Adding two equivalence relations to the interval temporal logic *AB*

Angelo Montanari[1], Marco Pazzaglia[1] and Pietro Sala[2]

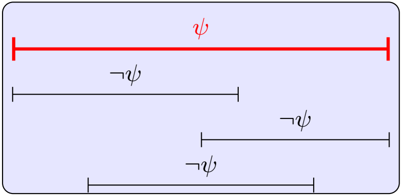
[1]University of Udine
Department of Mathematics and Computer Science
[2]University of Verona
Department of Computer Science

ICTCS 2014
Perugia, September 17, 2014

Interval Temporal Logics

Interval temporal logics: an alternative approach to point-based temporal representation and reasoning.

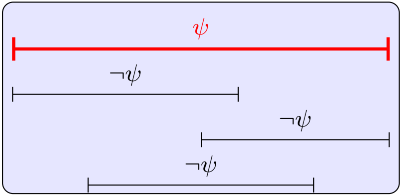
Truth of formulas is defined over intervals rather than points.



Interval Temporal Logics

Interval temporal logics: an alternative approach to point-based temporal representation and reasoning.

Truth of formulas is defined over intervals rather than points.

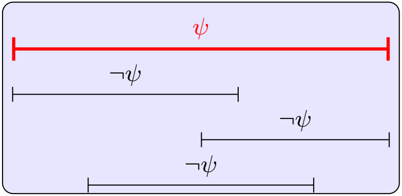


- Interval temporal logics are very expressive (compared to point-based temporal logics).

Interval Temporal Logics

Interval temporal logics: an alternative approach to point-based temporal representation and reasoning.

Truth of formulas is defined over intervals rather than points.

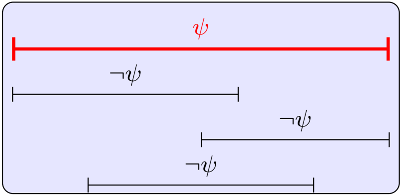


- Interval temporal logics are very expressive (compared to point-based temporal logics).
- Formulas of interval logics express properties of pairs of time points rather than of single time points, and are evaluated as sets of such pairs, i.e., as binary relations.

Interval Temporal Logics

Interval temporal logics: an alternative approach to point-based temporal representation and reasoning.

Truth of formulas is defined over intervals rather than points.



- Interval temporal logics are very expressive (compared to point-based temporal logics).
- Formulas of interval logics express properties of pairs of time points rather than of single time points, and are evaluated as sets of such pairs, i.e., as binary relations.
- Apart from very special (easy) cases, there is no reduction of the satisfiability/validity in interval logics to monadic second-order logic, and therefore Rabin's theorem is not applicable here.

The general picture

- Halpern and Shoham's modal logic of intervals (*HS*)
  - *HS* features 12 modalilities, one for each possible ordering of a pair of intervals (the so-called Allen's relations);
  - decidability and expressiveness of *HS* fragments (restrictions to subsets of *HS* modalities) have been systematically studied in the last decade.

The general picture

- Halpern and Shoham's modal logic of intervals (*HS*)
  - *HS* features 12 modalilities, one for each possible ordering of a pair of intervals (the so-called Allen's relations);
  - decidability and expressiveness of *HS* fragments (restrictions to subsets of *HS* modalities) have been systematically studied in the last decade.
- Decidability and expressiveness depend on two crucial factors: the selected set of modalities and the class of linear orders on which they are interpreted.

The general picture

- Halpern and Shoham's modal logic of intervals (*HS*)
  - *HS* features 12 modalilities, one for each possible ordering of a pair of intervals (the so-called Allen's relations);
  - decidability and expressiveness of *HS* fragments (restrictions to subsets of *HS* modalities) have been systematically studied in the last decade.
- Decidability and expressiveness depend on two crucial factors: the selected set of modalities and the class of linear orders on which they are interpreted.
- In the present work, we address the satisfiability problem for the logic *AB* of Allen's relation *meets* and *begun by* extended with two equivalence relations ($AB\sim_1\sim_2$ for short), interpreted over the class of finite linear orders.

2. $AB\sim_1\sim_2$

Syntax and Semantics

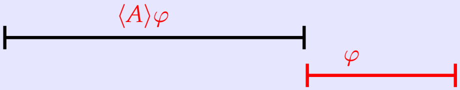
The formulas of the logic AB, from Allen's relations *meets* and *begun by*, are recursively defined as follows:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle A\rangle\varphi \mid \langle B\rangle\varphi$$

SYNTAX AND SEMANTICS

The formulas of the logic AB, from Allen's relations *meets* and *begun by*, are recursively defined as follows:

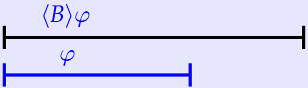
$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle A\rangle\varphi \mid \langle B\rangle\varphi$$



SYNTAX AND SEMANTICS

The formulas of the logic *AB*, from Allen's relations *meets* and *begun by*, are recursively defined as follows:

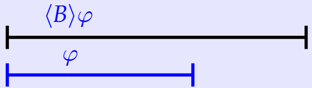
$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle A \rangle \varphi \mid \langle B \rangle \varphi$$



Syntax and Semantics

The formulas of the logic *AB*, from Allen's relations *meets* and *begun by*, are recursively defined as follows:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle A\rangle\varphi \mid \langle B\rangle\varphi$$



- $AB_{\sim}$
  - We extend the language of *AB* with a special proposition letter $\sim$ interpreted as an equivalence relation over the points of the domain.
  - An interval $[x, y]$ satisfies $\sim$ if and only if x and y belong to the same equivalence class.
- $AB_{\sim_1 \sim_2}$ is obtained from *AB* by adding two equivalence relations

Expressiveness

Examples of properties captured by AB:

EXPRESSIVENESS

Examples of properties captured by AB:

- To constrain the lenght of an interval to be equal to k ($k \in \mathbb{N}$):

$$\langle B\rangle^k \top \wedge [B]^{k+1} \perp$$

EXPRESSIVENESS

Examples of properties captured by AB:

- To constrain the lenght of an interval to be equal to k ($k \in \mathbb{N}$):

$$\langle B\rangle^k \top \wedge [B]^{k+1} \perp$$

- To constrain an interval to contain exactly one point (endpoints excluded) labeled with q:

$$\psi_{\exists! q} \equiv \langle B\rangle(\neg\pi \wedge \langle A\rangle(\pi \wedge q)) \wedge \big([B](\neg\pi \wedge \langle A\rangle(\pi \wedge q) \rightarrow [B]\langle A\rangle(\pi \wedge \neg q))\big)$$

EXPRESSIVENESS (CONT'D)

- The effects/benefits of the addition of one or more equivalence relations to a logic have been already studied in various settings, including (fragments of) first-order logic, linear temporal logic, metric temporal logic, and interval temporal logic.

EXPRESSIVENESS (CONT'D)

- The effects/benefits of the addition of one or more equivalence relations to a logic have been already studied in various settings, including (fragments of) first-order logic, linear temporal logic, metric temporal logic, and interval temporal logic.

  The increase in expressive power obtained from the extension of AB, interpreted over finite linear orders and $\mathbb{N}$, with an equivalence relation $\sim$ makes it possible to establish an original connection between interval temporal logics and extended regular languages of finite and infinite words (extended ω-regular languages).

Previous results

The satisfiability problem for:

- *AB* is **EXPSPACE**-complete on the class of finite linear orders (and on $\mathbb{N}$);

A. Montanari, G, Puppis, P. Sala, and G. Sciavicco. Decidability of the Interval Temporal Logic $AB\bar{B}$ over the Natural Numbers. Proc. of the 27th STACS, 2010.

Previous results

The satisfiability problem for:

- AB is **EXPSPACE**-complete on the class of finite linear orders (and on $\mathbb{N}$);

A. Montanari, G, Puppis, P. Sala, and G. Sciavicco. Decidability of the Interval Temporal Logic $AB\bar{B}$ over the Natural Numbers. Proc. of the 27th STACS, 2010.

- $AB\sim$ is decidable (but non-primitive recursive hard) on the class of finite linear orders (and undecidable on $\mathbb{N}$).

A. Montanari, and P. Sala. Adding an Equivalence Relation to the Interval Logic $AB\bar{B}$: Complexity and Expressiveness. Proc. of the 28th LICS, 2013.

UNDECIDABILITY OF $AB{\sim_1}{\sim_2}$

The results given in the paper complete the study of the extensions of AB with equivalence relations.

Teorema

The satisfiability problem for $AB{\sim_1}{\sim_2}$, interpreted on the class of finite linear orders, is undecidable.

Undecidability of $AB{\sim_1}{\sim_2}$

The results given in the paper complete the study of the extensions of AB with equivalence relations.

Teorema

The satisfiability problem for $AB{\sim_1}{\sim_2}$, interpreted on the class of finite linear orders, is undecidable.

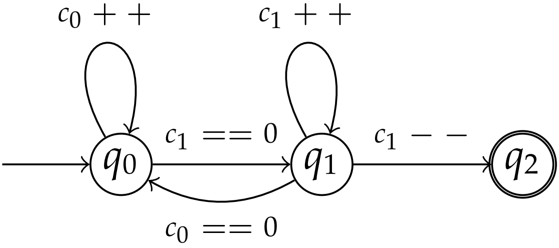
The proof relies on a reduction from the 0-0 reachability problem for counter machines (with two counters) to the satisfiability problem of $AB{\sim_1}{\sim_2}$ over finite linear orders.

Counter machines

Definizione

A counter machine is a triple of the form $M = (Q, k, \delta)$, where Q is a finite set of states, k is the number of counters, which assume values in $\mathbb{N}$, and δ is a function that maps $q \in Q$ in a transition rule of the following form:

1. $value(h) \leftarrow value(h) + 1;\ goto\ q'$, for some $1 \leq h \leq k$ and $q' \in Q$;
2. $if\ value(h) = 0\ then\ goto\ q'\ else\ value(h) \leftarrow value(h) - 1;\ goto\ q''$, for some $1 \leq h \leq k$ and $q', q'' \in Q$.



0-0 REACHABILITY AND ψ_M^{0-0}

Definizione

The 0-0 reachability problem for a counter machine M consists of determining, given two states $q_0, q_f \in Q$, if there exists a computation of M from the configuration $(q_0, 0, 0)$ to the configuration $(q_f, 0, 0)$.

0-0 reachability and ψ_M^{0-0}

Definizione

The 0-0 reachability problem for a counter machine M consists of determining, given two states $q_0, q_f \in Q$, if there exists a computation of M from the configuration $(q_0, 0, 0)$ to the configuration $(q_f, 0, 0)$.

Teorema (Minsky, 1967)

The problem of 0-0 reachability for counter machines with at least two counters is undecidable.

0-0 REACHABILITY AND ψ_M^{0-0}

Definizione

The 0-0 reachability problem for a counter machine M consists of determining, given two states $q_0, q_f \in Q$, if there exists a computation of M from the configuration $(q_0, 0, 0)$ to the configuration $(q_f, 0, 0)$.

Teorema (Minsky, 1967)

The problem of 0-0 reachability for counter machines with at least two counters is undecidable.

Given a counter machine M (with two counters), we build a formula ψ_M^{0-0} such that

ψ_M^{0-0} *is satisfiable iff there exists a computation from* $(q_0, 0, 0)$ *to* $(q_f, 0, 0)$ *in* M.

ENCODING

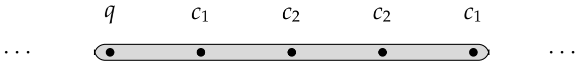
OUR MODEL OF COMPUTATION

- points (=point-intervals) are partitioned into two sets: state-points (points with label in Q) and counter-points (points with labels in $\{c_1, c_2\}$).

Encoding

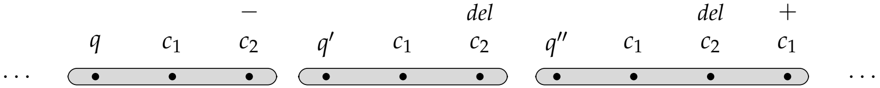
Our model of computation

- points (=point-intervals) are partitioned into two sets: state-points (points with label in Q) and counter-points (points with labels in $\{c_1, c_2\}$).
- A configuration (q, v_1, v_2) is represented by a sequence of consecutive points:
  - the first point is a state-point q;
  - the following points are counter-points (v_1 of them with label c_1 and v_2 of them with label c_2, in a random order).



ENCODING (CONT'D)

OUR MODEL OF COMPUTATION

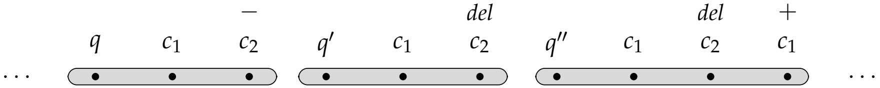


$$(q, 1, 1) \rightarrow (q', 1, 0) \rightarrow (q'', 2, 0)$$

- A computation (from q_0 to q_f) is given by a sequence of consecutive configurations.

Encoding (cont'd)

Our model of computation



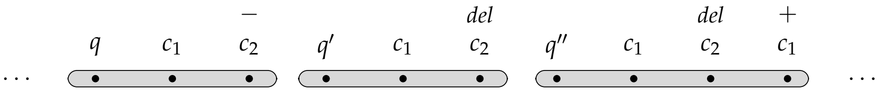
$$(q, 1, 1) \to (q', 1, 0) \to (q'', 2, 0)$$

- A computation (from q_0 to q_f) is given by a sequence of consecutive configurations.
- Counter-points with labels $+$ and $-$ respectively denote
  - points that are introduced in a configuration when a counter is increased;
  - points that are deleted from the next configuration when a counter is decreased

(increments and decrements must be consistent with state-points and transitions of M).

Encoding (cont'd)

Our model of computation



$$(q, 1, 1) \to (q', 1, 0) \to (q'', 2, 0)$$

- A computation (from q_0 to q_f) is given by a sequence of consecutive configurations.
- Counter-points with labels $+$ and $-$ respectively denote
  - points that are introduced in a configuration when a counter is increased;
  - points that are deleted from the next configuration when a counter is decreased

  (increments and decrements must be consistent with state-points and transitions of M).
- Deleted points are not removed from the configuration, but labeled with *del*.

ENCODING (CONT'D)

FORMULA ψ_M^{0-0}

We build a formula ψ_M^{0-0} as the conjunction of the following formulas:

- ψ_0 and ψ_f constrain the structure of the first $((q_0, 0, 0))$ and the last ($(q_f, 0, 0)$) configuration, respectively;

Encoding (cont'd)

Formula ψ_M^{0-0}

We build a formula ψ_M^{0-0} as the conjunction of the following formulas:

- ψ_0 and ψ_f constrain the structure of the first $((q_0, 0, 0))$ and the last ($(q_f, 0, 0)$) configuration, respectively;
- ψ_{points} forces the conditions on points;

Encoding (cont'd)

Formula ψ_M^{0-0}

We build a formula ψ_M^{0-0} as the conjunction of the following formulas:

- ψ_0 and ψ_f constrain the structure of the first $((q_0, 0, 0))$ and the last ($(q_f, 0, 0))$ configuration, respectively;
- ψ_{points} forces the conditions on points;
- ψ_δ ensures the consistency between state-points and $+/-$ labelings in the transitions of the machine M;

ENCODING (CONT'D)

FORMULA ψ_M^{0-0}

We build a formula ψ_M^{0-0} as the conjunction of the following formulas:

- ψ_0 and ψ_f constrain the structure of the first $((q_0, 0, 0))$ and the last ($(q_f, 0, 0)$) configuration, respectively;
- ψ_{points} forces the conditions on points;
- ψ_δ ensures the consistency between state-points and $+/-$ labelings in the transitions of the machine M;
- $\psi_\sim$ guarantees the consistency between counter-points and transitions in each configuration.

ENCODING (CONT'D)

FORMULA ψ_M^{0-0}

We build a formula ψ_M^{0-0} as the conjunction of the following formulas:

- ψ_0 and ψ_f constrain the structure of the first $((q_0, 0, 0))$ and the last ($(q_f, 0, 0)$) configuration, respectively;
- ψ_{points} forces the conditions on points;
- ψ_δ ensures the consistency between state-points and $+/-$ labelings in the transitions of the machine M;
- $\psi_\sim$ guarantees the consistency between counter-points and transitions in each configuration.

Formulas $\psi_0, \psi_f, \psi_{points}$, and ψ_δ can be expressed in the basic fragment AB (devoid of equivalence relations).

ENCODING (CONT'D)

FORMULA ψ_M^{0-0}

We build a formula ψ_M^{0-0} as the conjunction of the following formulas:

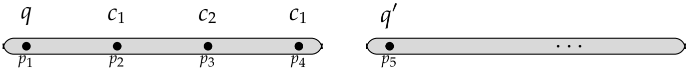
- ψ_0 and ψ_f constrain the structure of the first $((q_0, 0, 0))$ and the last ($(q_f, 0, 0)$) configuration, respectively;
- ψ_{points} forces the conditions on points;
- ψ_δ ensures the consistency between state-points and $+/-$ labelings in the transitions of the machine M;
- $\psi_\sim$ guarantees the consistency between counter-points and transitions in each configuration.

Formulas $\psi_0, \psi_f, \psi_{points}$, and ψ_δ can be expressed in the basic fragment AB (devoid of equivalence relations).

The most difficult condition to enforce is $\psi_\sim$: the number of points in a configuration is constrained by the number of points in the previous one and it depends on the fired transition.

Encoding (cont'd)

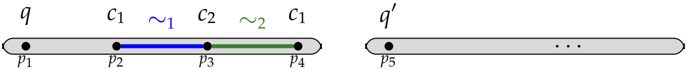
Constrains imposed by the formula $\psi_\sim$



ENCODING (CONT'D)

CONSTRAINS IMPOSED BY THE FORMULA $\psi_\sim$

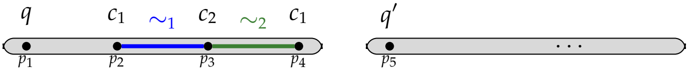
1. Counter-points belonging to a configuration form a chain of unit-length intervals that alternates $\sim_1$ and $\sim_2$ labeled intervals.



Encoding (cont'd)

Constrains imposed by the formula $\psi_\sim$

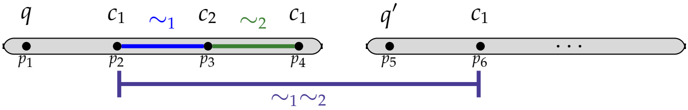
1. Counter-points belonging to a configuration form a chain of unit-length intervals that alternates $\sim_1$ and $\sim_2$ labeled intervals.
2. Inside a configuration, any interval of length greater than 1 makes neither $\sim_1$ nor $\sim_2$ true, and any interval of length equal to 1 makes either $\sim_1$ or $\sim_2$ true.



ENCODING (CONT'D)

CONSTRAINS IMPOSED BY THE FORMULA $\psi_\sim$

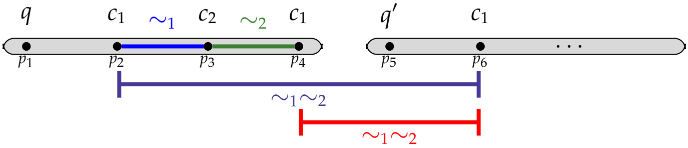
1. Counter-points belonging to a configuration form a chain of unit-length intervals that alternates $\sim_1$ and $\sim_2$ labeled intervals.
2. Inside a configuration, any interval of length greater than 1 makes neither $\sim_1$ nor $\sim_2$ true, and any interval of length equal to 1 makes either $\sim_1$ or $\sim_2$ true.
3. Each counter-point belonging to a non-final configuration begins an interval labeled with both $\sim_1$ and $\sim_2$, which crosses exactly one state-point and ends at another counter-point. Moreover, we constrain the two endpoints of such an interval to be labeled with the same label (we say that the two counter-points are linked). Finally, we impose that the first point in a configuration is linked to the first point in the next configuration.



Encoding (cont'd)

Constrains imposed by the formula $\psi_\sim$

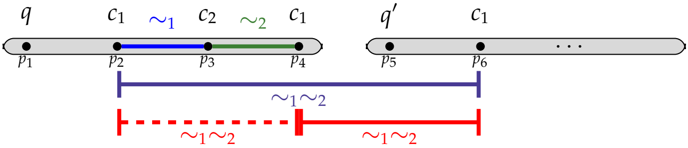
1. Counter-points belonging to a configuration form a chain of unit-length intervals that alternates $\sim_1$ and $\sim_2$ labeled intervals.
2. Inside a configuration, any interval of length greater than 1 makes neither $\sim_1$ nor $\sim_2$ true, and any interval of length equal to 1 makes either $\sim_1$ or $\sim_2$ true.
3. Each counter-point belonging to a non-final configuration begins an interval labeled with both $\sim_1$ and $\sim_2$, which crosses exactly one state-point and ends at another counter-point. Moreover, we constrain the two endpoints of such an interval to be labeled with the same label (we say that the two counter-points are linked). Finally, we impose that the first point in a configuration is linked to the first point in the next configuration.



ENCODING (CONT'D)

CONSTRAINS IMPOSED BY THE FORMULA $\psi_\sim$

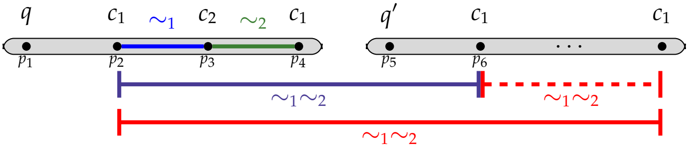
1. Counter-points belonging to a configuration form a chain of unit-length intervals that alternates $\sim_1$ and $\sim_2$ labeled intervals.
2. Inside a configuration, any interval of length greater than 1 makes neither $\sim_1$ nor $\sim_2$ true, and any interval of length equal to 1 makes either $\sim_1$ or $\sim_2$ true.
3. Each counter-point belonging to a non-final configuration begins an interval labeled with both $\sim_1$ and $\sim_2$, which crosses exactly one state-point and ends at another counter-point. Moreover, we constrain the two endpoints of such an interval to be labeled with the same label (we say that the two counter-points are linked). Finally, we impose that the first point in a configuration is linked to the first point in the next configuration.



Encoding (cont'd)

Constrains imposed by the formula $\psi_\sim$

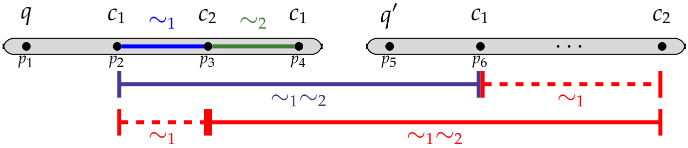
1. Counter-points belonging to a configuration form a chain of unit-length intervals that alternates $\sim_1$ and $\sim_2$ labeled intervals.
2. Inside a configuration, any interval of length greater than 1 makes neither $\sim_1$ nor $\sim_2$ true, and any interval of length equal to 1 makes either $\sim_1$ or $\sim_2$ true.
3. Each counter-point belonging to a non-final configuration begins an interval labeled with both $\sim_1$ and $\sim_2$, which crosses exactly one state-point and ends at another counter-point. Moreover, we constrain the two endpoints of such an interval to be labeled with the same label (we say that the two counter-points are linked). Finally, we impose that the first point in a configuration is linked to the first point in the next configuration.



ENCODING (CONT'D)

CONSTRAINS IMPOSED BY THE FORMULA $\psi_\sim$

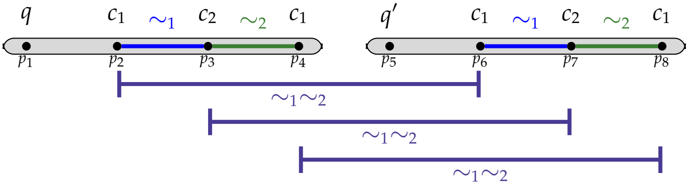
1. Counter-points belonging to a configuration form a chain of unit-length intervals that alternates $\sim_1$ and $\sim_2$ labeled intervals.
2. Inside a configuration, any interval of length greater than 1 makes neither $\sim_1$ nor $\sim_2$ true, and any interval of length equal to 1 makes either $\sim_1$ or $\sim_2$ true.
3. Each counter-point belonging to a non-final configuration begins an interval labeled with both $\sim_1$ and $\sim_2$, which crosses exactly one state-point and ends at another counter-point. Moreover, we constrain the two endpoints of such an interval to be labeled with the same label (we say that the two counter-points are linked). Finally, we impose that the first point in a configuration is linked to the first point in the next configuration.



ENCODING (CONT'D)

CONSTRAINS IMPOSED BY THE FORMULA $\psi_\sim$

1. Counter-points belonging to a configuration form a chain of unit-length intervals that alternates $\sim_1$ and $\sim_2$ labeled intervals.
2. Inside a configuration, any interval of length greater than 1 makes neither $\sim_1$ nor $\sim_2$ true, and any interval of length equal to 1 makes either $\sim_1$ or $\sim_2$ true.
3. Each counter-point belonging to a non-final configuration begins an interval labeled with both $\sim_1$ and $\sim_2$, which crosses exactly one state-point and ends at another counter-point. Moreover, we constrain the two endpoints of such an interval to be labeled with the same label (we say that the two counter-points are linked). Finally, we impose that the first point in a configuration is linked to the first point in the next configuration.



CONCLUSION (AND FUTURE WORK)

Logic	*Complexity (over finite linear orders)*
AB	**EXPSPACE**-complete
$AB\sim$	non-primitive recursive hard
$AB\sim_1\sim_2$	Undecidable
$\mathrm{PNL}(=\mathrm{A}\bar{\mathrm{A}})$	**NEXPTIME**-complete
$\mathrm{PNL}\sim$	**NEXPTIME**-complete
$\mathrm{PNL}\sim_1\sim_2$	?
$\mathrm{MPNL}\sim$	Decidable (VASS-reachability)

The End

Thank you!!