in a dense subinterval structure:
open saturated tableau $\Rightarrow$ pseudo-model $\Rightarrow$ standard model.

▷ Both transitions are provably successful and constructive, thus reducing the task to the construction of an open saturated tableau for the formula.

General features of tableau systems for dense subinterval structures

▷ All (standard) models are infinite.

▷ Every model can be assumed to be built on the rational interval [0,1].

▷ The tableaux require special looping control mechanism to guarantee termination.

▷ Open terminating tableaux do not produce models, but only finite pseudo-models, which then are expanded to infinite standard models.

▷ Thus, there are three stages in proving satisfiability of a formula in a dense subinterval structure:
open saturated tableau $\Rightarrow$ pseudo-model $\Rightarrow$ standard model.

▷ Both transitions are provably successful and constructive, thus reducing the task to the construction of an open saturated tableau for the formula.

The logic $D_{\sqsubset}$ of the strict subinterval relation

Syntax

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \langle D \rangle \varphi$$

A model for $D_{\sqsubset}$ is a pair $\langle\langle \mathbb{I}(\mathbf{D})^{-}, \mathrm{D}\rangle, V\rangle$ where

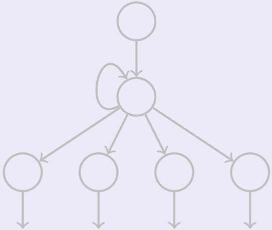
- $\langle \mathbb{I}(\mathbf{D})^{-}, \mathrm{D}\rangle$ is a dense subinterval structure
- V is a valuation function that assigns to every propositional variable p a set of intervals

Semantics

- $\langle D \rangle \varphi$ is true on $[a, b]$ iff there exists $[c, d] \sqsubset [a, b]$ where φ holds.

A $D_{\boxdot}$-structure is a special kind of rooted graph

- *reflexive* and *irreflexive* vertices



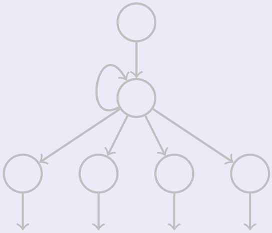
- the root is an irreflexive vertex
- every irreflexive vertex is followed by a unique reflexive one
- a reflexive vertex can have many irreflexive successors

- vertices are labelled with sets of formulas

$D_{\sqsubset}$-structures

A $D_{\sqsubset}$-structure is a special kind of rooted graph

-  *reflexive* and  *irreflexive* vertices



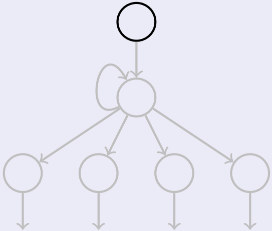
- the root is an irreflexive vertex
- every irreflexive vertex is followed by a unique reflexive one
- a reflexive vertex can have many irreflexive successors

- vertices are labelled with sets of formulas

$D_{\sqsubset}$-structures

A $D_{\sqsubset}$-structure is a special kind of rooted graph

-  *reflexive* and  *irreflexive* vertices



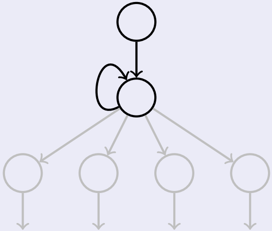
- the root is an irreflexive vertex
- every irreflexive vertex is followed by a unique reflexive one
- a reflexive vertex can have many irreflexive successors

- vertices are labelled with sets of formulas

$D_{\sqsubset}$-structures

A $D_{\sqsubset}$-structure is a special kind of rooted graph

-  *reflexive* and  *irreflexive* vertices



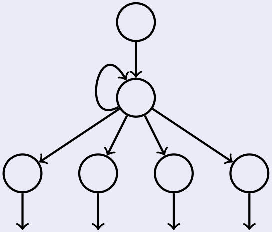
- the root is an irreflexive vertex
- every irreflexive vertex is followed by a unique reflexive one
- a reflexive vertex can have many irreflexive successors

- vertices are labelled with sets of formulas

$D_{\sqsubset}$-structures

A $D_{\sqsubset}$-structure is a special kind of rooted graph

-  *reflexive* and  *irreflexive* vertices



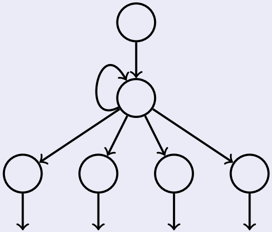
- the root is an irreflexive vertex
- every irreflexive vertex is followed by a unique reflexive one
- a reflexive vertex can have many irreflexive successors

- vertices are labelled with sets of formulas

$D_{\sqsubset}$-structures

A $D_{\sqsubset}$-structure is a special kind of rooted graph

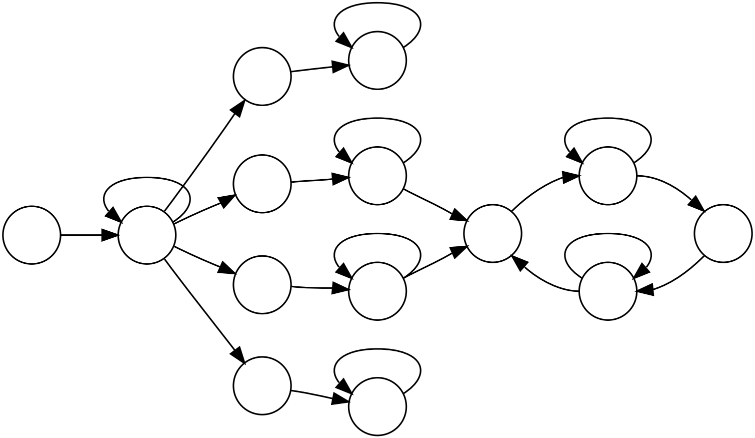
-  *reflexive* and  *irreflexive* vertices



- the root is an irreflexive vertex
- every irreflexive vertex is followed by a unique reflexive one
- a reflexive vertex can have many irreflexive successors

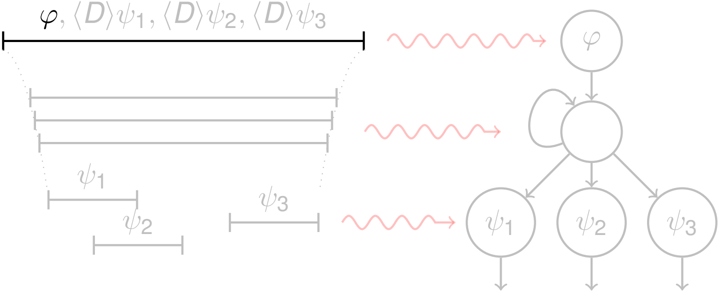
- vertices are labelled with sets of formulas

An example of a $D_{\sqsubset}$-structure



- A reflexive vertex may have no successors (different from itself)
- There can be loops involving irreflexive vertices

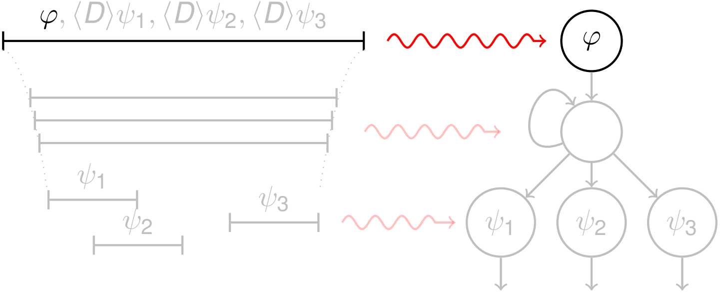
From interval models to $D_{\sqsubset}$-structures



Theorem

φ is satisfiable if and only if there exists a $D_{\sqsubset}$-structure for it.

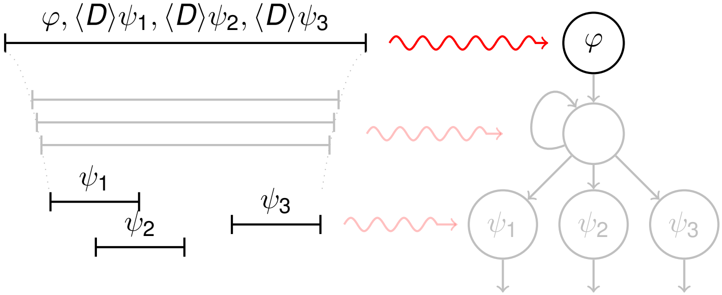
From interval models to $D_{\sqsubset}$-structures



Theorem

φ is satisfiable if and only if there exists a $D_{\sqsubset}$-structure for it.

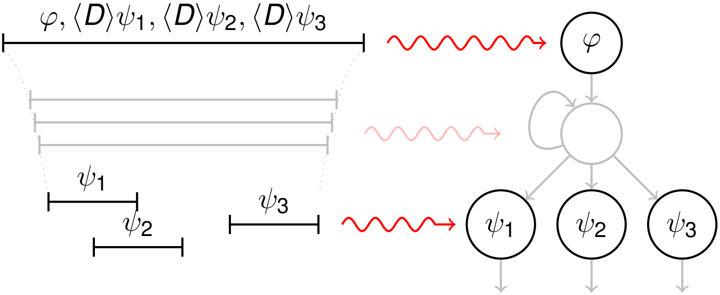
From interval models to $D_{\sqsubset}$-structures



Theorem

φ is satisfiable if and only if there exists a $D_{\sqsubset}$-structure for it.

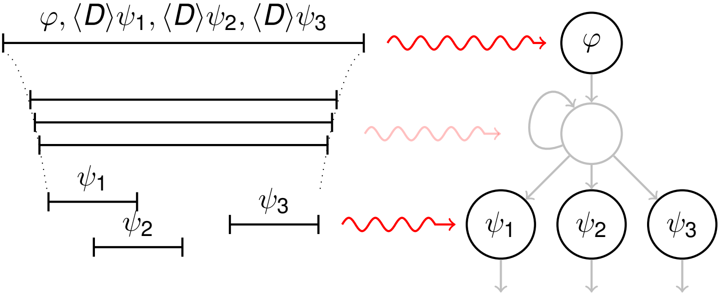
From interval models to $D_{\sqsubset}$-structures



Theorem

φ is satisfiable if and only if there exists a $D_{\sqsubset}$-structure for it.

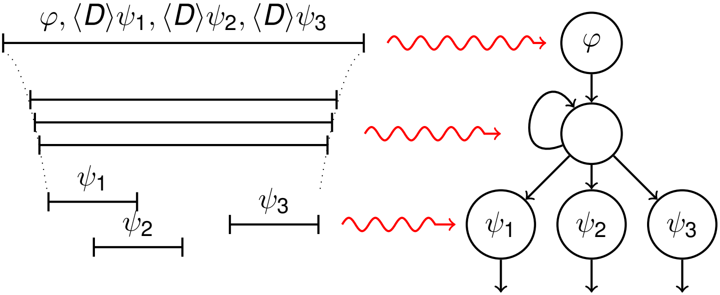
From interval models to $D_{\sqsubset}$-structures



Theorem

φ is satisfiable if and only if there exists a $D_{\sqsubset}$-structure for it.

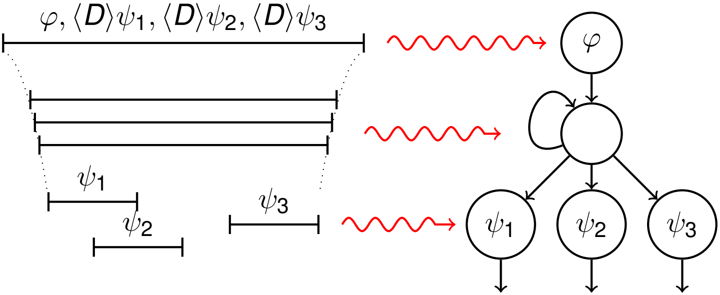
From interval models to $D_{\sqsubset}$-structures



Theorem

φ is satisfiable if and only if there exists a $D_{\sqsubset}$-structure for it.

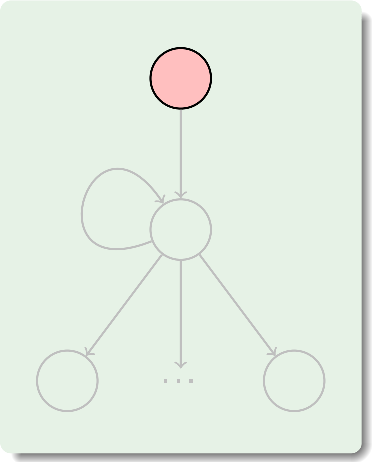
From interval models to $D_{\sqsubset}$-structures



Theorem

φ is satisfiable if and only if there exists a $D_{\sqsubset}$-structure for it.

Building the tableau for φ

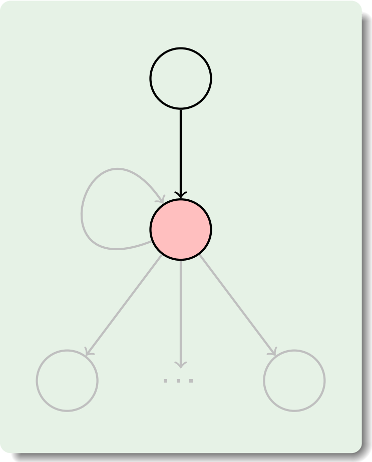


- Start from the initial tableau $\{\varphi\}$
- Apply the propositional rules:

(NOT) $\dfrac{\neg\neg\psi, F}{\psi, F}$

(OR) $\dfrac{\psi_1 \vee \psi_2, F}{\psi_1, F \mid \psi_2, F}$

(AND) $\dfrac{\psi_1 \wedge \psi_2, F}{\psi_1, \psi_2, F}$



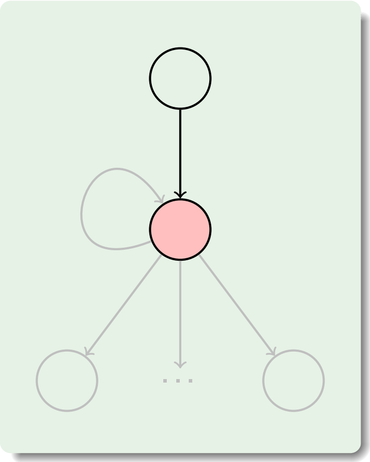
- Apply once the (2-DENS) rule:

$$\frac{\begin{array}{c}[D]\psi_1, \ldots, [D]\psi_m, \\ \langle D\rangle\varphi_1, \ldots, \langle D\rangle\varphi_n, F\end{array}}{\begin{array}{c}\psi_1, \ldots, \psi_m, [D]\psi_1, \ldots, [D]\psi_m, \\ \langle D\rangle\varphi_1, \ldots, \langle D\rangle\varphi_n\end{array}}$$

- Apply the propositional rules
- Apply the reflexivity rule:

$$\text{(REFL)}\ \frac{[D]\psi, F}{\psi, [D]\psi, F}$$

Building the tableau for φ



- Apply once the (2-DENS) rule:

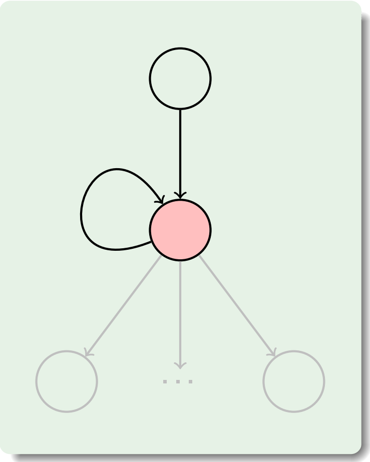
$$\frac{[D]\psi_1, \ldots, [D]\psi_m, \\ \langle D\rangle\varphi_1, \ldots, \langle D\rangle\varphi_n, F}{\psi_1, \ldots, \psi_m, [D]\psi_1, \ldots, [D]\psi_m, \\ \langle D\rangle\varphi_1, \ldots, \langle D\rangle\varphi_n}$$

- Apply the propositional rules

- Apply the reflexivity rule:

$$\text{(REFL)}\ \frac{[D]\psi, F}{\psi, [D]\psi, F}$$

Building the tableau for φ



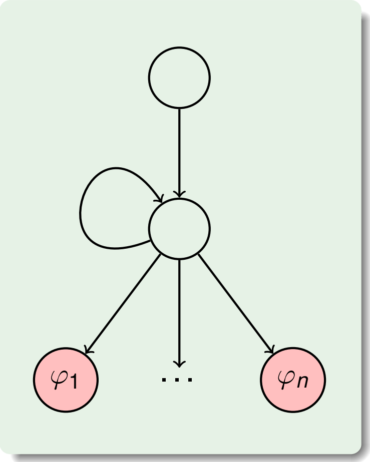
- Apply once the (2-DENS) rule:

$$\frac{[D]\psi_1, \ldots, [D]\psi_m, \quad \langle D\rangle\varphi_1, \ldots, \langle D\rangle\varphi_n, F}{\psi_1, \ldots, \psi_m, [D]\psi_1, \ldots, [D]\psi_m, \quad \langle D\rangle\varphi_1, \ldots, \langle D\rangle\varphi_n}$$

- Apply the propositional rules
- Apply the reflexivity rule:

$$(\text{REFL}) \ \frac{[D]\psi, F}{\psi, [D]\psi, F}$$

Building the tableau for φ

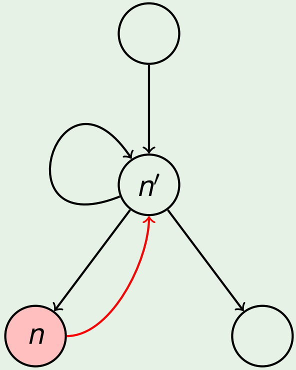


- Apply once the (STEP) rule:

$$\frac{\begin{array}{c}[D]\psi_1,\dots,[D]\psi_m,\\ \langle D\rangle\varphi_1,\dots,\langle D\rangle\varphi_n, F\end{array}}{\begin{array}{c}\varphi_1,\psi_1,\dots,\psi_m,[D]\psi_1,\dots,[D]\psi_m \mid \\ \dots \\ \mid \varphi_i,\psi_1,\dots,\psi_m,[D]\psi_1,\dots,[D]\psi_m \mid \\ \dots \\ \mid \varphi_n,\psi_1,\dots,\psi_m,[D]\psi_1,\dots,[D]\psi_m\end{array}}$$

- Proceed recursively in the expansion

Looping condition



When the application of the 2-DENS rule to a node n would generate a new reflexive node such that there exists another reflexive node n' in the tableau with the same set of temporal formulas, add an edge from n to n' instead of generating such a new node.

Open and closed tableau

A node *n* in a tableau is closed if one of the following conditions holds:

1. there exists a formula ψ such that $\psi, \neg\psi \in n$;
2. in the tableau construction, the NOT, OR, AND, 2-DENS, or REFL rules has been applied to *n* and *all* the immediate successors of *n* are closed;
3. in the tableau construction, the STEP rule has been applied to *n* and *at least one* of the immediate successors of *n* is closed.

A tableau is closed iff the root is closed, otherwise it is open.

Theorem (Soundness and completeness)

A formula φ is satisfiable if and only if its tableau is open

Open and closed tableau

A node *n* in a tableau is closed if one of the following conditions holds:

1. there exists a formula ψ such that $\psi, \neg\psi \in n$;
2. in the tableau construction, the NOT, OR, AND, 2-DENS, or REFL rules has been applied to *n* and *all* the immediate successors of *n* are closed;
3. in the tableau construction, the STEP rule has been applied to *n* and *at least one* of the immediate successors of *n* is closed.

A tableau is closed iff the root is closed, otherwise it is open.

Theorem (Soundness and completeness)

A formula φ is satisfiable if and only if its tableau is open

Computational complexity

- Every node has a number of successors that is bounded by $|\varphi|$
- The length of every path without repetition is linear in $|\varphi|$

$\Rightarrow$ The tableau can be explored using a polynomial amount of space.

- The logic is PSPACE-hard: the validity problem for prenex quantified boolean formulas can be reduced to the satisfiability problem for $D_{\sqsubset}$

$\Rightarrow$ The proposed method is of optimal complexity (PSPACE).

The logic $D_{\sqsubset}$ of the proper subinterval relation

The case of $D_{\sqsubset}$ is much more complex than that of $D_{\dot{\sqsubset}}$.

▷ The main complication is the presence of three distinct types of proper subintervals of the current interval: beginning, ending, and middle.

▷ Formulas of $D_{\sqsubset}$ can impose conditions on the type of subintervals needed to satisfy subformulas.

For instance, the formula

$$\langle D\rangle(p \wedge [D]q) \wedge \langle D\rangle(p \wedge [D]\neg q) \wedge [D]\neg(\langle D\rangle(p \wedge [D]q) \wedge \langle D\rangle(p \wedge [D]\neg q))$$

forces p to be true at some beginning and at some ending subinterval.

▷ This brings additional complications on the construction of the tableau and makes the pseudo-models for $D_{\sqsubset}$ quite complicate.

The logic $D_{\sqsubset}$ of the proper subinterval relation

The case of $D_{\sqsubset}$ is much more complex than that of $D_{\dot\sqsubset}$.

▷ The main complication is the presence of three distinct types of proper subintervals of the current interval: beginning, ending, and middle.

▷ Formulas of $D_{\sqsubset}$ can impose conditions on the type of subintervals needed to satisfy subformulas.

For instance, the formula

$$\langle D\rangle(p \wedge [D]q) \wedge \langle D\rangle(p \wedge [D]\neg q) \wedge [D]\neg(\langle D\rangle(p \wedge [D]q) \wedge \langle D\rangle(p \wedge [D]\neg q))$$

forces p to be true at some beginning and at some ending subinterval.

▷ This brings additional complications on the construction of the tableau and makes the pseudo-models for $D_{\sqsubset}$ quite complicate.

The logic $D_{\sqsubset}$ of the proper subinterval relation

The case of $D_{\sqsubset}$ is much more complex than that of $D_{\dot{\sqsubset}}$.

▷ The main complication is the presence of three distinct types of proper subintervals of the current interval: beginning, ending, and middle.

▷ Formulas of $D_{\sqsubset}$ can impose conditions on the type of subintervals needed to satisfy subformulas.

For instance, the formula

$$\langle D\rangle(p \wedge [D]q) \wedge \langle D\rangle(p \wedge [D]\neg q) \wedge [D]\neg(\langle D\rangle(p \wedge [D]q) \wedge \langle D\rangle(p \wedge [D]\neg q))$$

forces p to be true at some beginning and at some ending subinterval.

▷ This brings additional complications on the construction of the tableau and makes the pseudo-models for $D_{\sqsubset}$ quite complicate.

The logic $D_{\sqsubset}$ of the proper subinterval relation

The case of $D_{\sqsubset}$ is much more complex than that of $D_{\sqsubseteq}$.

▷ The main complication is the presence of three distinct types of proper subintervals of the current interval: beginning, ending, and middle.

▷ Formulas of $D_{\sqsubset}$ can impose conditions on the type of subintervals needed to satisfy subformulas.

For instance, the formula

$$\langle D\rangle(p \wedge [D]q) \wedge \langle D\rangle(p \wedge [D]\neg q) \wedge [D]\neg(\langle D\rangle(p \wedge [D]q) \wedge \langle D\rangle(p \wedge [D]\neg q))$$

forces p to be true at some beginning and at some ending subinterval.

▷ This brings additional complications on the construction of the tableau and makes the pseudo-models for $D_{\sqsubset}$ quite complicate.

The logic $D_{\sqsubset}$ of the proper subinterval relation

The case of $D_{\sqsubset}$ is much more complex than that of $D_{\sqsubseteq}$.

▷ The main complication is the presence of three distinct types of proper subintervals of the current interval: beginning, ending, and middle.

▷ Formulas of $D_{\sqsubset}$ can impose conditions on the type of subintervals needed to satisfy subformulas.

For instance, the formula

$$\langle D\rangle(p \wedge [D]q) \wedge \langle D\rangle(p \wedge [D]\neg q) \wedge [D]\neg(\langle D\rangle(p \wedge [D]q) \wedge \langle D\rangle(p \wedge [D]\neg q))$$

forces p to be true at some beginning and at some ending subinterval.

▷ This brings additional complications on the construction of the tableau and makes the pseudo-models for $D_{\sqsubset}$ quite complicate.

The general case (and beyond)

The logic of subinterval structures on discrete linear orders (as well as on finite linear orders) has been recently shown to be undecidable.

J. Marcinkowski and J. Michaliszyn, *The Ultimate Undecidability Result for the Halpern-Shoham Logic*, LICS 2011

The general case is still open.
It can combine the dense and discrete case in a recurrently interleaving way.

As an example, consider the formula

$$\langle D\rangle\langle D\rangle\top \wedge [D](\langle D\rangle\top \rightarrow \langle D\rangle\langle D\rangle\top \wedge \langle D\rangle[D]\bot)$$

It has neither discrete nor dense models, but it is satisfiable in Cantor's space.

The general case (and beyond)

The logic of subinterval structures on discrete linear orders (as well as on finite linear orders) has been recently shown to be undecidable.

J. Marcinkowski and J. Michaliszyn, *The Ultimate Undecidability Result for the Halpern-Shoham Logic*, LICS 2011

The general case is still open.

It can combine the dense and discrete case in a recurrently interleaving way.

As an example, consider the formula

$$\langle D\rangle\langle D\rangle\top \wedge [D](\langle D\rangle\top \rightarrow \langle D\rangle\langle D\rangle\top \wedge \langle D\rangle[D]\bot)$$

It has neither discrete nor dense models, but it is satisfiable in Cantor's space.

The general case (and beyond)

The logic of subinterval structures on discrete linear orders (as well as on finite linear orders) has been recently shown to be undecidable.

J. Marcinkowski and J. Michaliszyn, *The Ultimate Undecidability Result for the Halpern-Shoham Logic*, LICS 2011

The general case is still open.
It can combine the dense and discrete case in a recurrently interleaving way.

As an example, consider the formula

$$\langle D\rangle\langle D\rangle\top \wedge [D](\langle D\rangle\top \rightarrow \langle D\rangle\langle D\rangle\top \wedge \langle D\rangle[D]\bot)$$

It has neither discrete nor dense models, but it is satisfiable in Cantor's space.

The general case (and beyond)

The logic of subinterval structures on discrete linear orders (as well as on finite linear orders) has been recently shown to be undecidable.

J. Marcinkowski and J. Michaliszyn, *The Ultimate Undecidability Result for the Halpern-Shoham Logic*, LICS 2011

The general case is still open.
It can combine the dense and discrete case in a recurrently interleaving way.

As an example, consider the formula

$$\langle D\rangle\langle D\rangle\top \wedge [D](\langle D\rangle\top \rightarrow \langle D\rangle\langle D\rangle\top \wedge \langle D\rangle[D]\bot)$$

It has neither discrete nor dense models, but it is satisfiable in Cantor's space.

The general case (and beyond)

The logic of subinterval structures on discrete linear orders (as well as on finite linear orders) has been recently shown to be undecidable.

J. Marcinkowski and J. Michaliszyn, *The Ultimate Undecidability Result for the Halpern-Shoham Logic*, LICS 2011

The general case is still open.
It can combine the dense and discrete case in a recurrently interleaving way.

As an example, consider the formula

$$\langle D\rangle\langle D\rangle\top \wedge [D](\langle D\rangle\top \rightarrow \langle D\rangle\langle D\rangle\top \wedge \langle D\rangle[D]\bot)$$

It has neither discrete nor dense models, but it is satisfiable in Cantor's space.

The logic $AB\bar{B}$ of Allen's relations "meets", "begins", and "begun by" is quite expressive and decidable (EXPSPACE-complete).

A. Montanari, G. Puppis, P. Sala, and G. Sciavicco, Decidability of the interval temporal logic ABBbar over the natural numbers, STACS 2010

It allows one:

- to encode conditions of accomplishment (think of formula φ as the assertion: "Mr. Jones flew from Venice to Nancy"):

  $\langle A\rangle(\varphi \wedge [B](\neg\varphi \wedge [A]\neg\varphi) \wedge [\bar{B}]\neg\varphi)$;

Decidable interval logics: the logic $AB\bar{B}$ - 1

The logic $AB\bar{B}$ of Allen's relations "meets", "begins", and "begun by" is quite expressive and decidable (EXPSPACE-complete).

A. Montanari, G. Puppis, P. Sala, and G. Sciavicco, Decidability of the interval temporal logic ABBbar over the natural numbers, STACS 2010

It allows one:

- to encode conditions of accomplishment (think of formula φ as the assertion: "Mr. Jones flew from Venice to Nancy"):

  $\langle A\rangle\big(\varphi \ \wedge \ [B](\neg\varphi \ \wedge \ [A]\neg\varphi) \ \wedge \ [\bar{B}]\neg\varphi\big);$

Decidable interval logics: the logic $AB\bar{B}$ - 2

- to encode formulas of point-based temporal logics of the form $\psi \; U \; \varphi$, using the standard until operator (where atomic intervals are two-point intervals) as follows:

  $$\langle A\rangle\big([B]\perp \wedge \varphi\big) \;\vee\; \langle A\rangle\big(\langle A\rangle([B]\perp \wedge \varphi) \;\wedge\; [B](\langle A\rangle([B]\perp \wedge \psi))\big);$$

- to specify metric conditions like: "φ holds over a right neighbor interval of length greater than k (resp., less than k, equal to k)":

  $\langle A\rangle\big(\varphi \;\wedge\; \langle B\rangle^{k}\top\big)$ (resp., $\langle A\rangle\big(\varphi \;\wedge\; [B]^{k-1}\perp\big)$,
  $\langle A\rangle\big(\varphi \;\wedge\; [B]^{k}\perp \;\wedge\; \langle B\rangle^{k-1}\top\big)\big)$.

Decidable interval logics: the logic $AB\bar{B}$ - 2

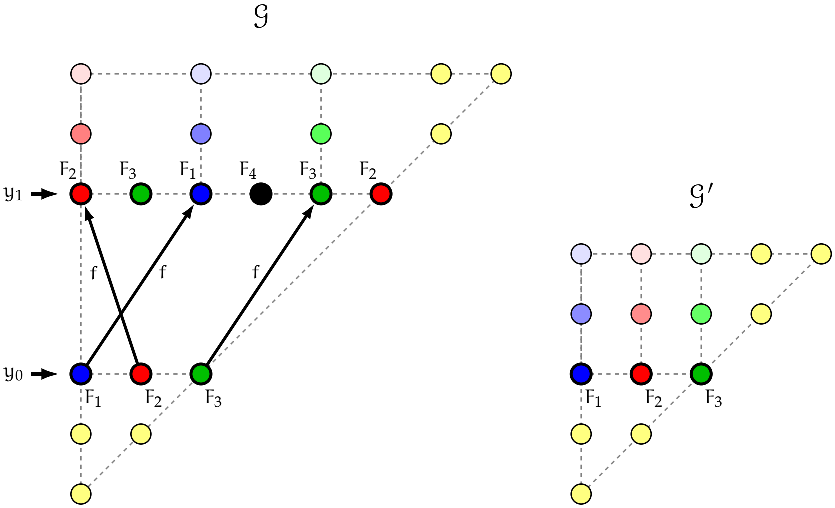
- to encode formulas of point-based temporal logics of the form $\psi\ U\ \varphi$, using the standard until operator (where atomic intervals are two-point intervals) as follows:

  $$\langle A\rangle\big([B]\bot \wedge \varphi\big) \ \vee\ \langle A\rangle\big(\langle A\rangle([B]\bot \wedge \varphi)\ \wedge\ [B](\langle A\rangle([B]\bot \wedge \psi))\big);$$

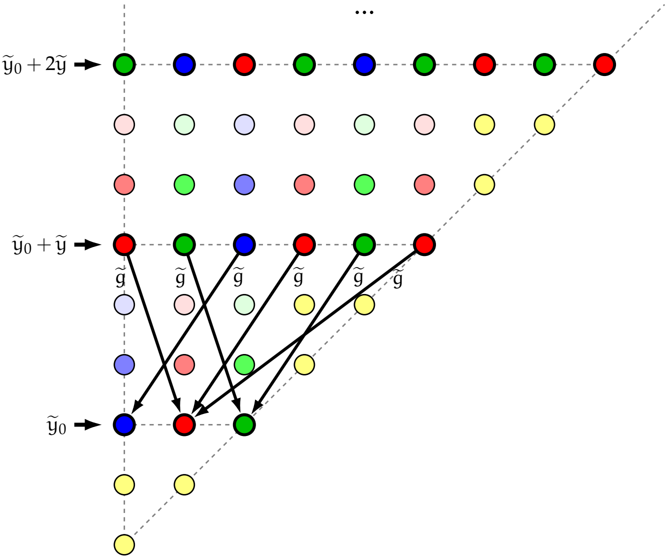
- to specify metric conditions like: "φ holds over a right neighbor interval of length greater than k (resp., less than k, equal to k)":

  $$\langle A\rangle\big(\varphi\ \wedge\ \langle B\rangle^{k}\top\big)\ \text{(resp., } \langle A\rangle\big(\varphi\ \wedge\ [B]^{k-1}\bot\big),$$
  $$\langle A\rangle\big(\varphi\ \wedge\ [B]^{k}\bot\ \wedge\ \langle B\rangle^{k-1}\top\big)\text{)}.$$

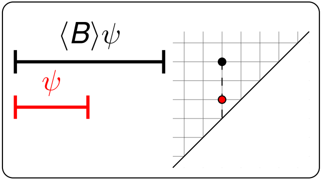
The finite case: a contraction method

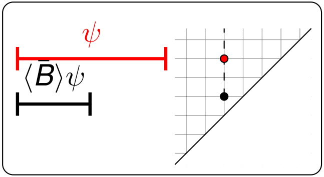


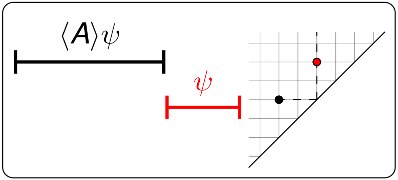
The infinite case: a periodic compass structure

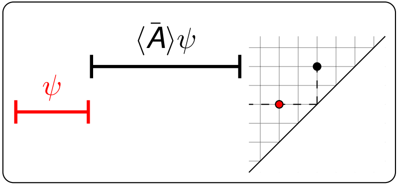


The maximal decidable fragment $AB\overline{BA}$



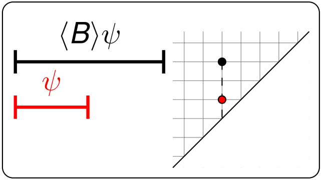


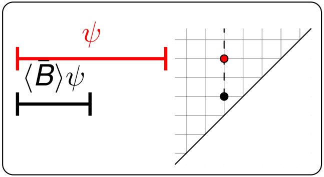


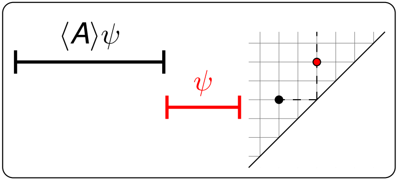


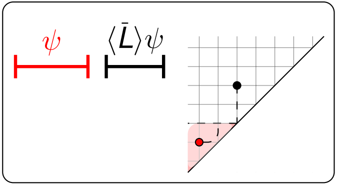
$AB\overline{BA}$ is NONPRIMITIVE RECURSIVE-hard over finite linear orders;
undecidable elsewhere

A. Montanari, G. Puppis, and P. Sala, *Maximal decidable fragments of Halpern and Shoham's modal logic of intervals*, ICALP 2010









We replace $\langle \overline{A}\rangle$ by $\langle \overline{L}\rangle$: $AB\overline{BL}$ is EXSPACE-complete over the classes of all, dense, and (weakly) discrete linear orders

D. Bresolin, A. Montanari, P. Sala, and G. Sciavicco, *What's decidable about Halpern and Shoham's interval logic? The maximal fragment $AB\overline{BL}$*, LICS 2011