A logical method for temporal knowledge representation and reasoning

Matei Popovici[1]

[1]POLITEHNICA University of Bucharest
Computer Science and Engineering Department

February 12, 2003

Outline

Outline

Problem statement

- **Objective:** Modelling domains that **change in time**

Problem statement

- **Objective:** Modelling domains that **change in time**
- Example:

Problem statement

- **Objective:** Modelling domains that **change in time**
- Example:

*Bob is **single***

Problem statement

- **Objective:** Modelling domains that **change in time**
- Example:

*Bob is **single***

*Bob **marries** Alice*

Problem statement

- **Objective:** Modelling domains that **change in time**
- Example:

*Bob is **single***

*Bob **marries** Alice*

*Alice **gives birth***

Problem statement

- **Objective:** Modelling domains that **change in time**
- Example:

*Bob is **single***

?

*Bob **marries** Alice*

?

*Alice **gives birth***

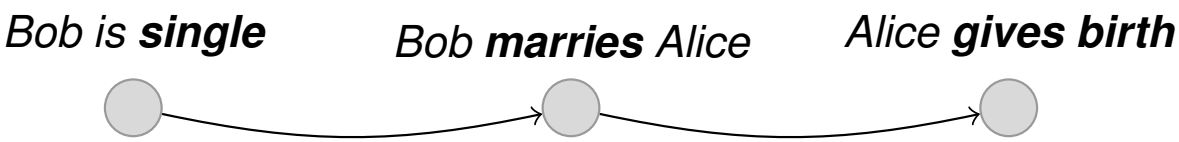
?

Distinctive features of our domains:

- No timestamps for events are known.

Problem statement

- **Objective:** Modelling domains that **change in time**
- Example:

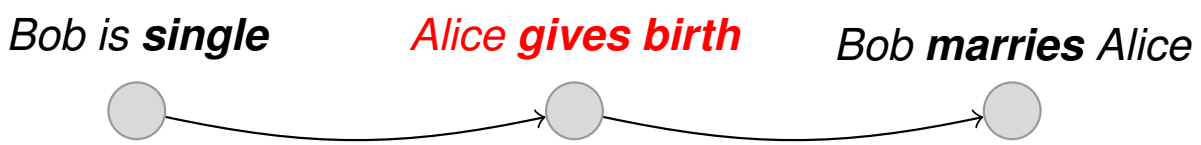


Distinctive features of our domains:

- No timestamps for events are known.
- The occurrence of an event can be related to previous events (partial ordering)

Problem statement

- **Objective:** Modelling domains that **change in time**
- Example:

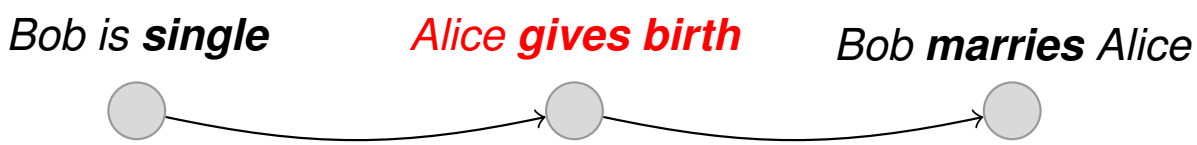


Distinctive features of our domains:

- No timestamps for events are known.
- The occurrence of an event can be related to previous events (partial ordering)

Problem statement

- **Objective:** Modelling domains that **change in time**
- Example:



Distinctive features of our domains:

- No timestamps for events are known.
- The occurrence of an event can be related to previous events (partial ordering)
- The domain's evolution is **non-Markovian**.

Problem statement

How do we **represent**

Problem statement

How do we **represent** and **reason** about changing domains

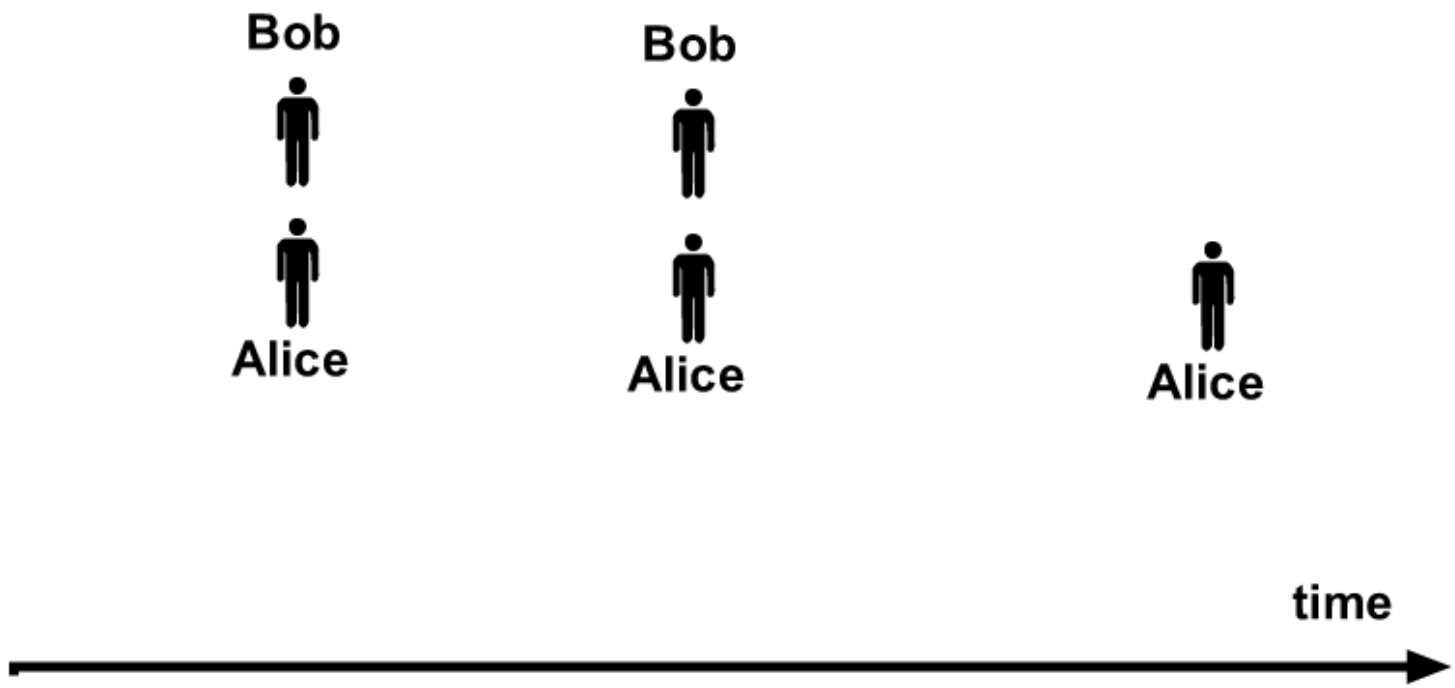
Problem statement

How do we **represent** and **reason** about changing domains in an **efficient** manner ?

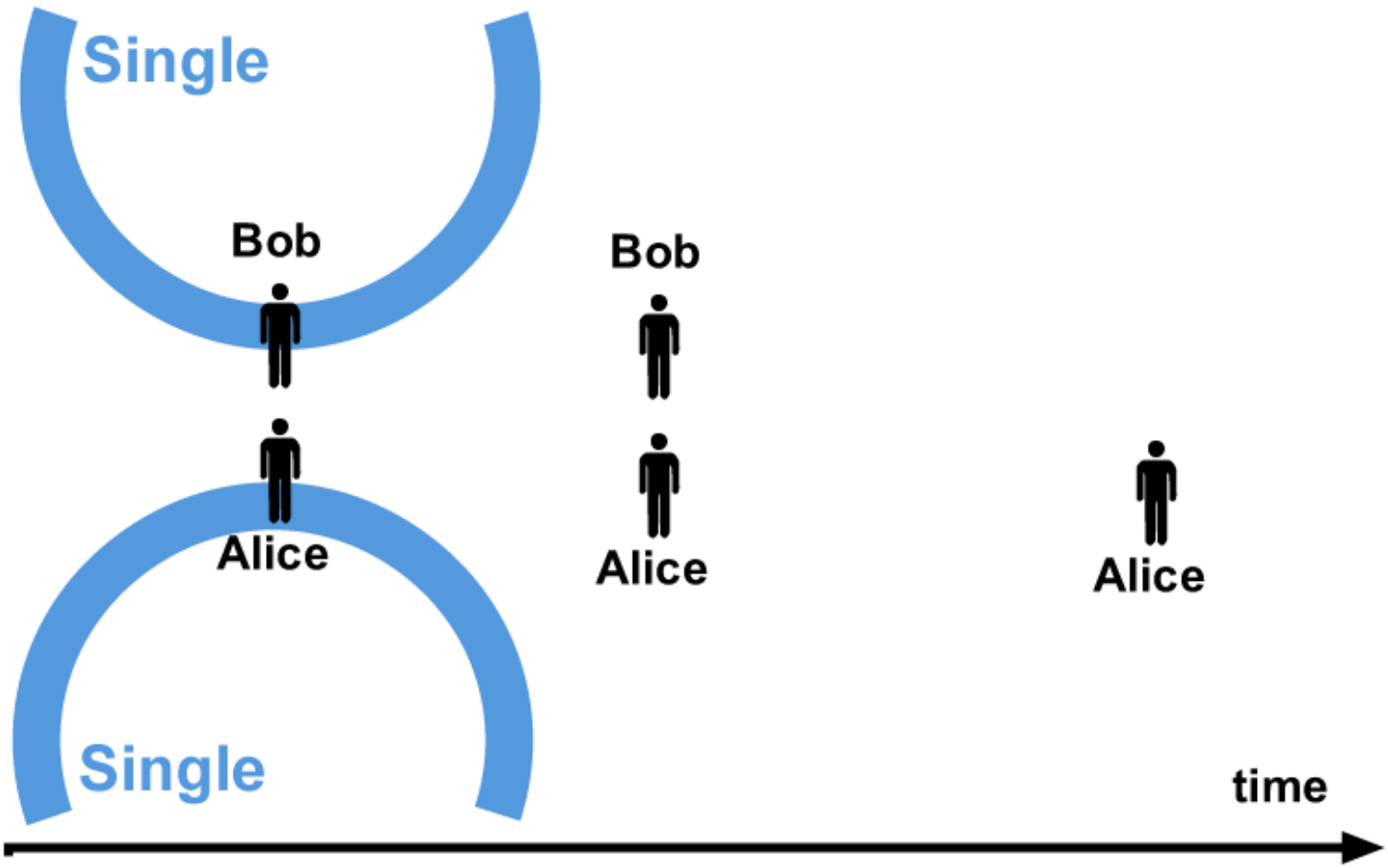
Outline

Intuition

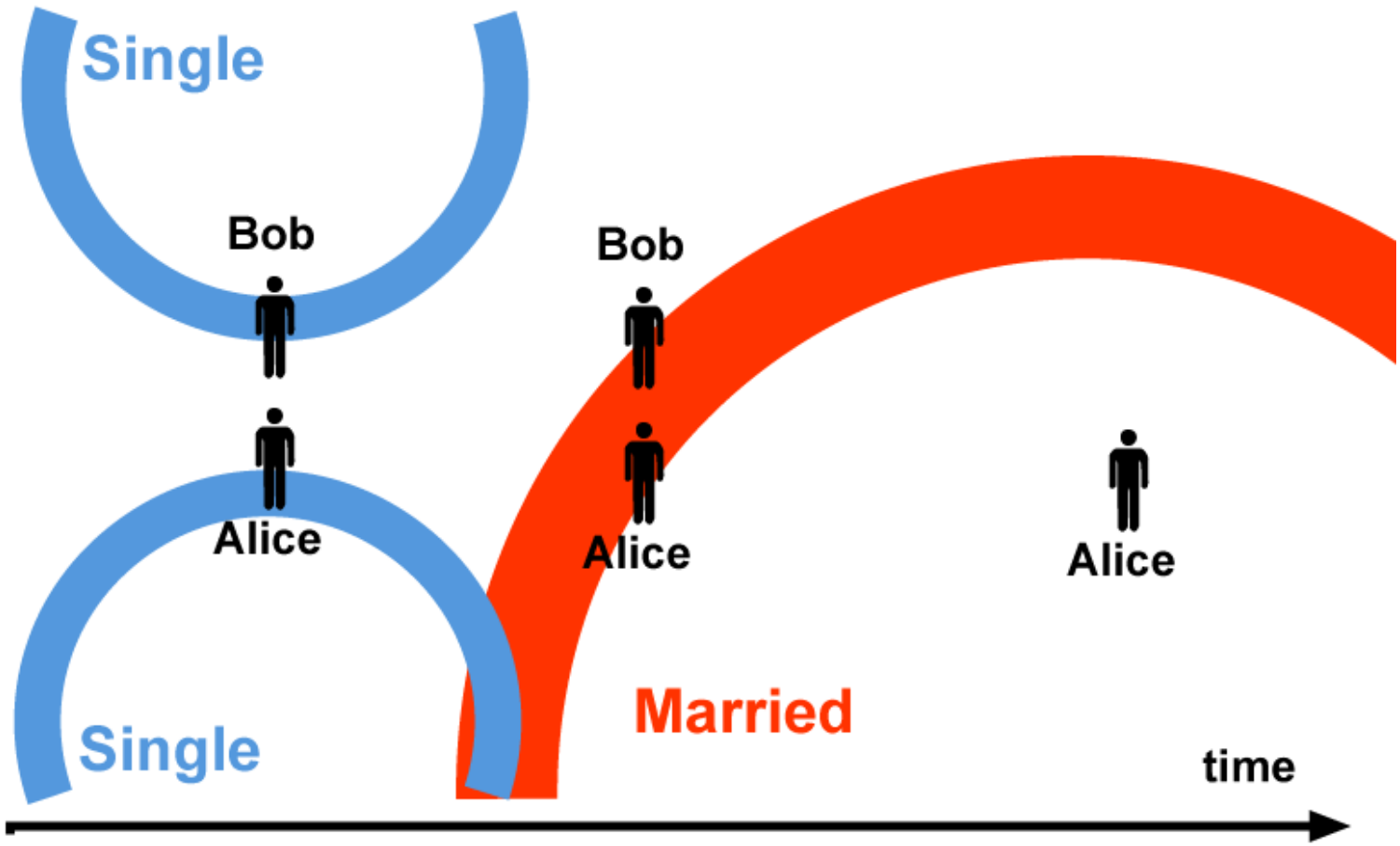
Intuition



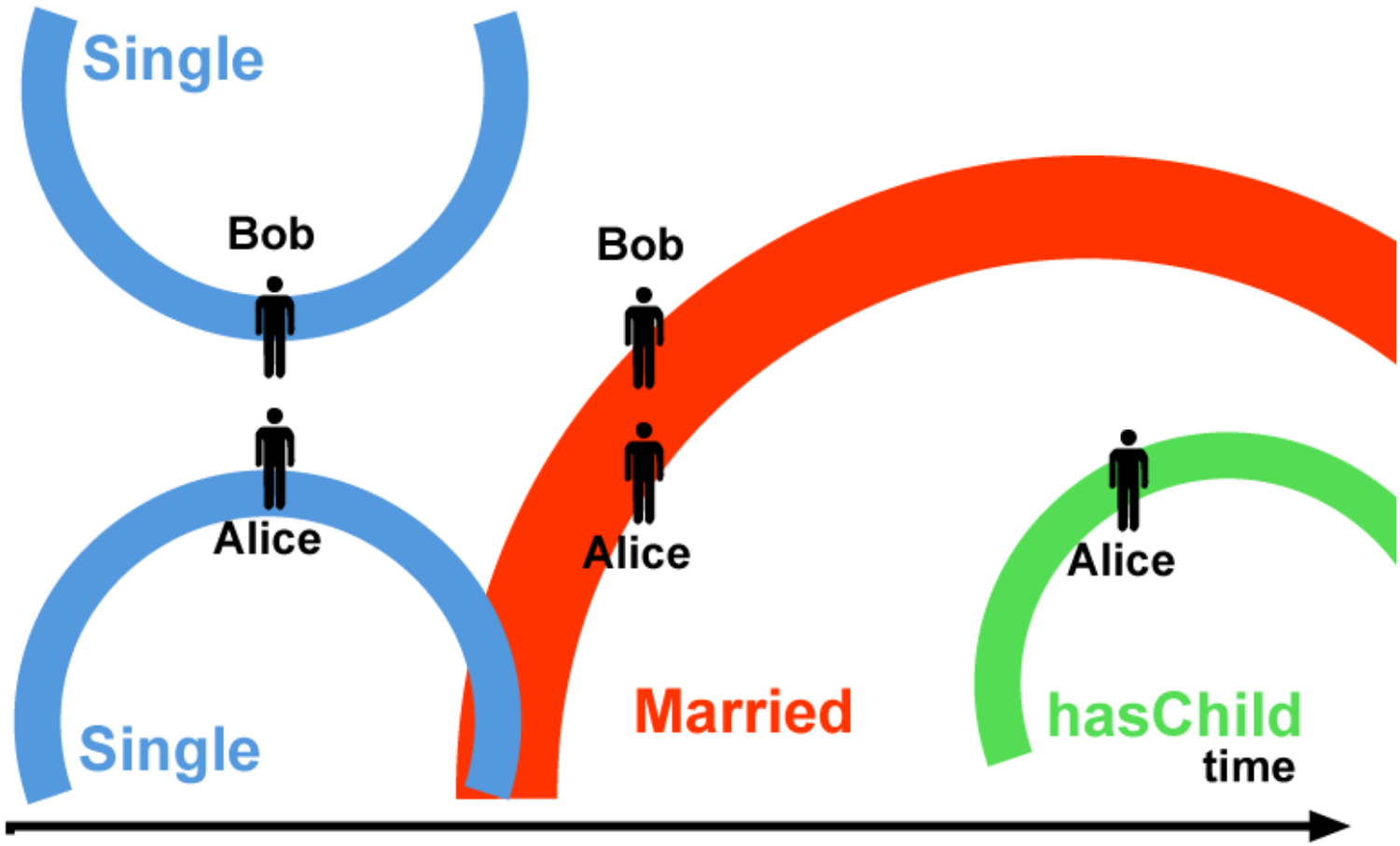
Intuition



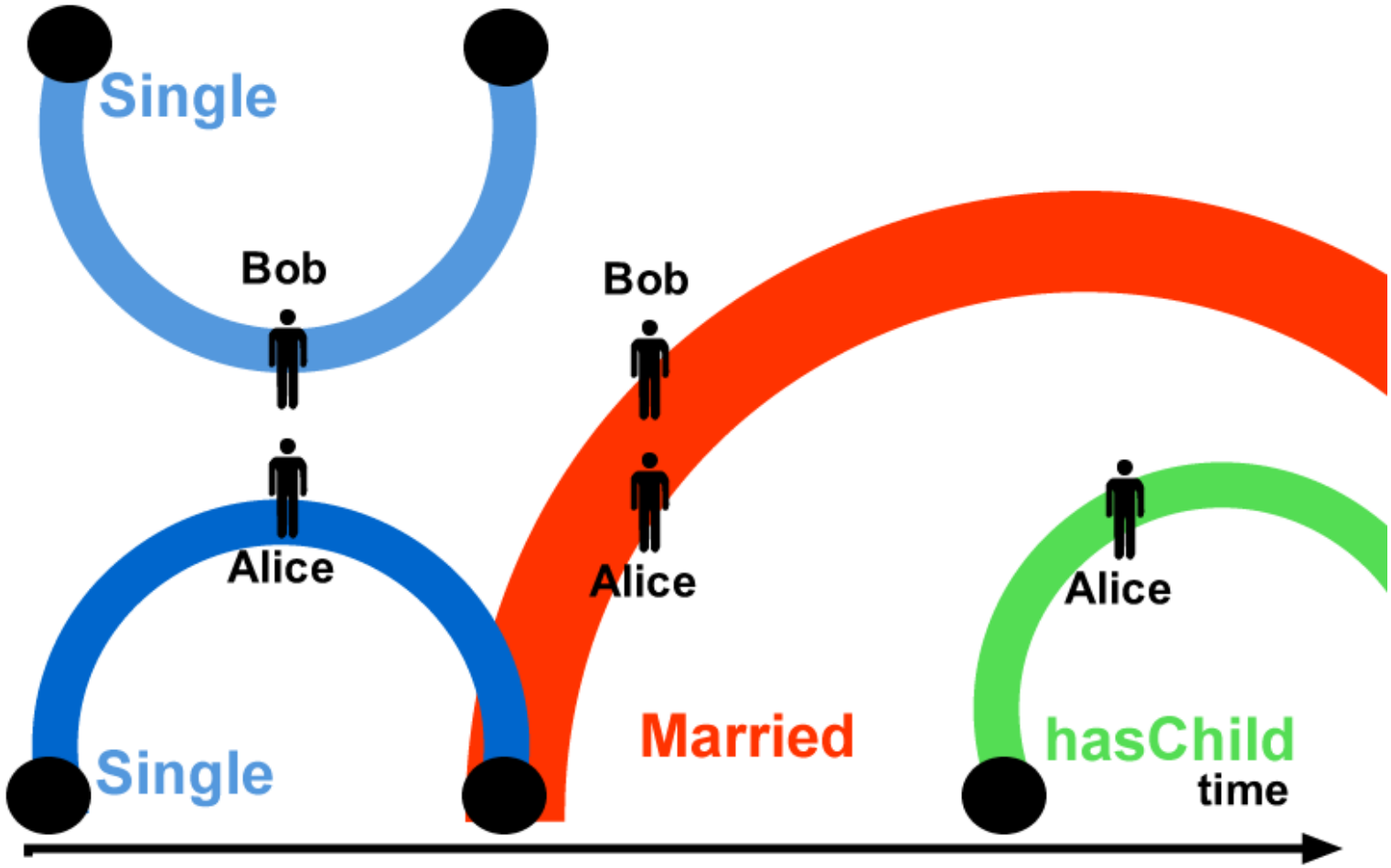
Intuition



Intuition


Single
Bob
Alice
Single
Bob
Alice
Married
Alice
hasChild
time

Intuition


Single
Bob
Alice
Single
Bob
Alice
Married
Alice
hasChild
time

Formal setting

Formal setting

- elements of a set $I = \{Bob, Alice\}$, called **universe**

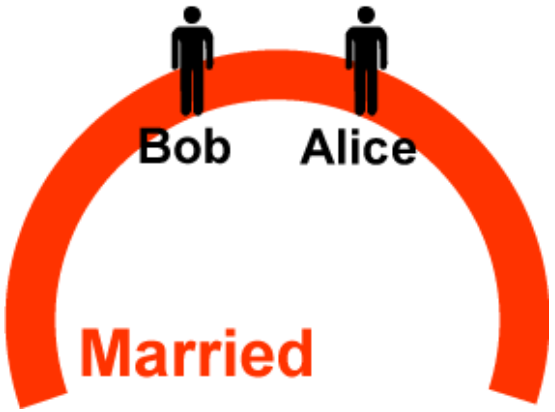
Formal setting

- elements of a set $I = \{Bob, Alice\}$, called **universe**
- **relation symbols** from a vocabulary
  $\sigma = \{Single, Married, hasChild\}$

Single **Married** **hasChild**

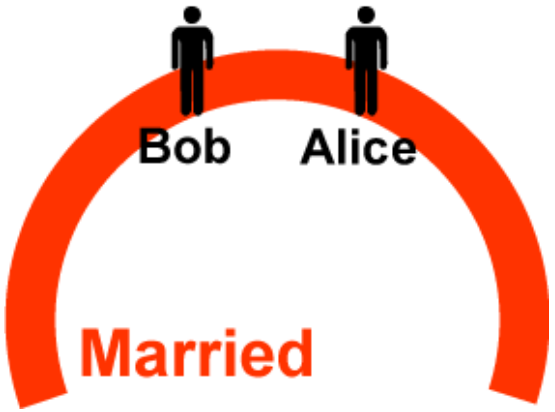
Formal setting

- elements of a set $I = \{Bob, Alice\}$, called **universe**
- **relation symbols** from a vocabulary $\sigma = \{Single, Married, hasChild\}$
- **relation instances** $(Bob, Alice) \in Married^I$ where $Married^I \subseteq I \times I$.



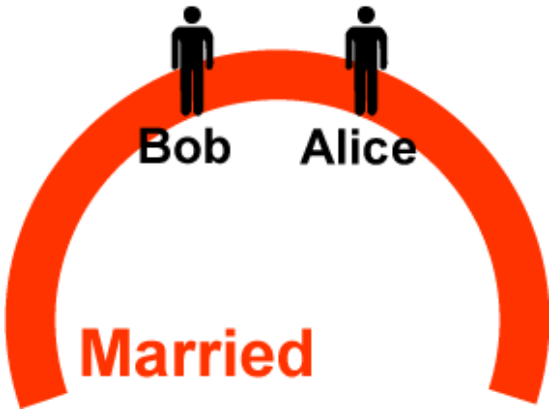
Formal setting

- elements of a set $I = \{Bob, Alice\}$, called **universe**
- **relation symbols** from a vocabulary $\sigma = \{Single, Married, hasChild\}$
- **relation instances** $(Bob, Alice) \in Married^I$ where $Married^I \subseteq I \times I$. We also write $Married(Bob, Alice)$



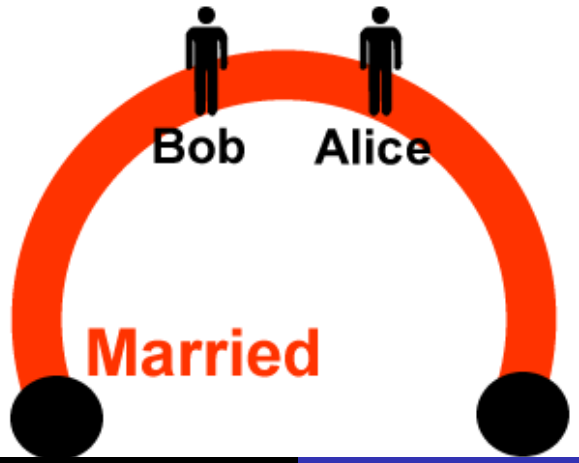
Formal setting

- elements of a set $I = \{Bob, Alice\}$, called **universe**
- **relation symbols** from a vocabulary $\sigma = \{Single, Married, hasChild\}$
- **relation instances** $(Bob, Alice) \in Married^I$ where $Married^I \subseteq I \times I$. We also write $Married(Bob, Alice)$
- $(I, Single^I, Married^I, hasChild^I)$ is a **labelling domain**



Formal setting

- elements of a set $I = \{Bob, Alice\}$, called **universe**
- **relation symbols** from a vocabulary $\sigma = \{Single, Married, hasChild\}$
- **relation instances** $(Bob, Alice) \in Married^I$ where $Married^I \subseteq I \times I$. We also write $Married(Bob, Alice)$
- $(I, Single^I, Married^I, hasChild^I)$ is a **labelling domain**
- **(labelled) quality edges** that span **(labelled) action nodes**

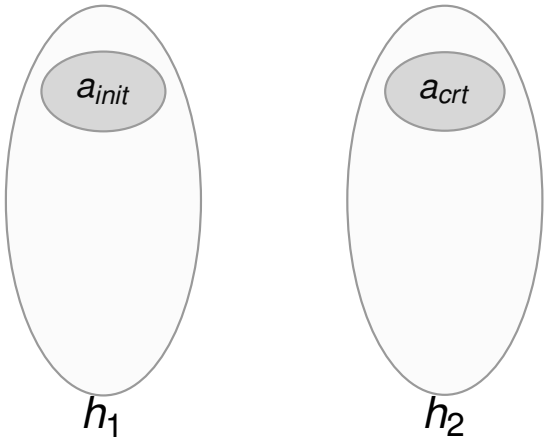


Temporal graphs

Temporal graphs

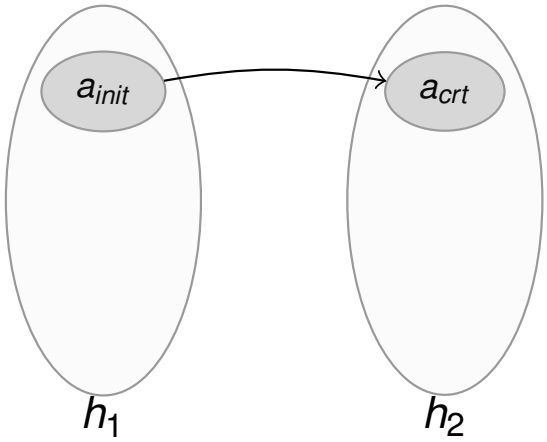
- A set A of **action nodes**

Temporal graphs



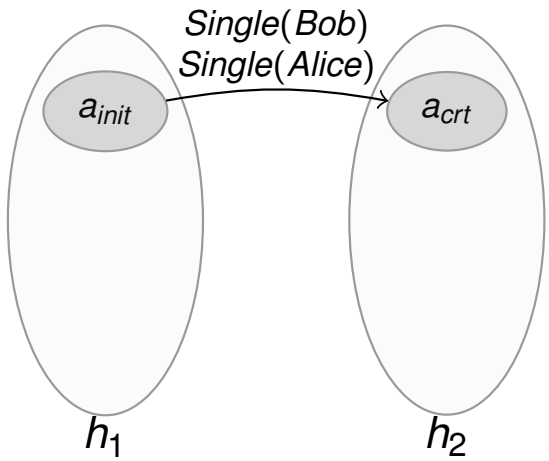
- A set A of **action nodes**
- A set H of **hypernodes** and $\mathcal{T} : A \rightarrow H$

Temporal graphs



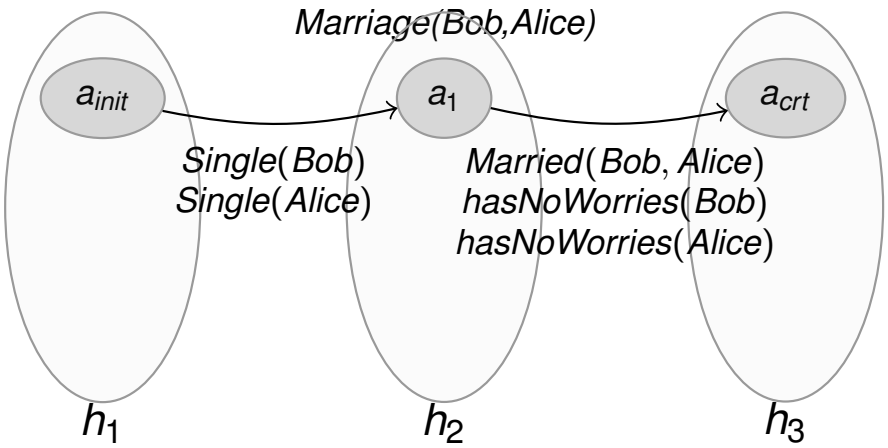
- A set A of **action nodes**
- A set H of **hypernodes** and $\mathcal{T} : A \to H$
- A set $E \subseteq A^2$ of **quality edges**

Temporal graphs



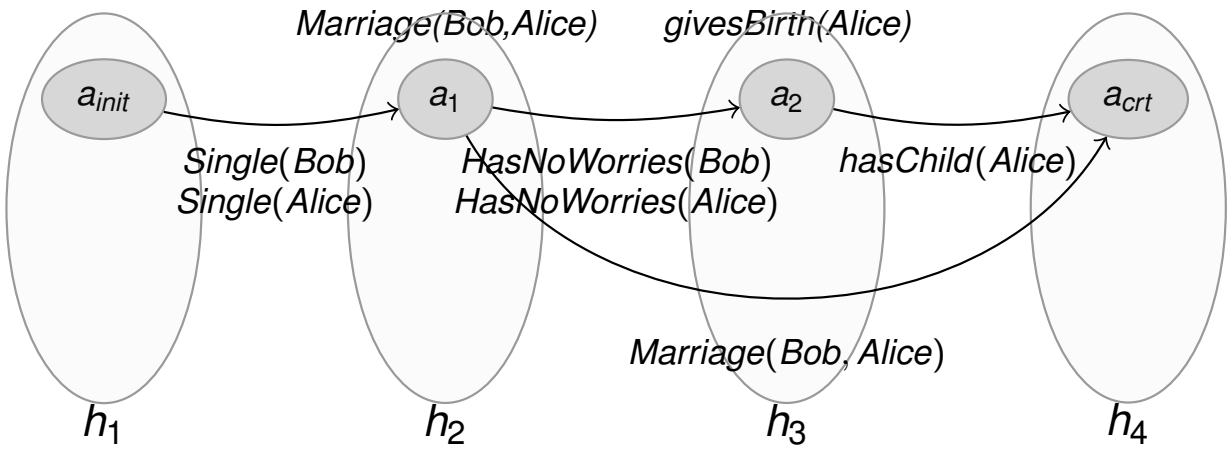
- A set A of **action nodes**
- A set H of **hypernodes** and $\mathcal{T} : A \to H$
- A set $E \subseteq A^2$ of **quality edges**, **labelled** with **relation instances** from a **labeling domain**

Temporal graphs



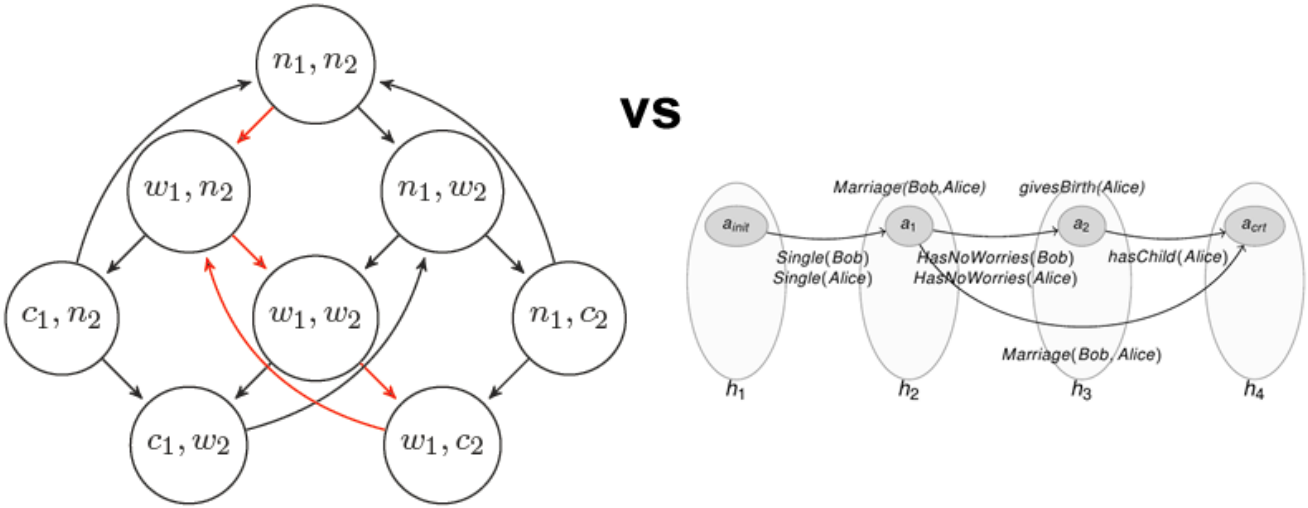
- A set A of **action nodes**
- A set H of **hypernodes** and $\mathcal{T} : A \to H$
- A set $E \subseteq A^2$ of **quality edges**, **labelled** with **relation instances** from a **labeling domain**

Temporal graphs



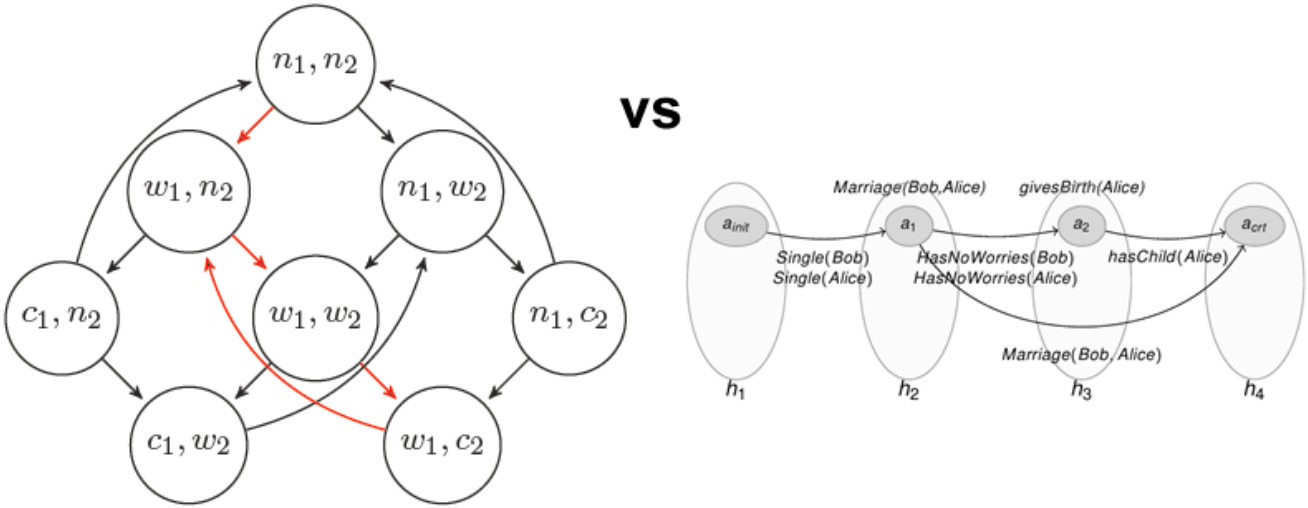
- A set A of **action nodes**
- A set H of **hypernodes** and $\mathcal{T} : A \rightarrow H$
- A set $E \subseteq A^2$ of **quality edges**, **labelled** with **relation instances** from a **labeling domain**

Temporal graphs and Kripke Structures



Temporal graphs **versus** Kripke Structures:

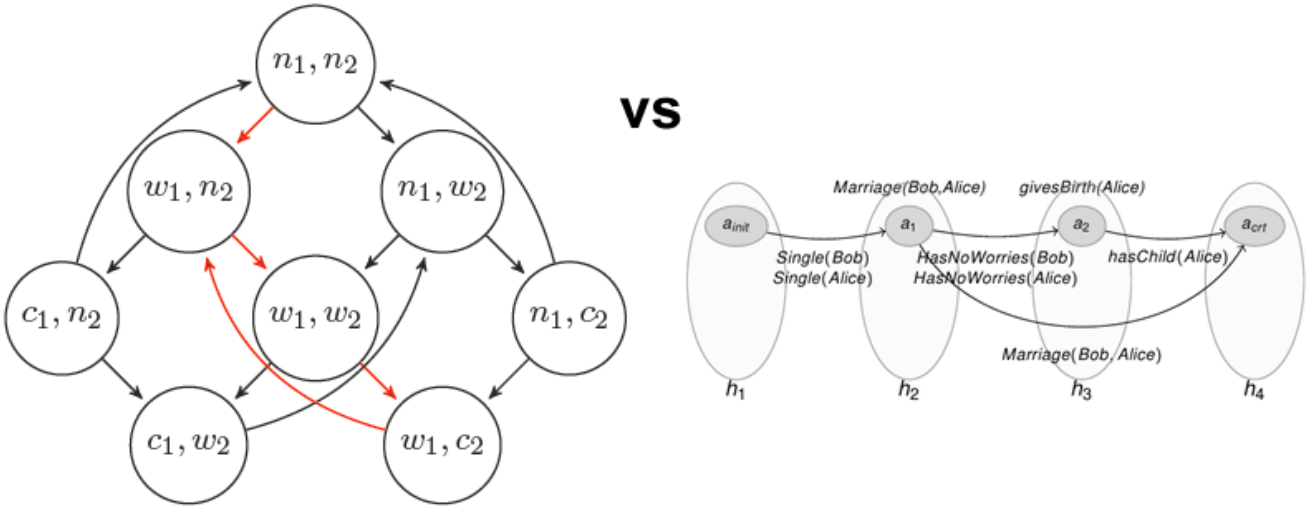
Temporal graphs and Kripke Structures



Temporal graphs **versus** Kripke Structures:

- Kripke Structures (KS) are **computational structures**

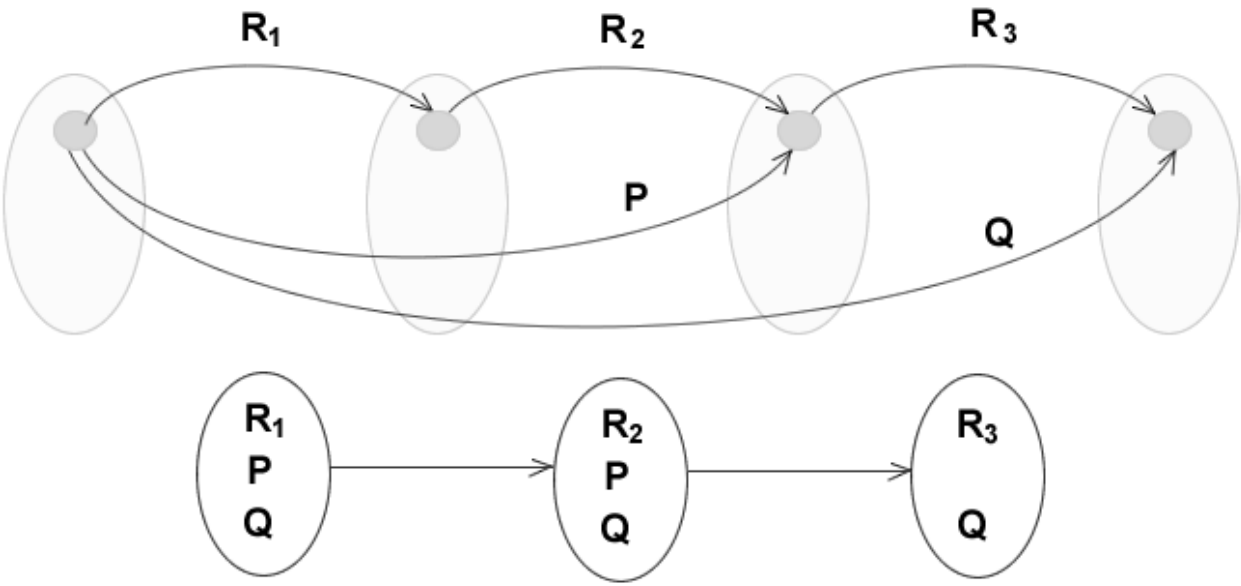
Temporal graphs and Kripke Structures



Temporal graphs **versus** Kripke Structures:

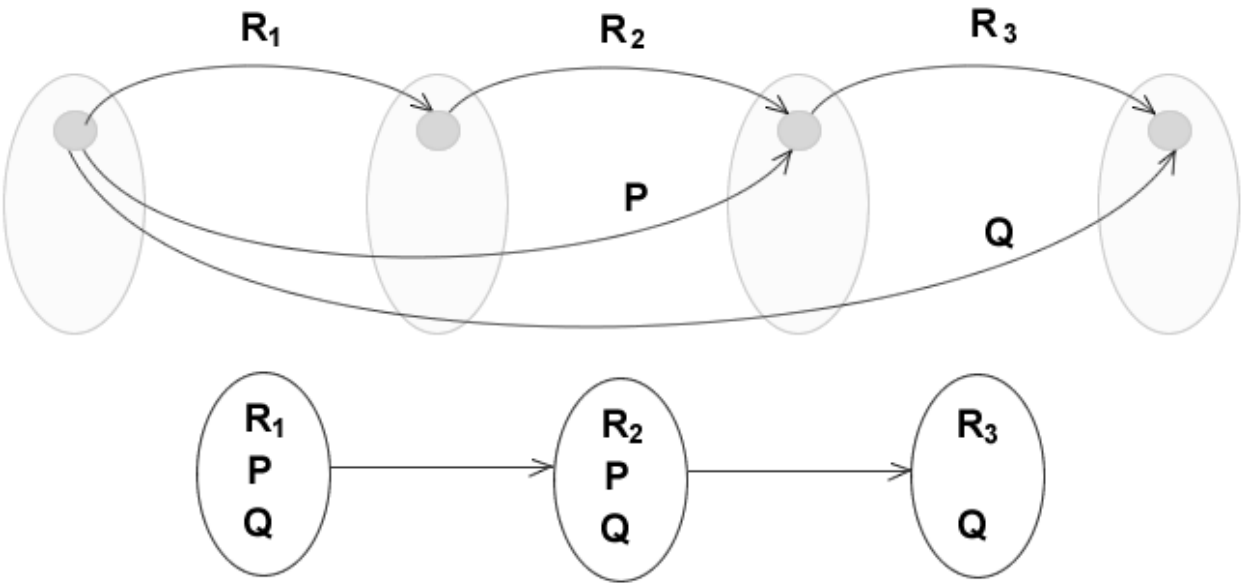
- Kripke Structures (KS) are **computational structures**
- temporal graphs are **behavioural structures** (similar to **paths** from a KS

Temporal graphs and Kripke Structures



Temporal graphs **versus** Kripke Structures:

Temporal graphs and Kripke Structures



Temporal graphs **versus** Kripke Structures:

- **explosion** of the number of labellings (scalability problem)

Outline

Why is theory important for our endeavour:

Why is theory important for our endeavour:

- It shows whether or not we can achieve tractable implementations

Why is theory important for our endeavour:

- It shows whether or not we can achieve tractable implementations
- It provides answers to questions such as:

Why is theory important for our endeavour:

- It shows whether or not we can achieve tractable implementations
- It provides answers to questions such as:
  - *Why not **First-Order Logics** ?*

Why is theory important for our endeavour:

- It shows whether or not we can achieve tractable implementations
- It provides answers to questions such as:
  - *Why not **First-Order Logics** ?*
  - *Why not **Description Logics** ?*

Outline

Temporal graphs and FOL undefinability

There exists no formula from First Order Logic (FOL) which can **express**:

Temporal graphs and FOL undefinability

There exists no formula from First Order Logic (FOL) which can **express**: **connectivity**

Temporal graphs and FOL undefinability

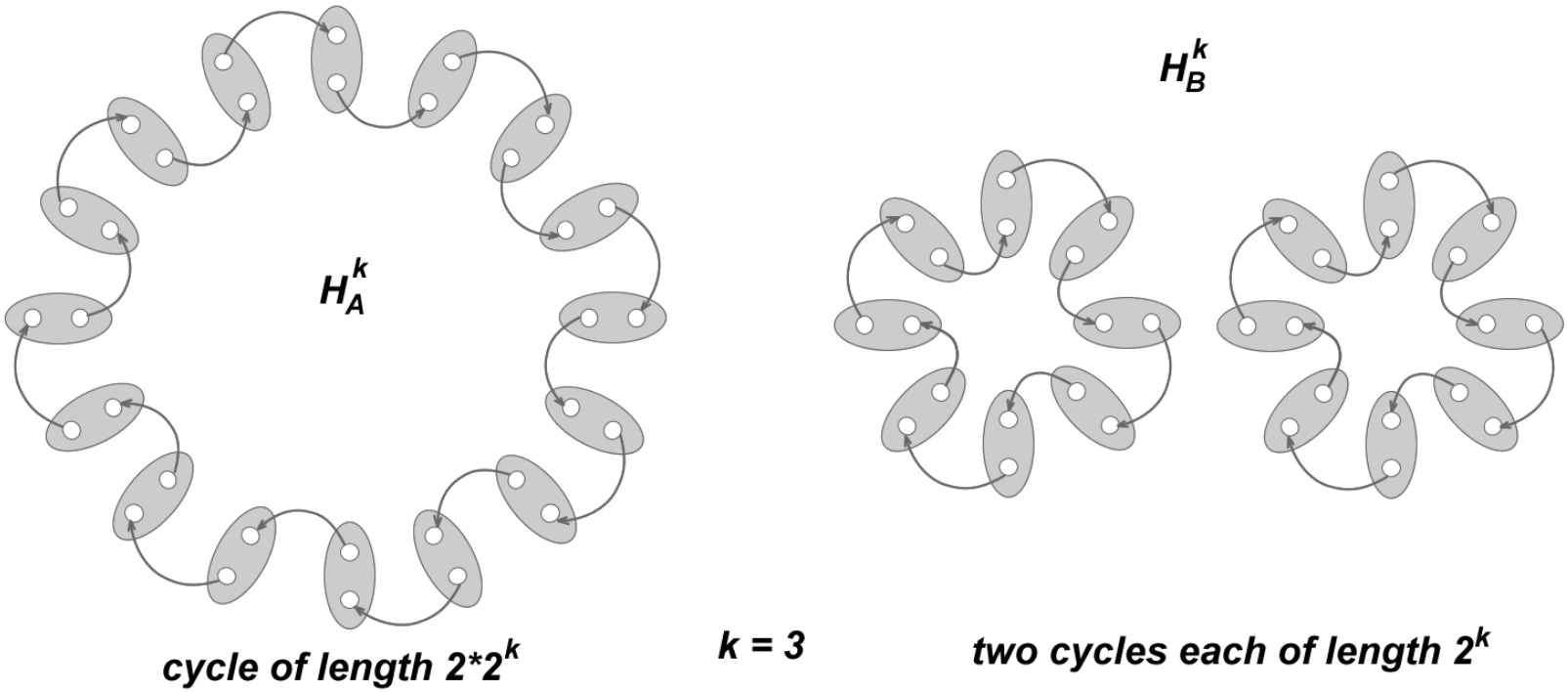
There exists no formula from First Order Logic (FOL) which can **express**: **connectivity** or the **existence of a path** between two components of a temporal graph

The Ehrenfeucht-Fraïssé method (connectivity)

fix a natural number k

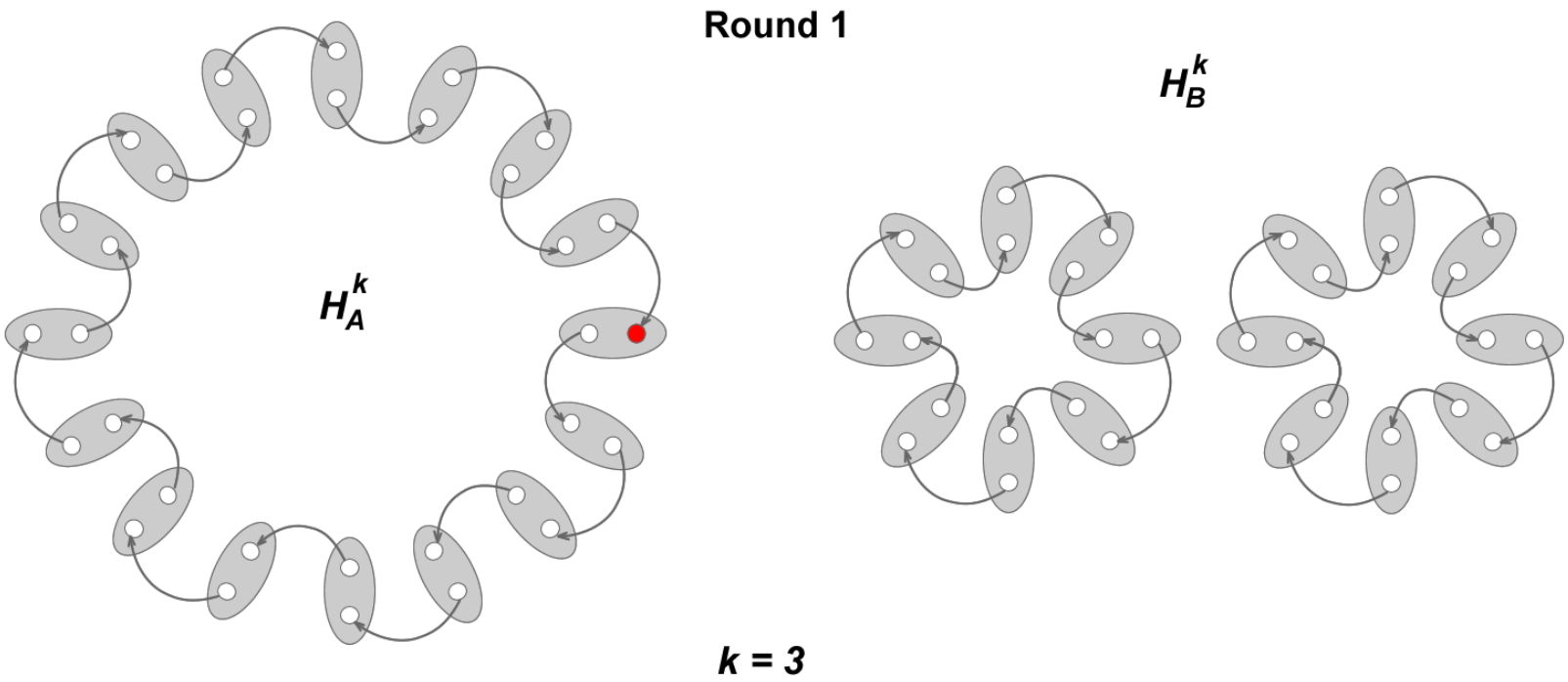
The Ehrenfeucht-Fraïssé method (connectivity)

build two temporal graphs $\mathcal{H}_A^k$ and $\mathcal{H}_B^k$



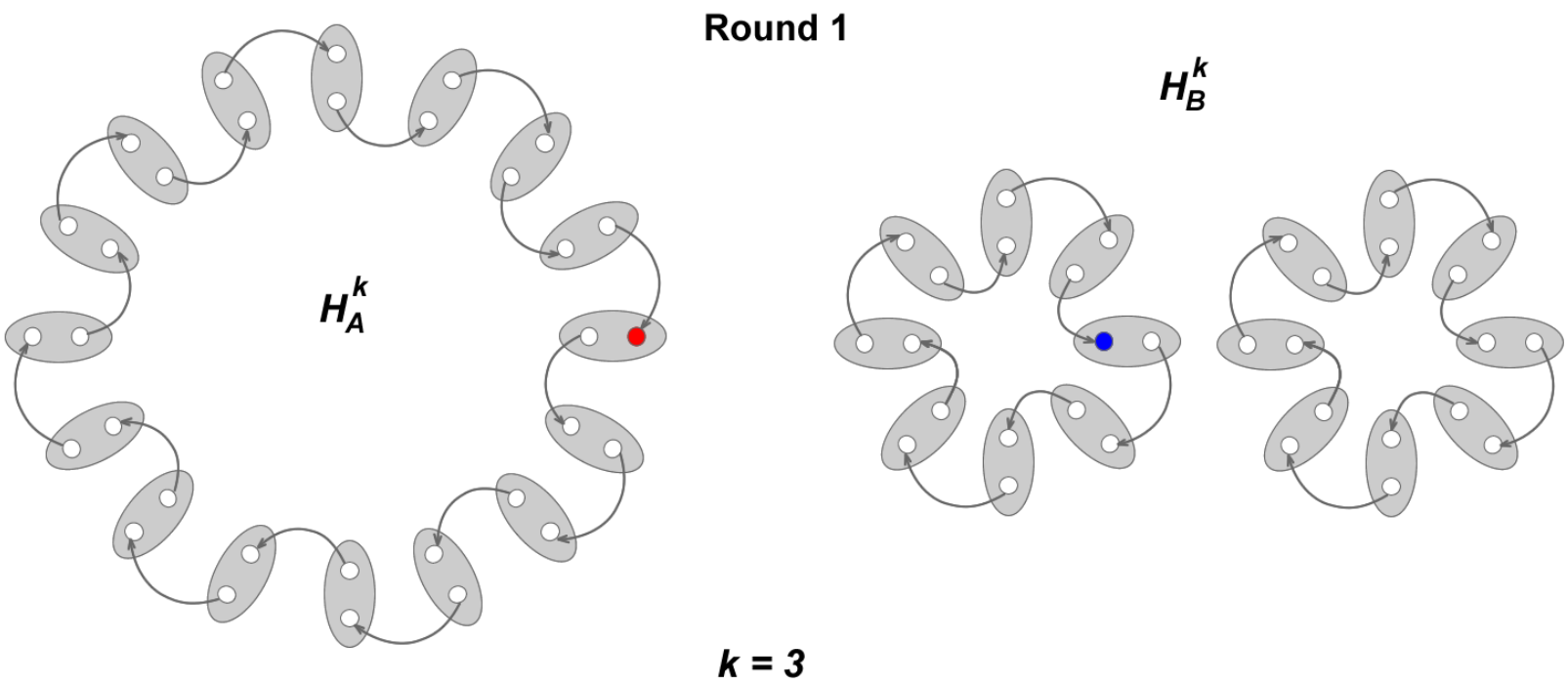
The Ehrenfeucht-Fraïssé method (connectivity)

play a k-round, two-player **game**: red vs blue



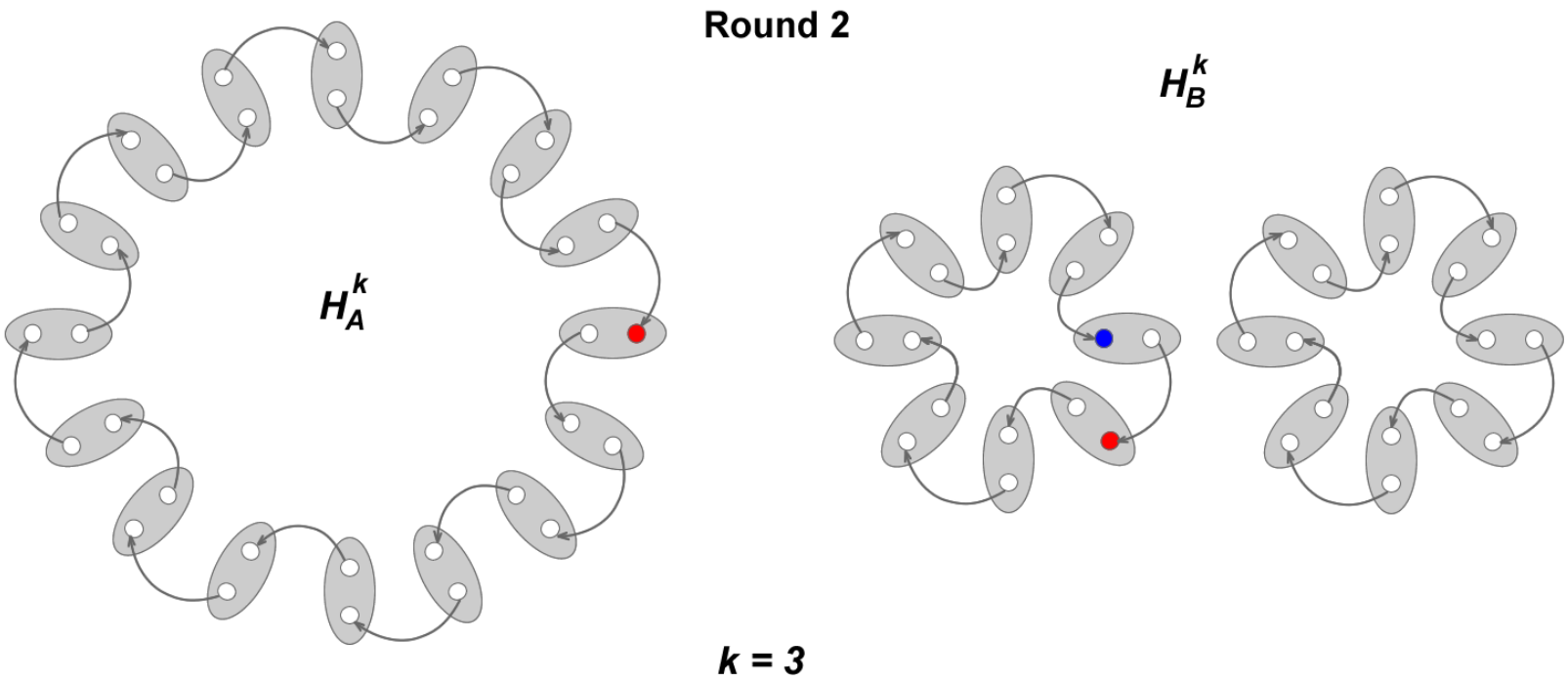
The Ehrenfeucht-Fraïssé method (connectivity)

play a k-round, two-player **game**: red vs blue



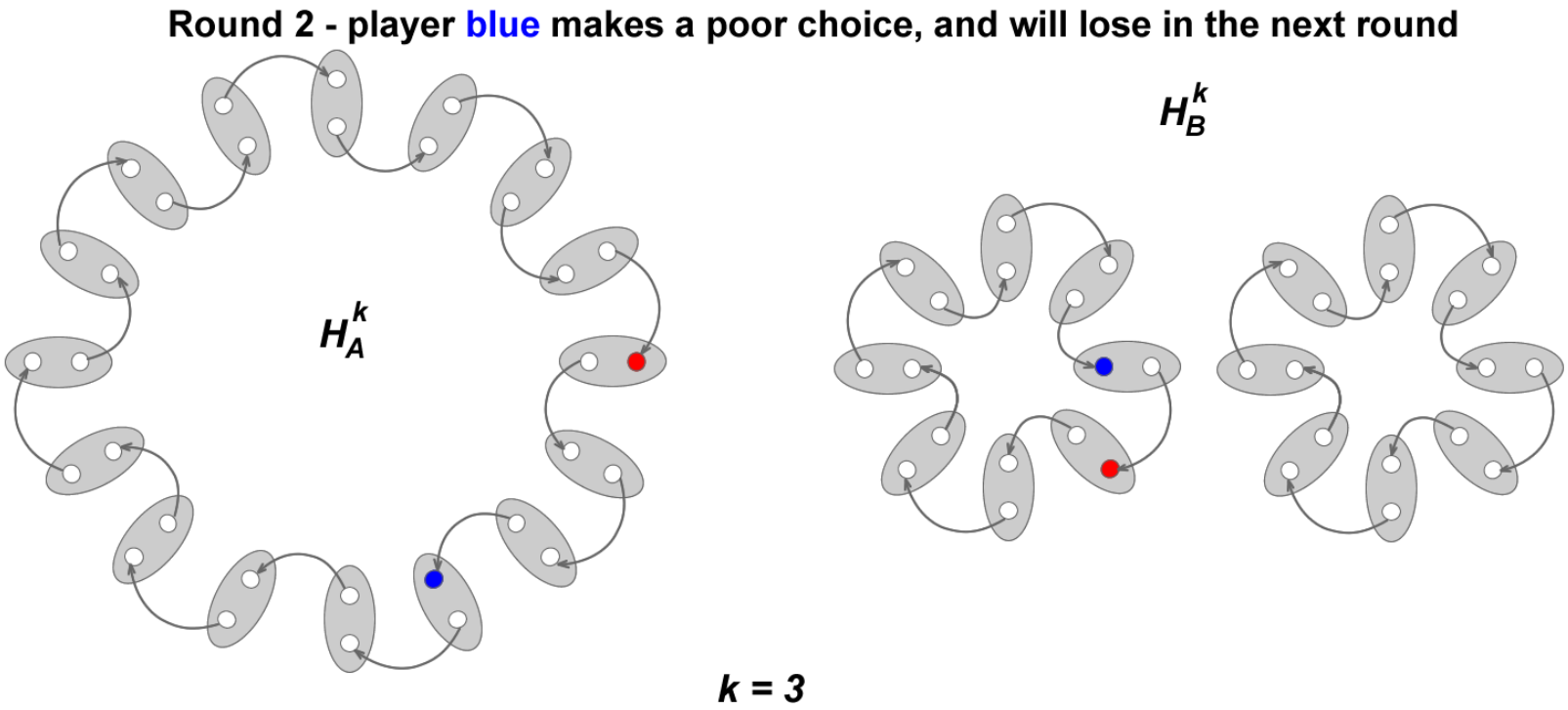
The Ehrenfeucht-Fraïssé method (connectivity)

play a k-round, two-player **game**: red vs blue



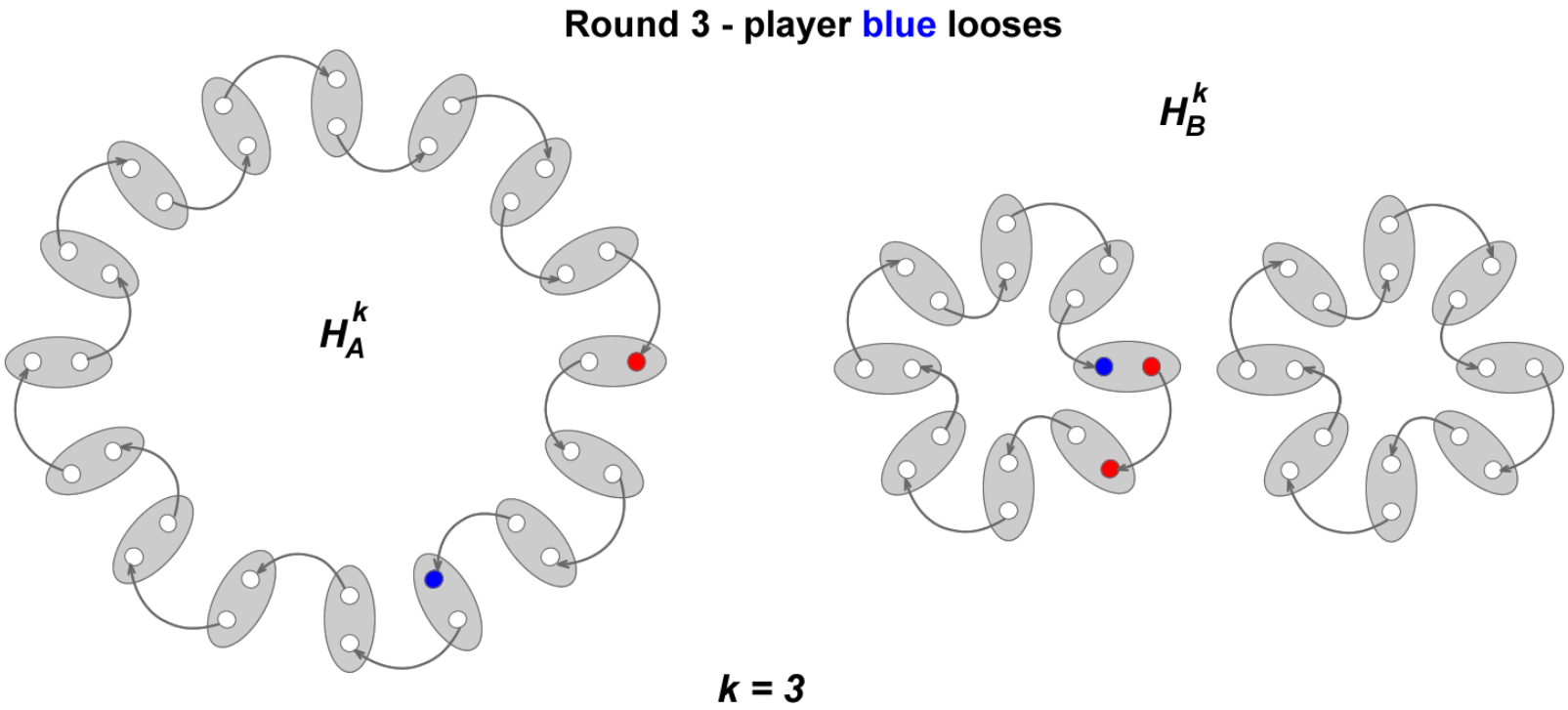
The Ehrenfeucht-Fraïssé method (connectivity)

play a k-round, two-player **game**: red vs blue



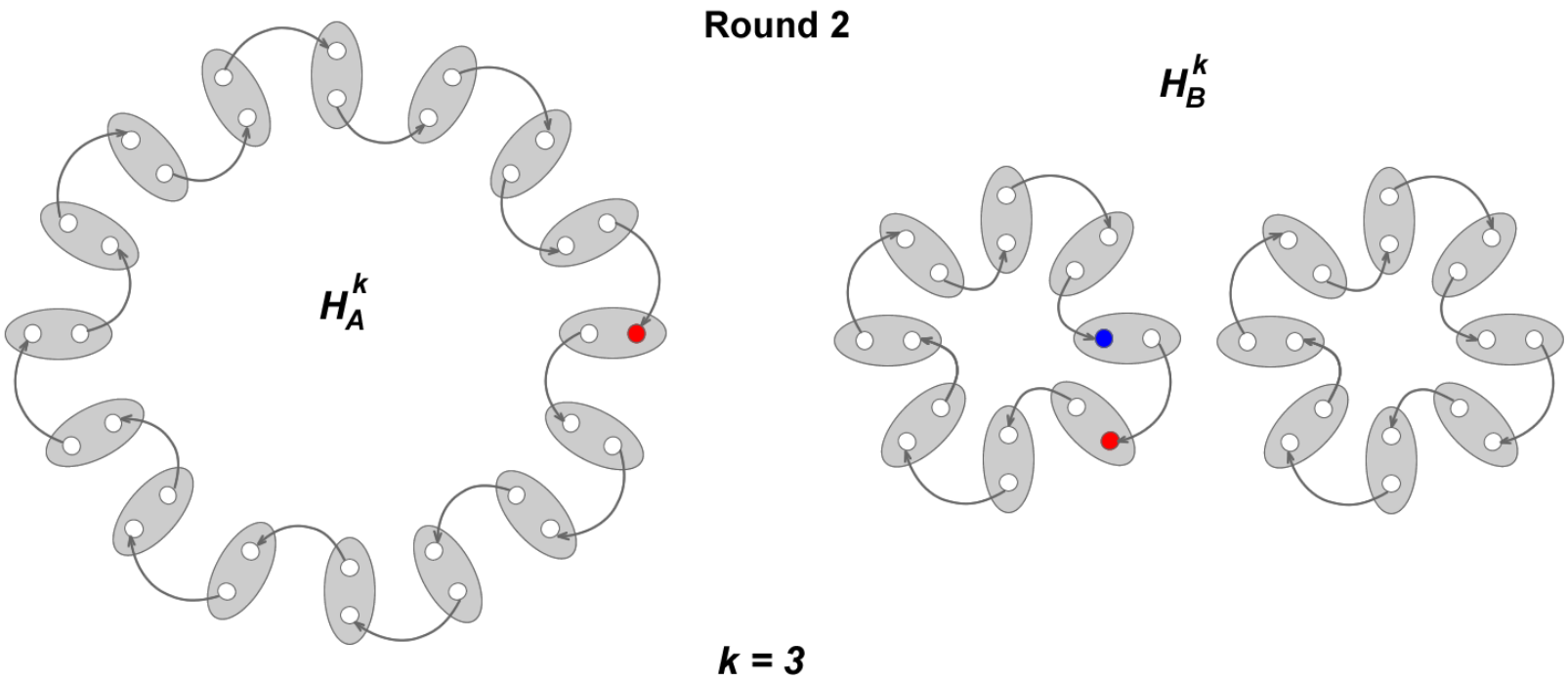
The Ehrenfeucht-Fraïssé method (connectivity)

play a k-round, two-player **game**: red vs blue



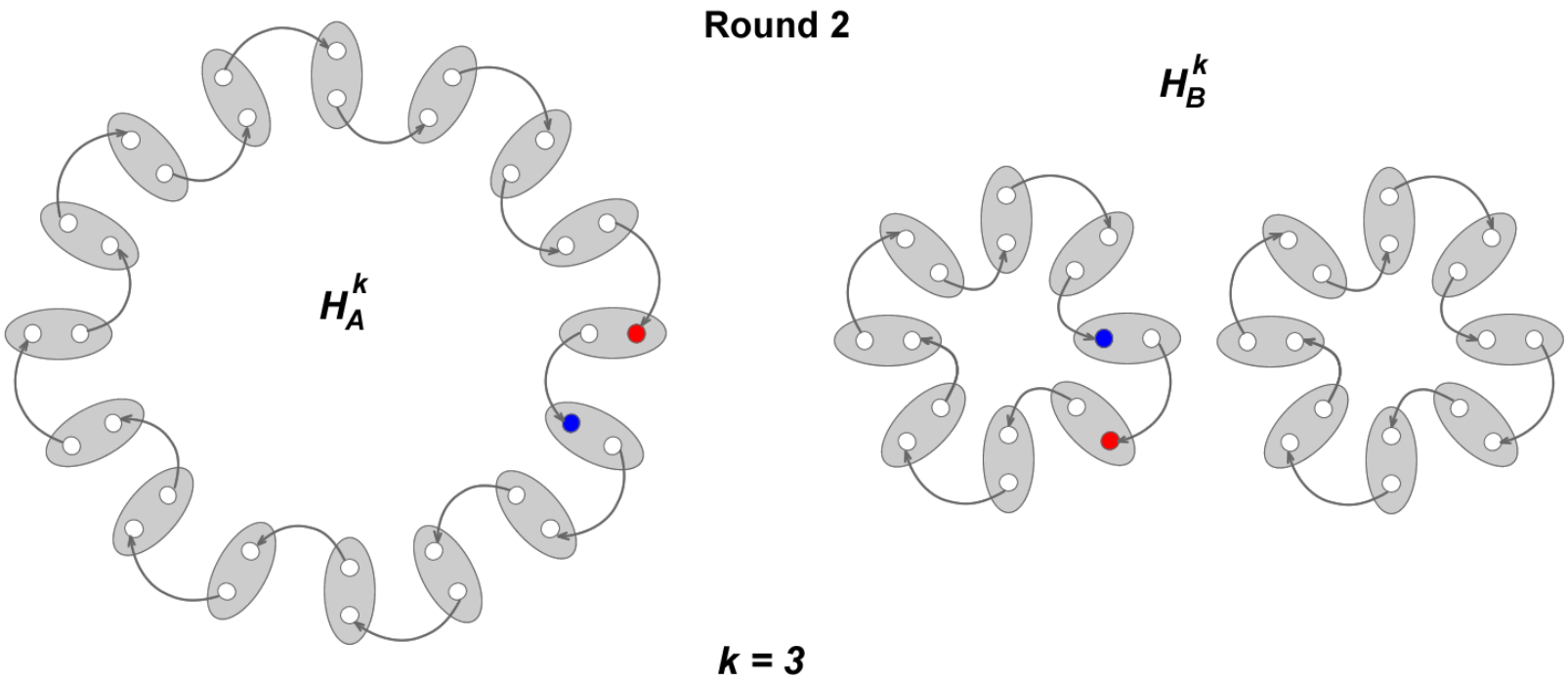
The Ehrenfeucht-Fraïssé method (connectivity)

play a k-round, two-player **game**: red vs blue



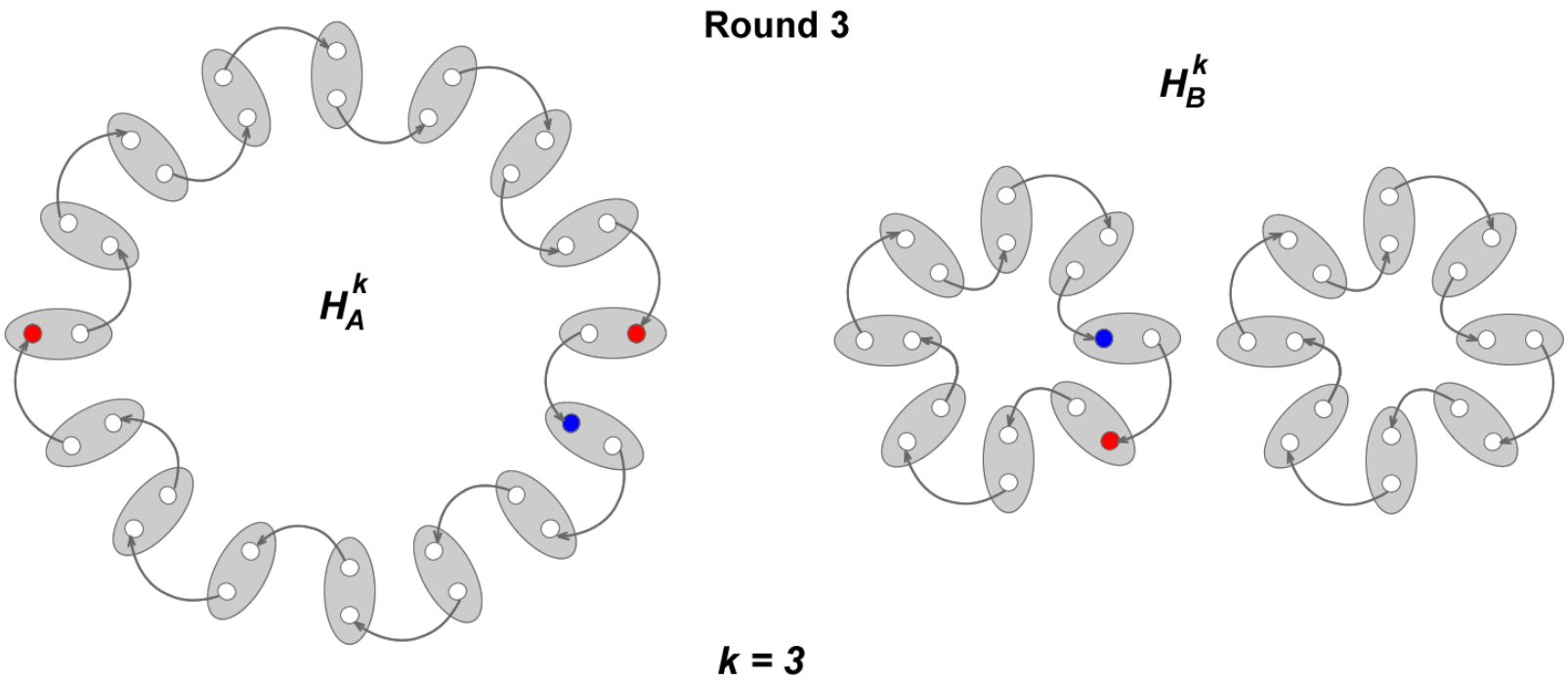
The Ehrenfeucht-Fraïssé method (connectivity)

play a k-round, two-player **game**: red vs blue



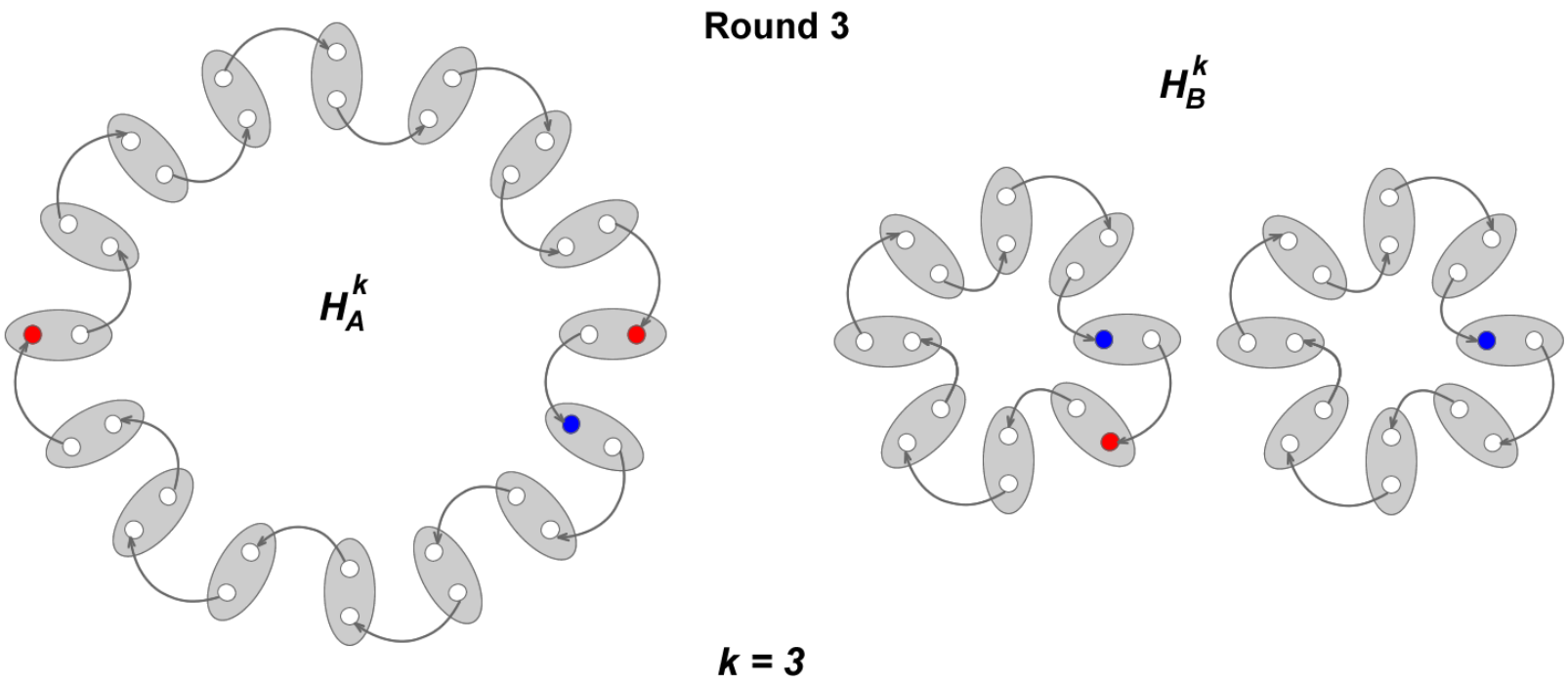
The Ehrenfeucht-Fraïssé method (connectivity)

play a k-round, two-player **game**: red vs blue



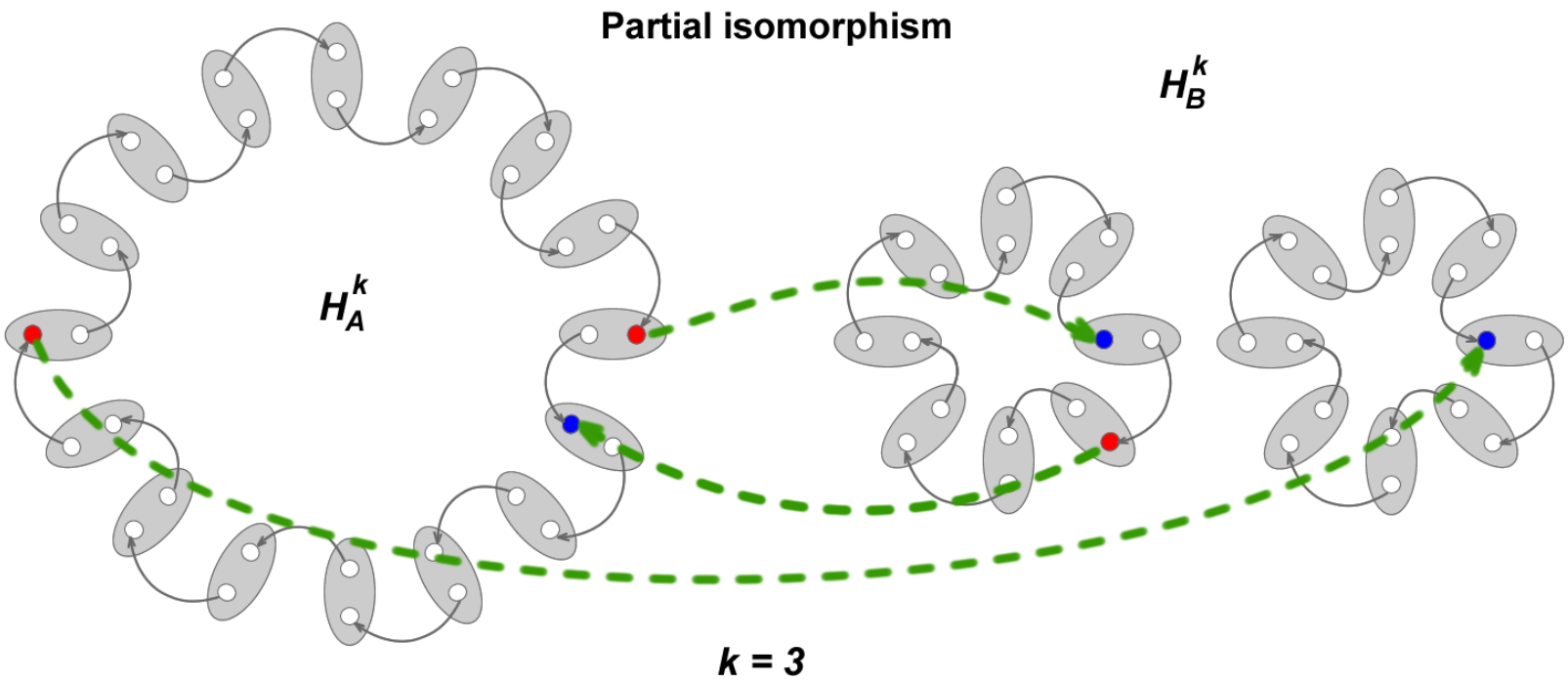
The Ehrenfeucht-Fraïssé method (connectivity)

play a k-round, two-player **game**: red vs blue



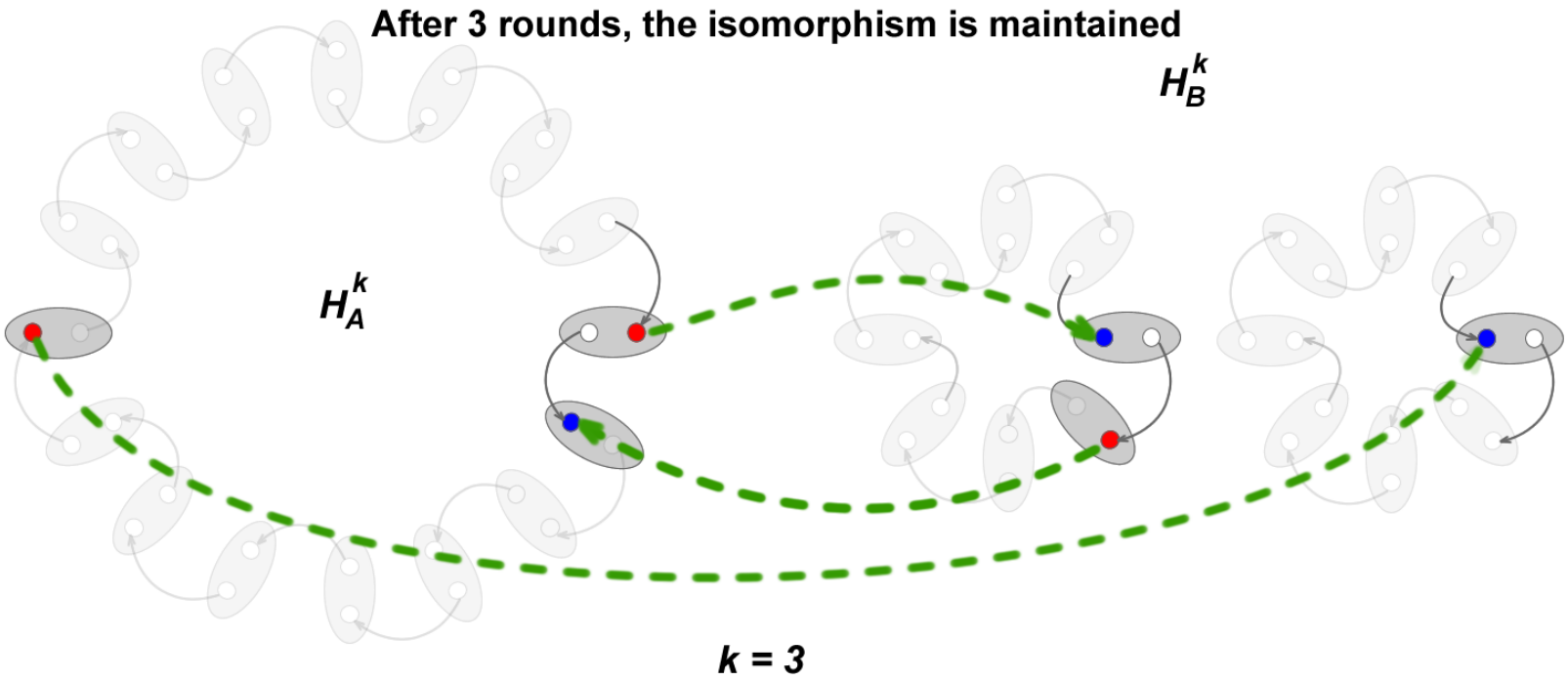
The Ehrenfeucht-Fraïssé method (connectivity)

play a k-round, two-player **game**: red vs blue



The Ehrenfeucht-Fraïssé method (connectivity)

play a k-round, two-player **game**: red vs blue



The Ehrenfeucht-Fraïssé method (connectivity)

Theorem (Ehrenfeucht-Fraïssé):

The Ehrenfeucht-Fraïssé method (connectivity)

Theorem (Ehrenfeucht-Fraïssé): If blue has a winning strategy

The Ehrenfeucht-Fraïssé method (connectivity)

Theorem (Ehrenfeucht-Fraïssé): If blue has a winning strategy on a *k*-**round** game

The Ehrenfeucht-Fraïssé method (connectivity)

Theorem (Ehrenfeucht-Fraïssé): If blue has a winning strategy on a k-**round** game, played over $\mathcal{H}_A^k$ and $\mathcal{H}_B^k$

The Ehrenfeucht-Fraïssé method (connectivity)

Theorem (Ehrenfeucht-Fraïssé): If blue has a winning strategy on a k-**round** game, played over $\mathcal{H}_A^k$ and $\mathcal{H}_B^k$ then there exists **no formula** from $FO[k]$

The Ehrenfeucht-Fraïssé method (connectivity)

Theorem (Ehrenfeucht-Fraïssé): If blue has a winning strategy on a k-**round** game, played over $\mathcal{H}_A^k$ and $\mathcal{H}_B^k$ then there exists **no formula** from $FO[k]$ which is true in $\mathcal{H}_A^k$ and false $\mathcal{H}_B^k$

The Ehrenfeucht-Fraïssé method (connectivity)

Importance of our result:

The Ehrenfeucht-Fraïssé method (connectivity)

Importance of our result:

- SQL **cannot** be used for querying about temporal graph properties

The Ehrenfeucht-Fraïssé method (connectivity)

Importance of our result:

- SQL **cannot** be used for querying about temporal graph properties
- Description Logics (without cyclic TBoxes) **cannot** be used to reason about temporal graphs

The Ehrenfeucht-Fraïssé method (connectivity)

Importance of our result:

- SQL **cannot** be used for querying about temporal graph properties
- Description Logics (without cyclic TBoxes) **cannot** be used to reason about temporal graphs
- A **language** which is suitable for reasoning about temporal graphs is *outside* FOL

Outline

$L_{\mathcal{H}}$ Syntax

Let $\mathbb{V}$ars be a set of variables, $\mathcal{A}$ be a $\sigma_E \cup \sigma_A$-structure and $\mathcal{H}_{\mathcal{A}}$ be a $\mathcal{A}$-labelled **t**-graph. Also, let $X \in \{E, A\}$, and $\dagger_E \in \mathfrak{C}_Q$ and $\dagger_A \in \mathfrak{C}_A$. The syntax of a X-formula is recursively defined with respect to $\mathcal{A}$, as follows:

- if $R \in \sigma_X$ with arity$(R) = n$ and $\bar{t} \in (\mathbb{V}\text{ars} \cup I)^n$, then $R(\bar{t})$ is an **atomic** Q-formula (or an atom).
- if ϕ is a X-formula then (ϕ) is also a X-formula;
- if ϕ, ψ are X-formulae then $\phi \dagger_X \psi$ and $\phi \neg \dagger_X \psi$ are also X-formulae. We call $\dagger_X$ a **positive** relation and $\neg\dagger_X$ a **negative/negated** relation.
- Let $R \in \sigma_X$, ϕ, ψ, ω be X-formulae and $\dagger_X, \dagger'_X$ designate positive or negated relations. If ϕ has any of the following forms: (i) $\phi = R(\bar{t})$, (ii) $\phi = R(\bar{t}) \dagger_X \psi$ or (iii) $\phi = (R(\bar{t}) \dagger_X \omega) \dagger'_X \psi$, then it is R**-compatible**.

If ϕ and φ are both R**-compatible**, then $\phi \wedge \varphi$ and $\phi \vee \varphi$ are X formulae, and R**-compatible** as well.

If ϕ is a E or A-formula and $\overline{x}$ are variables occurring in ϕ, then we also write $\phi(\overline{x})$, to highlight these variables.

$L_{\mathcal{H}}$ Semantics

Let $\mathcal{A}$ be a $\sigma_E \cup \sigma_A$ (labelling) structure, and $\mathcal{H}$ be a $\mathcal{A}$-labelled **t**-graph. $\bar{i} \in \mathcal{A}$, $R \in \sigma_E \cup \sigma_A$, a, a' denote action nodes from $\mathcal{H}$, and q, q' denote quality edges. We write $\phi_{[\overline{x}\setminus\bar{i}]}$, to refer to the formula obtained from ϕ by replacing all variables from $\overline{x}$ with individuals from $\bar{i}$. Also, let $X \in \{E, A\}$. The semantics of X-formulae is defined as follows:

1. $\|\phi(\overline{x})\|^X_{\mathcal{H}} = \bigcup_{\bar{i}} \{q \in \|\phi_{[\overline{x}\setminus\bar{i}]}\|^Q_{\mathcal{H}}\}$;
2. $\|R(\bar{i})\|^X_{\mathcal{H}} = \{q : R(\bar{i}) \in \mathcal{L}_X(q)\}$;
3. $\|(\phi)\dagger_Q \psi\|^Q_{\mathcal{H}} = \{q \in \|\phi\|^Q_{\mathcal{H}} : \exists q' \in \|\psi\|^Q_{\mathcal{H}}$ such that $\lambda_{\dagger_Q}(q, q')\}$
4. $\|(\phi)\dagger_A \psi\|^A_{\mathcal{H}} = \{a \in \|\phi\|^A_{\mathcal{H}} : \exists a' \in \|\psi\|^A_{\mathcal{H}}$ such that $\lambda_{\dagger_A}(a, a')\}$
5. $\|\phi\neg \dagger_X \psi\|^X_{\mathcal{H}} = \|(\phi)\neg \dagger_X \psi\|^X_{\mathcal{H}} = \|\phi\|^X_{\mathcal{H}} \setminus \|\phi \dagger_X \psi\|^X_{\mathcal{H}}$;
6. $\|\phi \dagger_X (\psi)\|^X_{\mathcal{H}} = \|\phi \dagger_X \psi\|^X_{\mathcal{H}}$
7. $\|R(\bar{i}) \dagger_X \psi\|^X_{\mathcal{H}} = \|(R(\bar{i})) \dagger_X \psi\|^X_{\mathcal{H}}$;
8. $\|\phi \vee \psi\|^X_{\mathcal{H}} = \|\phi\|^X_{\mathcal{H}} \cup \|\psi\|^X_{\mathcal{H}}$;
9. $\|\phi \wedge \psi\|^X_{\mathcal{H}} = \|\phi\|^X_{\mathcal{H}} \cap \|\psi\|^X_{\mathcal{H}}$;

The language $L_{\mathcal{H}}$

Let $\varphi \in L_{\mathcal{H}}$ be a formula

The language $L_{\mathcal{H}}$

Let $\varphi \in L_{\mathcal{H}}$ be a formula and $\mathcal{H}$ be a temporal graph.

The language $L_{\mathcal{H}}$

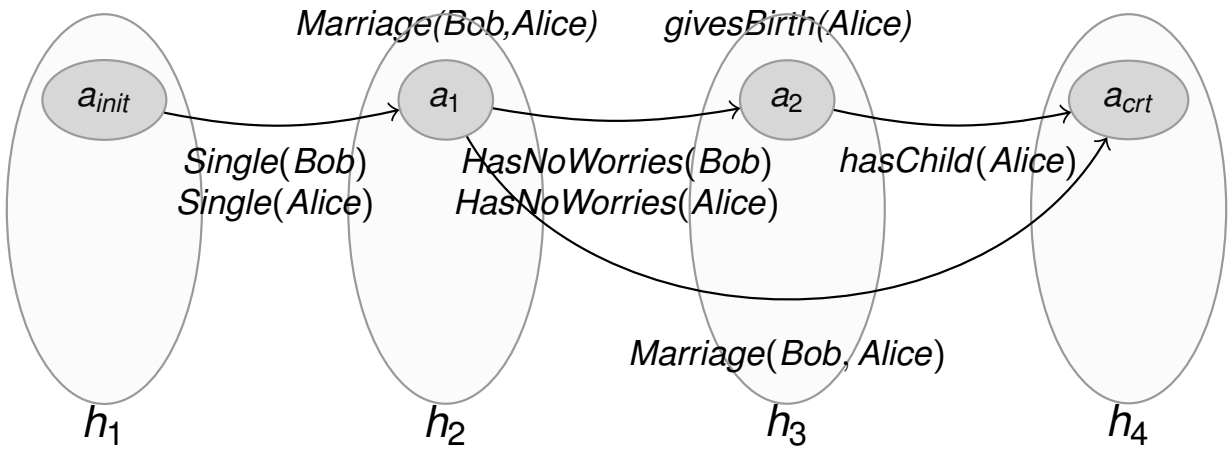
Let $\varphi \in L_{\mathcal{H}}$ be a formula and $\mathcal{H}$ be a temporal graph.

- $\|\varphi\|^E_{\mathcal{H}}$ is the **set** of quality edges that satisfy φ

The language $L_{\mathcal{H}}$

Let $\varphi \in L_{\mathcal{H}}$ be a formula and $\mathcal{H}$ be a temporal graph.

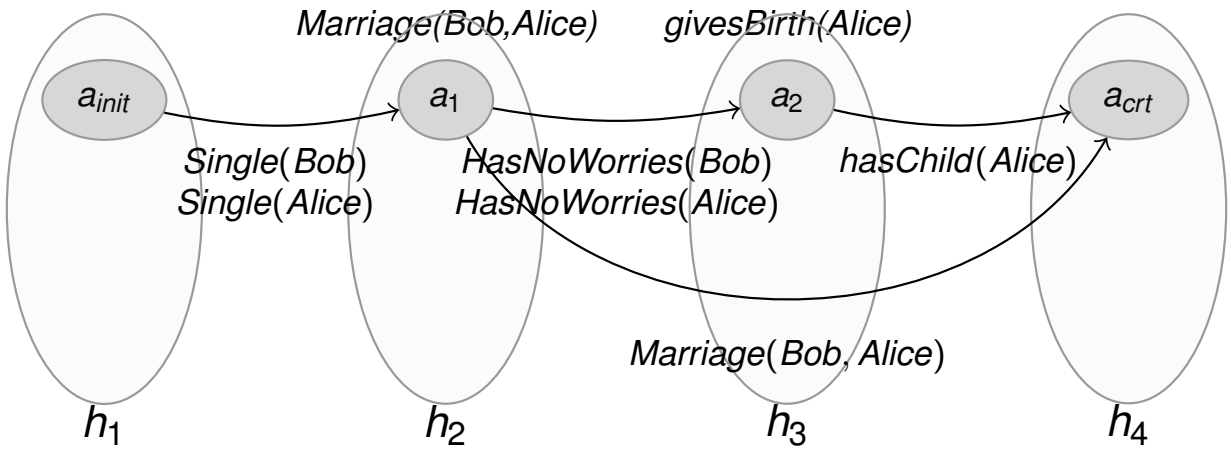
- $\|\varphi\|^{E}_{\mathcal{H}}$ is the **set** of quality edges that satisfy φ



The language $L_{\mathcal{H}}$

Let $\varphi \in L_{\mathcal{H}}$ be a formula and $\mathcal{H}$ be a temporal graph.

- $\|\varphi\|^E_{\mathcal{H}}$ is the **set** of quality edges that satisfy φ

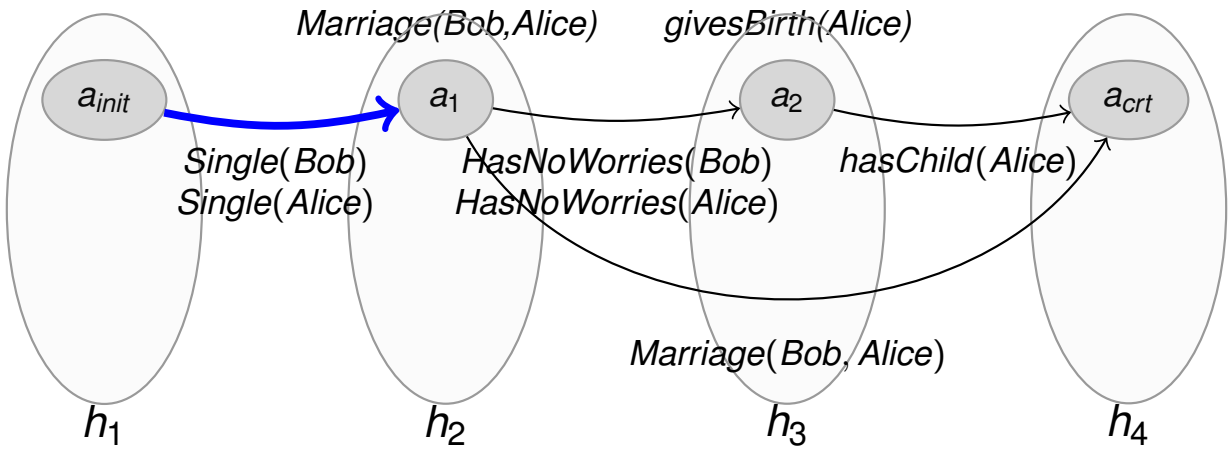


$\|\text{Single(x)}\|^E_{\mathcal{H}}$

The language $L_{\mathcal{H}}$

Let $\varphi \in L_{\mathcal{H}}$ be a formula and $\mathcal{H}$ be a temporal graph.

- $\|\varphi\|^{E}_{\mathcal{H}}$ is the **set** of quality edges that satisfy φ

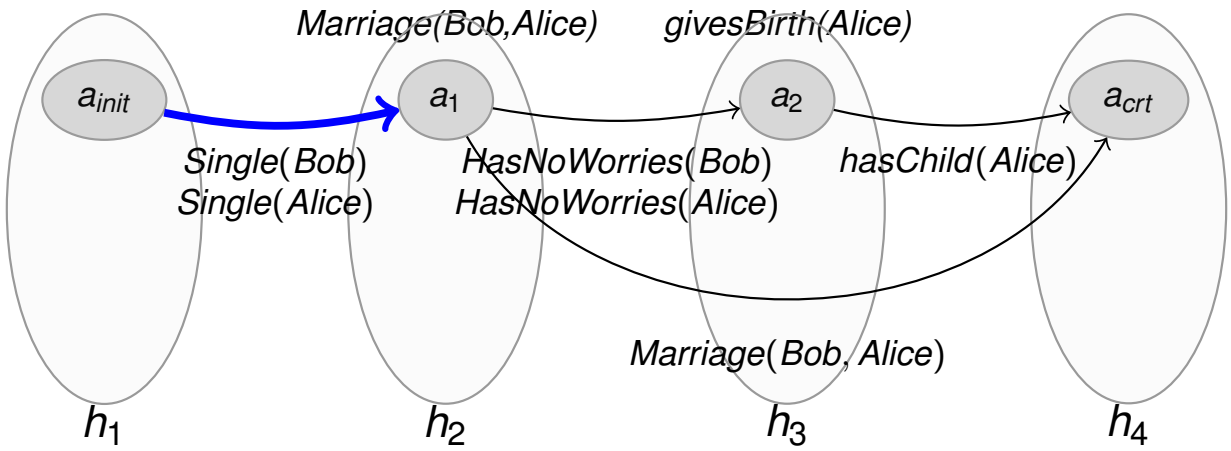


$\|\text{Single(x)}\|^{E}_{\mathcal{H}}$

The language $L_{\mathcal{H}}$

Let $\varphi \in L_{\mathcal{H}}$ be a formula and $\mathcal{H}$ be a temporal graph.

- $\|\varphi\|^{E}_{\mathcal{H}}$ is the **set** of quality edges that satisfy φ

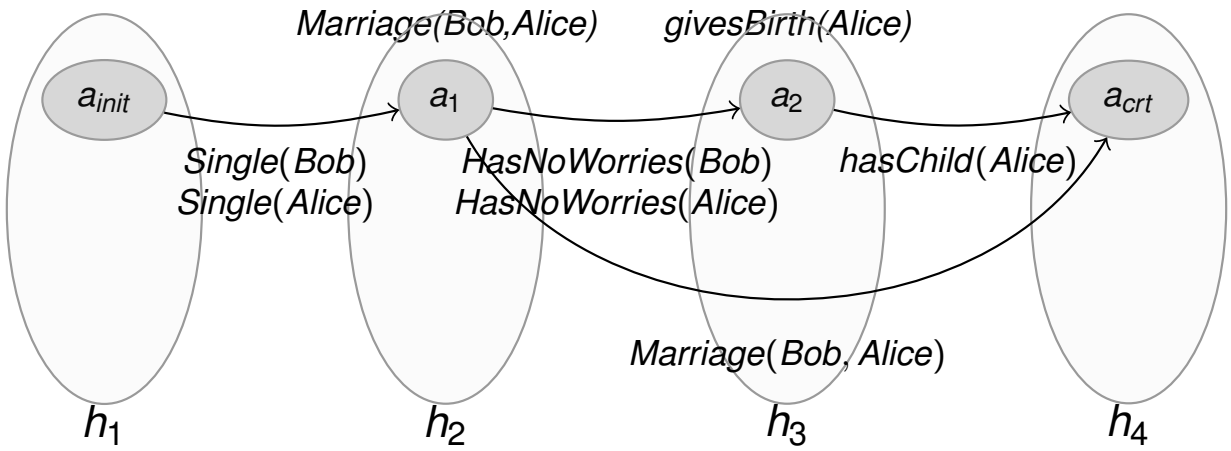


$\|\text{Single(x)}\|^{E}_{\mathcal{H}} = \{(a_{init}, a_1)\}$

The language $L_{\mathcal{H}}$

Let $\varphi \in L_{\mathcal{H}}$ be a formula and $\mathcal{H}$ be a temporal graph.

- $\|\varphi\|^{E}_{\mathcal{H}}$ is the **set** of quality edges that satisfy φ

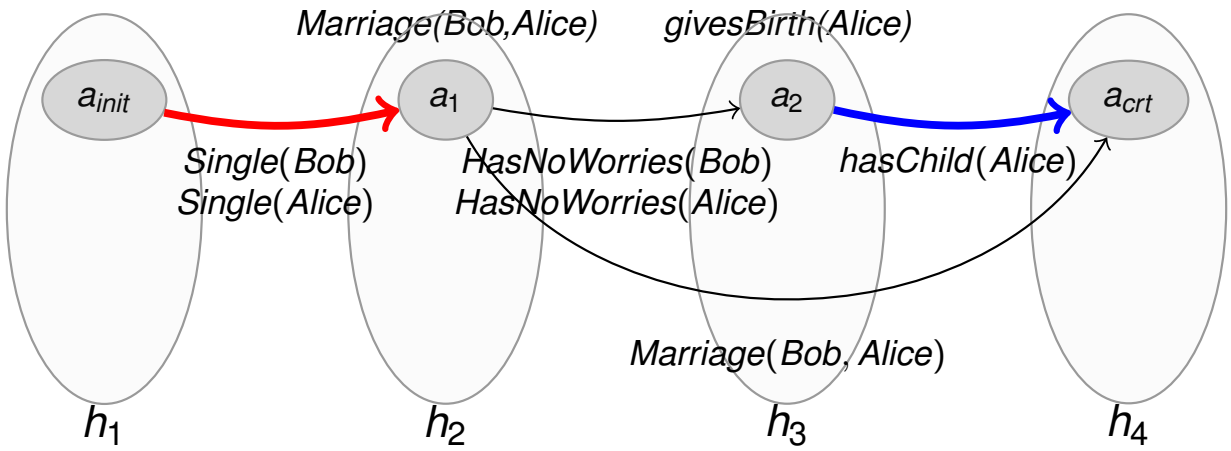


$\|\text{HasChild(x)}\ \ \textbf{after}\ \text{Single(x)}\|^{E}_{\mathcal{H}}$

The language $L_{\mathcal{H}}$

Let $\varphi \in L_{\mathcal{H}}$ be a formula and $\mathcal{H}$ be a temporal graph.

- $\|\varphi\|^{E}_{\mathcal{H}}$ is the **set** of quality edges that satisfy φ

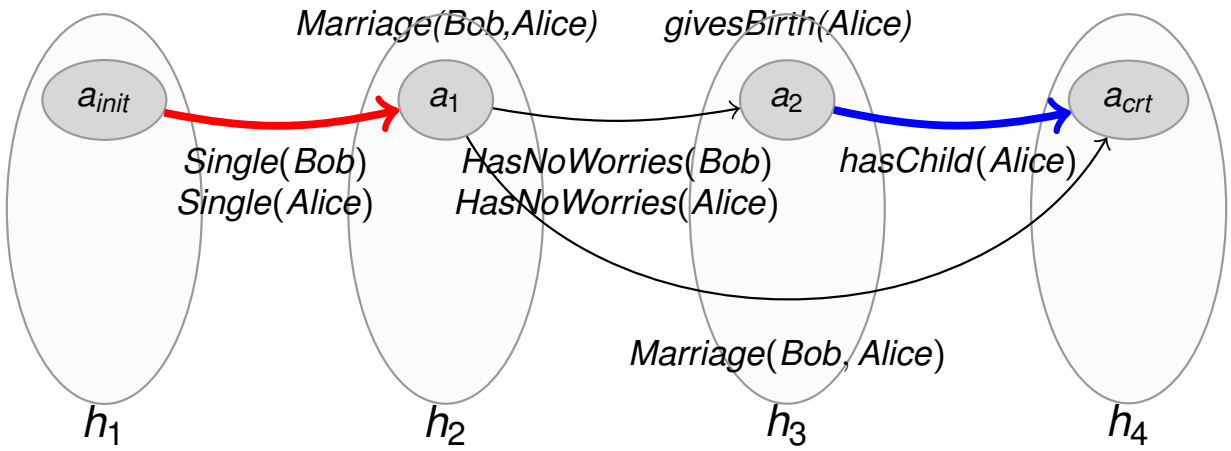


$\|\text{HasChild(x)}\ \ \textbf{after}\ \text{Single(x)}\|^{E}_{\mathcal{H}}$

The language $L_{\mathcal{H}}$

Let $\varphi \in L_{\mathcal{H}}$ be a formula and $\mathcal{H}$ be a temporal graph.

- $\|\varphi\|^{E}_{\mathcal{H}}$ is the **set** of quality edges that satisfy φ

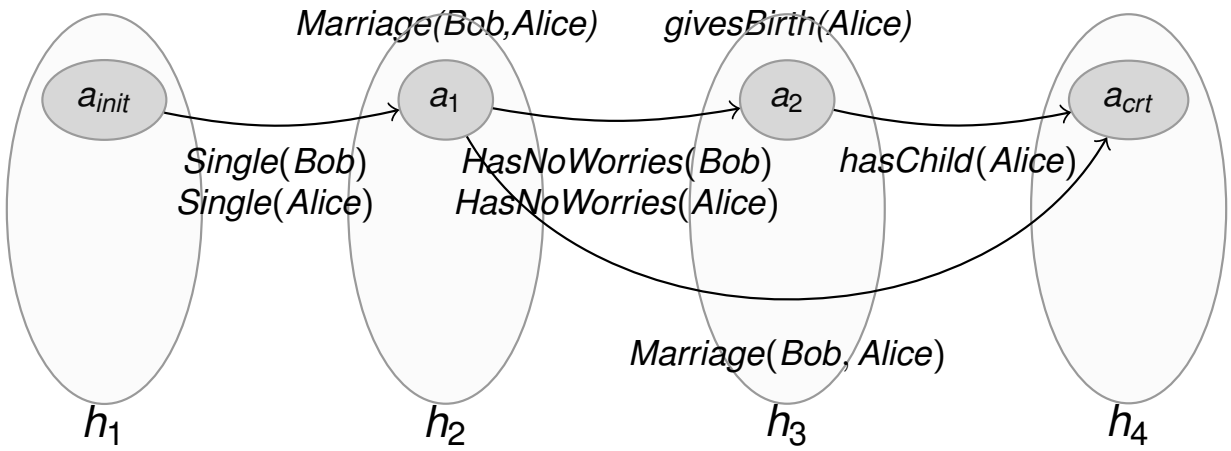


$\|\text{HasChild(x)} \ \ \textbf{after} \ \text{Single(x)}\|^{E}_{\mathcal{H}} = \{(a_2, a_{crt})\}$

The language $L_{\mathcal{H}}$

Let $\varphi \in L_{\mathcal{H}}$ be a formula and $\mathcal{H}$ be a temporal graph.

- $\|\varphi\|^{E}_{\mathcal{H}}$ is the **set** of quality edges that satisfy φ

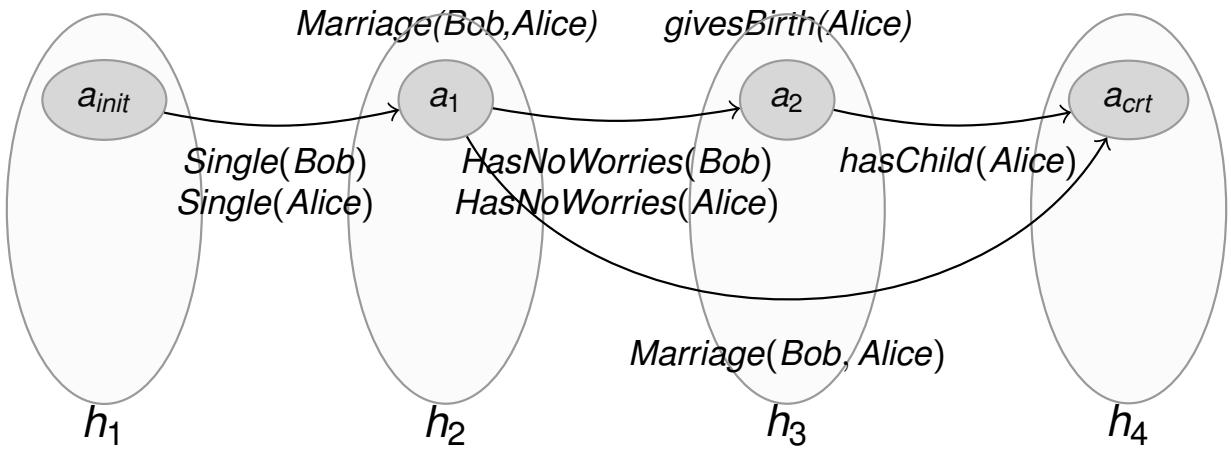


$\|\mathit{HasNoWorries}(x)$ **after** $\mathit{Single}(x) \wedge$
$\mathit{HasNoWorries}(x)$ **before** $\mathit{HasChild}(x)\|^{E}_{\mathcal{H}}$

The language $L_{\mathcal{H}}$

Let $\varphi \in L_{\mathcal{H}}$ be a formula and $\mathcal{H}$ be a temporal graph.

- $\|\varphi\|^{E}_{\mathcal{H}}$ is the **set** of quality edges that satisfy φ

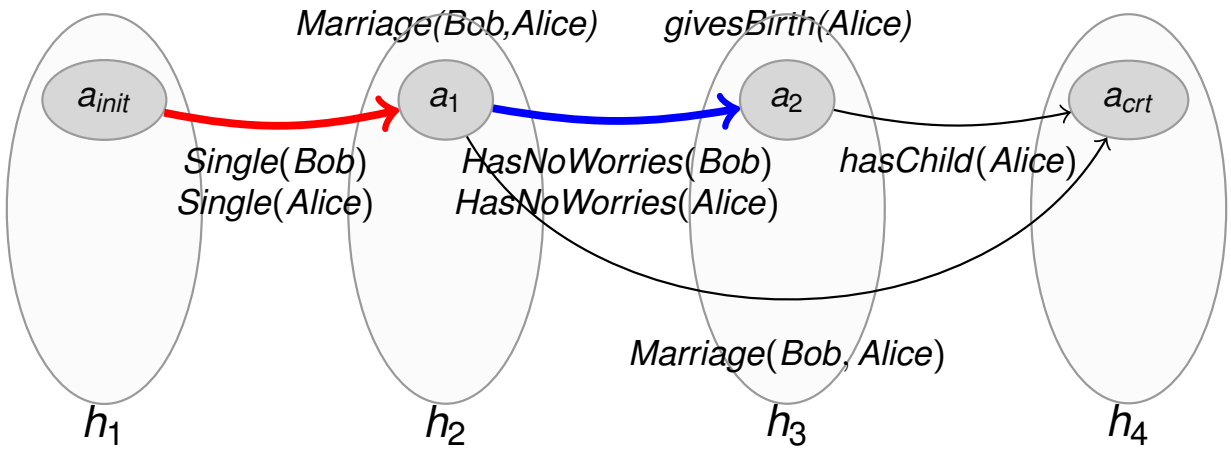


$\|\mathit{HasNoWorries}(x)$ **after** $\mathit{Single}(x) \wedge$
$\mathit{HasNoWorries}(x)$ **before** $\mathit{HasChild}(x)\|^{E}_{\mathcal{H}}$

The language $L_{\mathcal{H}}$

Let $\varphi \in L_{\mathcal{H}}$ be a formula and $\mathcal{H}$ be a temporal graph.

- $\|\varphi\|^{E}_{\mathcal{H}}$ is the **set** of quality edges that satisfy φ

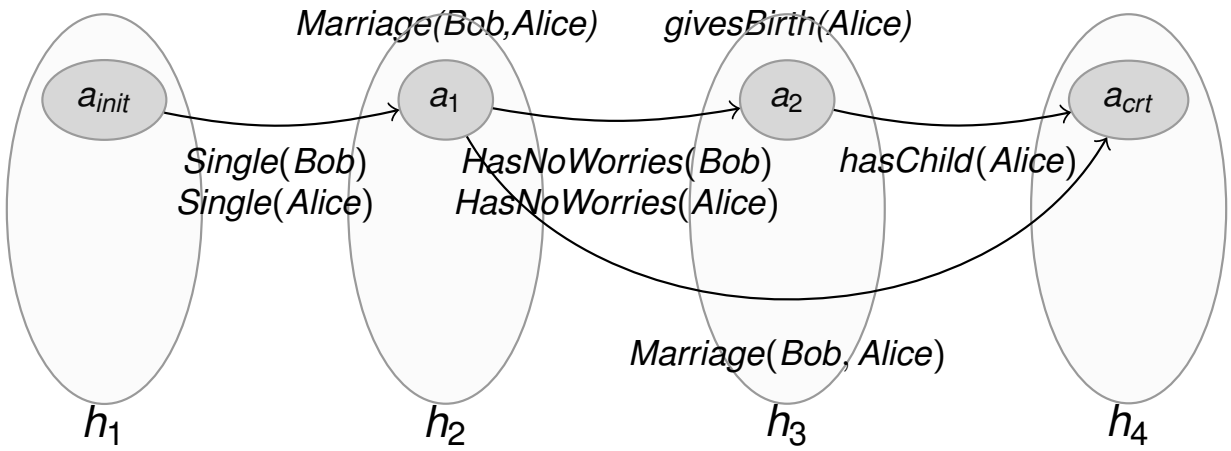


$\|\mathit{HasNoWorries}(x)$ **after** $\mathit{Single}(x) \wedge$
$\mathit{HasNoWorries}(x)$ **before** $\mathit{HasChild}(x)\|^{E}_{\mathcal{H}}$

The language $L_{\mathcal{H}}$

Let $\varphi \in L_{\mathcal{H}}$ be a formula and $\mathcal{H}$ be a temporal graph.

- $\|\varphi\|^{E}_{\mathcal{H}}$ is the **set** of quality edges that satisfy φ

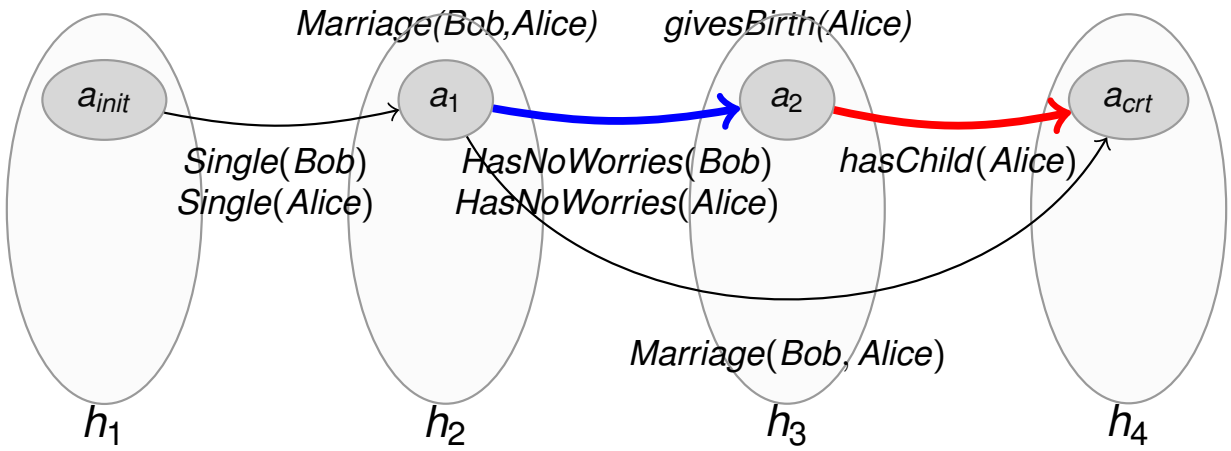


$\|\mathit{HasNoWorries}(x)$ **after** $\mathit{Single}(x) \wedge$
$\mathit{HasNoWorries}(x)$ **before** $\mathit{HasChild}(x)\|^{E}_{\mathcal{H}}$

The language $L_{\mathcal{H}}$

Let $\varphi \in L_{\mathcal{H}}$ be a formula and $\mathcal{H}$ be a temporal graph.

- $\|\varphi\|^{E}_{\mathcal{H}}$ is the **set** of quality edges that satisfy φ

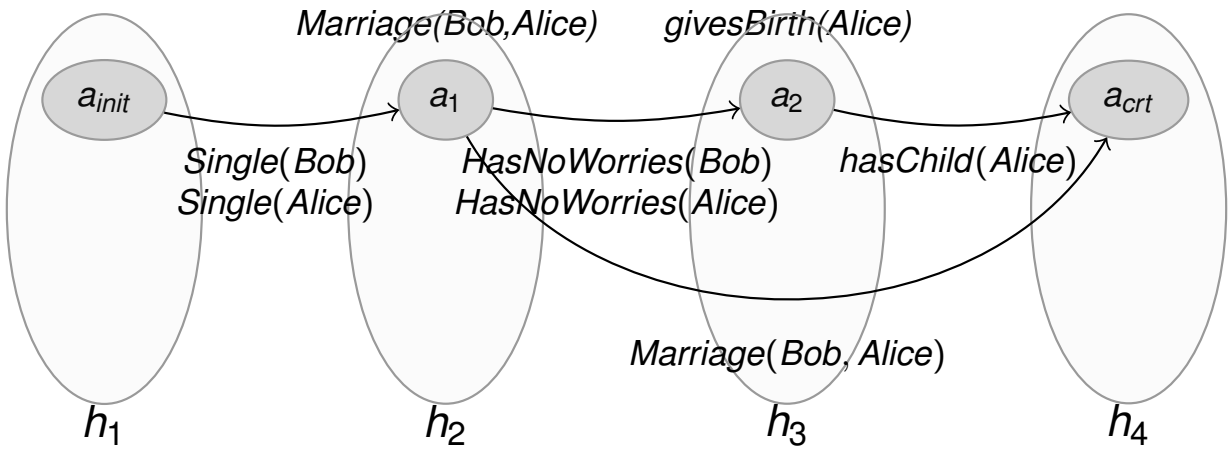


$\|\mathit{HasNoWorries}(x)$ **after** $\mathit{Single}(x) \wedge$
$\mathit{HasNoWorries}(x)$ **before** $\mathit{HasChild}(x)\|^{E}_{\mathcal{H}}$

The language $L_{\mathcal{H}}$

Let $\varphi \in L_{\mathcal{H}}$ be a formula and $\mathcal{H}$ be a temporal graph.

- $\|\varphi\|^{E}_{\mathcal{H}}$ is the **set** of quality edges that satisfy φ

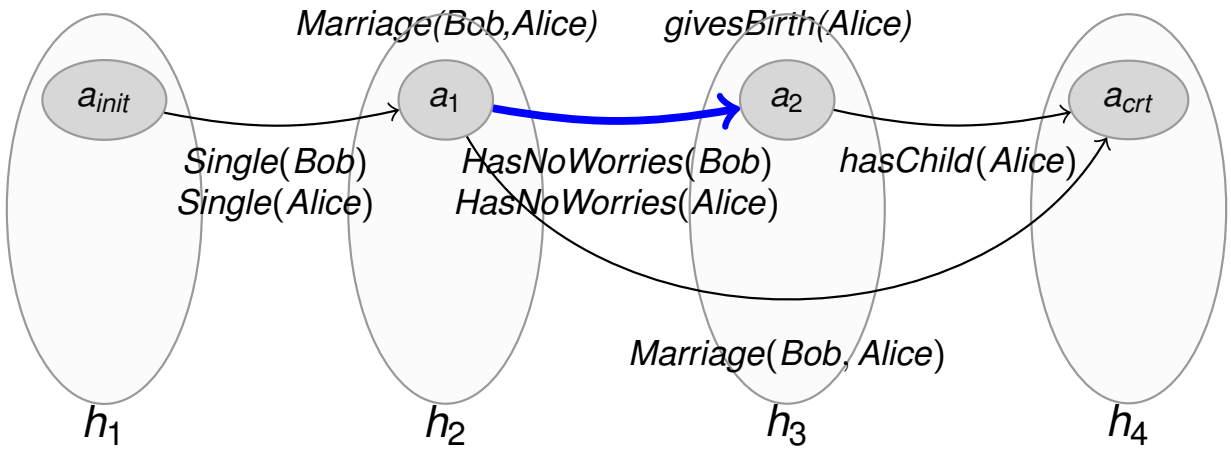


$\|\mathit{HasNoWorries}(x)$ **after** $\mathit{Single}(x) \wedge$
$\mathit{HasNoWorries}(x)$ **before** $\mathit{HasChild}(x)\|^{E}_{\mathcal{H}}$

The language $L_{\mathcal{H}}$

Let $\varphi \in L_{\mathcal{H}}$ be a formula and $\mathcal{H}$ be a temporal graph.

- $\|\varphi\|^{E}_{\mathcal{H}}$ is the **set** of quality edges that satisfy φ



$\|\mathit{HasNoWorries}(x)$ **after** $\mathit{Single}(x) \wedge$
$\mathit{HasNoWorries}(x)$ **before** $\mathit{HasChild}(x)\|^{E}_{\mathcal{H}} = \{(a_1, a_2)\}$

Outline

Complexity results

Computing $\|\varphi\|^{E}_{\mathcal{H}}$

Complexity results

Computing $\|\varphi\|^{E}_{\mathcal{H}} \equiv$ **model checking** the formula φ (w.r.t. a temporal graph $\mathcal{H}$)

Complexity results

Computing $\|\varphi\|^{E}_{\mathcal{H}} \equiv$ **model checking** the formula φ (w.r.t. a temporal graph $\mathcal{H}$)
The $L_{\mathcal{H}}$ **model checking** problem is:

Complexity results

Computing $\|\varphi\|^{E}_{\mathcal{H}} \equiv$ **model checking** the formula φ (w.r.t. a temporal graph $\mathcal{H}$)
The $L_{\mathcal{H}}$ **model checking** problem is:

- NP-complete, for the full language $\mathcal{L}_{\mathcal{H}}$;

Complexity results

Computing $\|\varphi\|^{E}_{\mathcal{H}} \equiv$ **model checking** the formula φ (w.r.t. a temporal graph $\mathcal{H}$)

The $L_{\mathcal{H}}$ **model checking** problem is:

- NP-complete, for the full language $\mathcal{L}_{\mathcal{H}}$;

Methodology:

Complexity results

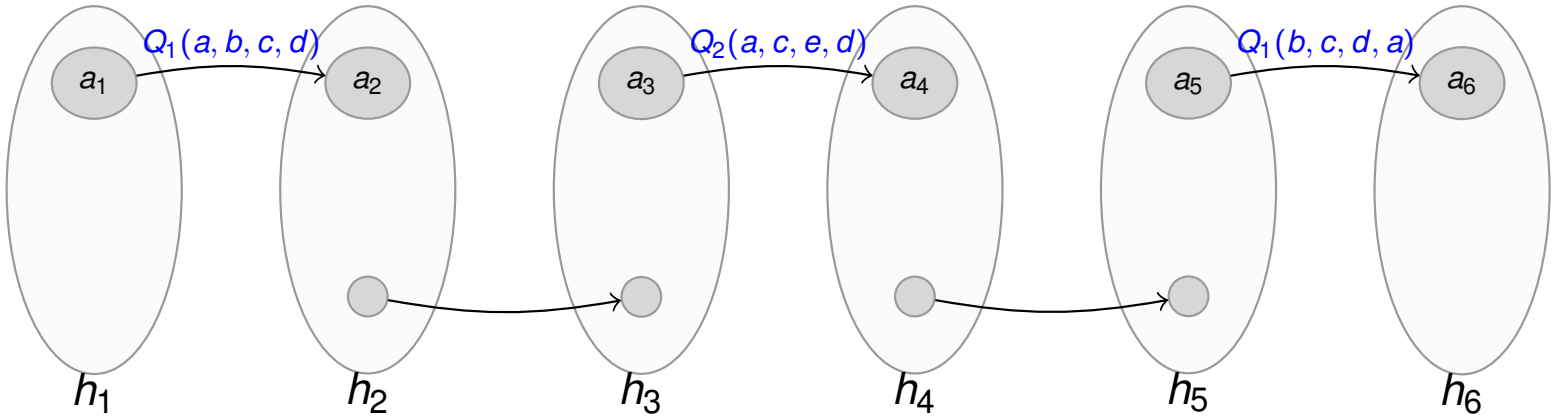
Computing $\|\varphi\|^E_{\mathcal{H}} \equiv$ **model checking** the formula φ (w.r.t. a temporal graph $\mathcal{H}$)

The $L_{\mathcal{H}}$ **model checking** problem is:

- NP-complete, for the full language $\mathcal{L}_{\mathcal{H}}$;

Methodology:

- Make labellings trivial:



Complexity results

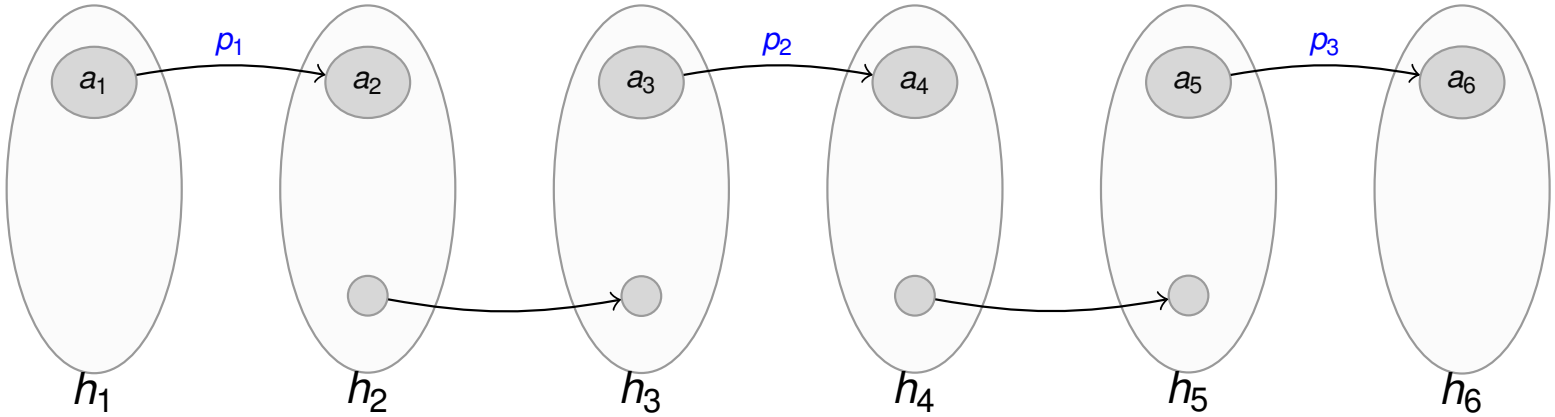
Computing $\|\varphi\|^E_{\mathcal{H}} \equiv$ **model checking** the formula φ (w.r.t. a temporal graph $\mathcal{H}$)

The $L_{\mathcal{H}}$ **model checking** problem is:

- NP-complete, for the full language $\mathcal{L}_{\mathcal{H}}$;

Methodology:

- Make labellings trivial: $L^*_{\mathcal{H}}$



Complexity results

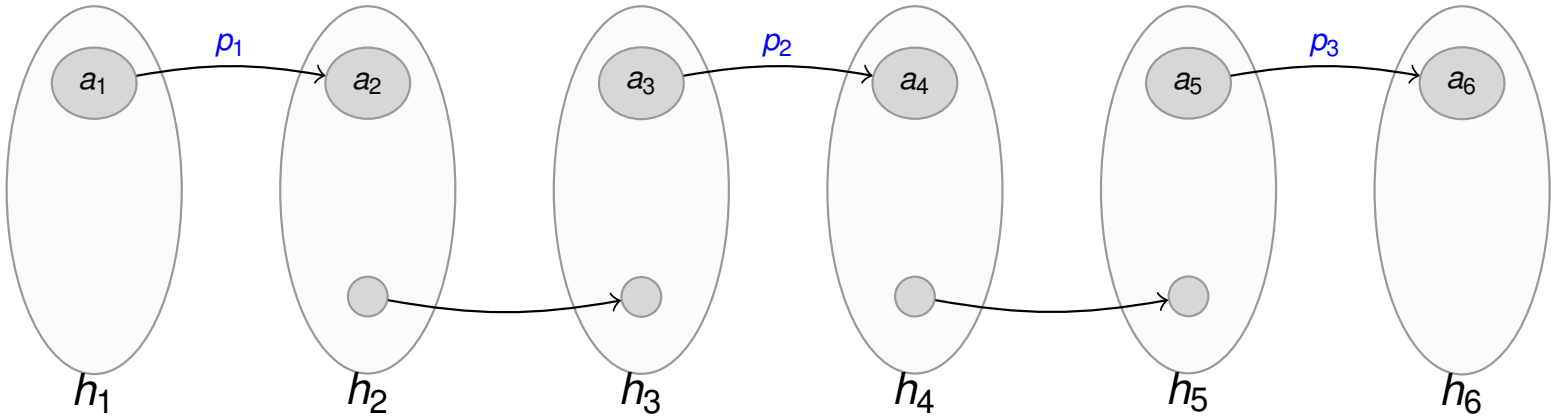
Computing $\|\varphi\|^E_{\mathcal{H}} \equiv$ **model checking** the formula φ (w.r.t. a temporal graph $\mathcal{H}$)

The $L_{\mathcal{H}}$ **model checking** problem is:

- NP-complete, for the full language $\mathcal{L}_{\mathcal{H}}$;

Methodology:

- Make labellings trivial: $L^*_{\mathcal{H}}$ PTIME model-checking



Complexity results

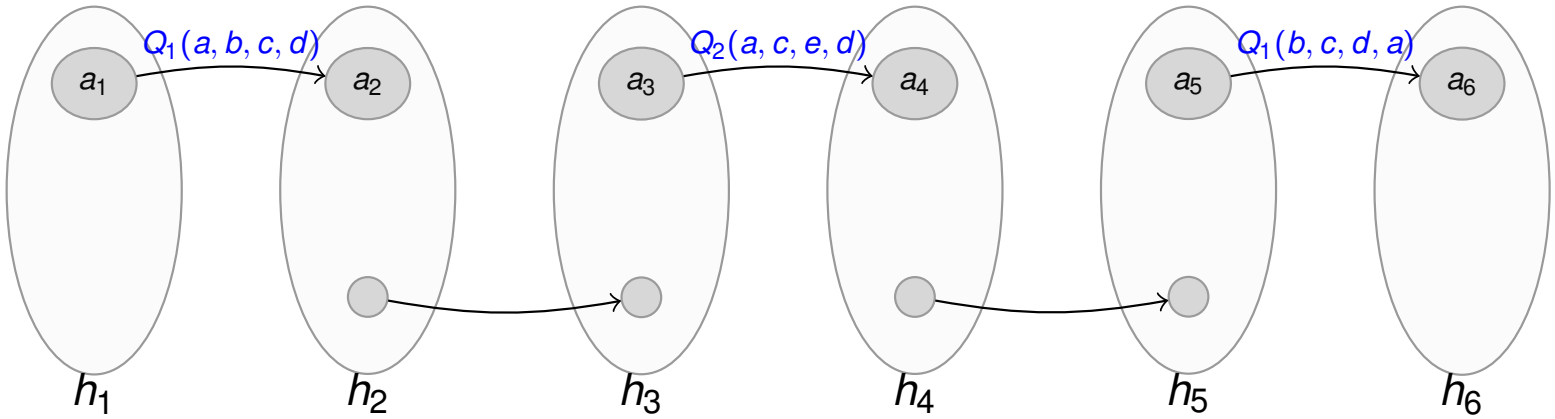
Computing $\|\varphi\|^{E}_{\mathcal{H}} \equiv$ **model checking** the formula φ (w.r.t. a temporal graph $\mathcal{H}$)

The $L_{\mathcal{H}}$ **model checking** problem is:

- NP-complete, for the full language $\mathcal{L}_{\mathcal{H}}$;

Methodology:

- Assume $\langle H, > \rangle$ is known:



Complexity results

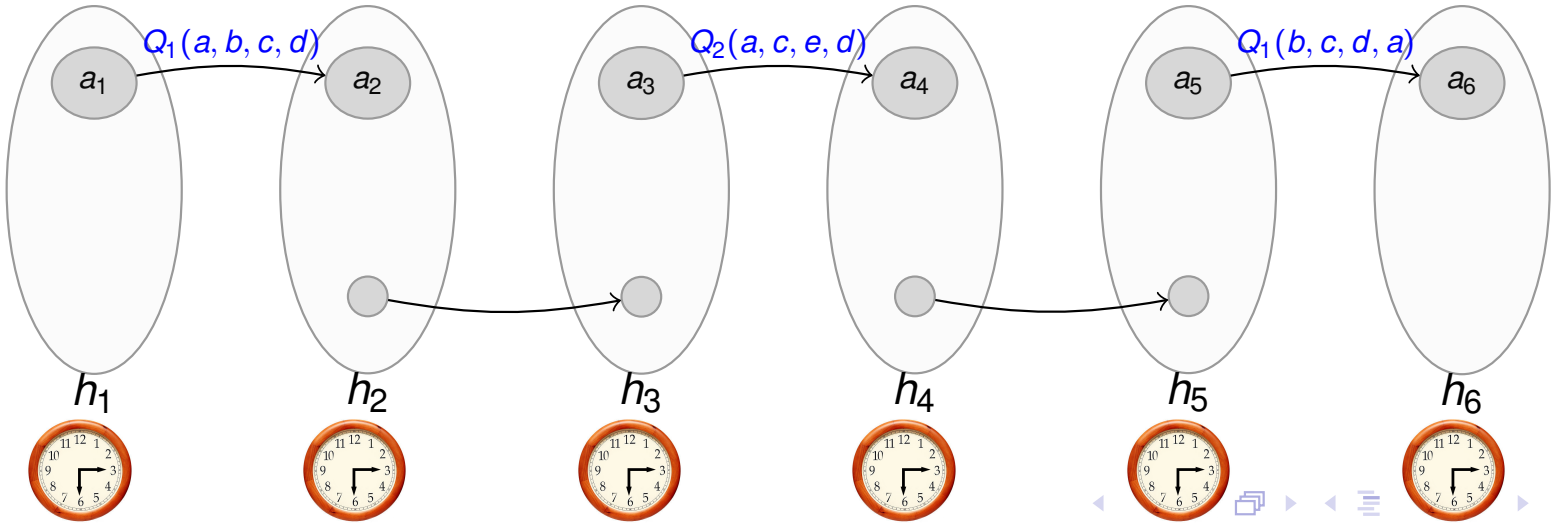
Computing $\|\varphi\|^{E}_{\mathcal{H}} \equiv$ **model checking** the formula φ (w.r.t. a temporal graph $\mathcal{H}$)

The $L_{\mathcal{H}}$ **model checking** problem is:

- NP-complete, for the full language $\mathcal{L}_{\mathcal{H}}$;

Methodology:

- Assume $\langle H, > \rangle$ is known: $L^{<}_{\mathcal{H}}$



Complexity results

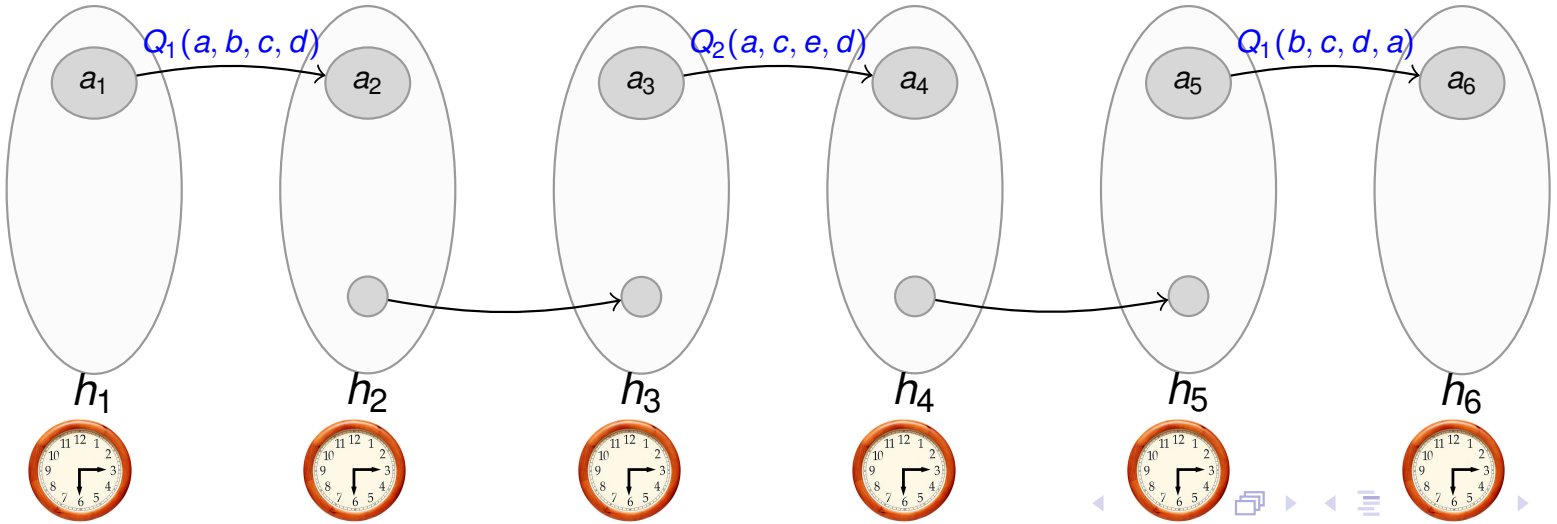
Computing $\|\varphi\|^{E}_{\mathcal{H}} \equiv$ **model checking** the formula φ (w.r.t. a temporal graph $\mathcal{H}$)

The $L_{\mathcal{H}}$ **model checking** problem is:

- NP-complete, for the full language $\mathcal{L}_{\mathcal{H}}$;

Methodology:

- Assume $\langle H, > \rangle$ is known: $L^{<}_{\mathcal{H}}$ NP-complete model-checking



Complexity results

Comparing $L_{\mathcal{H}}$ model checking with other approaches:

Method	**Reasoning problem**	**Complexity**
The TEDL $\mathcal{HS}$	satisfiability	*undecidable*
The TEDL $\mathcal{ALCT}$	satisfiability	*PSPACE – complete*
The DL $\mathcal{ALC}$(cyclic TBoxes)	subsumption	*EXPTIME*
The DL $\mathcal{FL}_0$(cyclic TBoxes)	subsumption	*PSPACE – complete*
$L_{\mathcal{H}}$	model-checking	*NP – complete*
$L^*_{\mathcal{H}}$	model-checking	*PTIME*

Applications of temporal graphs and $L_{\mathcal{H}}$

- Specifying **time-dependent** device behaviour in **intelligent buildings**
- Data mining logs from HPC systems
- Specifying **optimal** behaviour in **MAS** (Multi-Agent Systems)

Applications of temporal graphs and $L_{\mathcal{H}}$

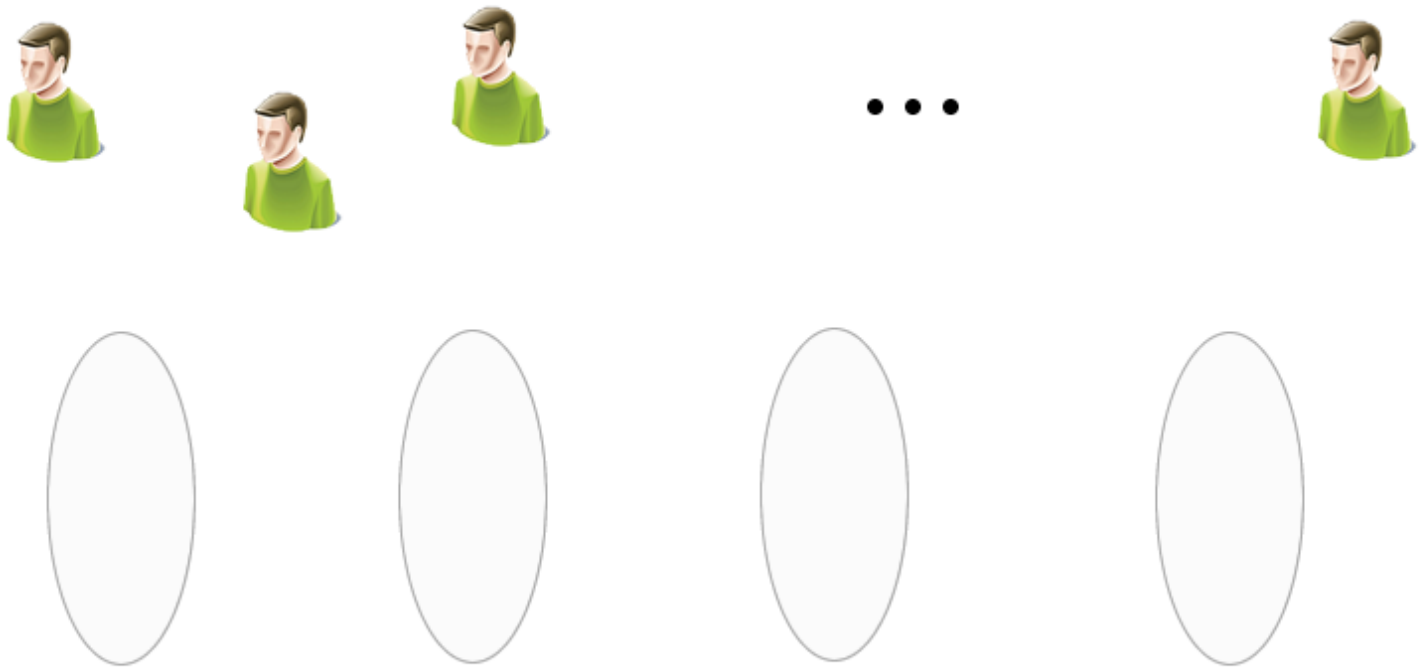
- Specifying **time-dependent** device behaviour in **intelligent buildings**
- Data mining logs from HPC systems
- Specifying **optimal** behaviour in **MAS** (Multi-Agent Systems)

Temporal graphs and $L^*_{\mathcal{H}}$ in MAS

Setting: a finite set of **players**

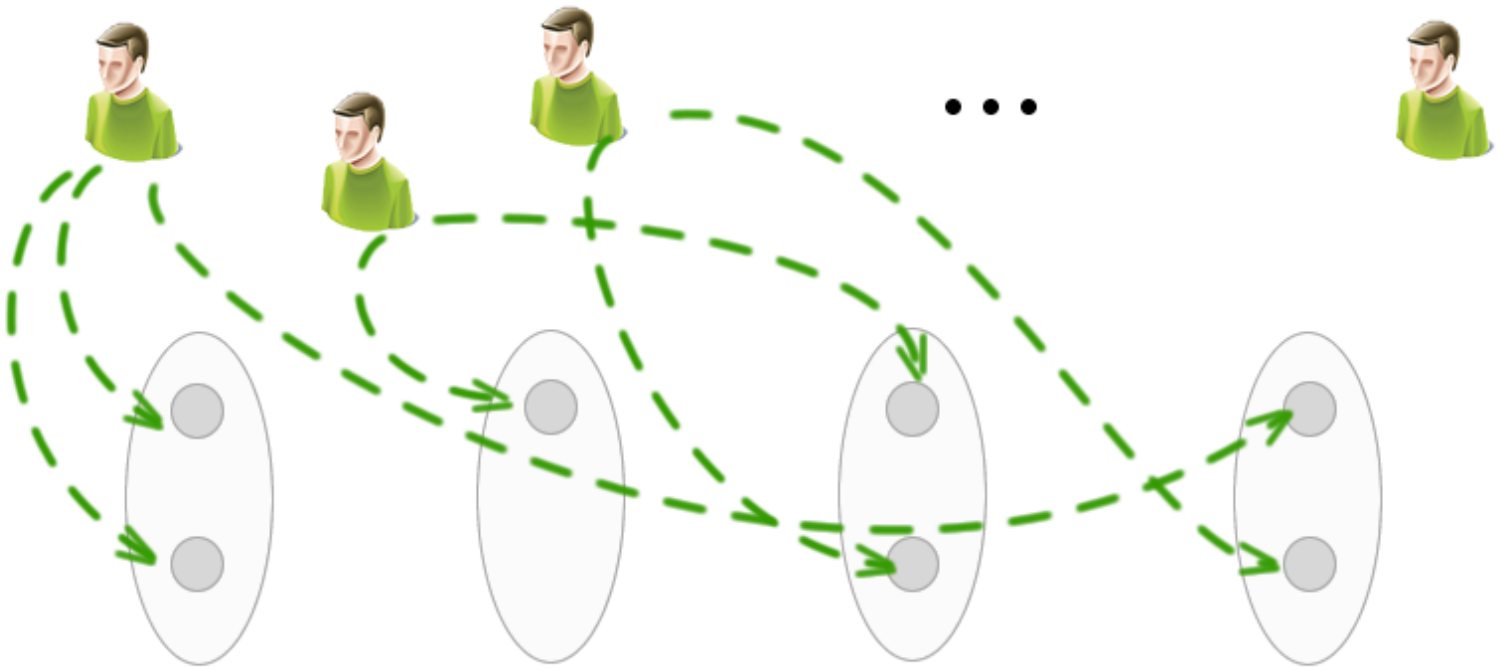
Temporal graphs and $L^*_{\mathcal{H}}$ in MAS

a finite set of **hypernodes**



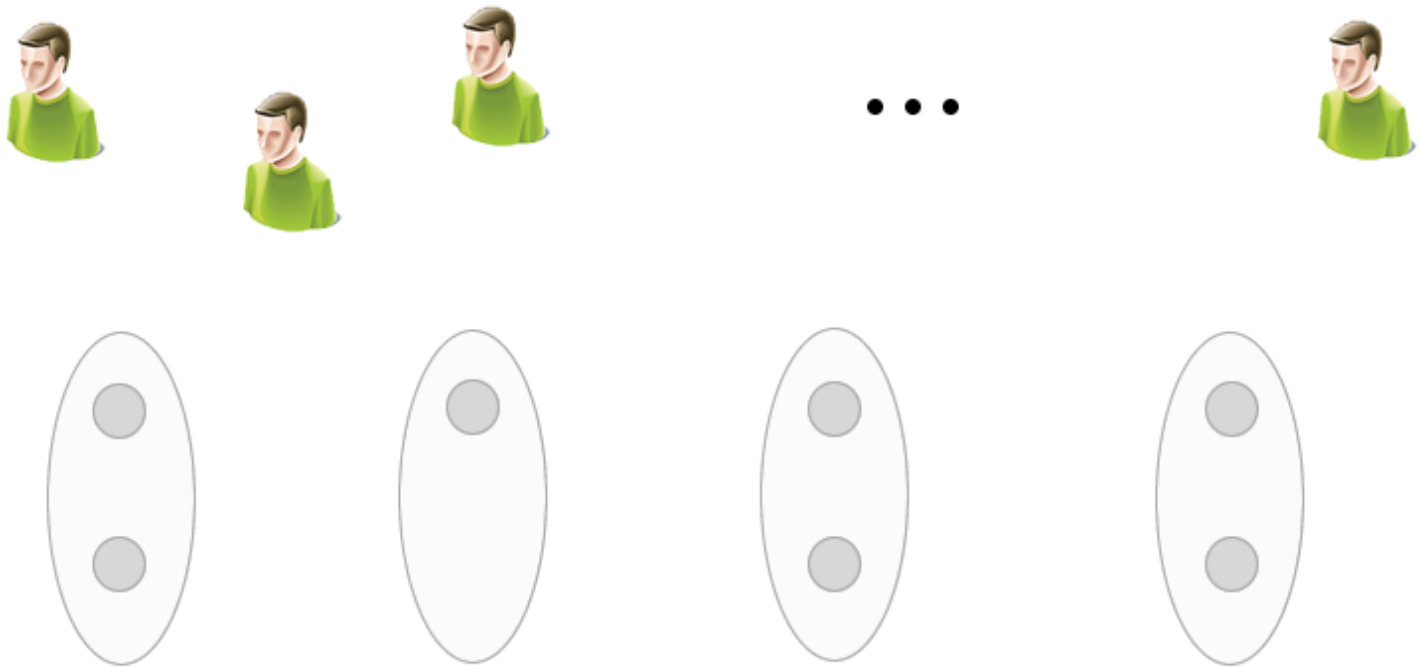
Temporal graphs and $L^*_{\mathcal{H}}$ in MAS

possible actions, for each player



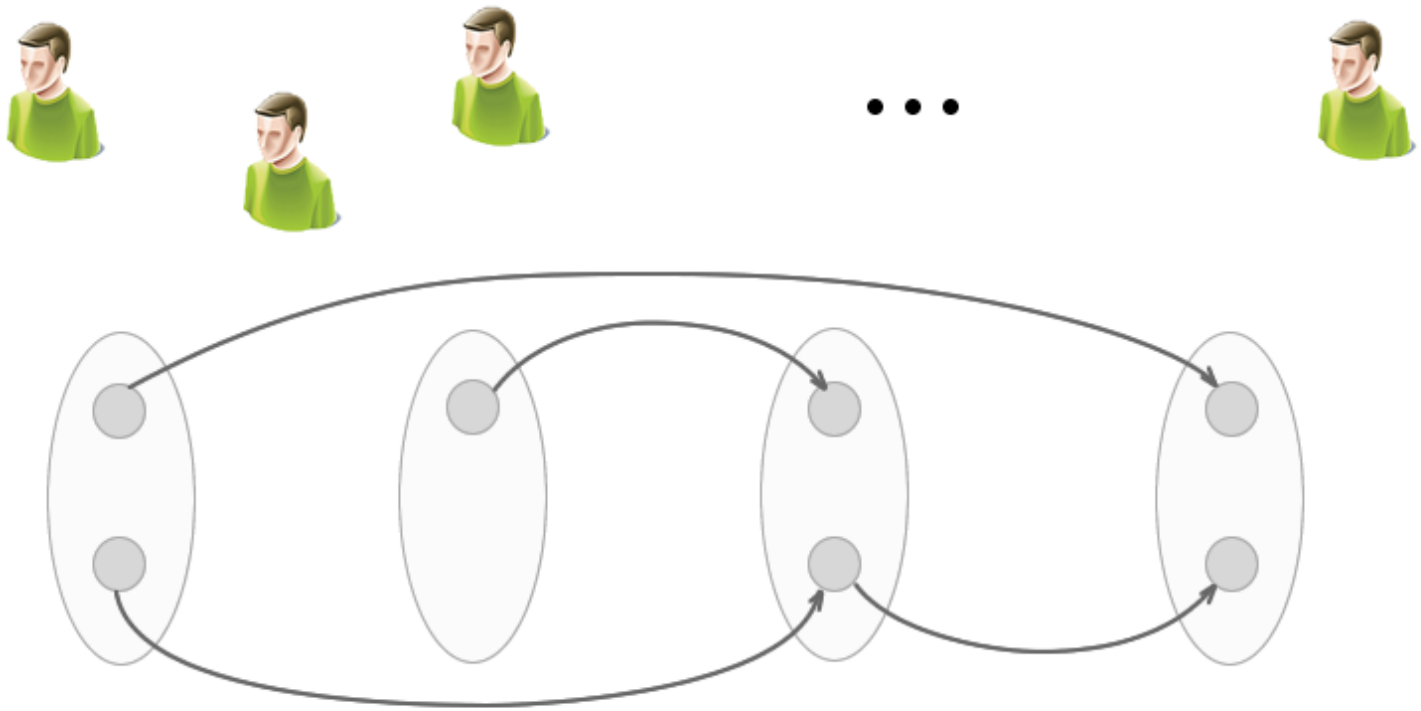
Temporal graphs and $L^*_{\mathcal{H}}$ in MAS

an *action* a_i consists of a **set** of **labelled** action nodes



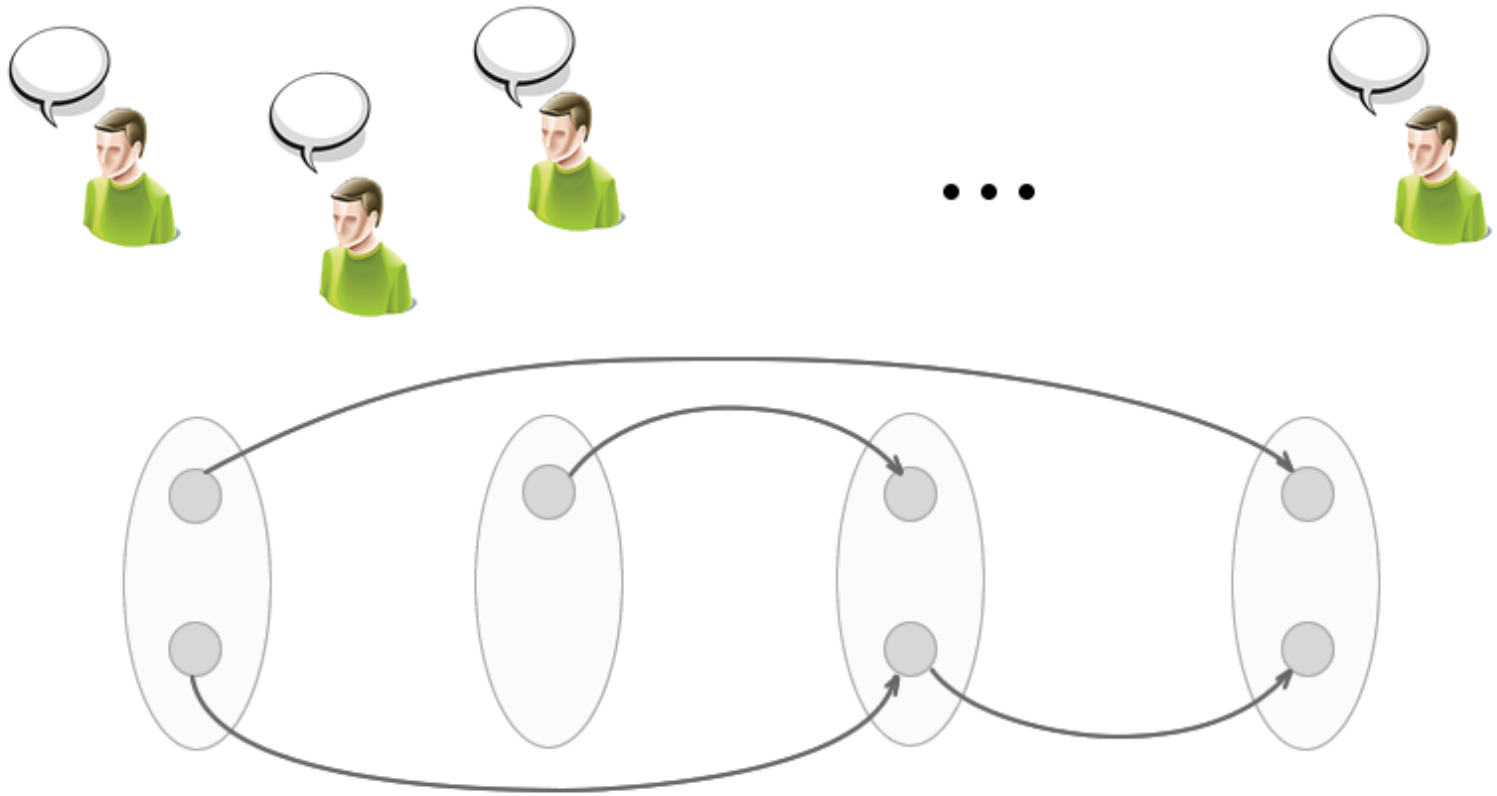
Temporal graphs and $L^*_{\mathcal{H}}$ in MAS

quality edges created/destroyed by action nodes



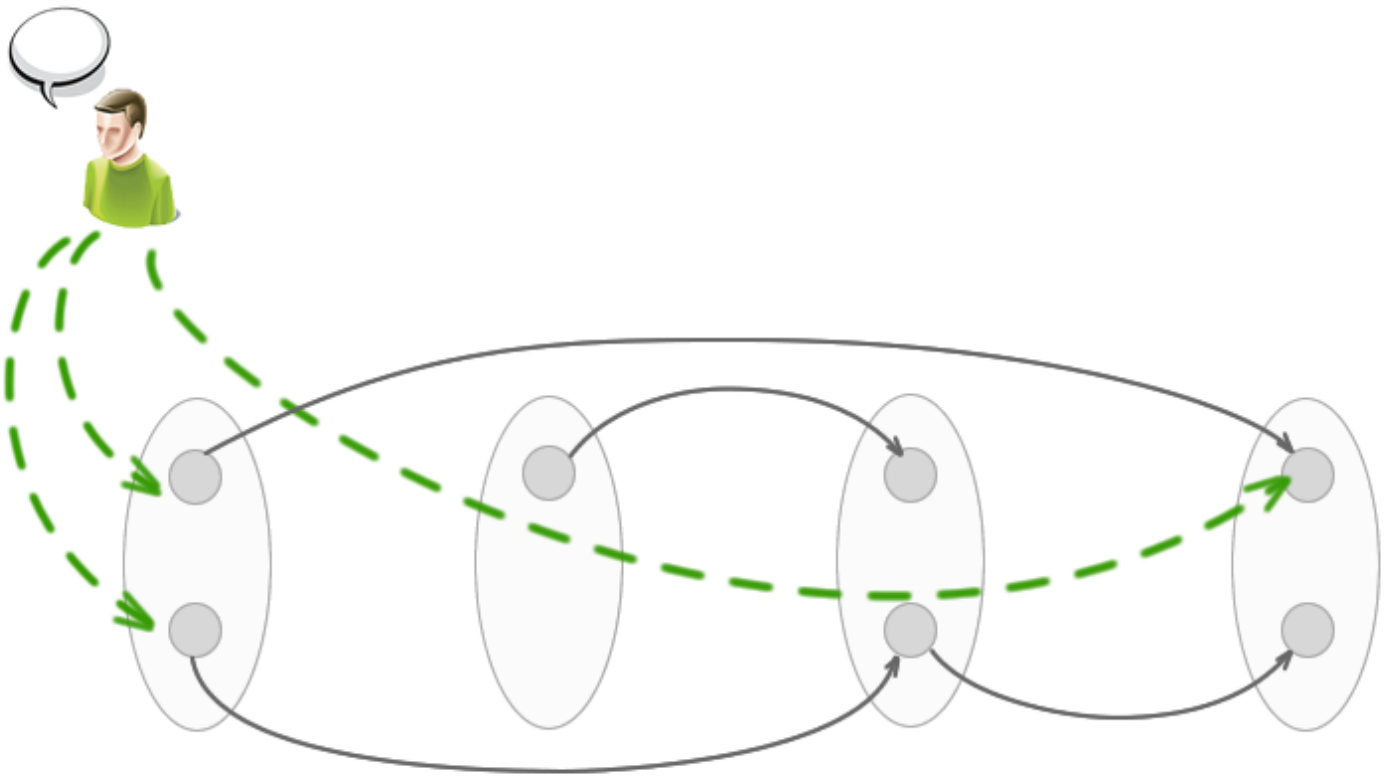
Temporal graphs and $L^*_{\mathcal{H}}$ in MAS

each player i has a **goal** φ_i. The goal is satisfied iff $\|\varphi_i\|^E_{\mathcal{H}} \neq \emptyset$



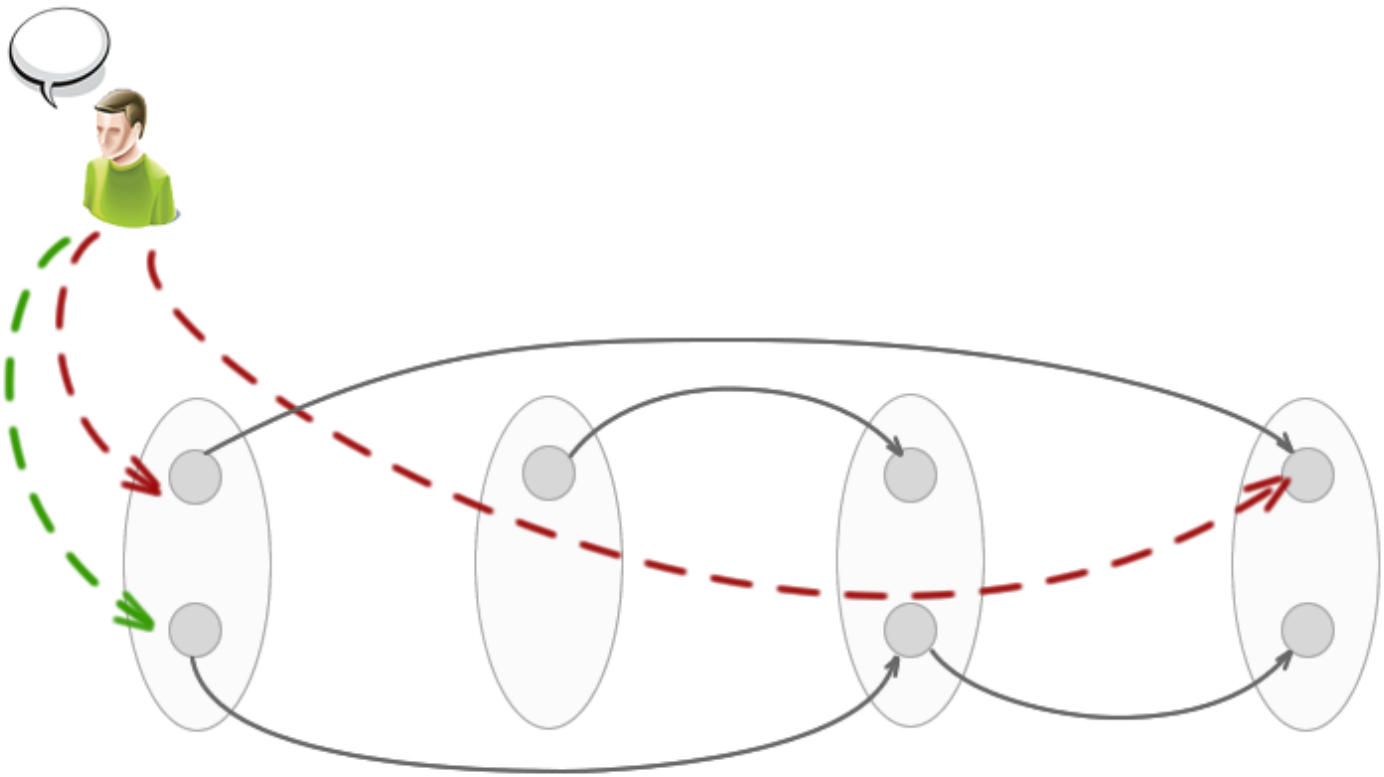
Temporal graphs and $L^*_{\mathcal{H}}$ in MAS

Deviation: Is a player incentivised to **change** his action, given the actions of others ?



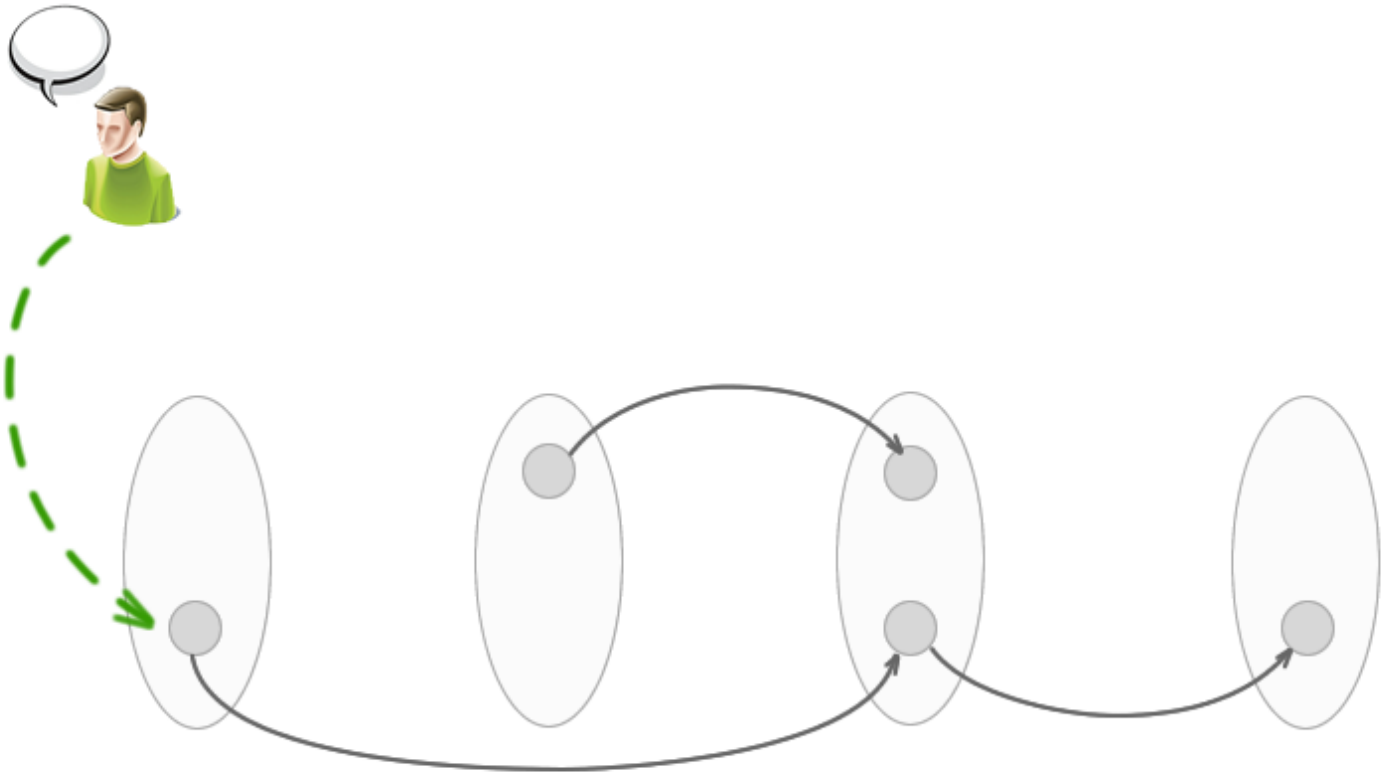
Temporal graphs and $L^*_{\mathcal{H}}$ in MAS

Deviation: Is a player incentivised to **change** his action, given the actions of others ?



Temporal graphs and $L^*_{\mathcal{H}}$ in MAS

Deviation: Is a player incentivised to **change** his action, given the actions of others ?



Temporal graphs and $L_{\mathcal{H}}$ in MAS

- *Remark:* The goal satisfaction of a player is affected by what **other players** do
- *Question:* What is the **proper** action each player should take, in order to **satisfy his goal** ?

Temporal graphs and $L_{\mathcal{H}}$ in MAS

An action profile $(a_1, \ldots, a_n)$ is a:

Temporal graphs and $L_{\mathcal{H}}$ in MAS

An action profile $(a_1, \ldots, a_n)$ is a:

- **Nash Equilibrium** (NE) iff there is no individual player that can **deviate**

Temporal graphs and $L_{\mathcal{H}}$ in MAS

An action profile $(a_1, \ldots, a_n)$ is a:

- **Nash Equilibrium** (NE) iff there is no individual player that can **deviate**
- **Strong Nash Equilibrium** (SNE) iff there is no **coalition** C of players that can **jointly deviate**

Temporal graphs and $L_{\mathcal{H}}$ in MAS

An action profile $(a_1, \ldots, a_n)$ is a:

- **Nash Equilibrium** (NE) iff there is no individual player that can **deviate**
- **Strong Nash Equilibrium** (SNE) iff there is no **coalition** C of players that can **jointly deviate**

Complexity results (for $L_{\mathcal{H}}$ with propositional symbols):

Temporal graphs and $L_{\mathcal{H}}$ in MAS

An action profile $(a_1, \ldots, a_n)$ is a:

- **Nash Equilibrium** (NE) iff there is no individual player that can **deviate**
- **Strong Nash Equilibrium** (SNE) iff there is no **coalition** C of players that can **jointly deviate**

Complexity results (for $L_{\mathcal{H}}$ with propositional symbols):

- The **verification** problem:

Temporal graphs and $L_{\mathcal{H}}$ in MAS

An action profile $(a_1, \ldots, a_n)$ is a:

- **Nash Equilibrium** (NE) iff there is no individual player that can **deviate**
- **Strong Nash Equilibrium** (SNE) iff there is no **coalition** C of players that can **jointly deviate**

Complexity results (for $L_{\mathcal{H}}$ with propositional symbols):

- The **verification** problem: coNP for NE and SNE

Temporal graphs and $L_{\mathcal{H}}$ in MAS

An action profile $(a_1, \ldots, a_n)$ is a:

- **Nash Equilibrium** (NE) iff there is no individual player that can **deviate**
- **Strong Nash Equilibrium** (SNE) iff there is no **coalition** C of players that can **jointly deviate**

Complexity results (for $L_{\mathcal{H}}$ with propositional symbols):

- The **verification** problem: coNP for NE and SNE
- The **synthesis** problem:

Temporal graphs and $L_{\mathcal{H}}$ in MAS

An action profile $(a_1, \ldots, a_n)$ is a:

- **Nash Equilibrium** (NE) iff there is no individual player that can **deviate**
- **Strong Nash Equilibrium** (SNE) iff there is no **coalition** C of players that can **jointly deviate**

Complexity results (for $L_{\mathcal{H}}$ with propositional symbols):

- The **verification** problem: coNP for NE and SNE
- The **synthesis** problem: Σ_2 for NE and SNE

Applications of temporal graphs and $L_{\mathcal{H}}$

- Specifying **time-dependent** device behaviour in **intelligent buildings**
- Data mining logs from HPC systems
- Specifying **optimal** behaviour in **MAS** (Multi-Agent Systems)

Practical application: intelligent buildings

Specifying **time-dependent** device behaviour in intelligent buildings

Practical application: intelligent buildings

Specifying **time-dependent** device behaviour in intelligent buildings using a Domain-Specific Language (DSL).

Practical application: intelligent buildings

Specifying **time-dependent** device behaviour in intelligent buildings using a Domain-Specific Language (DSL). The implementation relies on:

Practical application: intelligent buildings

Specifying **time-dependent** device behaviour in intelligent buildings using a Domain-Specific Language (DSL). The implementation relies on:

- **a model-checker** for the language $L^*_{\mathcal{H}}$

Practical application: intelligent buildings

Specifying **time-dependent** device behaviour in intelligent buildings using a Domain-Specific Language (DSL). The implementation relies on:

- **a model-checker** for the language $L^*_{\mathcal{H}}$
  - written in Java & Haskell

Practical application: intelligent buildings

Specifying **time-dependent** device behaviour in intelligent buildings using a Domain-Specific Language (DSL). The implementation relies on:

- **a model-checker** for the language $L^*_{\mathcal{H}}$
  - written in Java & Haskell
  - **tractable** implementation

Practical application: intelligent buildings

Specifying **time-dependent** device behaviour in intelligent buildings using a Domain-Specific Language (DSL). The implementation relies on:

- **a model-checker** for the language $L^*_{\mathcal{H}}$
  - written in Java & Haskell
  - **tractable** implementation
- interaction with devices, which are abstracted by Web Services

Practical application: intelligent buildings

Specifying **time-dependent** device behaviour in intelligent buildings using a Domain-Specific Language (DSL). The implementation relies on:

- **a model-checker** for the language $L^*_{\mathcal{H}}$
  - written in Java & Haskell
  - **tractable** implementation
- interaction with devices, which are abstracted by Web Services

In it's initial phase, DSL was developed under the **FCINT POSCCE project** (Ontology-based Service Composition Framework for Syndicating Building Intelligence)

Conclusions

- **Temporal graphs** are suitable for storing information about the history of a domain
- **The language** $L^*_{\mathcal{H}}$ can be used to capture classes of system properties, that are involved in complex temporal relation.