Synthesis for fragments of first-order logic on data words

Julien Grange[1], Mathieu Lehaut[2]

[1]LACL, Université Paris-Est Créteil, France
[2]University of Gothenburg, Sweden

May 7th, 2024

Motivation

- We want an **unbounded number of agents**...
  - processes
  - computers in a network
  - drones

Motivation

- We want an **unbounded number of agents**...
  - processes
  - computers in a network
  - drones
- ...acting in an **uncontrollable environment**...

Motivation

- We want an **unbounded number of agents**...
  - processes
  - computers in a network
  - drones
- ...acting in an **uncontrollable environment**...
- ...to satisfy some **specification**

- We want an **unbounded number of agents**...
  - processes
  - computers in a network
  - drones
- ...acting in an **uncontrollable environment**...
- ...to satisfy some **specification**

System and **Environment**, playing actions (a and b for System, c and d for Environment) in turn on shared or proper agents :

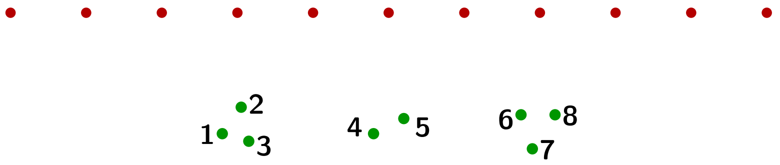
(1,a)(8,b)(7,d)(4,c)(6,a)(6,c)(7,a)(6,d)(2,b)(7,d)(7,a)

Executions : finite or infinite **data words**

`(1,a)(8,b)(7,d)(4,c)(6,a)(6,c)(7,a)(6,d)(2,b)(7,d)(7,a)`

Executions : finite or infinite **data words**

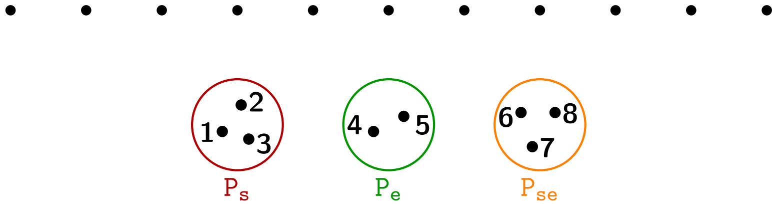
`(1,a)(8,b)(7,d)(4,c)(6,a)(6,c)(7,a)(6,d)(2,b)(7,d)(7,a)`



- One element for each position
- One element for each agent

Executions : finite or infinite **data words**

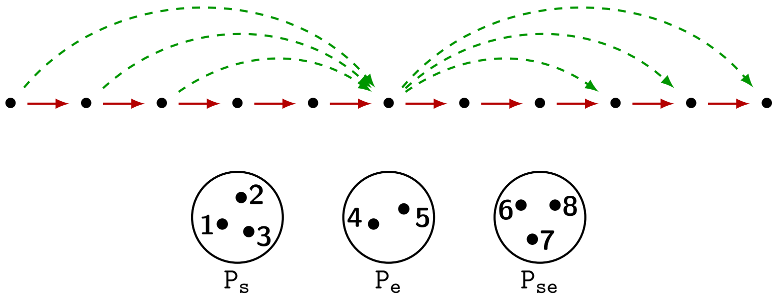
(1,a)(8,b)(7,d)(4,c)(6,a)(6,c)(7,a)(6,d)(2,b)(7,d)(7,a)



- Three unary relations P_s, P_e and P_{se} to denote ownership of the agents

Executions : finite or infinite **data words**

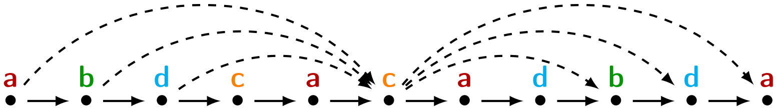
`(1,a)(8,b)(7,d)(4,c)(6,a)(6,c)(7,a)(6,d)(2,b)(7,d)(7,a)`



- A binary relation $+1$ between successive positions
- A binary relation $<$ for its transitive closure

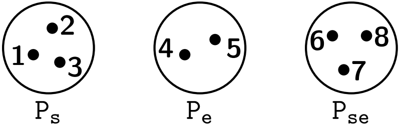
Executions : finite or infinite **data words**

(1,a)(8,b)(7,d)(4,c)(6,a)(6,c)(7,a)(6,d)(2,b)(7,d)(7,a)



$A_s = \{a, b\}$

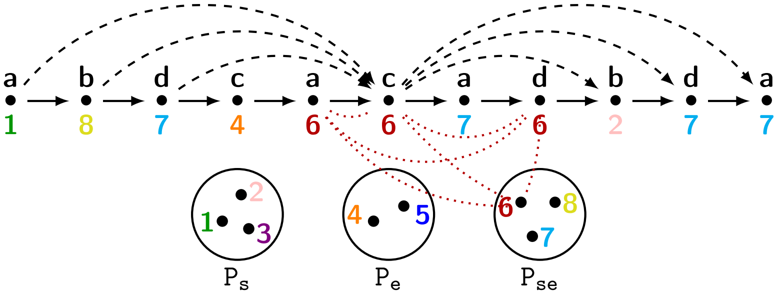
$A_e = \{c, d\}$



- A unary relation for each action

Executions : finite or infinite **data words**

(1,a)(8,b)(7,d)(4,c)(6,a)(6,c)(7,a)(6,d)(2,b)(7,d)(7,a)



- An equivalence relation $\sim$ with a class for each agent

Specification language

Fragment of first-order logic , with a subset of the binary predicates

Specification language

Fragment of first-order logic , with a subset of the binary predicates

- $\mathsf{FO}^2[\sim, <, +1]$

Specification language

$\overbrace{\text{Fragment of first-order logic}}^{\text{two variables}}$, with a subset of the binary predicates

- $\mathsf{FO}^2[\sim, <, +1]$

Specification language

Fragment of first-order logic, with a $\overbrace{\text{subset of the binary predicates}}^{\text{all predicates}}$

- $\mathsf{FO}^2[\sim, <, +1]$

Specification language

Fragment of first-order logic , with a subset of the binary predicates

- $FO^2[\sim, <, +1]$

> *Every agent requesting a resource eventually gets it :*

Specification language

Fragment of first-order logic , with a subset of the binary predicates

- $\mathsf{FO}^2[\sim, <, +1]$

Every agent requesting a resource eventually gets it :

$\forall x,\ \texttt{req(x)} \rightarrow\ \exists y,\ y \sim x\ \wedge\ y > x\ \wedge\ \texttt{gets(y)}$

Specification language

Fragment of first-order logic , with a subset of the binary predicates

- $\mathsf{FO}^2[\sim, <, +1]$

> *Every agent requesting a resource eventually gets it :*
>
> $\forall x,\ \texttt{req(x)} \to\ \exists y,\ y \sim x\ \wedge\ y > x\ \wedge\ \texttt{gets(y)}$

- $\mathsf{FO}[\sim]$

Specification language

$\overbrace{\text{Fragment of first-order logic}}^{\text{no restriction}}$, with a subset of the binary predicates

- $\mathsf{FO}^2[\sim, <, +1]$

> *Every agent requesting a resource eventually gets it :*
> $\forall x,\ \mathtt{req(x)} \rightarrow\ \exists y,\ y \sim x\ \wedge\ y > x\ \wedge\ \mathtt{gets(y)}$

- $\mathsf{FO}[\sim]$

Specification language

no positional predicate

Fragment of first-order logic, with a subset of the binary predicates

- $\mathsf{FO}^2[\sim, <, +1]$

> *Every agent requesting a resource eventually gets it :*
>
> $\forall x,\ \texttt{req(x)} \rightarrow\ \exists y,\ y \sim x\ \wedge\ y > x\ \wedge\ \texttt{gets(y)}$

- $\mathsf{FO}[\sim]$

Specification language

Fragment of first-order logic , with a subset of the binary predicates

- $\mathrm{FO}^2[\sim, <, +1]$

> *Every agent requesting a resource eventually gets it :*
>
> $\forall x,\ \mathtt{req(x)} \rightarrow\ \exists y,\ y \sim x\ \wedge\ y > x\ \wedge\ \mathtt{gets(y)}$

- $\mathrm{FO}[\sim]$

> *Every System agent requests at most twice a resource :*

Specification language

Fragment of first-order logic , with a subset of the binary predicates

- $\mathsf{FO}^2[\sim, <, +1]$

> *Every agent requesting a resource eventually gets it :*
>
> $\forall x,\ \mathtt{req(x)} \rightarrow\ \exists y,\ y \sim x\ \wedge\ y > x\ \wedge\ \mathtt{gets(y)}$

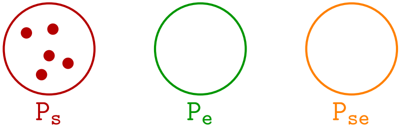
- $\mathsf{FO}[\sim]$

> *Every System agent requests at most twice a resource :*
>
> $\forall x, \mathrm{P_s}(x) \rightarrow \left[\forall y_1, y_2, y_3,\ \bigwedge_i \left(x \sim y_i \wedge \mathtt{req}(y_i)\right) \rightarrow \bigvee_{i \neq j} y_i = y_j\right]$

Agent control

We consider three configurations :

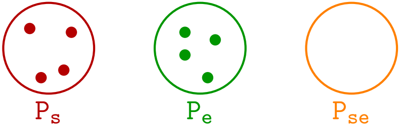
1. All the agents belong to System



Agent control

We consider three configurations :

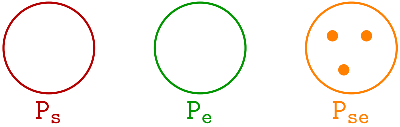
1. All the agents belong to System
2. There is no shared agent



Agent control

We consider three configurations :

1. All the agents belong to System
2. There is no shared agent
3. All the agents are shared by System and Environment



Synthesis problem

Parameters :

- a logic (specification language) $\mathcal{L}$
- a configuration for agent control (System only, partitioned or shared)

Synthesis problem

Parameters :

- a logic (specification language) $\mathcal{L}$
- a configuration for agent control (System only, partitioned or shared)

Synthesis problem for $\mathcal{L}$ for this configuration :

> **Input :** a formula $\varphi \in \mathcal{L}$
>
> **Question :** does there exist a distribution of agents, complying with the configuration, such that System has a winning strategy for φ ?

Filling the gaps

Logic\Agents	System only [a]	Partitioned	Shared
$FO^2[\sim]$	decidable[1]	?	?
$FO[\sim]$	decidable[2]	?	undecidable[2]
$FO^2[\sim, <]$	decidable[1]	?	?
$FO^2[\sim, +1]$	decidable[1]	?	?
$FO^2[\sim, <, +1]$	decidable[1]	?	undecidable[2]

1 : [Bojańczyk et al. '06]
2 : [Bérard et al. '20]

a. this amounts to the satisfiability problem

Filling the gaps

Logic\Agents	System only [a]	Partitioned	Shared
$FO^2[\sim]$	decidable[1]	?	**undecidable**
$FO[\sim]$	decidable[2]	**decidable**	undecidable[2]
$FO^2[\sim, <]$	decidable[1]	**undecidable**	?
$FO^2[\sim, +1]$	decidable[1]	**undecidable**	?
$FO^2[\sim, <, +1]$	decidable[1]	?	undecidable[2]

1 : [Bojańczyk et al. '06]
2 : [Bérard et al. '20]

a. this amounts to the satisfiability problem

Filling the gaps

Logic\Agents	System only [a]	Partitioned	Shared
$FO^2[\sim]$	decidable[1]	decidable	undecidable
$FO[\sim]$	decidable[2]	decidable	undecidable[2]
$FO^2[\sim, <]$	decidable[1]	undecidable	undecidable
$FO^2[\sim, +1]$	decidable[1]	undecidable	undecidable
$FO^2[\sim, <, +1]$	decidable[1]	undecidable	undecidable[2]

1 : [Bojańczyk et al. '06]
2 : [Bérard et al. '20]

---

a. this amounts to the satisfiability problem

Two-counter Minksy machine :

- a finite set of states $\mathcal{Q}$ with $q_0, q_h \in \mathcal{Q}$

Synthesis problem for $\mathrm{FO}^2[\sim, <]$ with partitioned agents

Two-counter Minksy machine :

- a finite set of states $\mathcal{Q}$ with $q_0, q_h \in \mathcal{Q}$
- two non-negative counters c_0 and c_1

Synthesis problem for $FO^2[\sim, <]$ with partitioned agents

Two-counter Minksy machine :

- a finite set of states $\mathcal{Q}$ with $q_0, q_h \in \mathcal{Q}$
- two non-negative counters c_0 and c_1
- a set $\mathcal{T}$ of transitions between two states either
  - increasing a counter
  - decreasing a counter
  - zero-testing a counter

Synthesis problem for $FO^2[\sim, <]$ with partitioned agents

Two-counter Minksy machine :

- a finite set of states $\mathcal{Q}$ with $q_0, q_h \in \mathcal{Q}$
- two non-negative counters c_0 and c_1
- a set $\mathcal{T}$ of transitions between two states either
  - increasing a counter
  - decreasing a counter
  - zero-testing a counter

Run : sequence of states linked by transitions that do not

- decrease a counter below zero
- use a zero-testing transition on a non-zero counter

Two-counter Minksy machine :

- a finite set of states $\mathcal{Q}$ with $q_0, q_h \in \mathcal{Q}$
- two non-negative counters c_0 and c_1
- a set $\mathcal{T}$ of transitions between two states either
  - increasing a counter
  - decreasing a counter
  - zero-testing a counter

Run : sequence of states linked by transitions that do not

- decrease a counter below zero
- use a zero-testing transition on a non-zero counter

Halting run : run starting in q_0 with zero counters, and ending in q_h

Synthesis problem for $\mathrm{FO}^2[\sim, <]$ with partitioned agents

Halting problem for two-counter Minsky machines :

Input : a two-counter Minsky machine M

Question : does M have a halting run ?

This problem is **undecidable** : we reduce it to the Synthesis problem for $\mathrm{FO}^2[\sim, <]$ with partitioned agents

Synthesis problem for $\mathrm{FO}^2[\sim, <]$ with partitioned agents

Halting problem for two-counter Minsky machines :

Input : a two-counter Minsky machine M

Question : does M have a halting run ?

This problem is **undecidable** : we reduce it to the Synthesis problem for $\mathrm{FO}^2[\sim, <]$ with partitioned agents

- System is in charge of the simulation

Synthesis problem for $FO^2[\sim, <]$ with partitioned agents

Halting problem for two-counter Minsky machines :

Input : a two-counter Minsky machine M

Question : does M have a halting run ?

This problem is **undecidable** : we reduce it to the Synthesis problem for $FO^2[\sim, <]$ with partitioned agents

- System is in charge of the simulation
- Environment can interrupt if System is cheating

Synthesis problem for $FO^2[\sim, <]$ with partitioned agents

Halting problem for two-counter Minsky machines :

Input : a two-counter Minsky machine M

Question : does M have a halting run ?

This problem is **undecidable** : we reduce it to the Synthesis problem for $FO^2[\sim, <]$ with partitioned agents

- System is in charge of the simulation
- Environment can interrupt if System is cheating
- The value of c_i is encoded as the number of System agents

$$\begin{cases} \text{who have played } \mathrm{inc}_i \\ \text{who have not played } \mathrm{dec}_i \end{cases}$$

Synthesis problem for $\mathrm{FO}^2[\sim, <]$ with partitioned agents

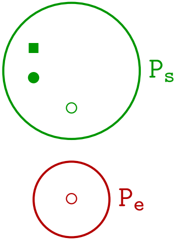
$\mathcal{Q} := \{q_0, q_1, q_2, q_h\}$ and $\mathcal{T} := \{t_0, t_1, t_2, t_3\}$, where $\begin{cases} t_0 : q_0 \xrightarrow{c_0++} q_0 \\ t_1 : q_0 \xrightarrow{c_0--} q_1 \\ t_2 : q_1 \xrightarrow{c_0--} q_2 \\ t_3 : q_2 \xrightarrow{c_0==0} q_h \end{cases}$

Synthesis problem for $\mathrm{FO}^2[\sim,<]$ with partitioned agents

$\mathcal{Q} := \{q_0, q_1, q_2, q_h\}$ and $\mathcal{T} := \{t_0, t_1, t_2, t_3\}$, where $\begin{cases} t_0 : q_0 \xrightarrow{c_0{+}{+}} q_0 \\ t_1 : q_0 \xrightarrow{c_0{-}{-}} q_1 \\ t_2 : q_1 \xrightarrow{c_0{-}{-}} q_2 \\ t_3 : q_2 \xrightarrow{c_0{=}{=}0} q_h \end{cases}$

q_0	$c_0 : 0$	$c_1 : 0$

$(\circ, \mathrm{ok}_S)(\circ, \mathrm{ok}_E)(\circ, q_0)$



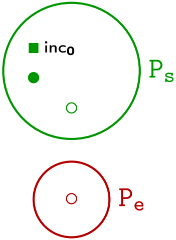
Synthesis problem for $\mathrm{FO}^2[\sim, <]$ with partitioned agents

$\mathcal{Q} := \{q_0, q_1, q_2, q_h\}$ and $\mathcal{T} := \{t_0, t_1, t_2, t_3\}$, where $\begin{cases} t_0 : q_0 \xrightarrow{c_0{+}{+}} q_0 \\ t_1 : q_0 \xrightarrow{c_0{-}{-}} q_1 \\ t_2 : q_1 \xrightarrow{c_0{-}{-}} q_2 \\ t_3 : q_2 \xrightarrow{c_0{=}{=}0} q_h \end{cases}$

$q_0 \xrightarrow{t_0} q_0$ $\qquad c_0 : 1 \qquad c_1 : 0$

$(\circ, \mathrm{ok}_S)(\circ, \mathrm{ok}_E)(\circ, q_0)(\circ, t_0)(\blacksquare, \mathrm{inc}_0)(\circ, \mathrm{ok}_S)(\circ, \mathrm{ok}_E)$

$(\circ, q_0)$



Synthesis problem for $\mathrm{FO}^2[\sim, <]$ with partitioned agents

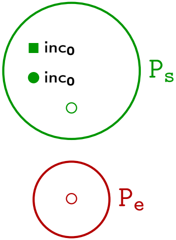
$\mathcal{Q} := \{q_0, q_1, q_2, q_h\}$ and $\mathcal{T} := \{t_0, t_1, t_2, t_3\}$, where $\begin{cases} t_0 : q_0 \xrightarrow{c_0++} q_0 \\ t_1 : q_0 \xrightarrow{c_0--} q_1 \\ t_2 : q_1 \xrightarrow{c_0--} q_2 \\ t_3 : q_2 \xrightarrow{c_0==0} q_h \end{cases}$

$q_0 \xrightarrow{t_0} q_0 \xrightarrow{t_0} q_0$ $\qquad c_0 : 2 \qquad c_1 : 0$

$(\circ, \mathrm{ok}_S)(\circ, \mathrm{ok}_E)(\circ, q_0)(\circ, t_0)(\blacksquare, \mathrm{inc}_0)(\circ, \mathrm{ok}_S)(\circ, \mathrm{ok}_E)$

$(\circ, q_0)(\circ, t_0)(\bullet, \mathrm{inc}_0)(\circ, \mathrm{ok}_S)(\circ, \mathrm{ok}_E)$

$(\circ, q_0)$

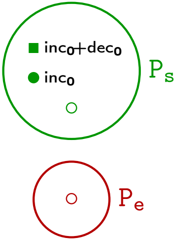


Synthesis problem for $\mathrm{FO}^2[\sim, <]$ with partitioned agents

$\mathcal{Q} := \{q_0, q_1, q_2, q_h\}$ and $\mathcal{T} := \{t_0, t_1, t_2, t_3\}$, where $\begin{cases} t_0 : q_0 \xrightarrow{c_0++} q_0 \\ t_1 : q_0 \xrightarrow{c_0--} q_1 \\ t_2 : q_1 \xrightarrow{c_0--} q_2 \\ t_3 : q_2 \xrightarrow{c_0==0} q_h \end{cases}$

$q_0 \xrightarrow{t_0} q_0 \xrightarrow{t_0} q_0 \xrightarrow{t_1} q_1$ $\quad c_0 : 1 \quad c_1 : 0$

$(\circ, \mathrm{ok}_S)(\circ, \mathrm{ok}_E)(\circ, q_0)(\circ, t_0)(\blacksquare, \mathrm{inc}_0)(\circ, \mathrm{ok}_S)(\circ, \mathrm{ok}_E)$
$(\circ, q_0)(\circ, t_0)(\bullet, \mathrm{inc}_0)(\circ, \mathrm{ok}_S)(\circ, \mathrm{ok}_E)$
$(\circ, q_0)(\circ, t_1)(\blacksquare, \mathrm{dec}_0)(\circ, \mathrm{ok}_S)(\circ, \mathrm{ok}_E)$
$(\circ, q_1)$

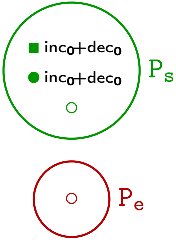


Synthesis problem for $FO^2[\sim, <]$ with partitioned agents

$\mathcal{Q} := \{q_0, q_1, q_2, q_h\}$ and $\mathcal{T} := \{t_0, t_1, t_2, t_3\}$, where $\begin{cases} t_0 : q_0 \xrightarrow{c_0++} q_0 \\ t_1 : q_0 \xrightarrow{c_0--} q_1 \\ t_2 : q_1 \xrightarrow{c_0--} q_2 \\ t_3 : q_2 \xrightarrow{c_0==0} q_h \end{cases}$

$$q_0 \xrightarrow{t_0} q_0 \xrightarrow{t_0} q_0 \xrightarrow{t_1} q_1 \xrightarrow{t_2} q_2 \qquad c_0 : 0 \qquad c_1 : 0$$

$$(\circ, \mathrm{ok}_S)(\circ, \mathrm{ok}_E)(\circ, q_0)(\circ, t_0)(\blacksquare, \mathrm{inc}_0)(\circ, \mathrm{ok}_S)(\circ, \mathrm{ok}_E)$$
$$(\circ, q_0)(\circ, t_0)(\bullet, \mathrm{inc}_0)(\circ, \mathrm{ok}_S)(\circ, \mathrm{ok}_E)$$
$$(\circ, q_0)(\circ, t_1)(\blacksquare, \mathrm{dec}_0)(\circ, \mathrm{ok}_S)(\circ, \mathrm{ok}_E)$$
$$(\circ, q_1)(\circ, t_2)(\bullet, \mathrm{dec}_0)(\circ, \mathrm{ok}_S)(\circ, \mathrm{ok}_E)$$
$$(\circ, q_2)$$



Synthesis problem for $\mathrm{FO}^2[\sim, <]$ with partitioned agents

$\mathcal{Q} := \{q_0, q_1, q_2, q_h\}$ and $\mathcal{T} := \{t_0, t_1, t_2, t_3\}$, where
$$\begin{cases} t_0 : q_0 \xrightarrow{c_0{+}{+}} q_0 \\ t_1 : q_0 \xrightarrow{c_0{-}{-}} q_1 \\ t_2 : q_1 \xrightarrow{c_0{-}{-}} q_2 \\ t_3 : q_2 \xrightarrow{c_0{=}{=}0} q_h \end{cases}$$

$$q_0 \xrightarrow{t_0} q_0 \xrightarrow{t_0} q_0 \xrightarrow{t_1} q_1 \xrightarrow{t_2} q_2 \xrightarrow{t_3} q_h \qquad c_0 : 0 \qquad c_1 : 0$$

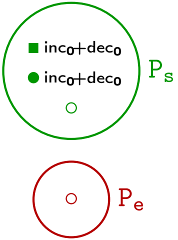
$(\circ, \mathrm{ok}_S)(\circ, \mathrm{ok}_E)(\circ, q_0)(\circ, t_0)(\blacksquare, \mathrm{inc}_0)(\circ, \mathrm{ok}_S)(\circ, \mathrm{ok}_E)$

$(\circ, q_0)(\circ, t_0)(\bullet, \mathrm{inc}_0)(\circ, \mathrm{ok}_S)(\circ, \mathrm{ok}_E)$

$(\circ, q_0)(\circ, t_1)(\blacksquare, \mathrm{dec}_0)(\circ, \mathrm{ok}_S)(\circ, \mathrm{ok}_E)$

$(\circ, q_1)(\circ, t_2)(\bullet, \mathrm{dec}_0)(\circ, \mathrm{ok}_S)(\circ, \mathrm{ok}_E)$

$(\circ, q_2)(\circ, t_3)(\circ, \mathrm{noop})(\circ, \mathrm{ok}_S)(\circ, \mathrm{ok}_E)(\circ, q_h)$



Decidability boundary

Logic\Agents	System only	Partitioned	Shared
$FO^2[\sim]$	decidable	**decidable**	**undecidable**
$FO[\sim]$	decidable	**decidable**	undecidable
$FO^2[\sim, <]$	decidable	**undecidable**	**undecidable**
$FO^2[\sim, +1]$	decidable	**undecidable**	**undecidable**
$FO^2[\sim, <, +1]$	decidable	**undecidable**	undecidable

Decidability boundary

Logic\Agents	Partitioned
$FO^2[\sim]$	decidable
$FO[\sim]$	decidable
$FO^2[\sim, <]$	undecidable
$FO^2[\sim, +1]$	undecidable
$FO^2[\sim, <, +1]$	undecidable

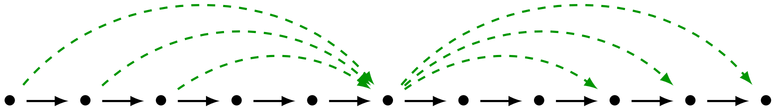
Decidability boundary

Logic\Agents	Partitioned
$FO^2[\sim]$	decidable
$FO[\sim]$	decidable
$FO^{pref}[\lesssim]$	?
$FO^2[\sim, <]$	undecidable
$FO^2[\sim, +1]$	undecidable
$FO^2[\sim, <, +1]$	undecidable

Prefix first-order logic on *words* : $\mathsf{FO}^{\mathsf{pref}}$

Definition ($\mathsf{FO}^{\mathsf{pref}}$)

As $\mathsf{FO}[<]$ on words, where...

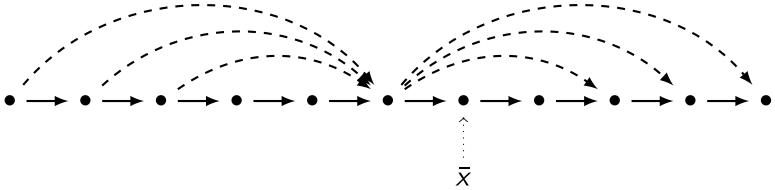


Prefix first-order logic on *words* : FO^{pref}

Definition (FO^{pref})

As FO[<] on words, where...

- first quantifier : $\forall \bar{x}$ or $\exists \bar{x}$

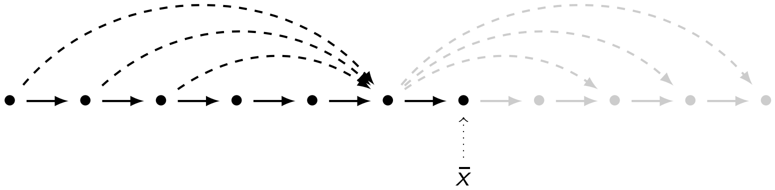


Prefix first-order logic on *words* : $\mathrm{FO}^{\mathrm{pref}}$

Definition ($\mathrm{FO}^{\mathrm{pref}}$)

As FO[<] on words, where...

- first quantifier : $\forall \bar{x}$ or $\exists \bar{x}$
- following quantifiers : $\forall x < \bar{x}$ or $\exists x < \bar{x}$



Prefix first-order logic on *words* : $\mathrm{FO}^{\mathrm{pref}}$

Definition ($\mathrm{FO}^{\mathrm{pref}}$)

As FO[$<$] on words, where...

- first quantifier : $\forall \bar{x}$ or $\exists \bar{x}$
- following quantifiers : $\forall x < \bar{x}$ or $\exists x < \bar{x}$

Some factor never appears before some other factor

Prefix first-order logic on *words* : FO^{pref}

Definition (FO^{pref})

As $FO[<]$ on words, where...

- first quantifier : $\forall \bar{x}$ or $\exists \bar{x}$
- following quantifiers : $\forall x < \bar{x}$ or $\exists x < \bar{x}$

Some factor never appears before some other factor

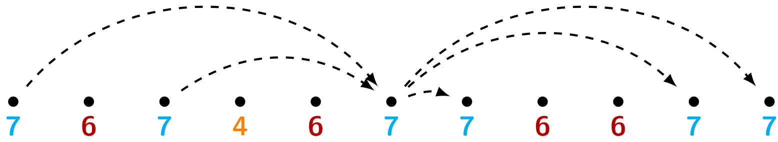
Not expressible in FO^{pref} :

There is an infinite number of a

Prefix first-order logic on *data words* : $FO^{pref}[\lesssim]$

Definition ($FO^{pref}[\lesssim]$)

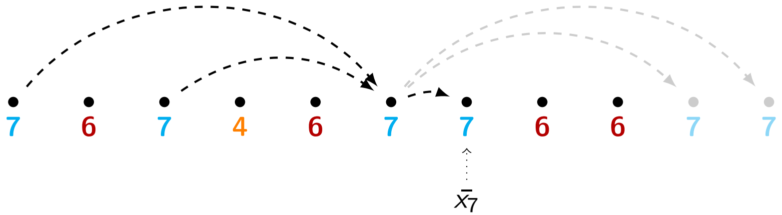
$FO^{pref}[\lesssim]$: FO^{pref} independently on each data class



Prefix first-order logic on *data words* : $FO^{pref}[\lesssim]$

Definition ($FO^{pref}[\lesssim]$)

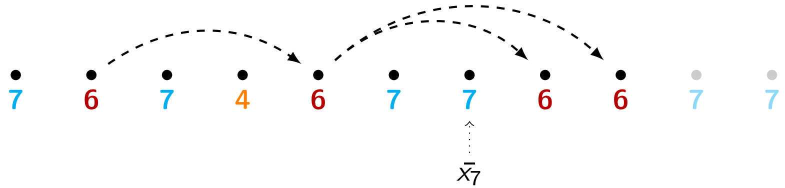
$FO^{pref}[\lesssim]$: FO^{pref} independently on each data class



Prefix first-order logic on *data words* : $FO^{pref}[\lesssim]$

Definition ($FO^{pref}[\lesssim]$)

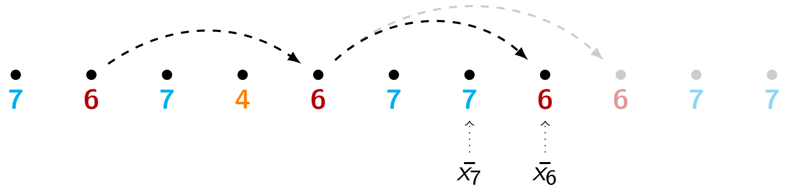
$FO^{pref}[\lesssim]$: FO^{pref} independently on each data class



Prefix first-order logic on *data words* : $\mathrm{FO}^{\mathrm{pref}}[\lesssim]$

Definition ($\mathrm{FO}^{\mathrm{pref}}[\lesssim]$)

$\mathrm{FO}^{\mathrm{pref}}[\lesssim]$: $\mathrm{FO}^{\mathrm{pref}}$ independently on each data class



Prefix first-order logic on *data words* : $\mathrm{FO}^{\mathrm{pref}}[\lesssim]$

Definition ($\mathrm{FO}^{\mathrm{pref}}[\lesssim]$)

$\mathrm{FO}^{\mathrm{pref}}[\lesssim]$: $\mathrm{FO}^{\mathrm{pref}}$ independently on each data class

An agent only closes a resource they opened and did not already close :

Prefix first-order logic on *data words* : $\mathrm{FO}^{\mathrm{pref}}[\lesssim]$

Definition ($\mathrm{FO}^{\mathrm{pref}}[\lesssim]$)

$\mathrm{FO}^{\mathrm{pref}}[\lesssim]$: $\mathrm{FO}^{\mathrm{pref}}$ independently on each data class

An agent only closes a resource they opened and did not already close :

$\forall \bar{x}, \mathrm{close}(\bar{x}) \rightarrow \big(\exists x \lesssim \bar{x}, \mathrm{open}(x) \ \wedge\ \forall y \lesssim \bar{x}, x \lesssim y \rightarrow \neg\mathrm{close}(y)\big)$

Prefix first-order logic on *data words* : $\mathrm{FO}^{\mathrm{pref}}[\lesssim]$

Definition ($\mathrm{FO}^{\mathrm{pref}}[\lesssim]$)

$\mathrm{FO}^{\mathrm{pref}}[\lesssim]$: $\mathrm{FO}^{\mathrm{pref}}$ independently on each data class

An agent only closes a resource they opened and did not already close :

$$\forall \bar{x}, \mathrm{close}(\bar{x}) \rightarrow \big(\exists x \lesssim \bar{x}, \mathrm{open}(x) \ \wedge \ \forall y \lesssim \bar{x}, x \lesssim y \rightarrow \neg\mathrm{close}(y)\big)$$

Not expressible in $\mathrm{FO}^{\mathrm{pref}}[\lesssim]$:

Two agents never have the same resource open simultaneously

An agent always ends up closing an open resource

Synthesis problem for $\mathsf{FO}^{\mathsf{pref}}[\lesssim]$

Input : a formula $\varphi \in \mathsf{FO}^{\mathsf{pref}}[\lesssim]$

Question : does there exist a distribution of partitioned agents such that System has a winning strategy for φ ?

Synthesis problem for $\mathsf{FO}^{\mathsf{pref}}[\lesssim]$

Input : a formula $\varphi \in \mathsf{FO}^{\mathsf{pref}}[\lesssim]$

Question : does there exist a distribution of partitioned agents such that System has a winning strategy for φ?

Theorem

The synthesis problem for $\mathsf{FO}^{\mathsf{pref}}[\lesssim]$ *with partitioned agents is decidable*

Synthesis problem for $\mathrm{FO}^{\mathrm{pref}}[\lesssim]$

Input : a formula $\varphi \in \mathrm{FO}^{\mathrm{pref}}[\lesssim]$

Question : does there exist a distribution of partitioned agents such that System has a winning strategy for φ?

Theorem

The synthesis problem for $\mathrm{FO}^{\mathrm{pref}}[\lesssim]$ *with partitioned agents is decidable*

Sketch of proof :

1. normalize the game (strict alternation between players)

Synthesis problem for $\mathrm{FO}^{\mathrm{pref}}[\lesssim]$

Input : a formula $\varphi \in \mathrm{FO}^{\mathrm{pref}}[\lesssim]$

Question : does there exist a distribution of partitioned agents such that System has a winning strategy for φ ?

Theorem

The synthesis problem for $\mathrm{FO}^{\mathrm{pref}}[\lesssim]$ *with partitioned agents is decidable*

Sketch of proof :

1. normalize the game
2. convert it to a token game

Synthesis problem for $\mathsf{FO}^{\mathsf{pref}}[\lesssim]$

Input : a formula $\varphi \in \mathsf{FO}^{\mathsf{pref}}[\lesssim]$

Question : does there exist a distribution of partitioned agents such that System has a winning strategy for φ ?

Theorem

The synthesis problem for $\mathsf{FO}^{\mathsf{pref}}[\lesssim]$ *with partitioned agents is decidable*

Sketch of proof :

1. normalize the game
2. convert it to a token game
3. solve the token game (by showing it admits some kind of cutoff)

Synthesis problem for $\mathrm{FO}^{\mathrm{pref}}[\lesssim]$

Definition ($\mathrm{FO}^{\mathrm{pref}}$ type of a word w)

Set of sentences of $\mathrm{FO}^{\mathrm{pref}}$

- with as many nested quantifiers as φ
- satisfied by w

Synthesis problem for $\mathrm{FO}^{\mathrm{pref}}[\lesssim]$

Definition ($\mathrm{FO}^{\mathrm{pref}}$ type of a word w)

Set of sentences of $\mathrm{FO}^{\mathrm{pref}}$

- with as many nested quantifiers as φ
- satisfied by w

$\mathrm{FO}^{\mathrm{pref}}$ types are stationary :

Lemma

For every infinite word w, there exists $i \in \mathbb{N}$ such that for every $j \geq i$, w and $w[1 \ldots j]$ have the same $\mathrm{FO}^{\mathrm{pref}}$ type

Synthesis problem for $FO^{pref}[\lesssim]$

Definition (FO^{pref} type of a word w)

Set of sentences of FO^{pref}

- with as many nested quantifiers as φ
- satisfied by w

Conversion to token game :

Arena : set of FO^{pref} types (finite)

Synthesis problem for $FO^{pref}[\lesssim]$

Definition (FO^{pref} type of a word w)

Set of sentences of FO^{pref}

- with as many nested quantifiers as φ
- satisfied by w

Conversion to token game :

Arena : set of FO^{pref} types (finite)

Tokens : one per agent, starting in the FO^{pref} type of ε

Synthesis problem for $\mathrm{FO}^{\mathrm{pref}}[\lesssim]$

Definition ($\mathrm{FO}^{\mathrm{pref}}$ type of a word w)

Set of sentences of $\mathrm{FO}^{\mathrm{pref}}$

- with as many nested quantifiers as φ
- satisfied by w

Conversion to token game :

Arena : set of $\mathrm{FO}^{\mathrm{pref}}$ types (finite)

Tokens : one per agent, starting in the $\mathrm{FO}^{\mathrm{pref}}$ type of ε

Transitions : $\tau \to \tau'$ iff there exists a finite word w of type τ,

there exists u such that wu has type τ'

Synthesis problem for $\mathrm{FO}^{\mathrm{pref}}[\lesssim]$

Definition ($\mathrm{FO}^{\mathrm{pref}}$ type of a word w)

Set of sentences of $\mathrm{FO}^{\mathrm{pref}}$

- with as many nested quantifiers as φ
- satisfied by w

Conversion to token game :

Arena : set of $\mathrm{FO}^{\mathrm{pref}}$ types (finite)

Tokens : one per agent, starting in the $\mathrm{FO}^{\mathrm{pref}}$ type of ε

Transitions : $\tau \to \tau'$ iff $\underbrace{\text{there exists a}}_{\text{equiv., for all}}$ finite word w of type τ,

there exists u such that wu has type τ'

Synthesis problem for $FO^{pref}[\lesssim]$

Definition (FO^{pref} type of a word w)

Set of sentences of FO^{pref}

- with as many nested quantifiers as φ
- satisfied by w

Conversion to token game :

Arena : set of FO^{pref} types (finite)

Tokens : one per agent, starting in the FO^{pref} type of ε

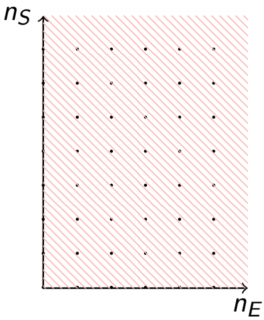
Transitions : $\tau \to \tau'$ iff $\underbrace{\text{there exists a}}_{\text{equiv., for all}}$ finite word w of type τ,

there exists u such that wu has type τ'

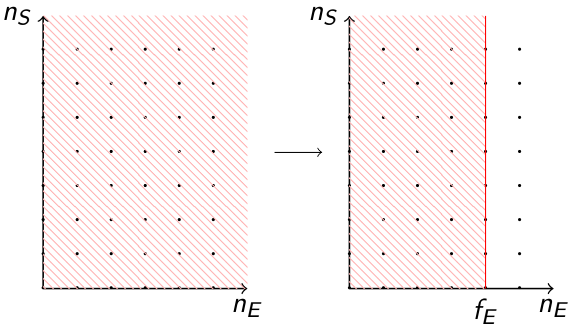
Win config : set of configurations,

with token counting up to the quantifier nesting of φ

Synthesis problem for $\mathrm{FO}^{\mathrm{pref}}[\lesssim]$



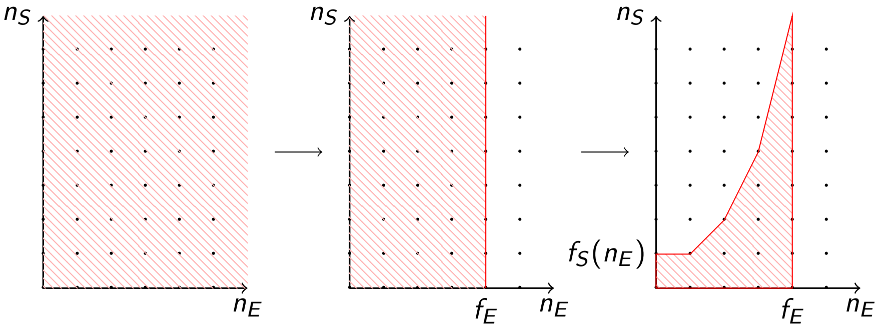
Synthesis problem for $\mathrm{FO}^{\mathrm{pref}}[\lesssim]$



Lemma

Beyond some threshold f_E, if Environment can win with some number of tokens, they can win with a larger number of tokens

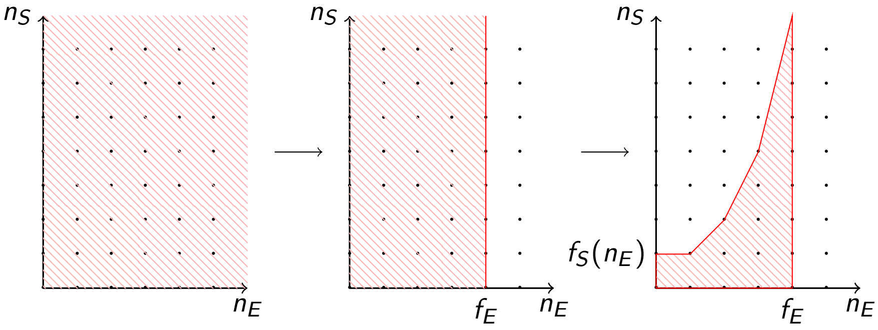
Synthesis problem for $FO^{pref}[\lesssim]$



Lemma

There exists $f_S : \mathbb{N} \to \mathbb{N}$ such that for every $n_E \in \mathbb{N}$,
if System can win with $> f_S(n_E)$ tokens when Environment has n_E tokens,
then System can already win with $f_S(n_E)$ tokens

Synthesis problem for $\mathrm{FO}^{\mathrm{pref}}[\lesssim]$



Lemma

For fixed $n_S, n_E \in \mathbb{N}$, one can decide whether System can win with n_S tokens when Environment has n_E tokens

Conclusion

Logic\Agents	System only	Partitioned	Shared
$FO^2[\sim]$	decidable	**decidable**	**undecidable**
$FO[\sim]$	decidable	**decidable**	undecidable
$FO^{pref}[\lesssim]$	**decidable**	**decidable**	**undecidable**
$FO^2[\sim, <]$	decidable	**undecidable**	**undecidable**
$FO^2[\sim, +1]$	decidable	**undecidable**	**undecidable**
$FO^2[\sim, <, +1]$	decidable	**undecidable**	undecidable

Conclusion

Logic\Agents	System only	Partitioned	Shared
$FO^2[\sim]$	decidable	**decidable**	**undecidable**
$FO[\sim]$	decidable	**decidable**	undecidable
$FO^{pref}[\lesssim]$	**decidable**	**decidable**	**undecidable**
$FO^2[\sim, <]$	decidable	**undecidable**	**undecidable**
$FO^2[\sim, +1]$	decidable	**undecidable**	**undecidable**
$FO^2[\sim, <, +1]$	decidable	**undecidable**	undecidable

Conjecture

The synthesis problem for $FO^2[\lesssim]$ *with partitioned agents is decidable*

Conclusion

Logic\Agents	System only	Partitioned	Shared
$FO^2[\sim]$	decidable	**decidable**	**undecidable**
$FO[\sim]$	decidable	**decidable**	undecidable
$FO^{pref}[\lesssim]$	**decidable**	**decidable**	**undecidable**
$FO^2[\sim, <]$	decidable	**undecidable**	**undecidable**
$FO^2[\sim, +1]$	decidable	**undecidable**	**undecidable**
$FO^2[\sim, <, +1]$	decidable	**undecidable**	undecidable

Conjecture

The synthesis problem for $FO^2[\lesssim]$ *with partitioned agents is decidable*

We considered a centralized strategy. What about distributed strategies ?