The Ultimate Undecidability Result for the Halpern–Shoham Logic

Jerzy Marcinkowski, **Jakub Michaliszyn**

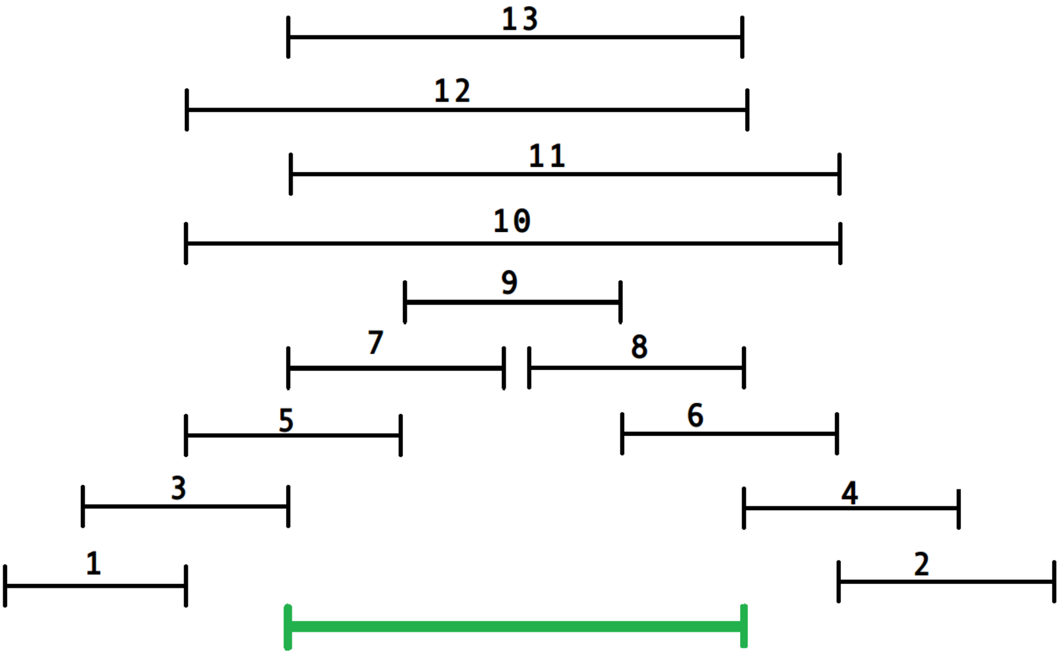
Instytut Informatyki
University of Wrocław

June 24, 2011

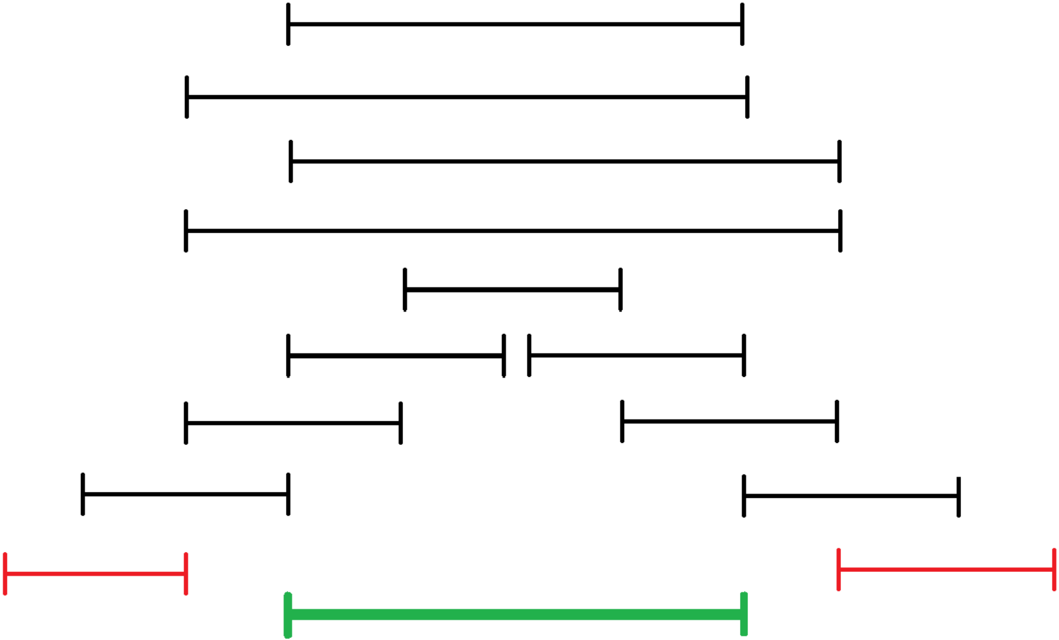
Intervals

- Given an arbitrary total order $\langle \mathbb{D}, \leq \rangle$.
- An interval: $[a, b]$ such that $a, b \in \mathbb{D}$ and $a \leq b$.
- What relative positions of two intervals can be expressed using $\leq$?

13 relative positions of intervals can be expressed using $\leq$

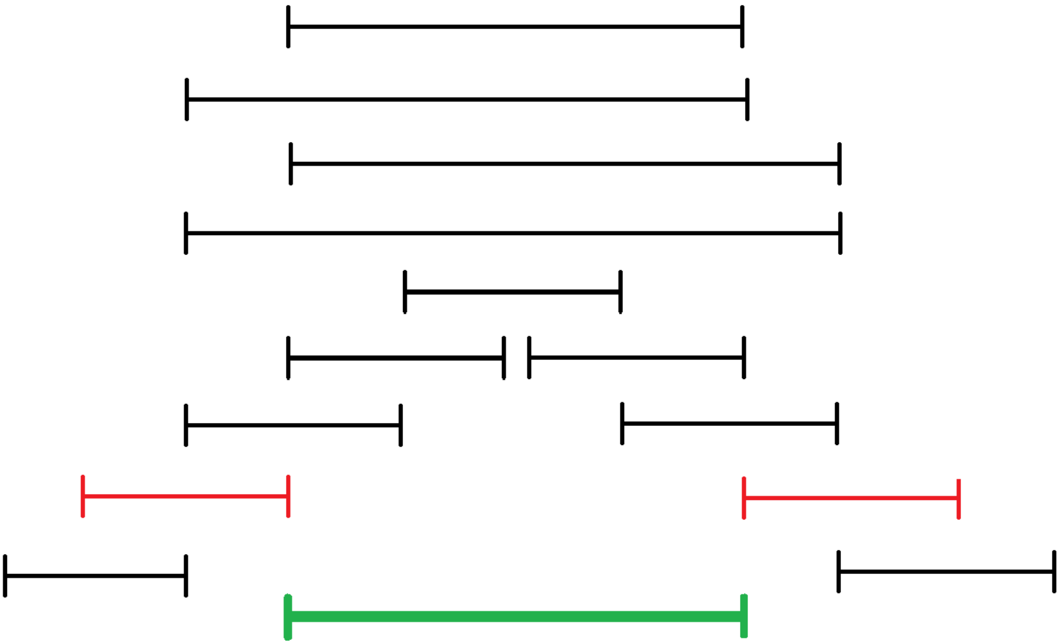


13 relative positions of intervals can be expressed using $\leq$



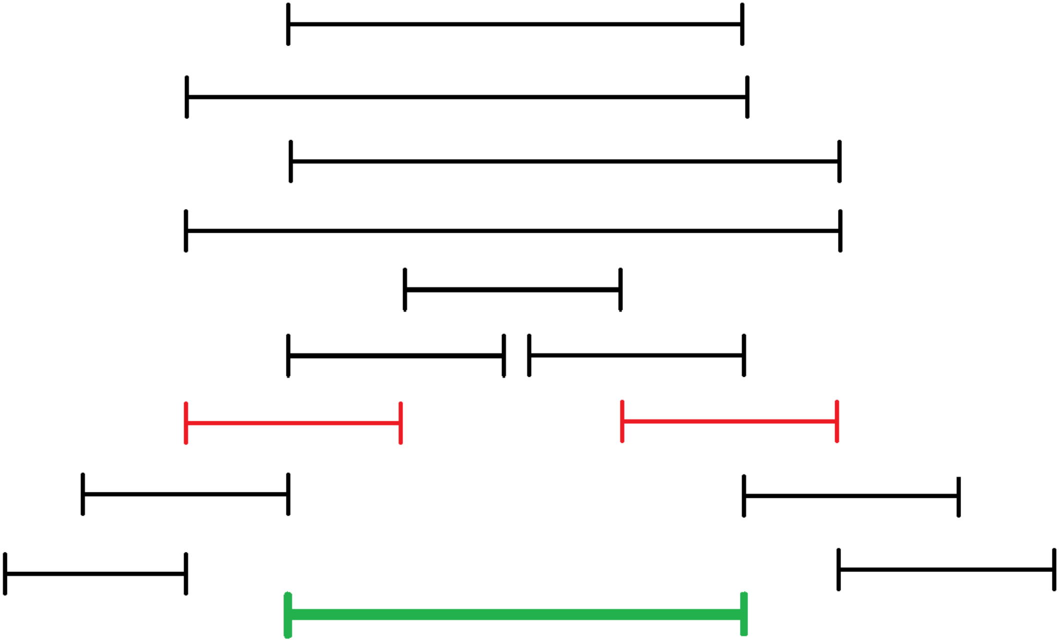
Before and after.

13 relative positions of intervals can be expressed using $\leq$



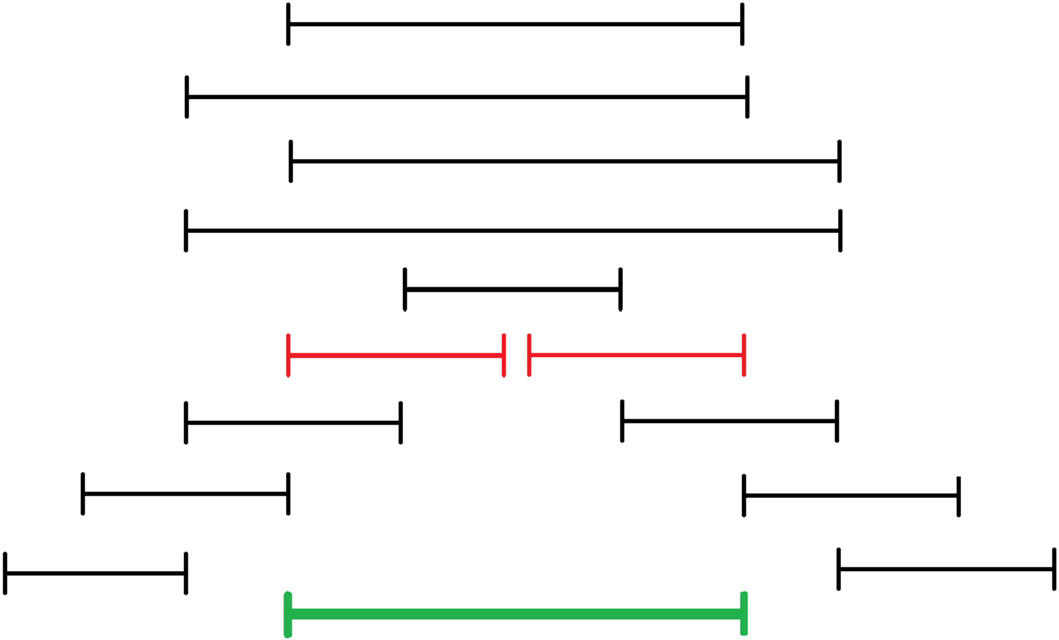
Meet and met by.

13 relative positions of intervals can be expressed using $\leq$



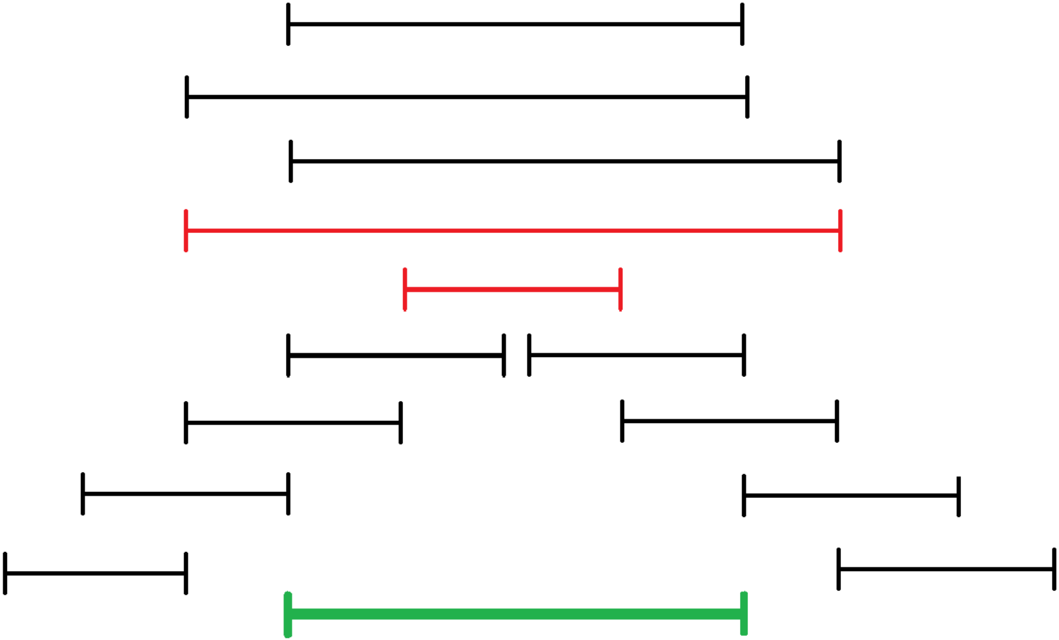
Overlaps and overlapped by.

13 relative positions of intervals can be expressed using $\leq$



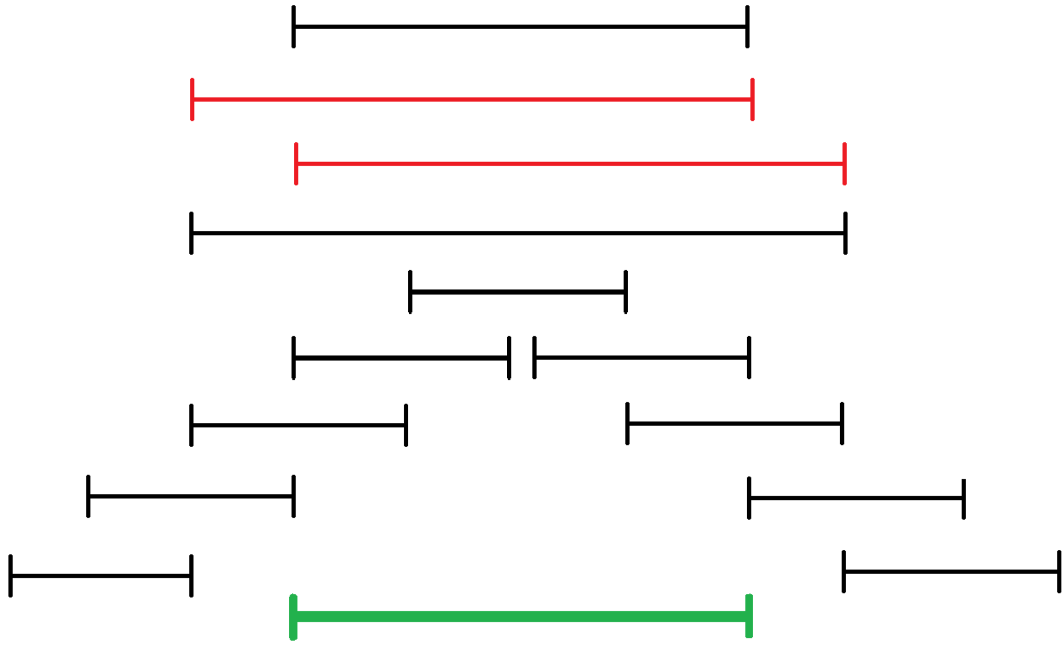
Starts and finishes.

13 relative positions of intervals can be expressed using $\leq$



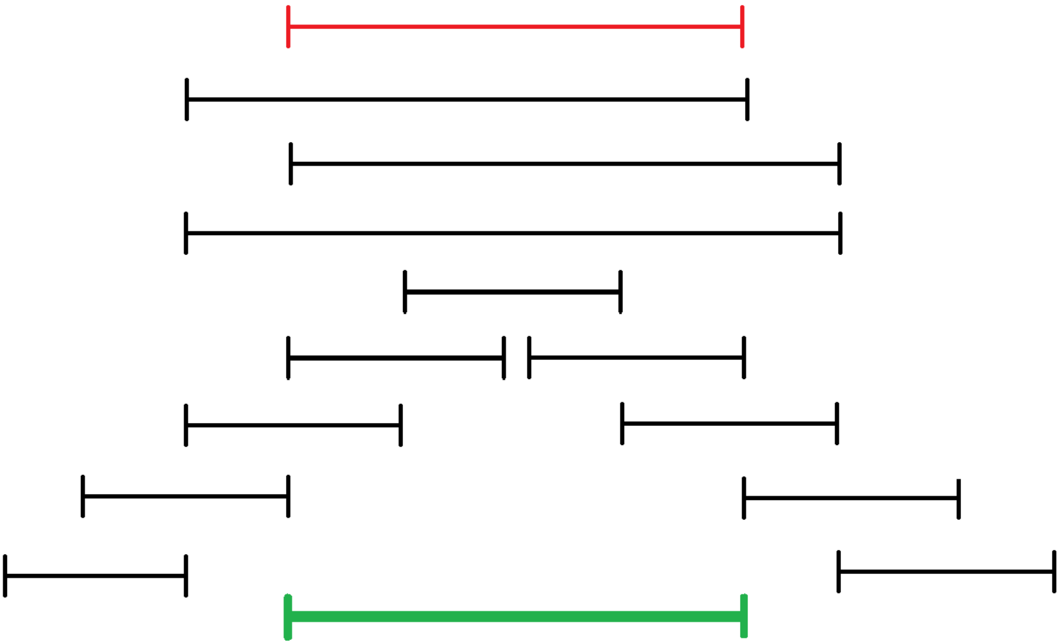
Contains and during.

13 relative positions of intervals can be expressed using $\leq$



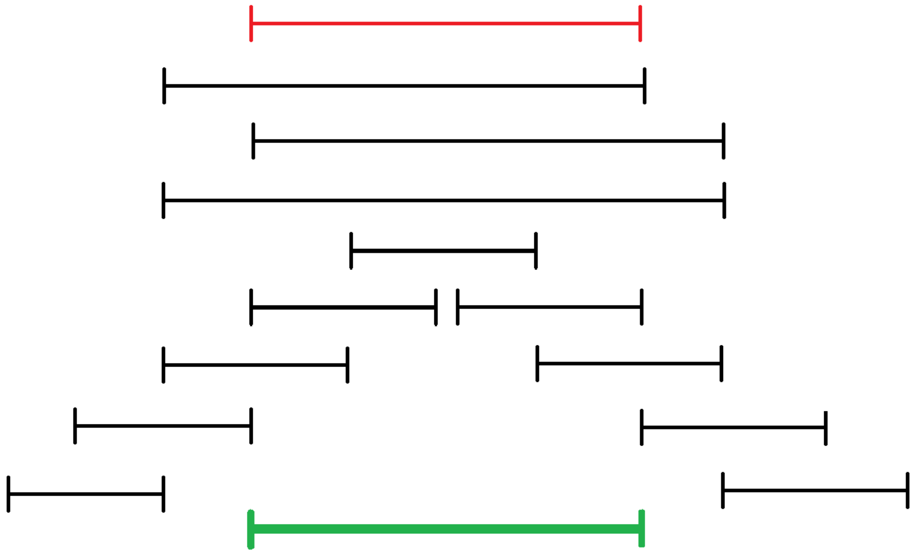
Started by and finished by.

13 relative positions of intervals can be expressed using $\leq$



Equals.

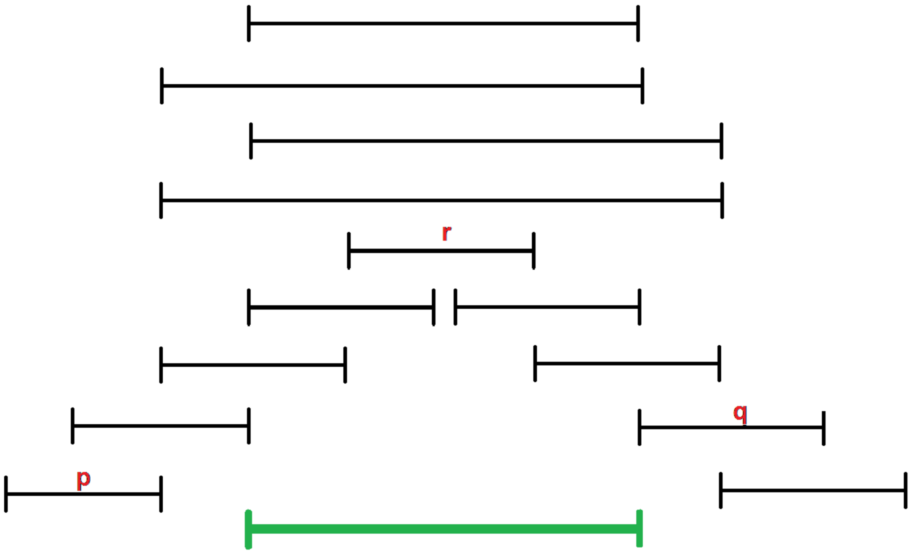
What can we do with those relations?



Allen's algebra.

$$\forall xy.x \textit{ before } y \Rightarrow \exists z.z \textit{ meet } y \wedge z \textit{ met by } x$$

What can we do with those relations?



Halpern–Shoham logic.

$$\langle before \rangle p \wedge \langle during \rangle (r \wedge \langle after \rangle q)$$

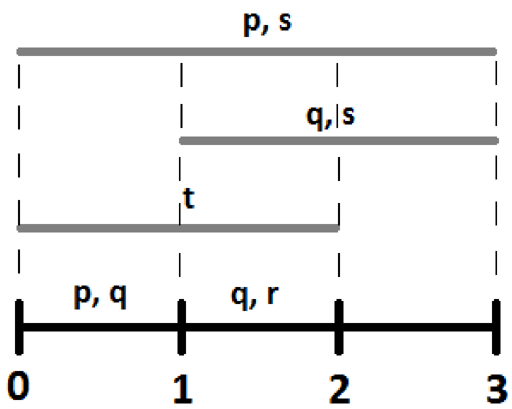
Formally - the models

- Any total order $\mathcal{D} = \langle \mathbb{D}, \leq \rangle$
- Set of propositional variables $\mathcal{V}ar$
- Classic temporal logics — labeling $\gamma : \mathbb{D} \rightarrow \mathcal{P}(\mathcal{V}ar)$

Formally - the models

- Any total order $\mathcal{D} = \langle \mathbb{D}, \leq \rangle$
- Set of propositional variables $\mathcal{V}ar$
- Classic temporal logics — labeling $\gamma : \mathbb{D} \to \mathcal{P}(\mathcal{V}ar)$
- Interval temporal logics — labeling $\gamma : \mathrm{I}(\mathbb{D}) \to \mathcal{P}(\mathcal{V}ar)$,

  where $\mathrm{I}(\mathbb{D}) = \{[a,b] | a, b \in \mathbb{D} \wedge a \leq b\}$



The Halpern–Shoham Logic

- Halpern–Shoham logic contains 12 operators $A, \bar{A}, B, \bar{B}, D, \bar{D}, E, \bar{E}, L, \bar{L}, O, \bar{O}$

The Halpern–Shoham Logic

- Halpern–Shoham logic contains 12 operators $A, \bar{A}, B, \bar{B}, D, \bar{D}, E, \bar{E}, L, \bar{L}, O, \bar{O}$
- Halpern–Shoham (LICS'86) — undecidability (of the satisfiability problem for the formulae of the HS logic) using "after", "begins" and "ends".

The Halpern–Shoham Logic

- Halpern–Shoham logic contains 12 operators $A, \bar{A}, B, \bar{B}, D, \bar{D}, E, \bar{E}, L, \bar{L}, O, \bar{O}$
- Halpern–Shoham (LICS'86) — undecidability (of the satisfiability problem for the formulae of the HS logic) using "after", "begins" and "ends".
- Lodaya ('00) — undecidability using "begins" and "ends".

The Halpern–Shoham Logic

- Halpern–Shoham logic contains 12 operators $A, \bar{A}, B, \bar{B}, D, \bar{D}, E, \bar{E}, L, \bar{L}, O, \bar{O}$
- Halpern–Shoham (LICS'86) — undecidability (of the satisfiability problem for the formulae of the HS logic) using "after", "begins" and "ends".
- Lodaya ('00) — undecidability using "begins" and "ends".
- Further results ('07-'10) — undecidability of O, $B\bar{E}$, $\bar{B}\bar{E}$, $A\bar{A}D$, $\bar{A}\bar{D}\bar{B}$, ...

The Halpern–Shoham Logic

- Halpern–Shoham logic contains 12 operators $A, \bar{A}, B, \bar{B}, D, \bar{D}, E, \bar{E}, L, \bar{L}, O, \bar{O}$
- Halpern–Shoham (LICS'86) — undecidability (of the satisfiability problem for the formulae of the HS logic) using "after", "begins" and "ends".
- Lodaya ('00) — undecidability using "begins" and "ends".
- Further results ('07-'10) — undecidability of O, $B\bar{E}$, $\bar{B}\bar{E}$, $A\bar{A}D$, $\bar{A}\bar{D}\bar{B}$, ...
- The positive side: decidability of $B\bar{B}$, $A\bar{A}$, $AB\bar{B}\bar{L}$ (next presentation), ...

The Halpern–Shoham Logic

- Halpern–Shoham logic contains 12 operators $A, \bar{A}, B, \bar{B}, D, \bar{D}, E, \bar{E}, L, \bar{L}, O, \bar{O}$
- Halpern–Shoham (LICS'86) — undecidability (of the satisfiability problem for the formulae of the HS logic) using “after”, “begins” and “ends”.
- Lodaya ('00) — undecidability using “begins” and “ends”.
- Further results ('07-'10) — undecidability of O, $B\bar{E}$, $\bar{B}\bar{E}$, $A\bar{A}D$, $\bar{A}\bar{D}\bar{B}$, ...
- The positive side: decidability of $B\bar{B}$, $A\bar{A}$, $AB\bar{B}\bar{L}$ (next presentation), ...
- The decidability of $B\bar{B}D\bar{D}L\bar{L}$ over the class of dense structures.

The Halpern–Shoham Logic

- Halpern–Shoham logic contains 12 operators $A, \bar{A}, B, \bar{B}, D, \bar{D}, E, \bar{E}, L, \bar{L}, O, \bar{O}$
- Halpern–Shoham (LICS'86) — undecidability (of the satisfiability problem for the formulae of the HS logic) using "after", "begins" and "ends".
- Lodaya ('00) — undecidability using "begins" and "ends".
- Further results ('07-'10) — undecidability of O, $B\bar{E}$, $\bar{B}\bar{E}$, $A\bar{A}D$, $\bar{A}\bar{D}\bar{B}$, ...
- The positive side: decidability of $B\bar{B}$, $A\bar{A}$, $AB\bar{B}\bar{L}$ (next presentation), ...
- The decidability of $B\bar{B}D\bar{D}L\bar{L}$ over the class of dense structures.

The Logic of Subintervals

- The logic of subintervals contains only one operator (D):
- $\langle D\rangle\varphi$ is satisfied if φ is satisfied in some subinterval.
- $[D]\varphi$ is satisfied if φ is satisfied in all subintervals.

Logic of subintervals results

- The logic of subintervals is PSPACE–complete over the class of **dense** orders (D. Bresolin, V. Goranko, A. Montanari, P. Sala, 2007).

Logic of subintervals results

- The logic of subintervals is PSPACE–complete over the class of **dense** orders (D. Bresolin, V. Goranko, A. Montanari, P. Sala, 2007).
- The logic of **reflexive** subintervals is PSPACE–complete over the class of (finite) discrete orders (A. Montanari, I. Pratt-Hartmann, P. Sala, 2010).

Logic of subintervals results

- The logic of subintervals is PSPACE–complete over the class of **dense** orders (D. Bresolin, V. Goranko, A. Montanari, P. Sala, 2007).
- The logic of **reflexive** subintervals is PSPACE–complete over the class of (finite) discrete orders (A. Montanari, I. Pratt-Hartmann, P. Sala, 2010).
- The logic of subintervals is in EXPSPACE over the class of discrete orders if we allow to **remove** intervals from models (follows from T. Schwentick, T. Zeume, 2010).

Logic of subintervals results

- The logic of subintervals is PSPACE–complete over the class of **dense** orders (D. Bresolin, V. Goranko, A. Montanari, P. Sala, 2007).
- The logic of **reflexive** subintervals is PSPACE–complete over the class of (finite) discrete orders (A. Montanari, I. Pratt-Hartmann, P. Sala, 2010).
- The logic of subintervals is in EXPSPACE over the class of discrete orders if we allow to **remove** intervals from models (follows from T. Schwentick, T. Zeume, 2010).
- The problem they left open: the complexity of the logic of subintervals over the class of **discrete** orders. Remark: No nontrivial lower bounds were known.

What can we express using the “subinterval” relation?

- “each morning I spend a while thinking of you”

What can we express using the "subinterval" relation?

- "each morning I spend a while thinking of you"
- "each nice period of my life contains an unpleasant fragment"

What can we express using the “subinterval” relation?

- “each morning I spend a while thinking of you”
- “each nice period of my life contains an unpleasant fragment”
- “there is no error while printing”

Logic of subintervals results

- The logic of subintervals is PSPACE–complete over the class of dense orders (D. Bresolin, V. Goranko, A. Montanari, P. Sala, 2007).
- The logic of reflexive subintervals is PSPACE–complete over the class of (finite) discrete orders (A. Montanari, I. Pratt-Hartmann, P. Sala, 2010).
- The logic of subintervals is in EXPSPACE over the class of discrete orders if we allow to remove intervals from models (follows from T. Schwentick, T. Zeume, 2010).
- The problem they left open: the complexity of the logic of subintervals over the class of discrete orders. Remark: No nontrivial lower bounds where known.

Logic of subintervals results

- The logic of subintervals is PSPACE–complete over the class of dense orders (D. Bresolin, V. Goranko, A. Montanari, P. Sala, 2007).
- The logic of reflexive subintervals is PSPACE–complete over the class of (finite) discrete orders (A. Montanari, I. Pratt-Hartmann, P. Sala, 2010).
- The logic of subintervals is in EXPSPACE over the class of discrete orders if we allow to remove intervals from models (follows from T. Schwentick, T. Zeume, 2010).
- The problem they left open: the complexity of the logic of subintervals over the class of discrete orders. Remark: No nontrivial lower bounds where known.
- Prefixes and subintervals are enough to make the HS logic undecidable over the class of discrete orders (E. Kieroński, J. Marcinkowski, J. Michaliszyn, ICALP 2010).

Logic of subintervals results

- The logic of subintervals is PSPACE–complete over the class of dense orders (D. Bresolin, V. Goranko, A. Montanari, P. Sala, 2007).
- The logic of reflexive subintervals is PSPACE–complete over the class of (finite) discrete orders (A. Montanari, I. Pratt-Hartmann, P. Sala, 2010).
- The logic of subintervals is in EXPSPACE over the class of discrete orders if we allow to remove intervals from models (follows from T. Schwentick, T. Zeume, 2010).
- The problem they left open: the complexity of the logic of subintervals over the class of discrete orders. Remark: No nontrivial lower bounds where known.
- Prefixes and subintervals are enough to make the HS logic undecidable over the class of discrete orders (E. Kieroński, J. Marcinkowski, J. Michaliszyn, ICALP 2010).
- The logic of subintervals is **undecidable** over the class of discrete orders (this presentation).

Our undecidability result

Assumptions

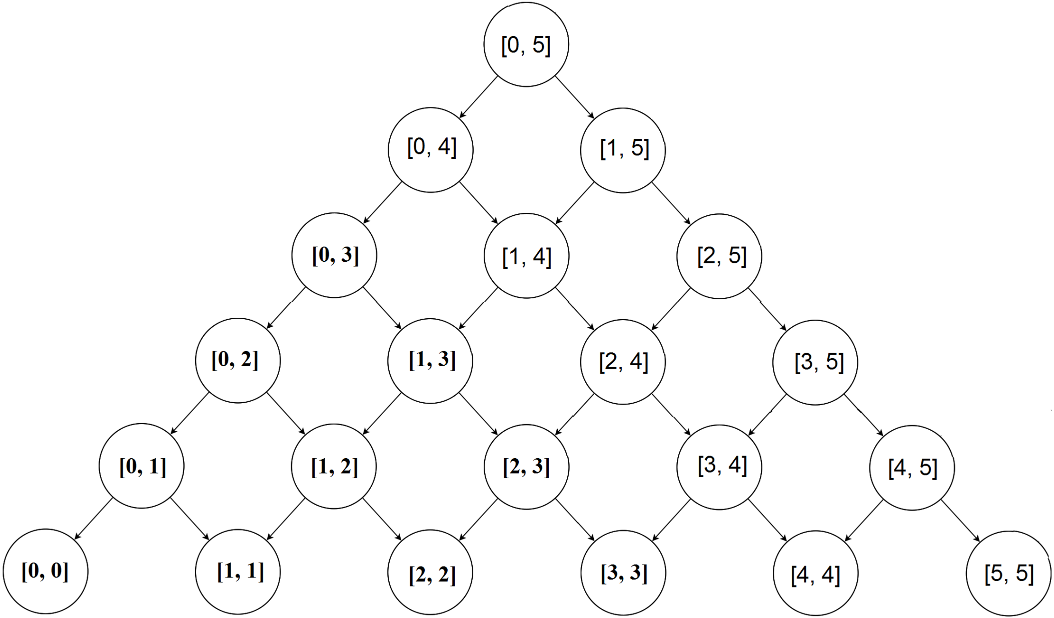
Assumption	In our paper	In this presentation
Order	All discrete	All finite
Do we allow point intervals ($[a, a]$)?	Whatever	Yes
Subinterval relation or superinterval relation?	Does not matter	Subinterval

An overview of the proof

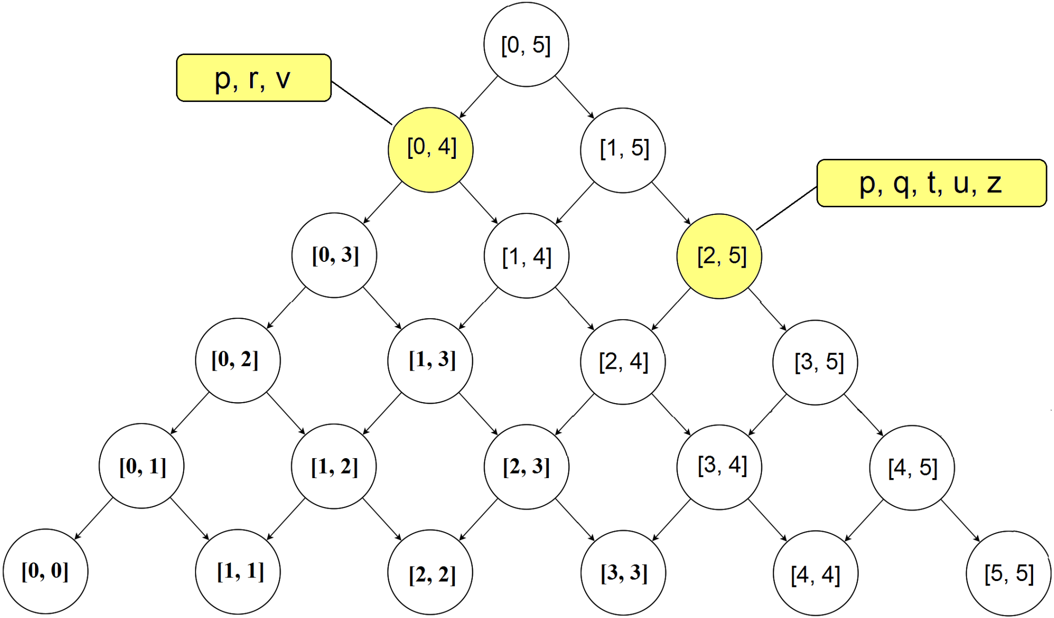
The logic of subintervals is **undecidable**

over the class of all finite structures.

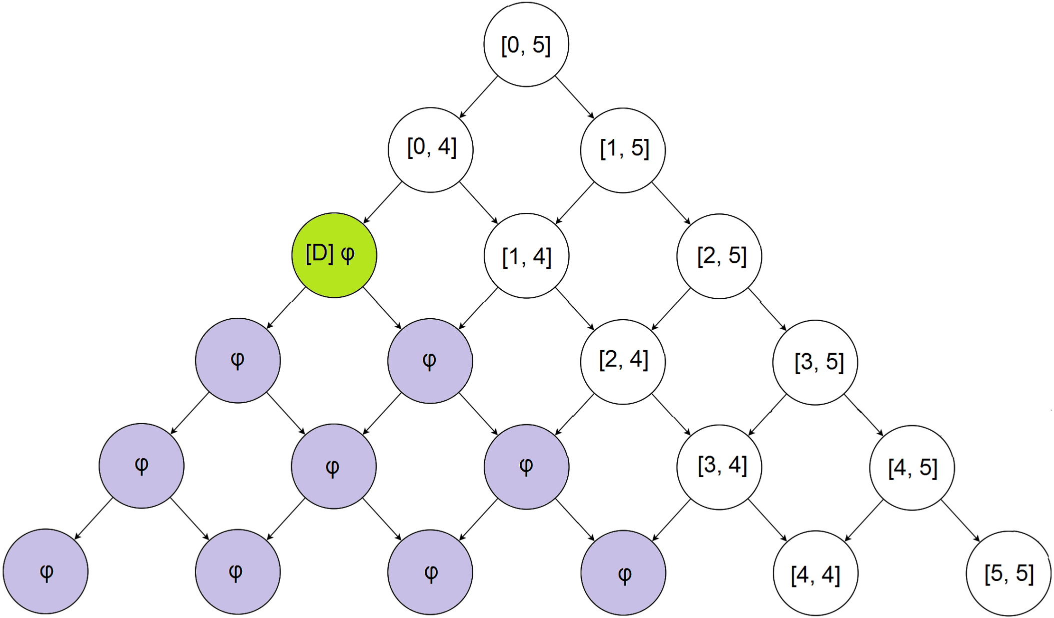
How we imagine that — "triangle structures"



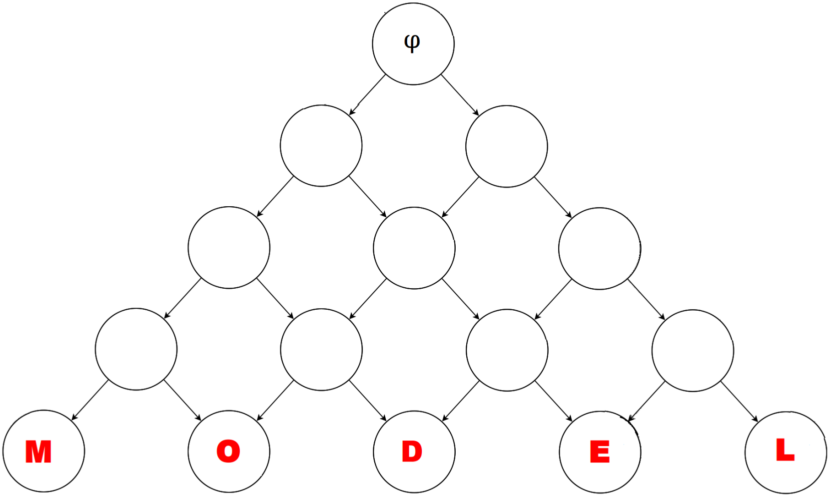
How we imagine that — "triangle structures"



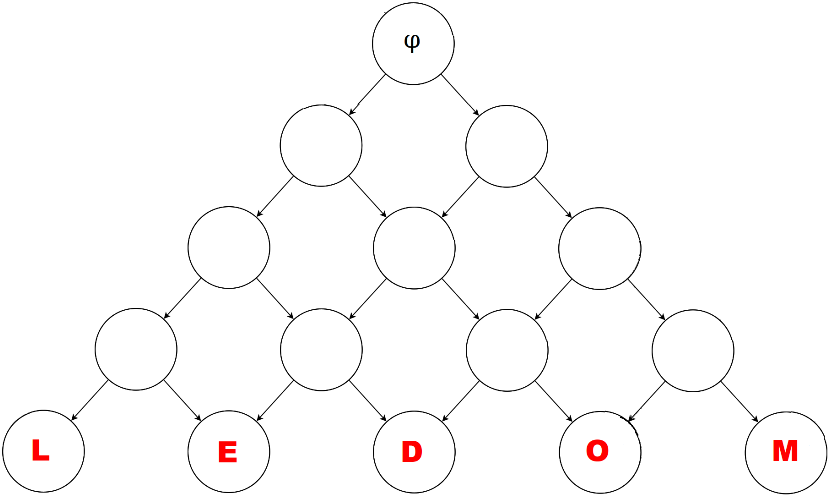
How we imagine that — “triangle structures”



Problem 1 — symmetry

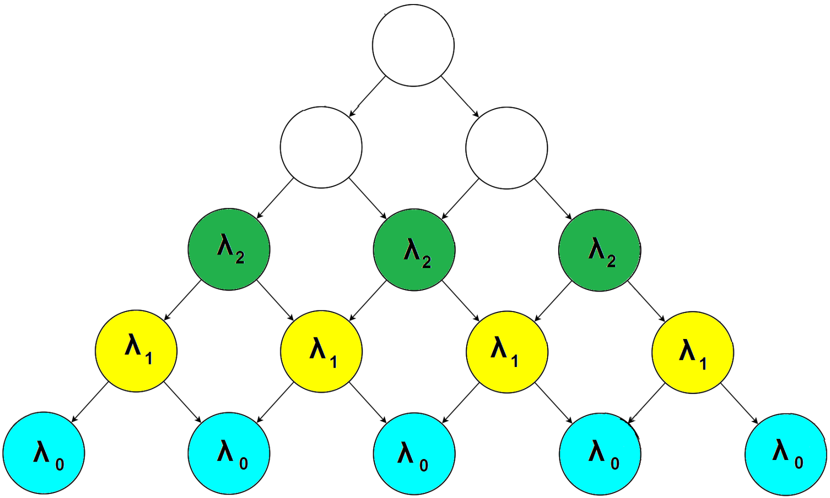


Problem 1 — symmetry



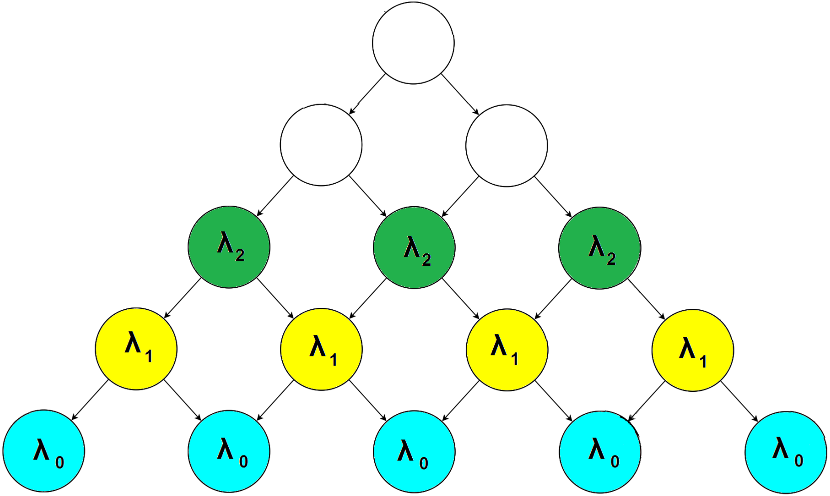
Our little library of formulae

- λ_0: $[D]\bot$



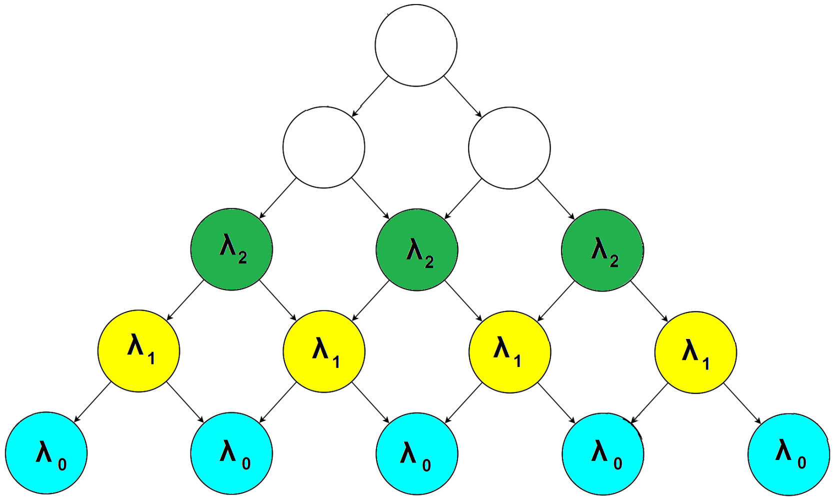
Our little library of formulae

- λ_0: $[D]\bot$
- $\lambda_{\leq 1}$: $[D]\lambda_0$



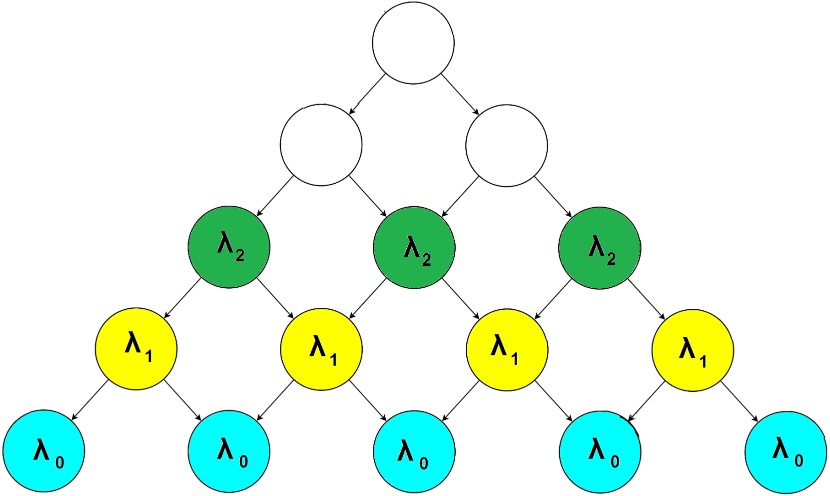
Our little library of formulae

- λ_0: $[D]\bot$
- $\lambda_{\leq 1}$: $[D]\lambda_0$



Our little library of formulae

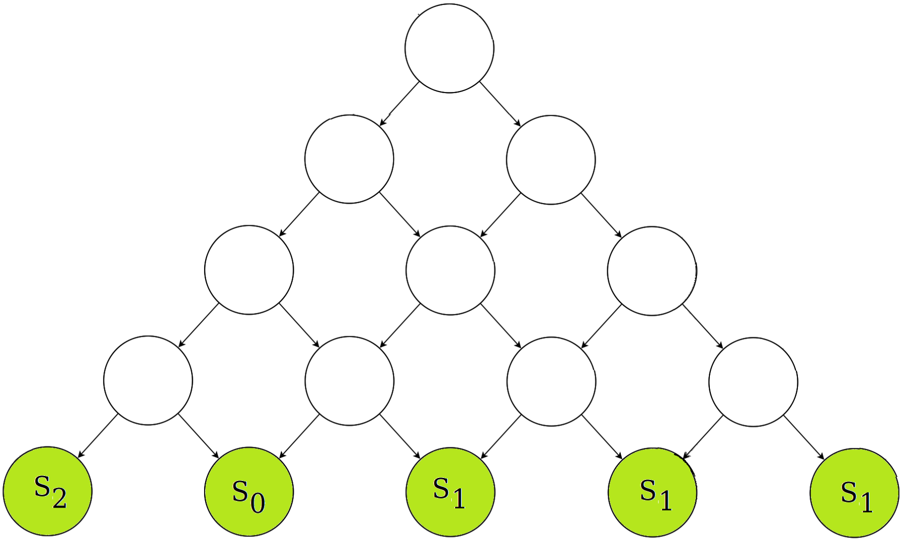
- λ_0: $[D]\bot$
- $\lambda_{\leq 1}$: $[D]\lambda_0$
- λ_1: $\lambda_{\leq 1} \wedge \neg\lambda_0$



Problem 1 — symmetry

Now we can deal with the symmetry

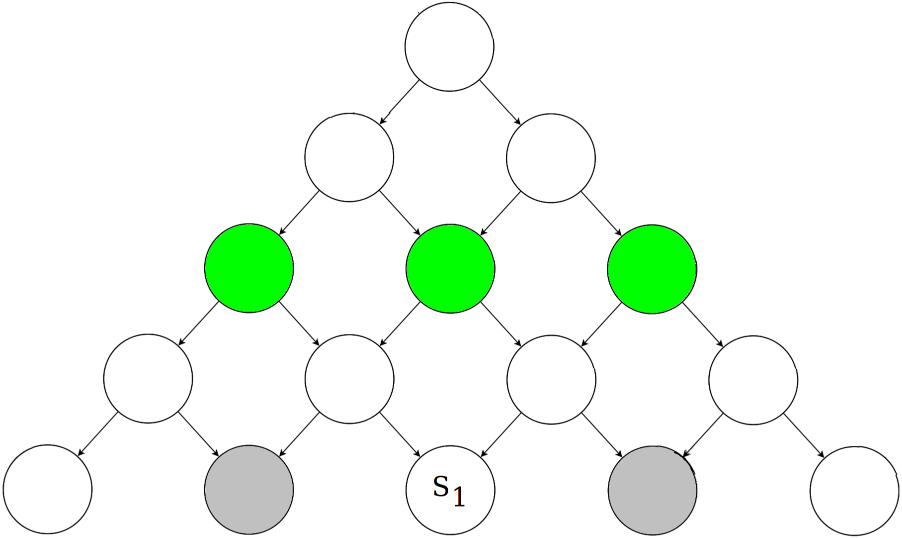
Problem 1 — symmetry



Each point-interval is labeled with exactly one of s_0, s_1, s_2, and each interval labeled with s_0, s_1, or s_2 is an point-interval.

$$[D](\lambda_0 \Leftrightarrow s_1 \vee s_2 \vee s_0) \wedge [D]\neg(s_0 \wedge s_1) \wedge [D]\neg(s_1 \wedge s_2) \wedge [D]\neg(s_0 \wedge s_2)$$

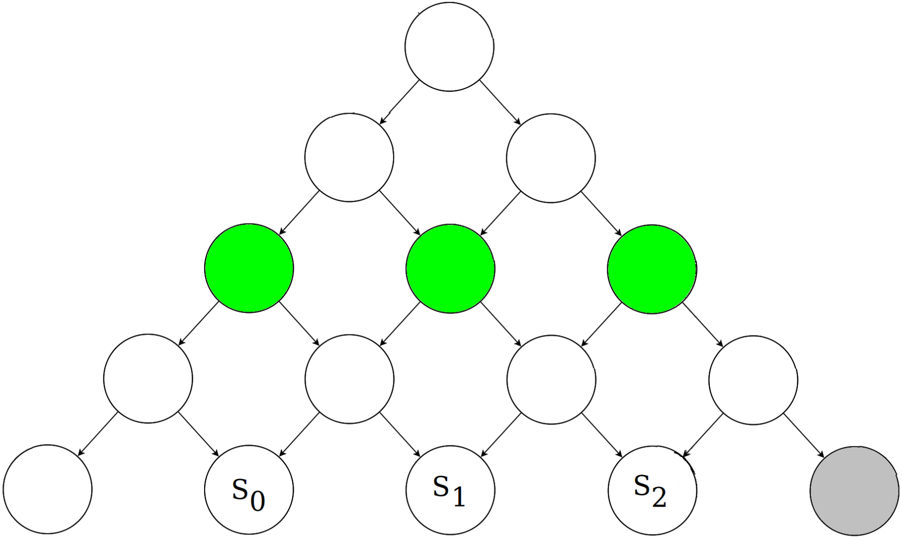
Problem 1 — symmetry



Each interval with length 2 contains intervals labeled with s_0, s_1, and s_2.

$$[D](\lambda_2 \Rightarrow \langle D \rangle s_0 \wedge \langle D \rangle s_1 \wedge \langle D \rangle s_2)$$

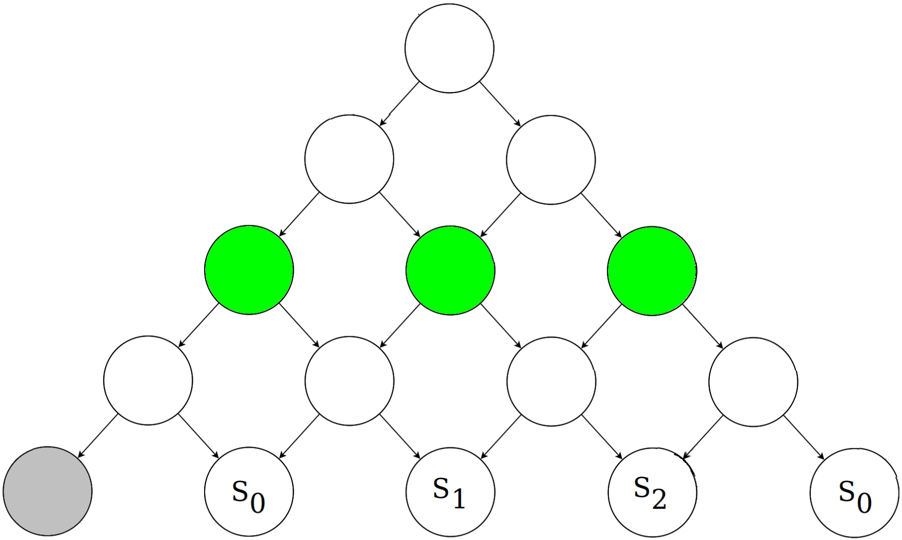
Problem 1 — symmetry



Each interval with length 2 contains intervals labeled with s_0, s_1, and s_2.

$$[D](\lambda_2 \Rightarrow \langle D \rangle s_0 \wedge \langle D \rangle s_1 \wedge \langle D \rangle s_2)$$

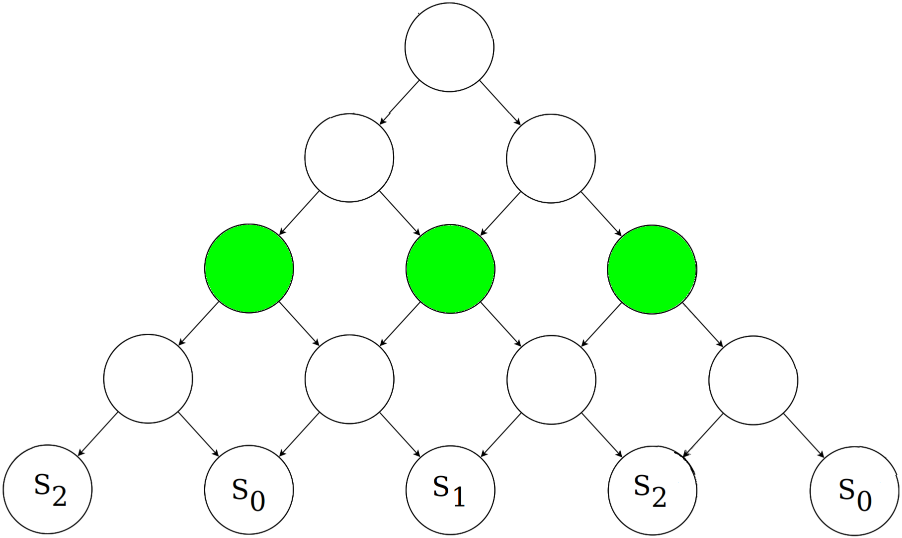
Problem 1 — symmetry



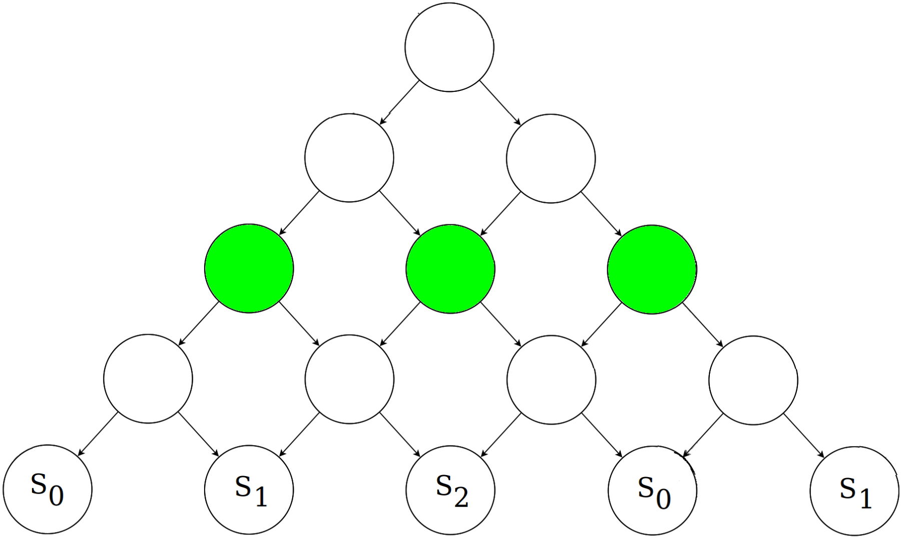
Each interval with length 2 contains intervals labeled with s_0, s_1, and s_2.

$$[D](\lambda_2 \Rightarrow \langle D\rangle s_0 \wedge \langle D\rangle s_1 \wedge \langle D\rangle s_2)$$

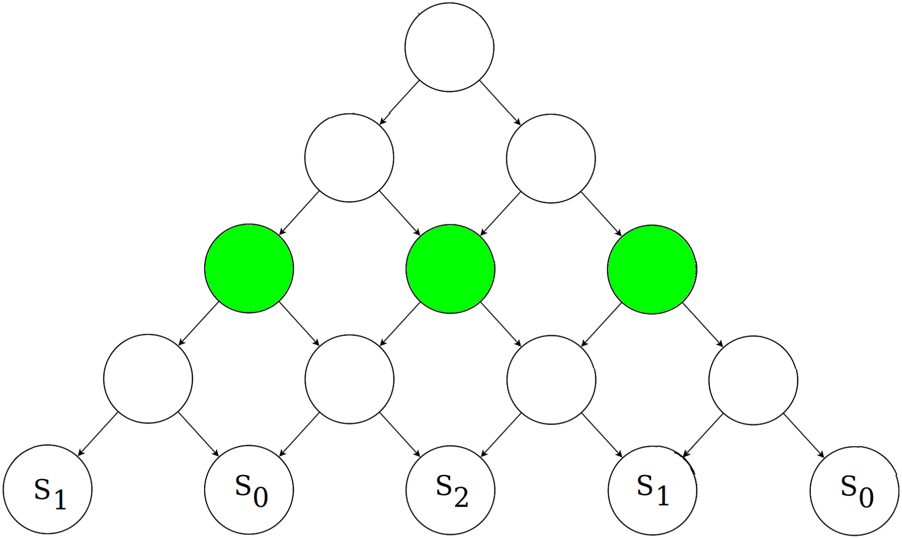
Problem 1 — symmetry



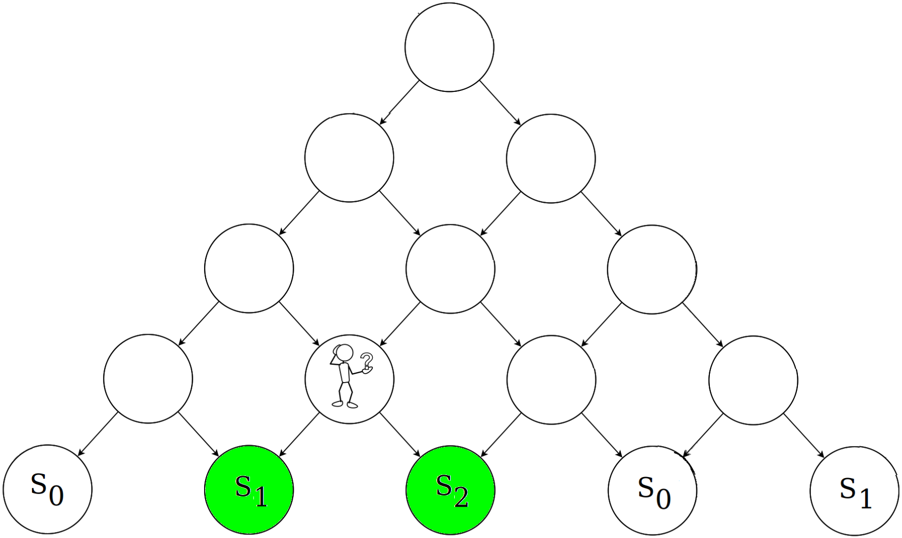
Problem 1 — symmetry



Problem 1 — symmetry



Problem 1 — symmetry



The source of the undecidability

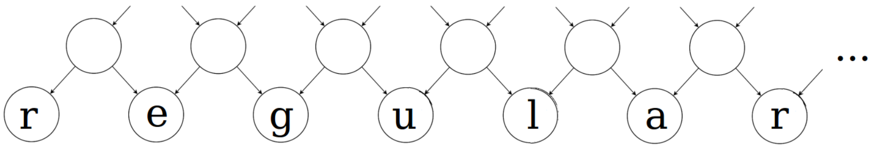
Regularity + ability to measure $\geq$ undecidability

The source of the undecidability

$$\text{Regularity} + \text{ability to measure} \geq \text{undecidability}$$

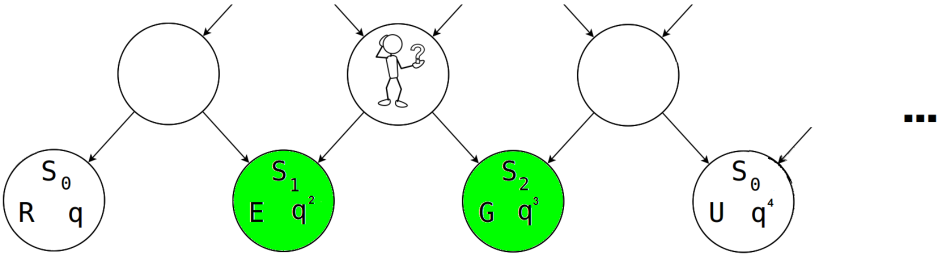
Actually, with D we only have very limited ability to measure. One of the technical lemmas is that this limited ability already leads to the undecidability.

Second step – regularity



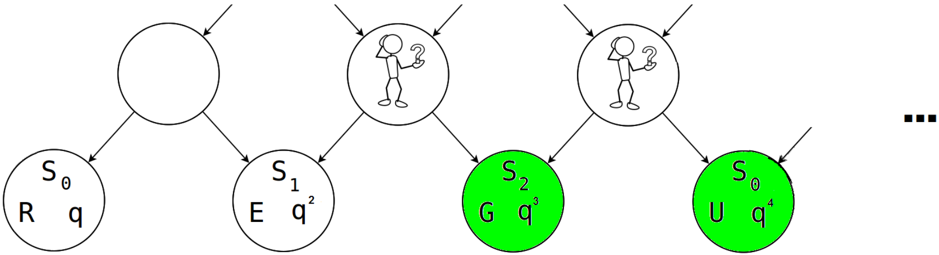
We can encode any finite automaton.

Second step – regularity



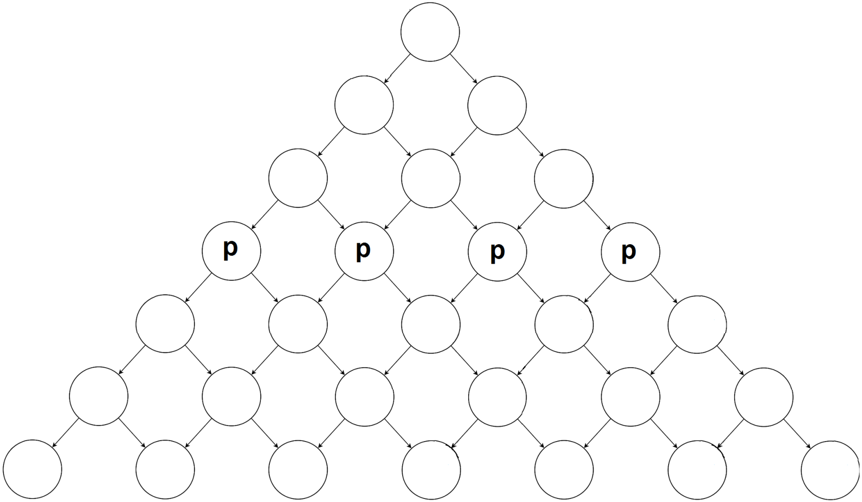
We can encode any finite automaton.

Second step – regularity

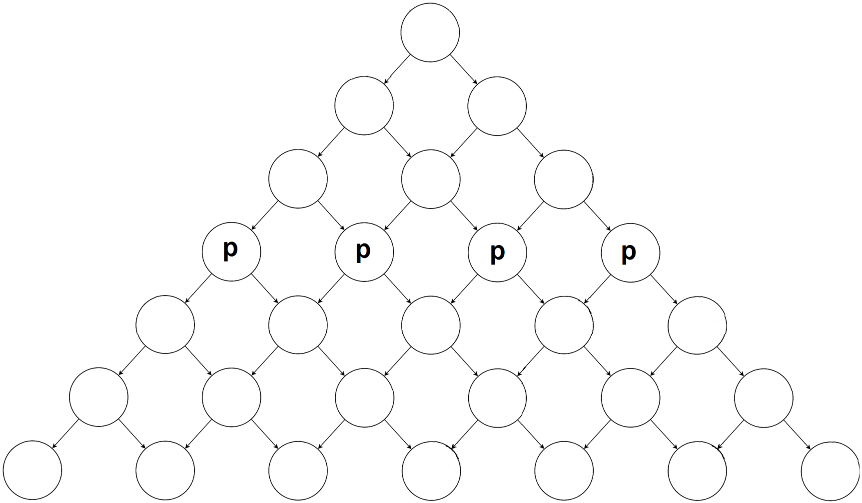


We can encode any finite automaton.

Third step — the cloud.



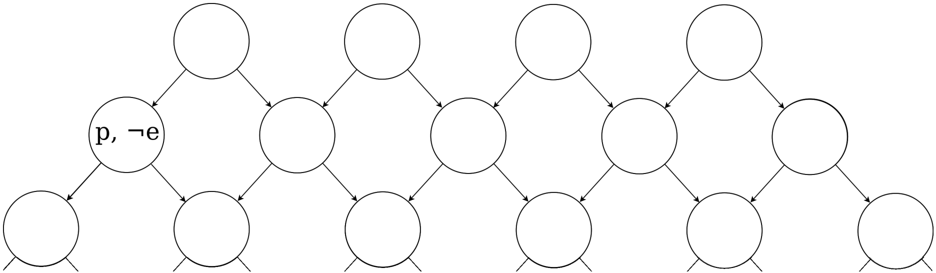
Third step — the cloud.



What remains to be explained:

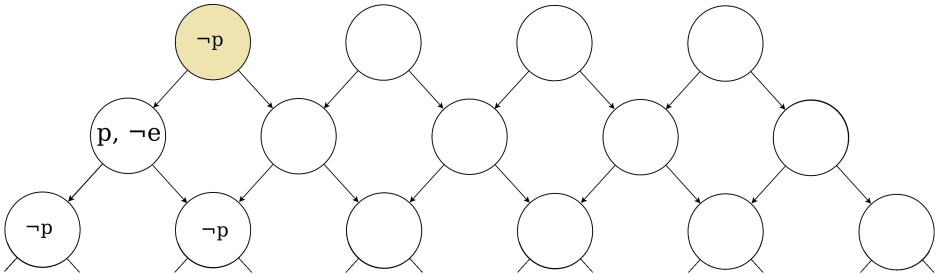
1. How to write a formula saying that a propositional variable p is a cloud.
2. How to use this cloud.

Third step — the cloud.



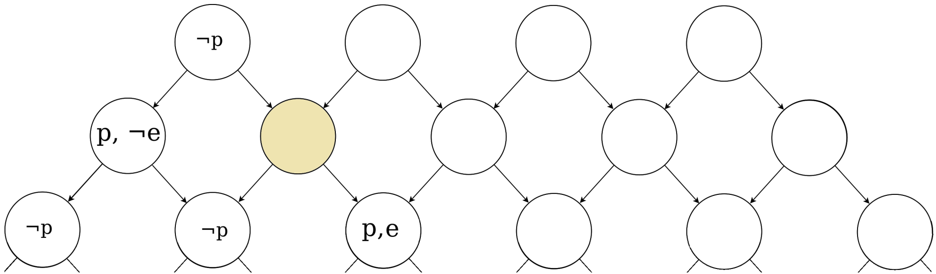
- There exists an interval labeled with p.
- Intervals labeled with p do not contain each other.
- Any interval that contains an interval with p, contains two such intervals (one with e and one with $\neg e$).

Third step — the cloud.



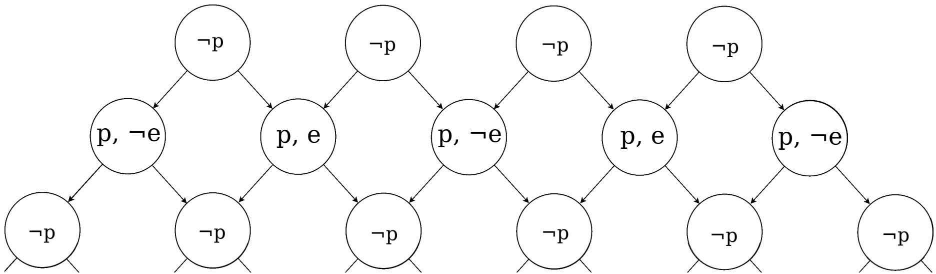
- There exists an interval labeled with p.
- Intervals labeled with p do not contain each other.
- Any interval that contains an interval with p, contains two such intervals (one with e and one with $\neg e$).

Third step — the cloud.



- There exists an interval labeled with p.
- Intervals labeled with p do not contain each other.
- Any interval that contains an interval with p, contains two such intervals (one with e and one with $\neg e$).

Third step — the cloud.



- There exists an interval labeled with p.
- Intervals labeled with p do not contain each other.
- Any interval that contains an interval with p, contains two such intervals (one with e and one with $\neg e$).

Third step — the cloud.

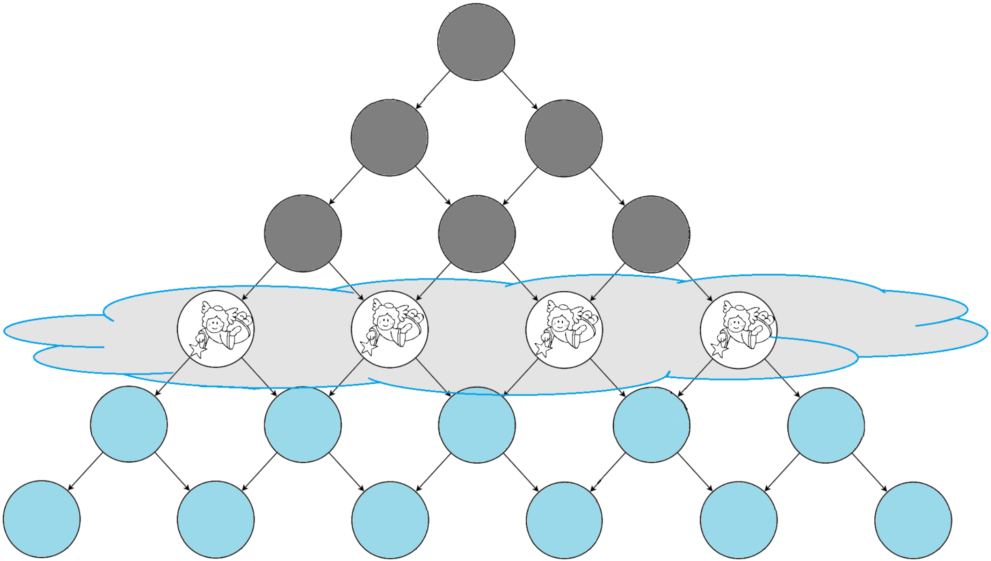
- There exists an interval labeled with p.
- Intervals labeled with p do not contain each other.
- Any interval that contains an interval with p, contains two such intervals (one with e and one with $\neg e$).

1. $\langle D \rangle p$
2. $[D](p \Rightarrow [D]\neg p)$
3. $[D](\langle D \rangle p \Rightarrow \langle D \rangle (p \wedge e) \wedge \langle D \rangle (p \wedge \neg e))$

Toy example

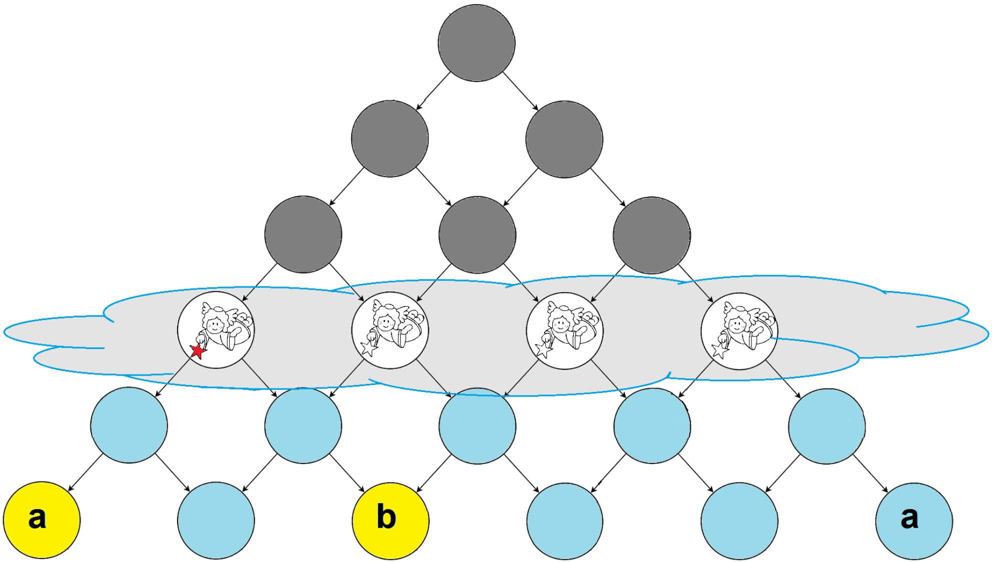
- We already know how to encode regularity.
- Consider the regular language defined by $(ac^*bc^*)^*$.
- We want to force that each maximal block of c has the same length.

Toy example



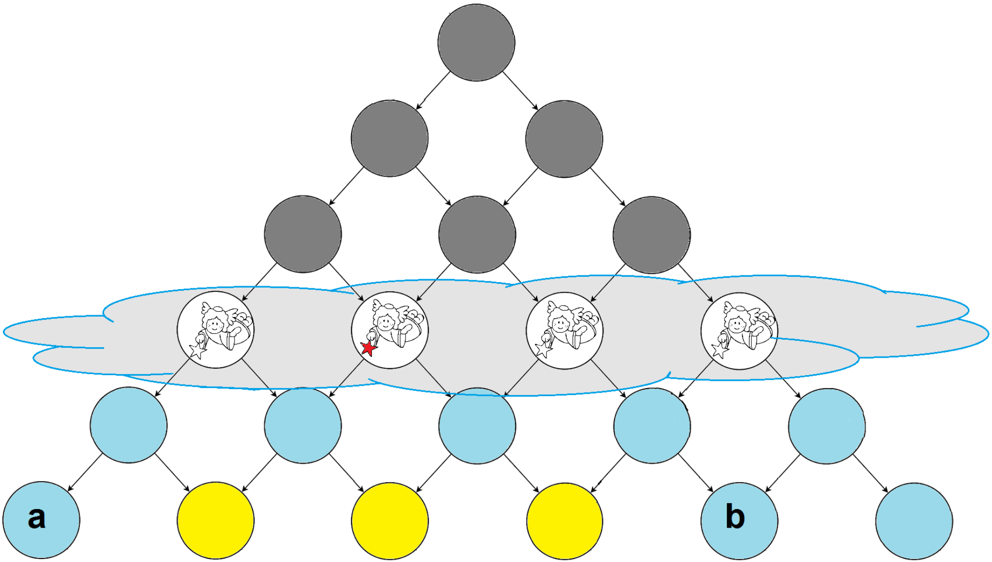
- No angel can see both *a* and *b*.
- Each angel has to see *a* or *b*.

Toy example



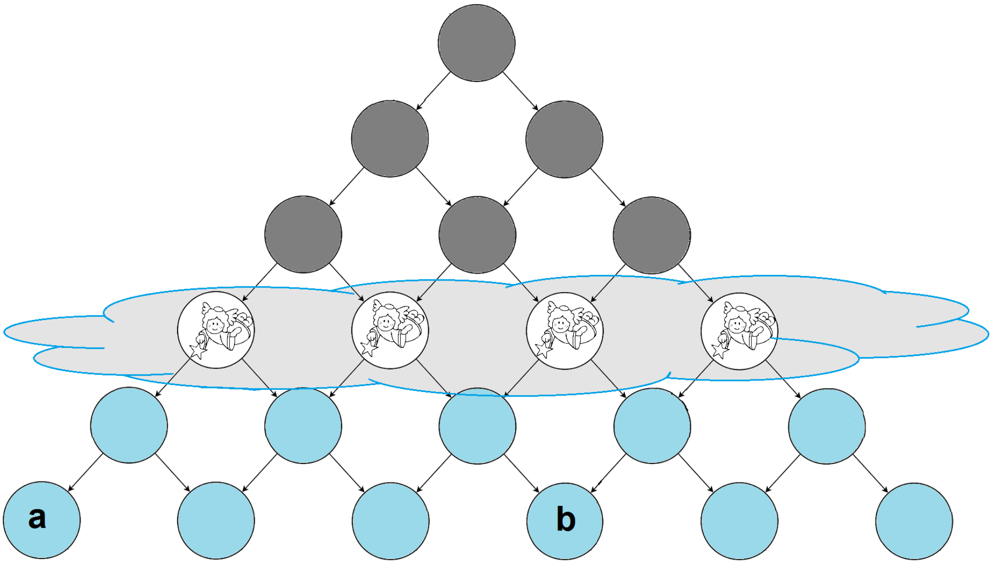
- No angel can see both *a* and *b*.
- Each angel has to see *a* or *b*.

Toy example



- No angel can see both *a* and *b*.
- Each angel has to see *a* or *b*.

Toy example



- No angel can see both *a* and *b*.
- Each angel has to see *a* or *b*.

Summary

- The logic of subintervals is **decidable** over the class of **dense** structures.
- The logic of subintervals is **undecidable** over the class of **discrete** structures.

Summary

- The logic of subintervals is **decidable** over the class of **dense** structures.
- The logic of subintervals is **undecidable** over the class of **discrete** structures.
- Is the logic of subintervals decidable over the class of all structures? (open)

Thank you!

Thank you!

Coming next: Davide Bresolin, Angelo Montanari, Pietro Sala and Guido Sciavicco. What's decidable about Halpern and Shoham's interval logic? The maximal fragment $AB\bar{B}L$