Synthesis with Binary Predicates

Pietro Sala
Department of Computer Science,
University of Verona

Department of Computer science, Mathematics and Physics, University of Udine

07/02/2020

Udine (Italy)

OUTLINE

A short recall of classical synthesis problem

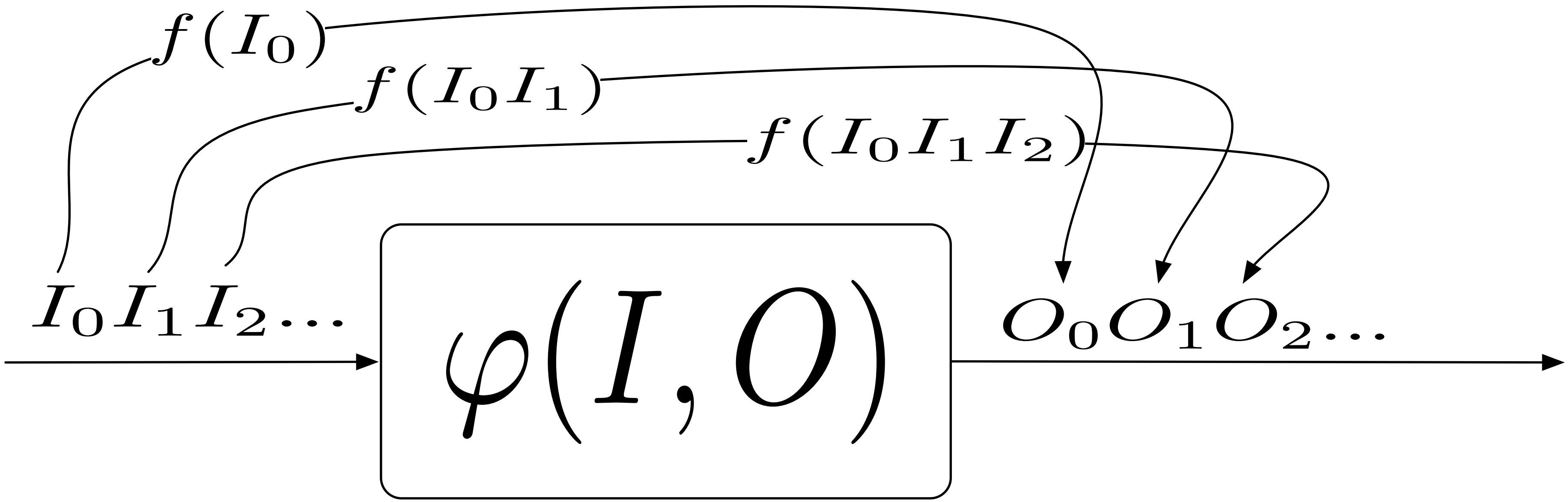
Introduction of synthesis with binary predicates

Expressiveness of synthesis with binary predicates: examples

Results and Open Problems

Classical Synthesis Problem

Classical Synthesis Problem



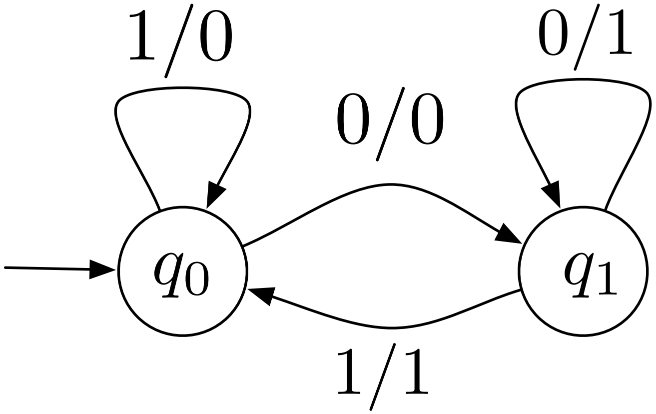
$$\begin{array}{c} I \\ O \end{array} \begin{pmatrix} I_0 \\ O_0 \end{pmatrix} \begin{pmatrix} I_1 \\ O_1 \end{pmatrix} \begin{pmatrix} I_2 \\ O_2 \end{pmatrix} \cdots \begin{pmatrix} I_t \\ O_t \end{pmatrix} \cdots \models \varphi(I,O)$$

Example

$$\begin{array}{c} I \\ O \end{array} \begin{pmatrix} 0 \\ 0 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \ldots \models \forall x(x \notin I \leftrightarrow \exists y(next(x,y) \wedge y \in O))$$

$0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5$

Classical Synthesis Problem



Mealy Automaton

$$(Q, q_0, \Sigma_I, \Sigma_O, \delta, \delta_{IO})$$

$$\delta : Q \times \Sigma_I \to Q \qquad \delta_{IO} : Q \times \Sigma_I \to \Sigma_O$$

$$MSO_0 \quad \varphi ::= X \subseteq Y \mid next(X, Y) \mid \|X\|_1 \mid \neg\varphi \mid \varphi \vee \varphi \mid \exists X \varphi$$

$Shorthands:$

$$\psi_1 \wedge \psi_2 ::= \neg(\neg\psi_1 \vee \neg\psi_2)$$

$$\psi_1 \to \psi_2 ::= \neg\psi_1 \vee \psi_2$$

$$\psi_1 \leftrightarrow \psi_2 ::= \psi_1 \to \psi_2 \wedge \psi_2 \to \psi_1$$

$$\forall X \varphi ::= \neg\exists X \neg\varphi$$

$$\exists x \varphi ::= \exists X(\|X\|_1 \wedge \varphi[x/X])$$

$$\forall x \varphi ::= \forall X(\|X\|_1 \to \varphi[x/X])$$

$$x \in S ::= \|X\|_1 \wedge X \subseteq S$$

$$x = y ::= \|X\|_1 \wedge X \subseteq Y \wedge Y \subseteq X$$

$$x < y ::= \exists S \left(\begin{array}{c} x \in S \wedge y \in S \wedge \forall w(next(w, x) \to w \notin S) \\ \wedge \forall z(z \in S \to \exists w(next(z, w) \wedge w \in S)) \end{array} \right)$$

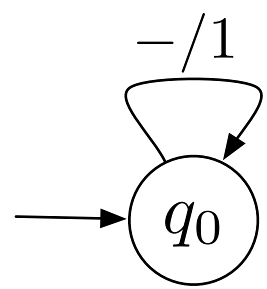
Classical Synthesis Problem

(R1) “every 1 on the input bit at position t will be matched by a 1 on the output bits between position t +1 and position t+ 5, and we cannot have a 1 on the output bits at position t which is not matched by a 1 on the input bits between positions t - 5 and t - 1“

REMARK

“every 1 on the input bit at position t it be produced produced a 1 on the output bits between position t +1 and position t + 5”

may solved using a Mealy automaton which is blind to its input



Classical Synthesis Problem

(R1) "every 1 on the input bit at position t will be matched by a 1 on the output bits between position t +1 and position t+ 5, and we cannot have a 1 on the output bits at position t > 0 which is not matched by a 1 on the input bits between positions t - 5 and t - 1"

$$\forall x_0 \left(x_0 \in I \rightarrow \exists x_1 \dots x_5 \left(\bigwedge_{i=0}^{4} next(x_i, x_{i+1}) \wedge \bigvee_{i=1}^{5} x_i \in O \right)\right)$$

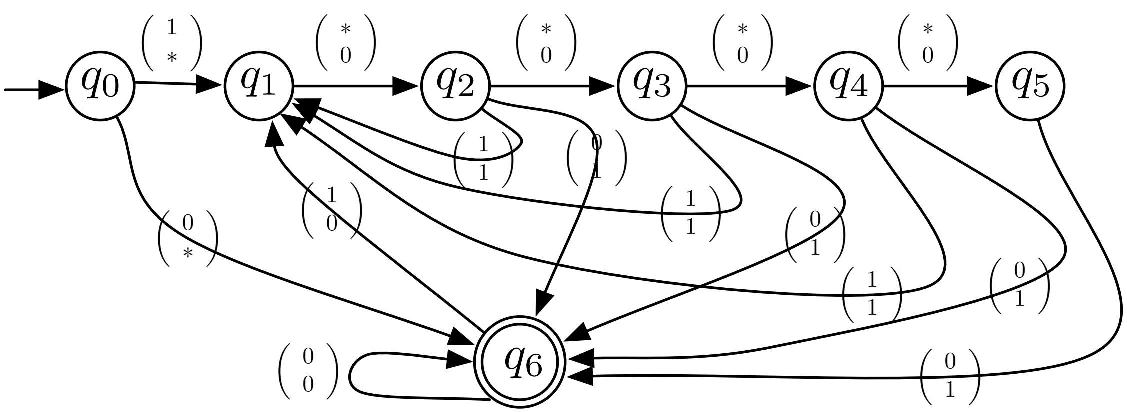
$$\forall x_5 \left(x_5 \in I \wedge x_5 > 0 \rightarrow \exists x_1 \dots x_4 \left(next(x_4, x_5) \wedge \bigwedge_{i=0}^{4} (x_i \leq x_{i+1}) \wedge \bigvee_{i=0}^{4} x_i \in O \right)\right)$$

Classical Synthesis Problem

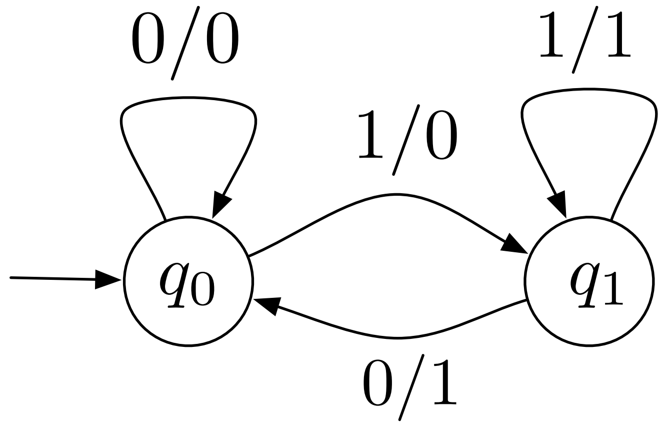
$$\forall x_0 \left(x_0 \in I \rightarrow \exists x_1 \ldots x_5 \left(\bigwedge_{i=0}^{4} next(x_i, x_{i+1}) \wedge \bigvee_{i=1}^{5} x_i \in O \right) \right)$$

$$\forall x_5 \left(x_5 \in I \wedge x_5 > 0 \rightarrow \exists x_1 \ldots x_4 \left(next(x_4, x_5) \wedge \bigwedge_{i=0}^{4} (x_i \leq x_{i+1}) \wedge \bigvee_{i=0}^{4} x_i \in O \right) \right)$$

The Büchi Automaton of the relative language



A Mealy Automaton that realises such specification.

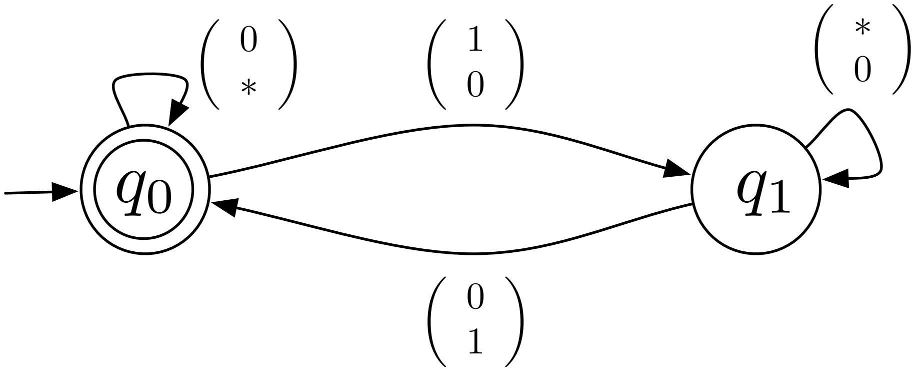


Classical Synthesis Problem

(R2) "every 1 on the input bit at position t must be followed by a 1 on the output bit but we cannot have 1 the input and 1 on the output at the same time"

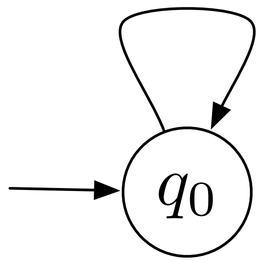
$$\forall x \left(x \in I \rightarrow \begin{array}{c} x \notin O \wedge \\ \exists z (x < z \wedge z \in O) \end{array} \right)$$

(R2) all its models are generated by the language accepted by the following **Büchi** automaton:



but (R2) is a negative instance of the synthesis problem

$$I = * / O = 1$$



Synthesis with Binary Predicates

Synthesis with Binary Predicates

In the classical (monadic) setting:

Specification	Successful Plays	Strategy
MSO formula	Accepting Traces of Büchi Automaton	Mealy Automaton

In the binary setting:

Specification	Successful Plays	Strategy
FO formula with binary predicates	Interval Models	Trees
Interval Temporal Logic (HS) Formula		

The input of the problem:

FO or HS formula + Partition of predicates/propositional variables

$$(\varphi, \mathcal{P}_E, \mathcal{P}_S), \mathcal{P}(\varphi) = \mathcal{P}_E \cup \mathcal{P}_S, \mathcal{P}_E \cap \mathcal{P}_S = \emptyset$$

Plays are done as follows:

at each round t the environment provides the values for the variables/predicates in $\mathcal{P}_E$ for all the intervals ending in t, namely, $[0, t], \ldots, [t-1, t], [t, t]$.

the system complete the partial labeling provided by the environment for the variables/predicates in $\mathcal{P}_S$ for all the intervals ending in t, namely, $[0, t], \ldots, [t-1, t], [t, t]$.

Example

$$\mathcal{P}_E = \{p\}, \mathcal{P}_S = \{q\}$$

$\circ$

0

Interval Based Synthesis

Plays are done as follows:

at each round t the environment provides the values for the variables/predicates in $\mathcal{P}_E$ for all the intervals ending in t, namely, $[0,t],\ldots,[t-1,t],[t,t]$.

the system complete the partial labeling provided by the environment for the variables/predicates in $\mathcal{P}_S$ for all the intervals ending in t, namely, $[0,t],\ldots,[t-1,t],[t,t]$.

Example

$$\mathcal{P}_E = \{p\}, \mathcal{P}_S = \{q\}$$

$\{p\}$

$\circ$

0

Interval Based Synthesis

Plays are done as follows:

at each round t the environment provides the values for the variables/predicates in $\mathcal{P}_E$ for all the intervals ending in t, namely, $[0,t],\ldots,[t-1,t],[t,t]$.

the system complete the partial labeling provided by the environment for the variables/predicates in $\mathcal{P}_S$ for all the intervals ending in t, namely, $[0,t],\ldots,[t-1,t],[t,t]$.

Example

$$\mathcal{P}_E = \{p\}, \mathcal{P}_S = \{q\}$$

$\{p\}$ $\{\}$

$\circ$

0

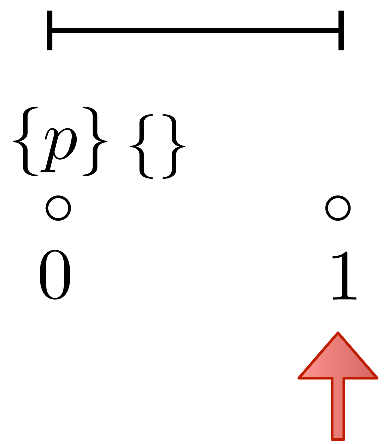
Interval Based Synthesis

Plays are done as follows:

at each round t the environment provides the values for the variables/predicates in $\mathcal{P}_E$ for all the intervals ending in t, namely, $[0,t], \ldots, [t-1,t], [t,t]$.

the system complete the partial labeling provided by the environment for the variables/predicates in $\mathcal{P}_S$ for all the intervals ending in t, namely, $[0,t], \ldots, [t-1,t], [t,t]$.

Example $$\mathcal{P}_E = \{p\}, \mathcal{P}_S = \{q\}$$



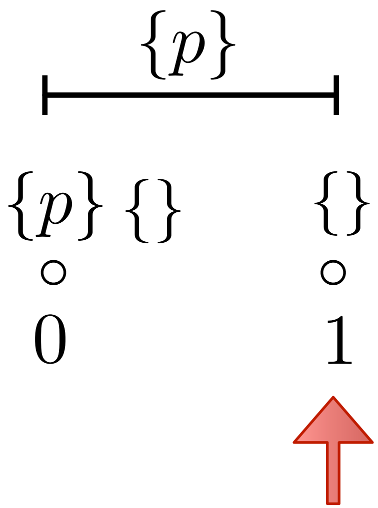
Interval Based Synthesis

at each round t the environment provides the values for the variables/predicates in $\mathcal{P}_E$ for all the intervals ending in t, namely, $[0,t],\ldots,[t-1,t],[t,t]$.

the system complete the partial labeling provided by the environment for the variables/predicates in $\mathcal{P}_S$ for all the intervals ending in t, namely, $[0,t],\ldots,[t-1,t],[t,t]$.

Example

$$\mathcal{P}_E = \{p\}, \mathcal{P}_S = \{q\}$$



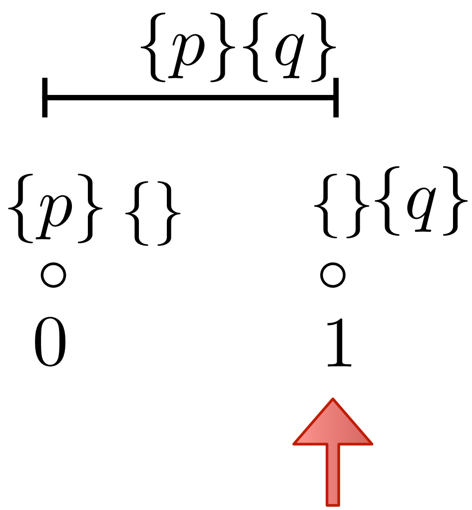
Interval Based Synthesis

at each round t the environment provides the values for the variables/predicates in $\mathcal{P}_E$ for all the intervals ending in t, namely, $[0,t], \ldots, [t-1,t], [t,t]$.

the system complete the partial labeling provided by the environment for the variables/predicates in $\mathcal{P}_S$ for all the intervals ending in t, namely, $[0,t], \ldots, [t-1,t], [t,t]$.

Example

$$\mathcal{P}_E = \{p\}, \mathcal{P}_S = \{q\}$$

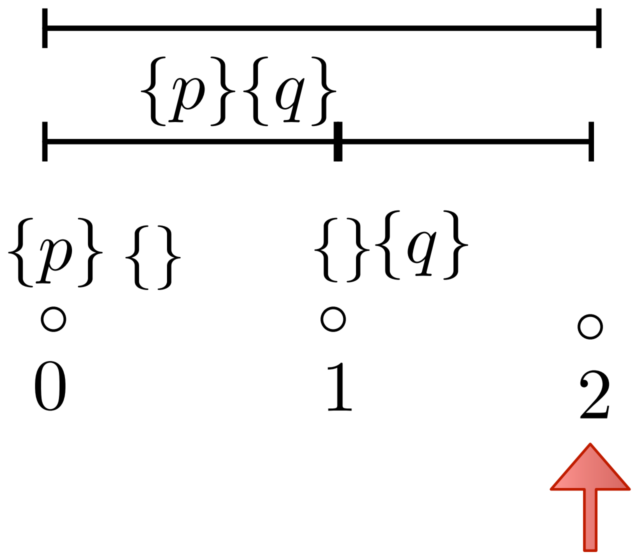


at each round t the environment provides the values for the variables/predicates in $\mathcal{P}_E$ for all the intervals ending in t, namely, $[0,t], \ldots, [t-1,t], [t,t]$.

the system complete the partial labeling provided by the environment for the variables/predicates in $\mathcal{P}_S$ for all the intervals ending in t, namely, $[0,t], \ldots, [t-1,t], [t,t]$.

Example

$$\mathcal{P}_E = \{p\}, \mathcal{P}_S = \{q\}$$

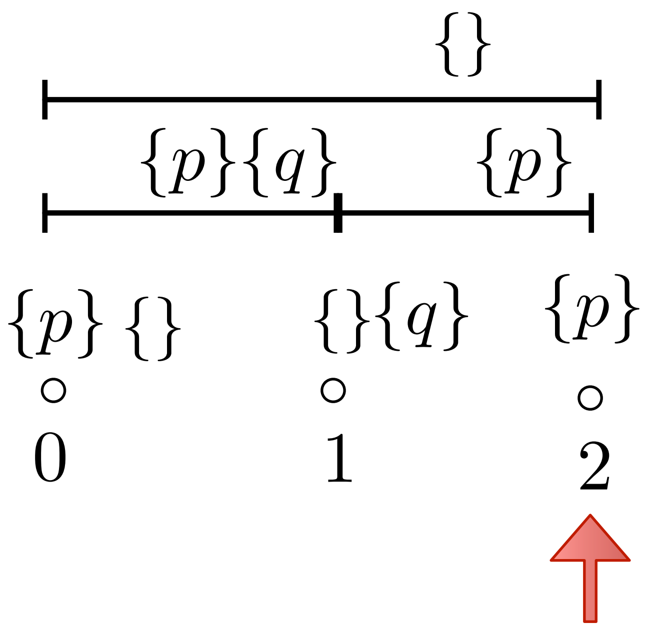


at each round t the environment provides the values for the variables/predicates in $\mathcal{P}_E$ for all the intervals ending in t, namely, $[0,t],\ldots,[t-1,t],[t,t]$.

the system complete the partial labeling provided by the environment for the variables/predicates in $\mathcal{P}_S$ for all the intervals ending in t, namely, $[0,t],\ldots,[t-1,t],[t,t]$.

Example

$$\mathcal{P}_E = \{p\}, \mathcal{P}_S = \{q\}$$

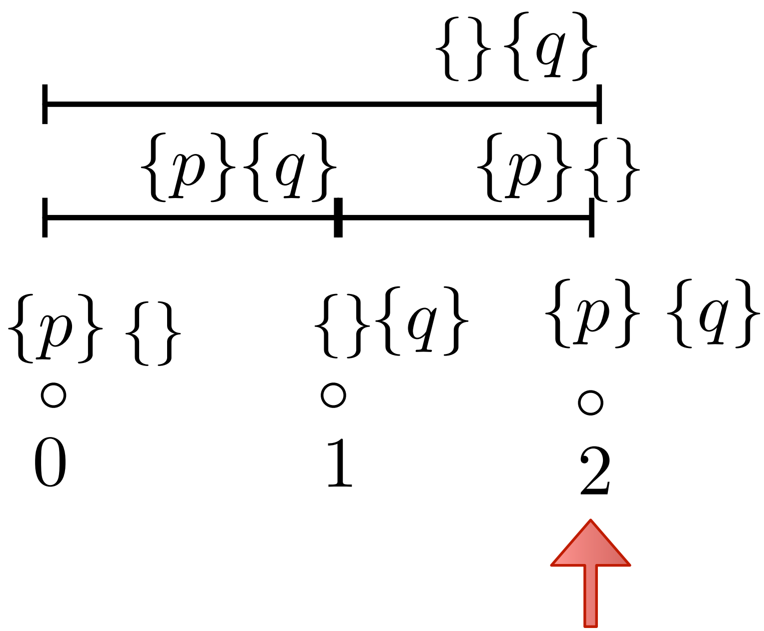


Interval Based Synthesis

at each round t the environment provides the values for the variables/predicates in $\mathcal{P}_E$ for all the intervals ending in t, namely, $[0,t],\ldots,[t-1,t],[t,t]$.

the system complete the partial labeling provided by the environment for the variables/predicates in $\mathcal{P}_S$ for all the intervals ending in t, namely, $[0,t],\ldots,[t-1,t],[t,t]$.

Example $$\mathcal{P}_E = \{p\}, \mathcal{P}_S = \{q\}$$



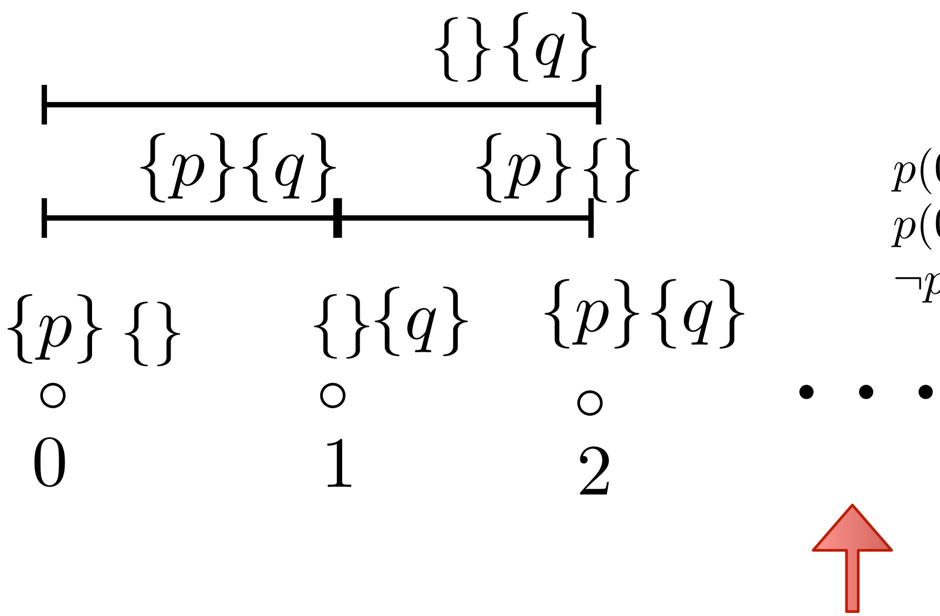
A Synthesis Formulation for HS

at each round t the environment provides the values for the variables/predicates in $\mathcal{P}_E$ for all the intervals ending in t, namely, $[0, t], \ldots, [t-1, t], [t, t]$.

the system complete the partial labeling provided by the environment for the variables/predicates in $\mathcal{P}_S$ for all the intervals ending in t, namely, $[0, t], \ldots, [t-1, t], [t, t]$.

Example

$$\mathcal{P}_E = \{p\}, \mathcal{P}_S = \{q\}$$



$p(0,0), \neg q(0,0)$
$p(0,1), q(0,1), \neg p(1,1), q(1,1)$
$\neg p(0,2), q(0,2), p(1,2), \neg q(1,2), p(2,2), q(2,2)$

Strategies with Binary Predicates

$$(\varphi, \mathcal{P}_E, \mathcal{P}_S), \mathcal{P}(\varphi) = \mathcal{P}_E \cup \mathcal{P}_S, \mathcal{P}_E \cap \mathcal{P}_S = \emptyset$$

A t-system-move is a function $\{0 \ldots t\} \to 2^{\mathcal{P}_S}$.

A t-environment-move is a function $\{0 \ldots t\} \to 2^{\mathcal{P}_E}$.

A system-strategy-tree is an infinite tree where:

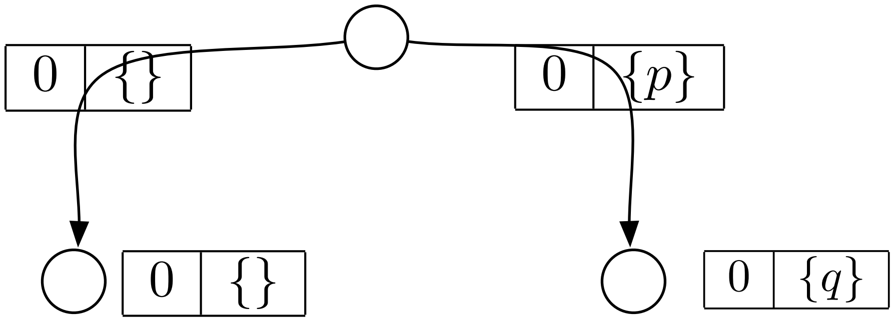
- the root is empty;
- every node at level t is labelled with a $(t-1)$-system-move and has many successors as the possible t-environment moves;
- every edge departing from a node at level t is labelled with a t-environment-move and every pair of distinct edges departing from the same node feature different t-environment-moves.

A Synthesis Formulation for HS

$\mathcal{P}_E = \{p\}, \mathcal{P}_S = \{q\}$

A Synthesis Formulation for HS

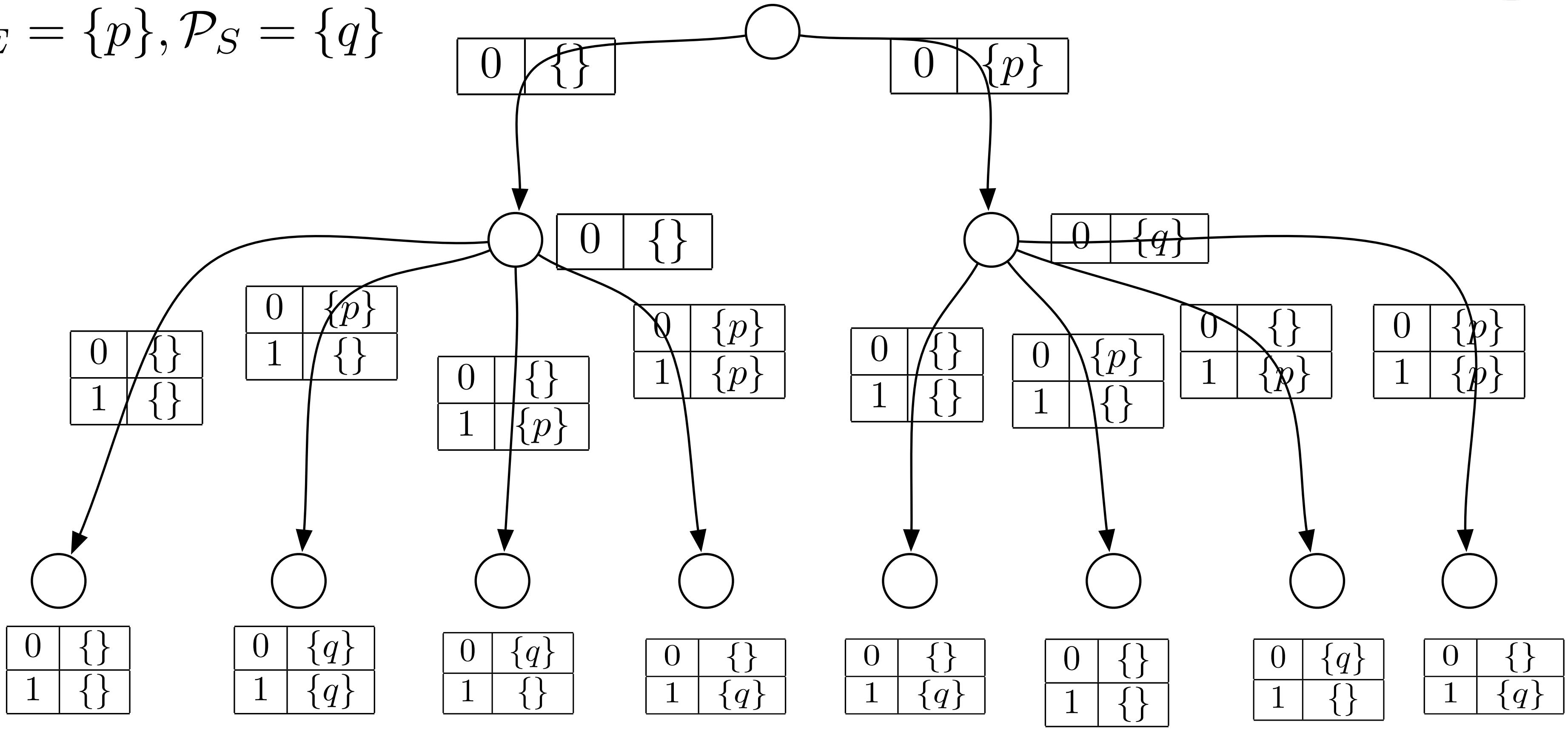
$\mathcal{P}_E = \{p\}, \mathcal{P}_S = \{q\}$



Interval Based Synthesis

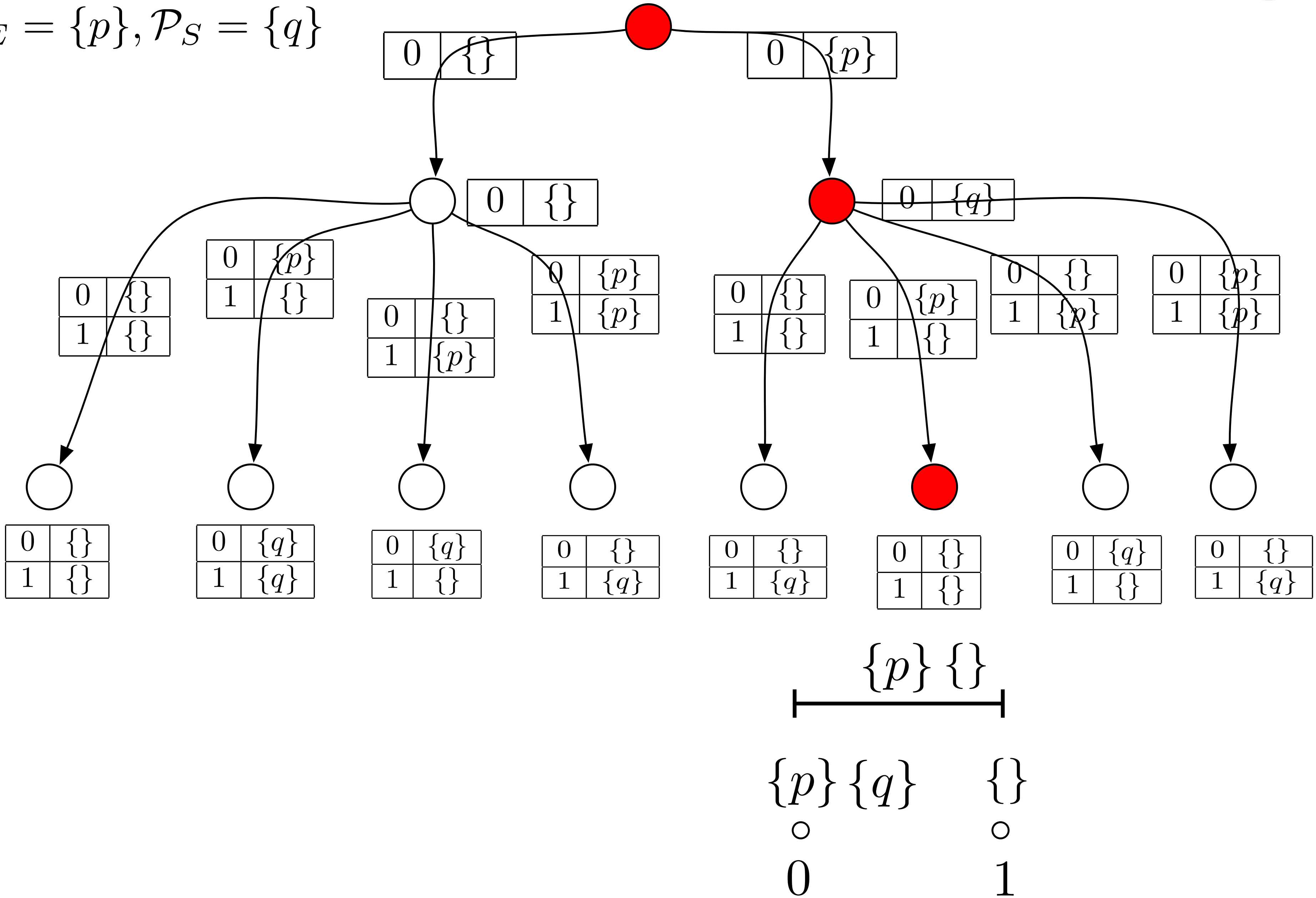
A Synthesis Formulation for HS

$$\mathcal{P}_E = \{p\}, \mathcal{P}_S = \{q\}$$



A Synthesis Formulation for HS

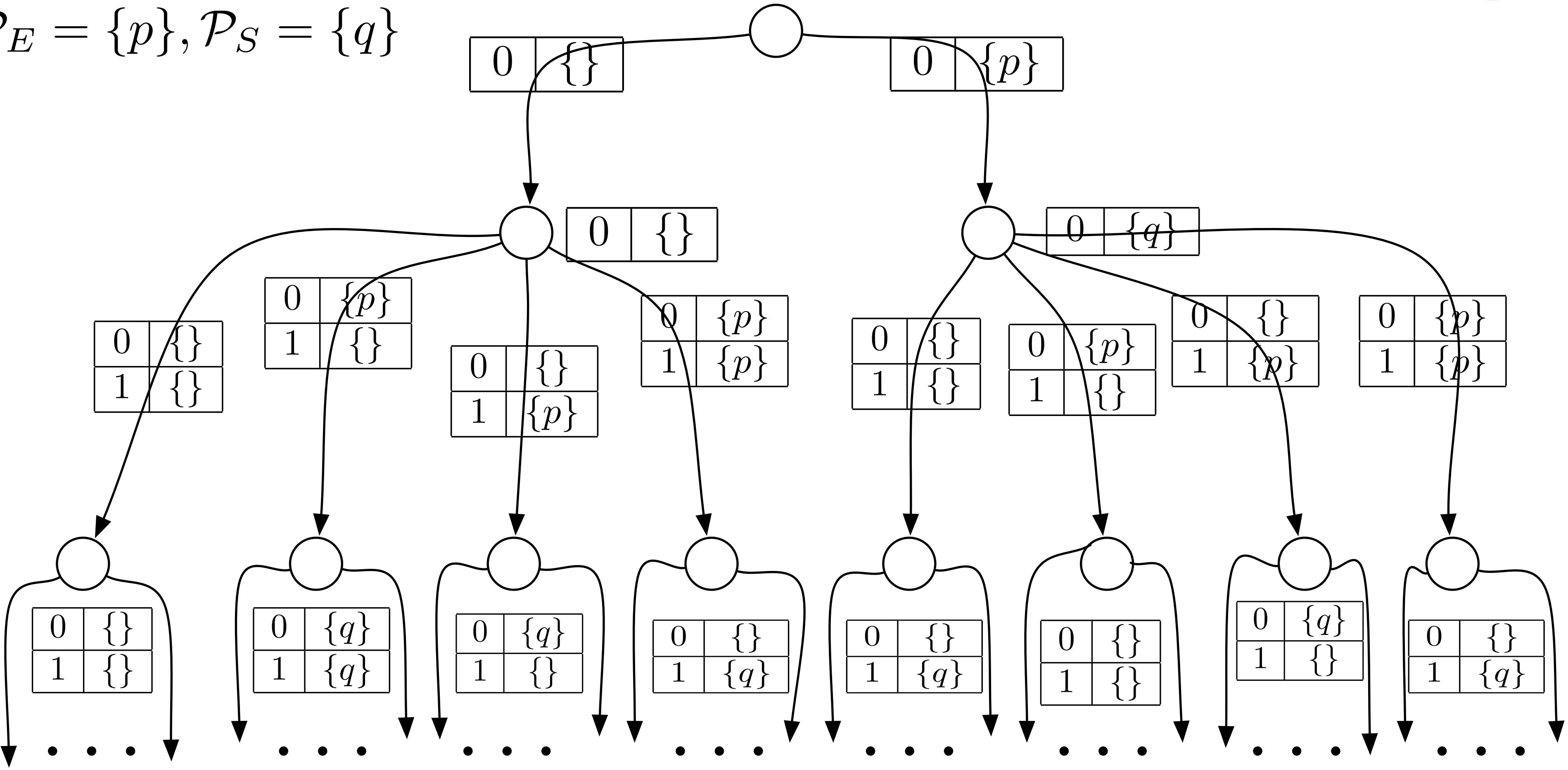
$\mathcal{P}_E = \{p\}, \mathcal{P}_S = \{q\}$



Interval Based Synthesis

A Synthesis Formulation for HS

$$\mathcal{P}_E = \{p\}, \mathcal{P}_S = \{q\}$$

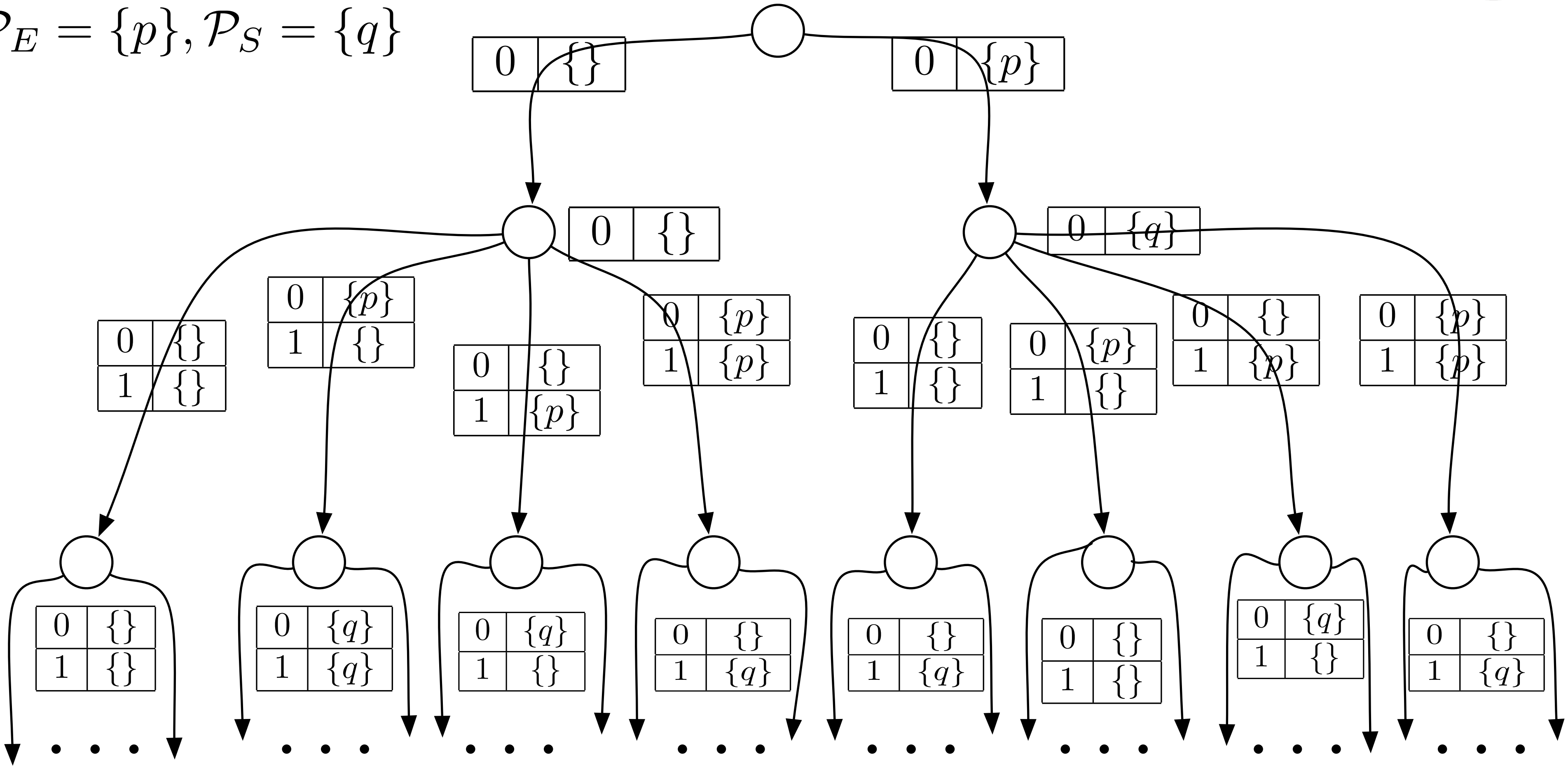


Pietro Sala Ph.D.

Interval Based Synthesis

A Synthesis Formulation for HS

$\mathcal{P}_E = \{p\}, \mathcal{P}_S = \{q\}$



Every infinite path represents a play.

If for every infinite path P the model induced by P is a model for φ then the tree is a winning strategy for the System.

$(\varphi, \mathcal{P}_E, \mathcal{P}_S)$ is a positive instance of the synthesis problem if and only if there exists a winning strategy for it.

Halpern and Shoham's Modal Logic of Intervals (HS)

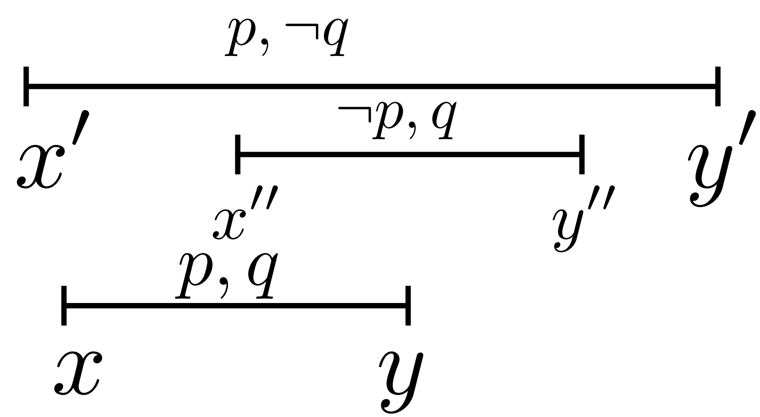
(R1) may be rephrased in an ``interval form'' as ``every interval that lasts 5 time points and begins with a point labelled by I it must contains a point labelled by O and if an interval that lasts 5 time points ends up in a point labelled by O must contain a point labelled by I".

We can express (R1) in HS the syntax of HS is the following:

$$HS \quad \varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle * \rangle \varphi$$

$$p \in \mathcal{P}, * \in \{A, B, D, E, L, O, \overline{A}, \overline{B}, \overline{D}, \overline{E}, \overline{L}, \overline{O}\}$$

$$\mathcal{V}([x, y]) = \{p, q\}$$
$$\mathcal{V}([x', y']) = \{p\}$$
$$\mathcal{V}([x'', y'']) = \{q\}$$



$$\mathbf{M} = (\mathbb{N}, \mathcal{V}), \mathcal{V} : \{[x, y] : x \leq y \in \mathbb{N}\} \rightarrow 2^{\mathcal{P}}$$

$$\mathbf{M}, [x, y] \models p \text{ if and only if } p \in \mathcal{V}([x, y])$$

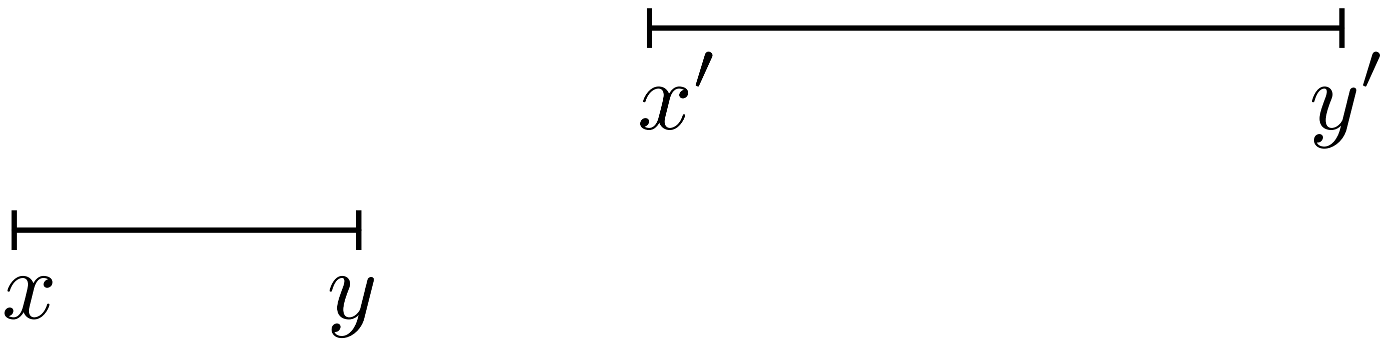
Halpern and Shoham's Modal Logic of Intervals (HS)

(R1) may be rephrased in an ``interval form" as ``every interval that lasts 5 time points and begins with a point labelled by I it must contains a point labelled by O and if an interval that lasts 5 time points ends up in a point labelled by O must contain a point labelled by I".

We can express (R1) in HS the syntax of HS is the following:

$$HS \quad \varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle * \rangle \varphi$$

$$p \in \mathcal{P}, * \in \{A, B, D, E, L, O, \overline{A}, \overline{B}, \overline{D}, \overline{E}, \overline{L}, \overline{O}\}$$



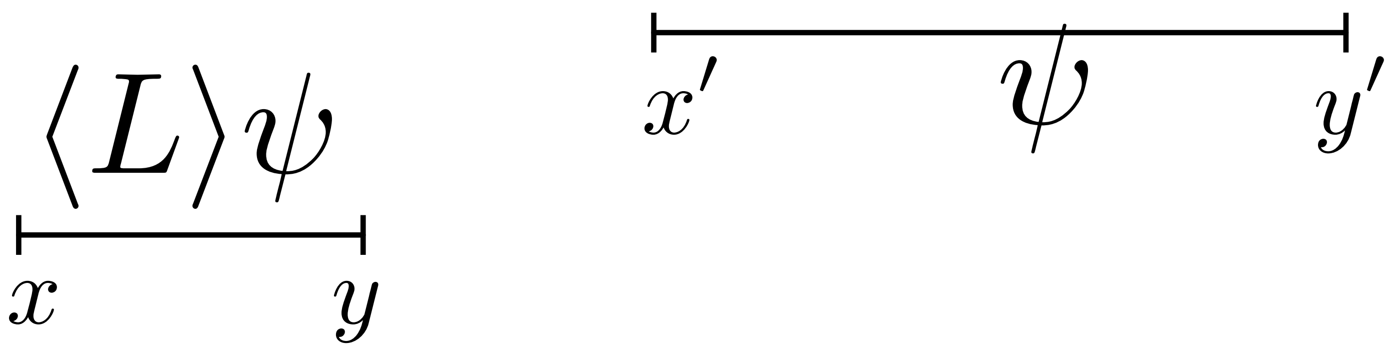
Halpern and Shoham's Modal Logic of Intervals (HS)

(R1) may be rephrased in an ``interval form" as ``every interval that lasts 5 time points and begins with a point labelled by I it must contains a point labelled by O and if an interval that lasts 5 time points ends up in a point labelled by O must contain a point labelled by I".

We can express (R1) in HS the syntax of HS is the following:

$$HS \quad \varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle * \rangle \varphi$$

$$p \in \mathcal{P}, * \in \{A, B, D, E, L, O, \overline{A}, \overline{B}, \overline{D}, \overline{E}, \overline{L}, \overline{O}\}$$



$$\mathbf{M}, [x, y] \models \langle L \rangle \psi$$

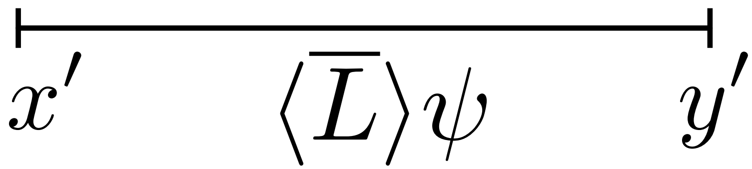
Halpern and Shoham's Modal Logic of Intervals (HS)

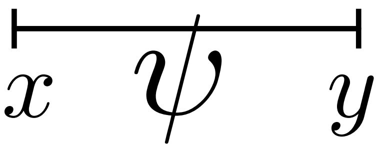
(R1) may be rephrased in an ``interval form" as ``every interval that lasts 5 time points and begins with a point labelled by I it must contains a point labelled by O and if an interval that lasts 5 time points ends up in a point labelled by O must contain a point labelled by I".

We can express (R1) in HS the syntax of HS is the following:

$$HS \quad \varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle * \rangle \varphi$$

$$p \in \mathcal{P}, * \in \{A, B, D, E, L, O, \overline{A}, \overline{B}, \overline{D}, \overline{E}, \overline{L}, \overline{O}\}$$





$$\mathbf{M}, [x', y] \models \langle \overline{L} \rangle \psi$$

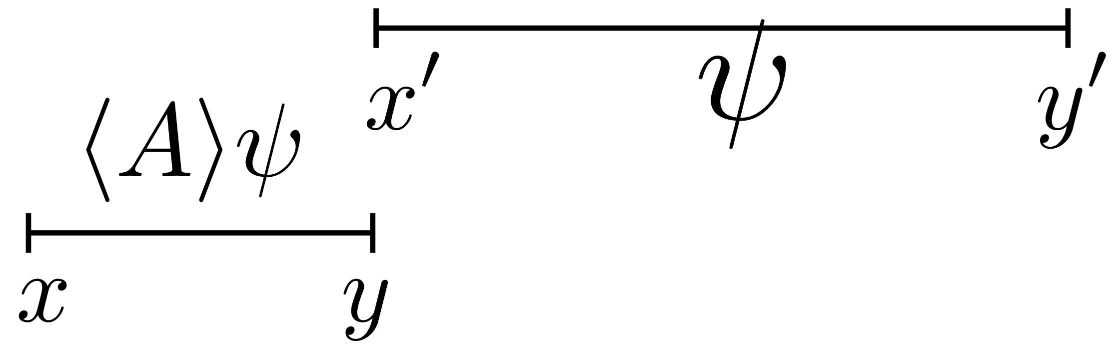
Halpern and Shoham's Modal Logic of Intervals (HS)

(R1) may be rephrased in an ``interval form" as ``every interval that lasts 5 time points and begins with a point labelled by I it must contains a point labelled by O and if an interval that lasts 5 time points ends up in a point labelled by O must contain a point labelled by I".

We can express (R1) in HS the syntax of HS is the following:

$$HS \quad \varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle * \rangle \varphi$$

$$p \in \mathcal{P}, * \in \{A, B, D, E, L, O, \overline{A}, \overline{B}, \overline{D}, \overline{E}, \overline{L}, \overline{O}\}$$



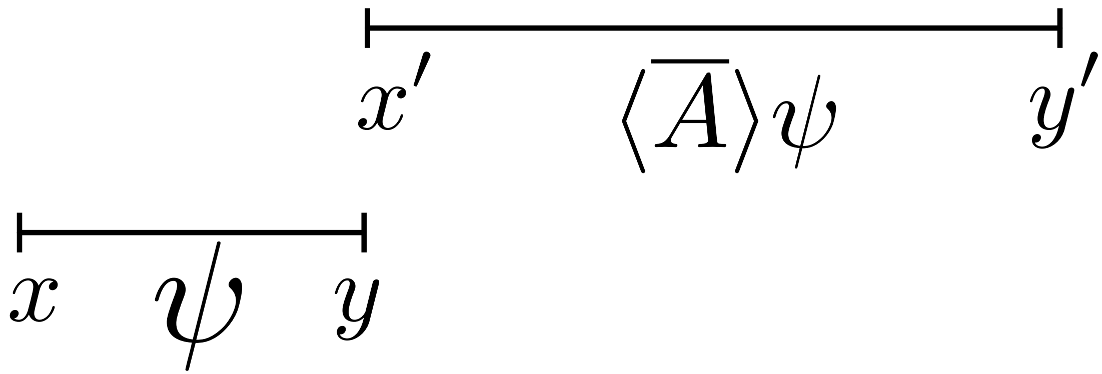
Halpern and Shoham's Modal Logic of Intervals (HS)

(R1) may be rephrased in an ``interval form" as ``every interval that lasts 5 time points and begins with a point labelled by I it must contains a point labelled by O and if an interval that lasts 5 time points ends up in a point labelled by O must contain a point labelled by I".

We can express (R1) in HS the syntax of HS is the following:

$$HS \quad \varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle * \rangle \varphi$$

$$p \in \mathcal{P}, * \in \{A, B, D, E, L, O, \overline{A}, \overline{B}, \overline{D}, \overline{E}, \overline{L}, \overline{O}\}$$



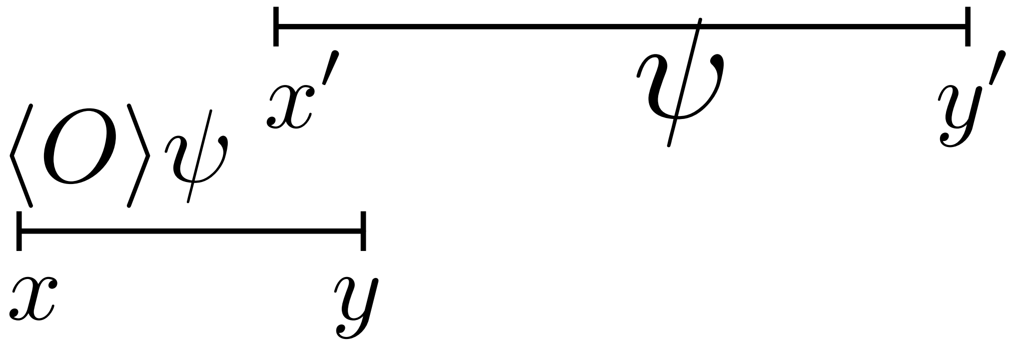
Halpern and Shoham's Modal Logic of Intervals (HS)

(R1) may be rephrased in an ``interval form" as ``every interval that lasts 5 time points and begins with a point labelled by I it must contains a point labelled by O and if an interval that lasts 5 time points ends up in a point labelled by O must contain a point labelled by I".

We can express (R1) in HS the syntax of HS is the following:

$$HS \quad \varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle * \rangle \varphi$$

$$p \in \mathcal{P}, * \in \{A, B, D, E, L, O, \overline{A}, \overline{B}, \overline{D}, \overline{E}, \overline{L}, \overline{O}\}$$



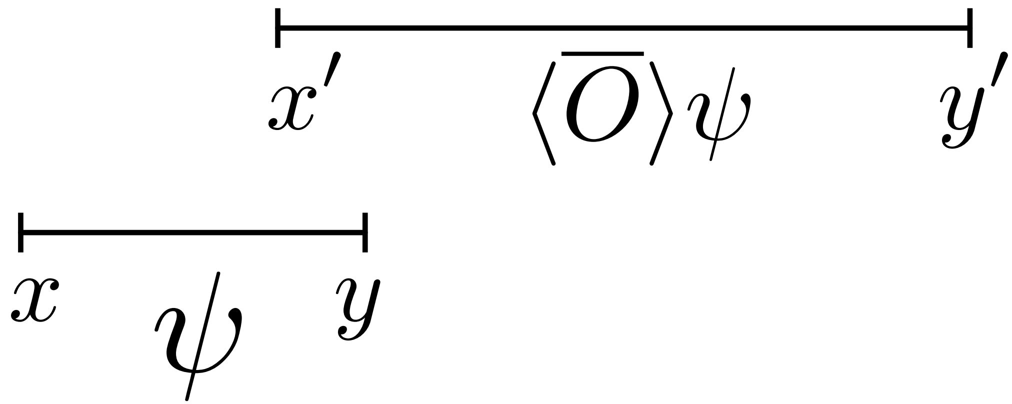
Halpern and Shoham's Modal Logic of Intervals (HS)

(R1) may be rephrased in an ``interval form" as ``every interval that lasts 5 time points and begins with a point labelled by I it must contains a point labelled by O and if an interval that lasts 5 time points ends up in a point labelled by O must contain a point labelled by I".

We can express (R1) in HS the syntax of HS is the following:

$$HS \quad \varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle * \rangle \varphi$$

$$p \in \mathcal{P}, * \in \{A, B, D, E, L, O, \overline{A}, \overline{B}, \overline{D}, \overline{E}, \overline{L}, \overline{O}\}$$



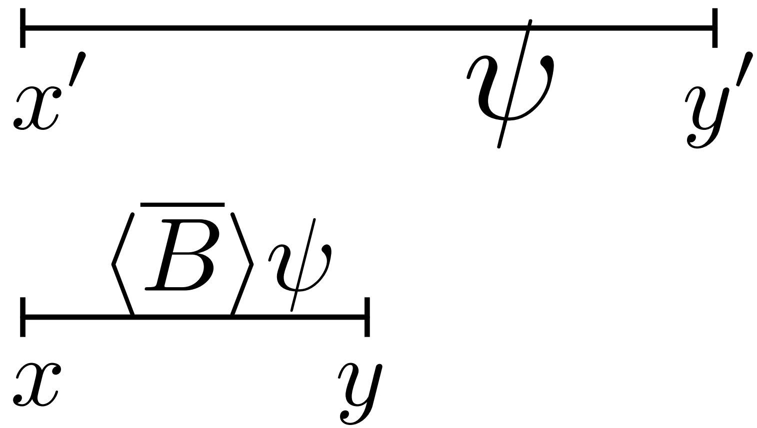
Halpern and Shoham's Modal Logic of Intervals (HS)

(R1) may be rephrased in an ``interval form" as ``every interval that lasts 5 time points and begins with a point labelled by I it must contains a point labelled by O and if an interval that lasts 5 time points ends up in a point labelled by O must contain a point labelled by I".

We can express (R1) in HS the syntax of HS is the following:

$$HS \quad \varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle * \rangle \varphi$$

$$p \in \mathcal{P}, * \in \{A, B, D, E, L, O, \overline{A}, \overline{B}, \overline{D}, \overline{E}, \overline{L}, \overline{O}\}$$



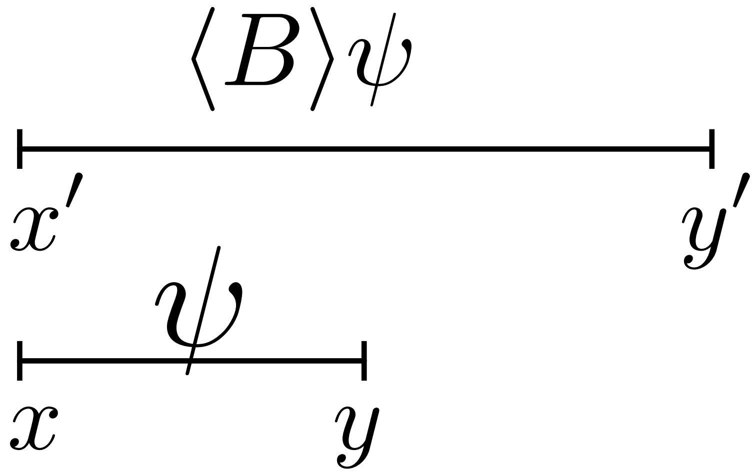
Halpern and Shoham's Modal Logic of Intervals (HS)

(R1) may be rephrased in an ``interval form'' as ``every interval that lasts 5 time points and begins with a point labelled by I it must contains a point labelled by O and if an interval that lasts 5 time points ends up in a point labelled by O must contain a point labelled by I".

We can express (R1) in HS the syntax of HS is the following:

$$HS \quad \varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle * \rangle \varphi$$

$$p \in \mathcal{P}, * \in \{A, B, D, E, L, O, \overline{A}, \overline{B}, \overline{D}, \overline{E}, \overline{L}, \overline{O}\}$$



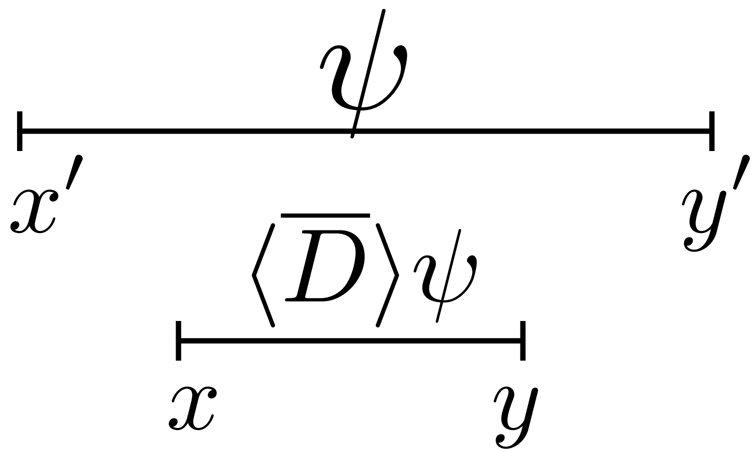
Halpern and Shoham's Modal Logic of Intervals (HS)

(R1) may be rephrased in an ``interval form''  as ``every interval that lasts 5 time points and begins with a point labelled by I it must contains a point labelled by O and if an interval that lasts 5 time points ends up in a point labelled by O must contain a point labelled by I".

We can express (R1) in HS the syntax of HS is the following:

$$HS \quad \varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle * \rangle \varphi$$

$$p \in \mathcal{P}, * \in \{A, B, D, E, L, O, \overline{A}, \overline{B}, \overline{D}, \overline{E}, \overline{L}, \overline{O}\}$$



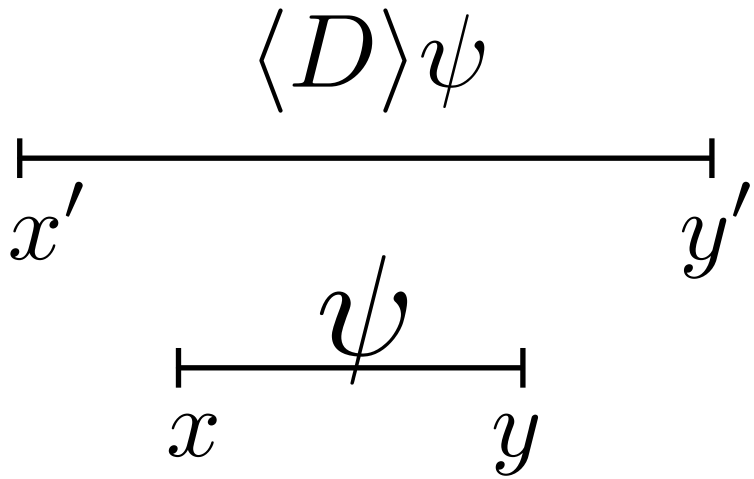
Halpern and Shoham’s Modal Logic of Intervals (HS)

(R1) may be rephrased in an ``interval form" as ``every interval that lasts 5 time points and begins with a point labelled by I it must contains a point labelled by O and if an interval that lasts 5 time points ends up in a point labelled by O must contain a point labelled by I”.

We can express (R1) in HS the syntax of HS is the following:

$$HS \quad \varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle * \rangle \varphi$$

$$p \in \mathcal{P}, * \in \{A, B, D, E, L, O, \overline{A}, \overline{B}, \overline{D}, \overline{E}, \overline{L}, \overline{O}\}$$



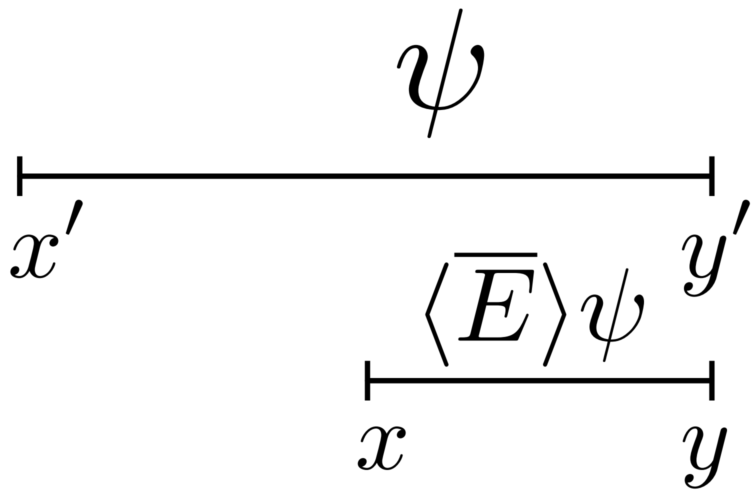
Halpern and Shoham's Modal Logic of Intervals (HS)

(R1) may be rephrased in an ``interval form" as ``every interval that lasts 5 time points and begins with a point labelled by I it must contains a point labelled by O and if an interval that lasts 5 time points ends up in a point labelled by O must contain a point labelled by I".

We can express (R1) in HS the syntax of HS is the following:

$$HS \quad \varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle * \rangle \varphi$$

$$p \in \mathcal{P}, * \in \{A, B, D, E, L, O, \overline{A}, \overline{B}, \overline{D}, \overline{E}, \overline{L}, \overline{O}\}$$



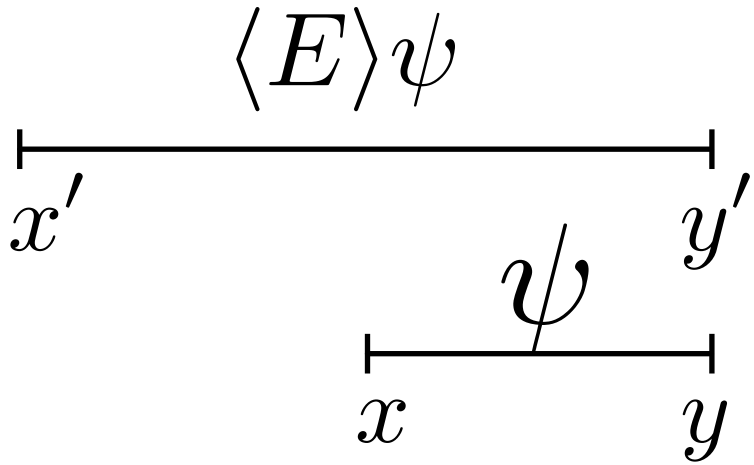
Halpern and Shoham's Modal Logic of Intervals (HS)

(R1) may be rephrased in an ``interval form" as ``every interval that lasts 5 time points and begins with a point labelled by I it must contains a point labelled by O and if an interval that lasts 5 time points ends up in a point labelled by O must contain a point labelled by I".

We can express (R1) in HS the syntax of HS is the following:

$$HS \quad \varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle * \rangle \varphi$$

$$p \in \mathcal{P}, * \in \{A, B, D, E, L, O, \overline{A}, \overline{B}, \overline{D}, \overline{E}, \overline{L}, \overline{O}\}$$



Halpern and Shoham's Modal Logic of Intervals (HS)

(R1) may be rephrased in an ``interval form'' as ``every interval that lasts 5 time points and begins with a point labelled by I it must contains a point labelled by O and if an interval that lasts 5 time points ends up in a point labelled by O must contain a point labelled by I''.

We can express (R1) in HS the syntax of HS is the following:

$$HS \quad \varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle * \rangle \varphi$$

$$p \in \mathcal{P}, * \in \{A, B, D, E, L, O, \overline{A}, \overline{B}, \overline{D}, \overline{E}, \overline{L}, \overline{O}\}$$

$$[*]\psi :== \neg\langle * \rangle \neg\psi$$

$$[G]\psi :== \psi \wedge [A]\psi \wedge [A][A]\psi$$

$$\langle S \rangle \psi :== \neg[G]\neg\psi$$

$$\perp = p \wedge \neg p$$

$$len_k = \begin{cases} [B]\perp & k = 0 \\ \langle B \rangle len_{k-1} \wedge [B][B]\neg len_{k-1} & k > 0 \end{cases}$$

Halpern and Shoham's Modal Logic of Intervals (HS)

(R1) may be rephrased in an ``interval form" as ``every interval that lasts 5 time points and begins with a point labelled by I it must contains a point labelled by O and if an interval that lasts 5 time points ends up in a point labelled by O must contain a point labelled by I".

We can express (R1) in HS the syntax of HS is the following:

$$HS \quad \varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle * \rangle \varphi$$

$$p \in \mathcal{P}, * \in \{A, B, D, E, L, O, \overline{A}, \overline{B}, \overline{D}, \overline{E}, \overline{L}, \overline{O}\}$$

$$[G](len_5 \wedge \langle B \rangle (I \wedge len_0) \rightarrow \langle B \rangle (\neg len_0 \wedge \langle A \rangle (I \wedge len_0))$$

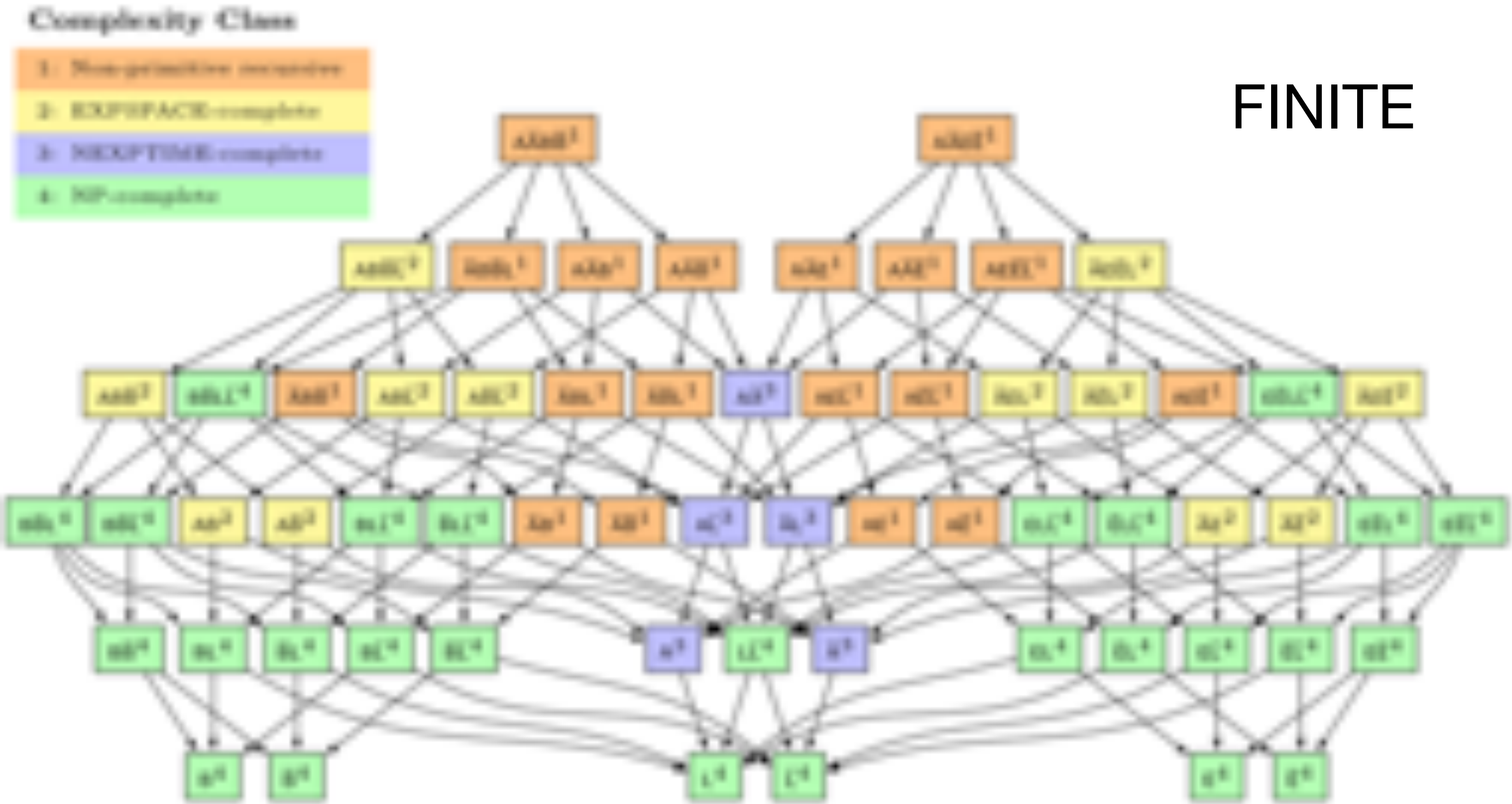
$$\wedge$$

$$[G](len_5 \wedge \langle A \rangle (I \wedge len_0) \rightarrow \langle B \rangle \langle A \rangle (I \wedge len_0))$$

HS Satisfiability Problem

HS synthesis problem entails the decidability problem (it suffices to put $\mathcal{P}_E = \emptyset$);

HS interpreted over (prefixes) of the natural numbers has very few fragments that feature a decidable satisfiability problem:

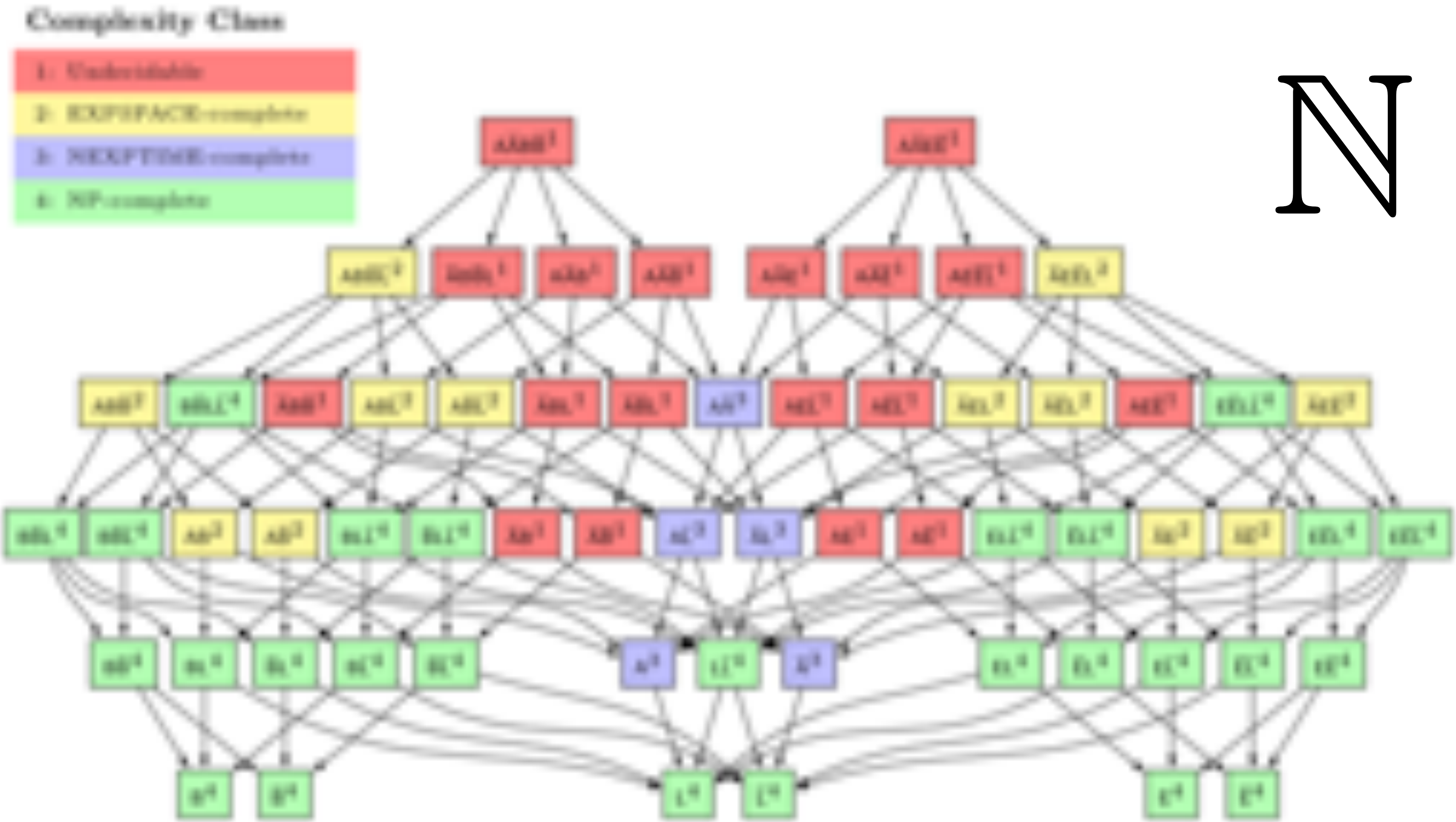


Davide Bresolin, Dario Della Monica, Angelo Montanari, Pietro Sala, Guido Sciavicco:
Interval temporal logics over strongly discrete linear orders: Expressiveness and complexity. Theor. Comput. Sci. 560: 269-291 (2014)

HS Satisfiability Problem

HS synthesis problem entails the decidability problem (it suffices to put $\mathcal{P}_E = \emptyset$);

HS interpreted over (prefixes) of the natural numbers has very few fragments that feature a decidable satisfiability problem:



$\mathbb{N}$

Davide Bresolin, Dario Della Monica, Angelo Montanari, Pietro Sala, Guido Sciavicco:
Interval temporal logics over strongly discrete linear orders: Expressiveness and complexity. Theor. Comput. Sci. 560: 269-291 (2014)

HS Synthesis Problem

The two main obstacles on the way to achieve decidability for HS synthesis Problem are:

nodes and edge-labels of the edges in the tree are grow along the path

WQO representation of labels

the number of successors in the tree are growing in size along the path

the strategy tree that witnesses positive instance in thenatural numbers case is infinite

Graph representation of trees

HS Synthesis Expressivity

HS is not just a rewriting of the properties that may be expressed in more classical logics like MSO.

Let us consider the following example taken from the medical domain:

“for some severe illness diagnosed in the Emergency Unit the patient must be kept under observation for at least as many days as it is the days elapsed from the day in which he has contracted the fever”

HS Synthesis Expressivity

HS is not just a rewriting of the properties that may be expressed in more classical logics like MSO.

Let us consider the following example taken from the medical domain:

“for some severe illness diagnosed in the Emergency Unit the patient must be kept under observation for at least as many days as it is the days elapsed from the day in which he has contracted the fever”

We assume that each point of our model is a day.

$$\begin{array}{cccccccccccc} & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ \\ day & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{array}$$

HS Synthesis Expressivity

HS is not just a rewriting of the properties that may be expressed in more classical logics like MSO.

Let us consider the following example taken from the medical domain:

"for some severe illness diagnosed in the Emergency Unit the patient must be kept under observation for at least as many days as it is the days elapsed from the day in which he has contracted the fever"

We assume that each point of our model is a day.

$$day \quad 0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8 \quad 9 \quad 10$$

HS Synthesis Expressivity

HS is not just a rewriting of the properties that may be expressed in more classical logics like MSO.

Let us consider the following example taken from the medical domain:

“for some severe illness diagnosed in the Emergency Unit the patient must be kept under observation for at least as many days as it is the days elapsed from the day in which he has contracted the fever”

We assume that each point of our model is a day.

$$day \quad 0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8 \quad 9 \quad 10$$

HS Synthesis Expressivity

HS is not just a rewriting of the properties that may be expressed in more classical logics like MSO.

Let us consider the following example taken from the medical domain:

“for some severe illness diagnosed in the Emergency Unit the patient must be kept under observation for at least as many days as it is the days elapsed from the day in which he has contracted the fever”

We assume that each point of our model is a day.

$$\begin{array}{cccccccccccc} & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ \\ day & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{array}$$

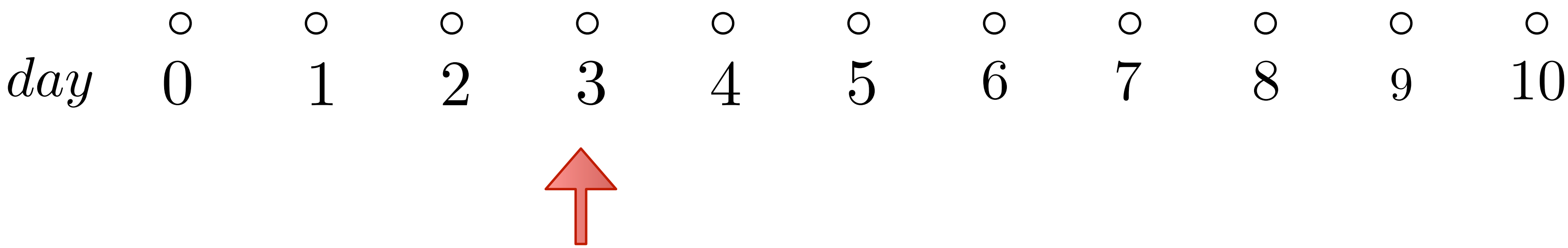
HS Synthesis Expressivity

HS is not just a rewriting of the properties that may be expressed in more classical logics like MSO.

Let us consider the following example taken from the medical domain:

"for some severe illness diagnosed in the Emergency Unit the patient must be kept under observation for at least as many days as it is the days elapsed from the day in which he has contracted the fever"

We assume that each point of our model is a day.



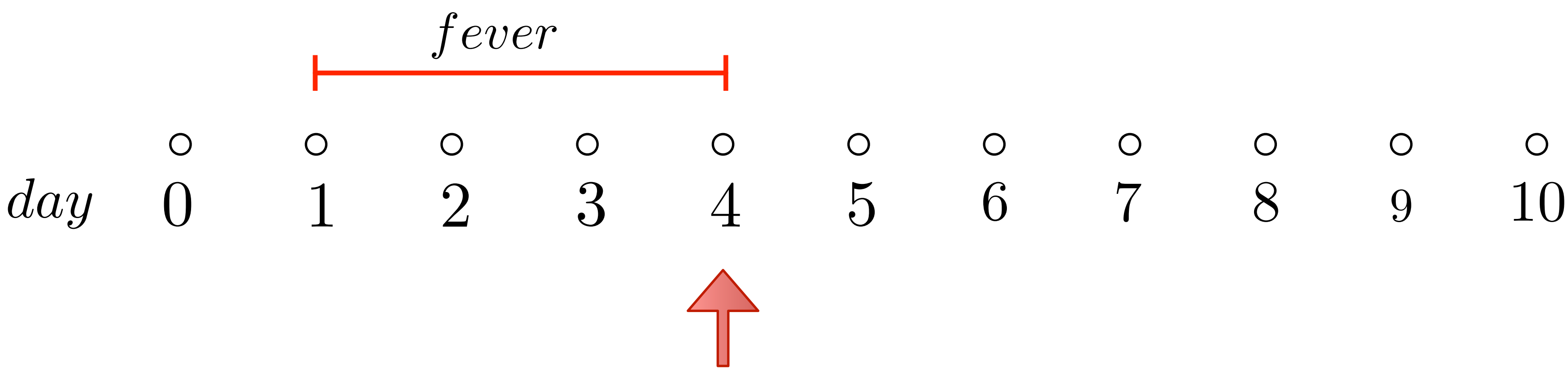
HS Synthesis Expressivity

HS is not just a rewriting of the properties that may be expressed in more classical logics like MSO.

Let us consider the following example taken from the medical domain:

“for some severe illness diagnosed in the Emergency Unit the patient must be kept under observation for at least as many days as it is the days elapsed from the day in which he has contracted the fever”

We assume that each point of our model is a day.



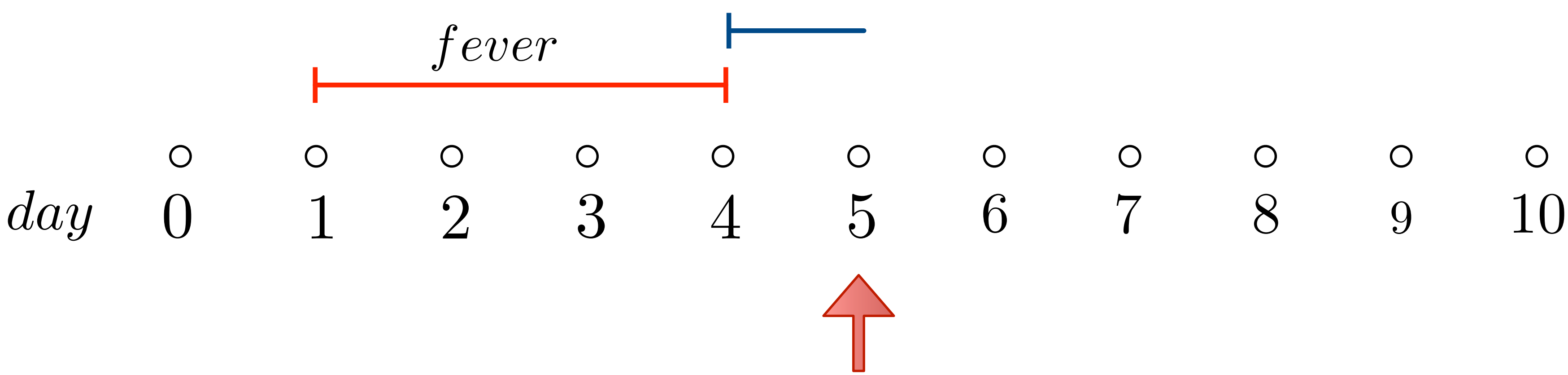
HS Synthesis Expressivity

HS is not just a rewriting of the properties that may be expressed in more classical logics like MSO.

Let us consider the following example taken from the medical domain:

“for some severe illness diagnosed in the Emergency Unit the patient must be kept under observation for at least as many days as it is the days elapsed from the day in which he has contracted the fever”

We assume that each point of our model is a day.



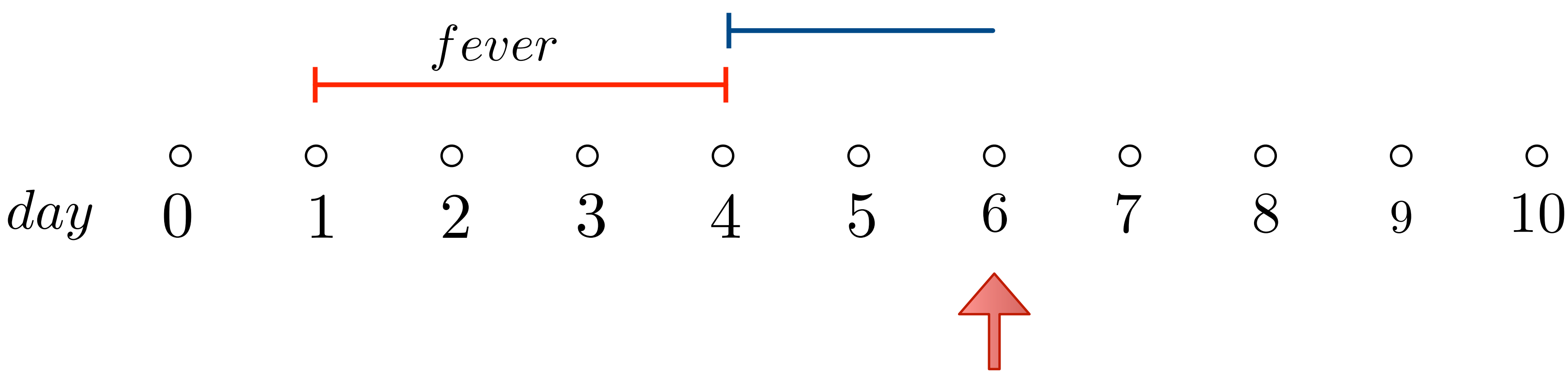
HS Synthesis Expressivity

HS is not just a rewriting of the properties that may be expressed in more classical logics like MSO.

Let us consider the following example taken from the medical domain:

"for some severe illness diagnosed in the Emergency Unit the patient must be kept under observation for at least as many days as it is the days elapsed from the day in which he has contracted the fever"

We assume that each point of our model is a day.



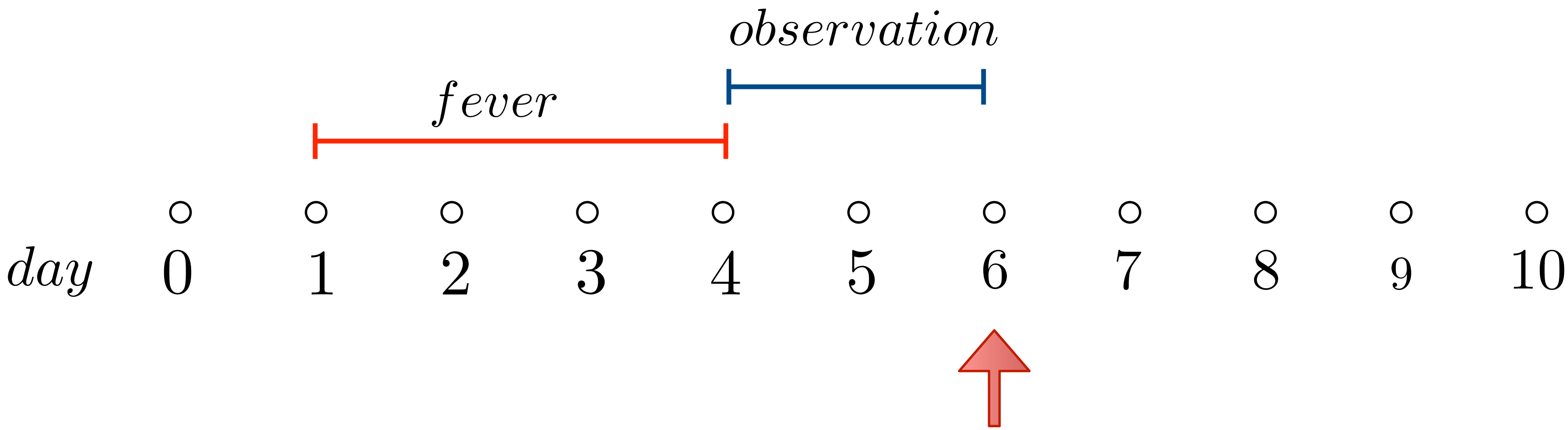
HS Synthesis Expressivity

HS is not just a rewriting of the properties that may be expressed in more classical logics like MSO.

Let us consider the following example taken from the medical domain:

“for some severe illness diagnosed in the Emergency Unit the patient must be kept under observation for at least as many days as it is the days elapsed from the day in which he has contracted the fever”

We assume that each point of our model is a day.



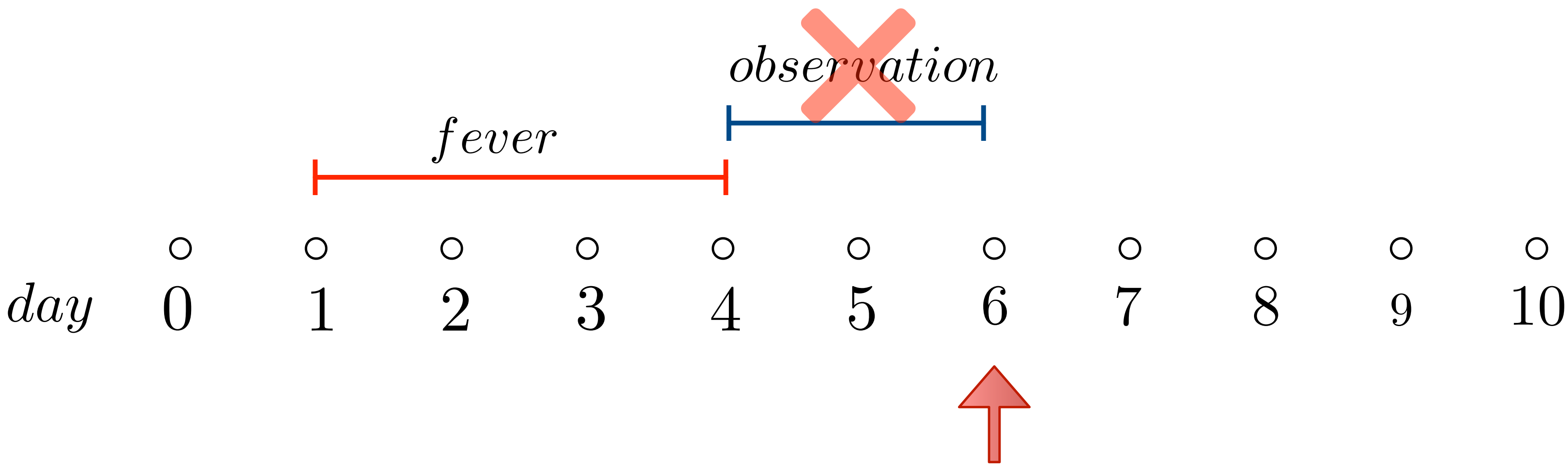
HS Synthesis Expressivity

HS is not just a rewriting of the properties that may be expressed in more classical logics like MSO.

Let us consider the following example taken from the medical domain:

“for some severe illness diagnosed in the Emergency Unit the patient must be kept under observation for at least as many days as it is the days elapsed from the day in which he has contracted the fever”

We assume that each point of our model is a day.



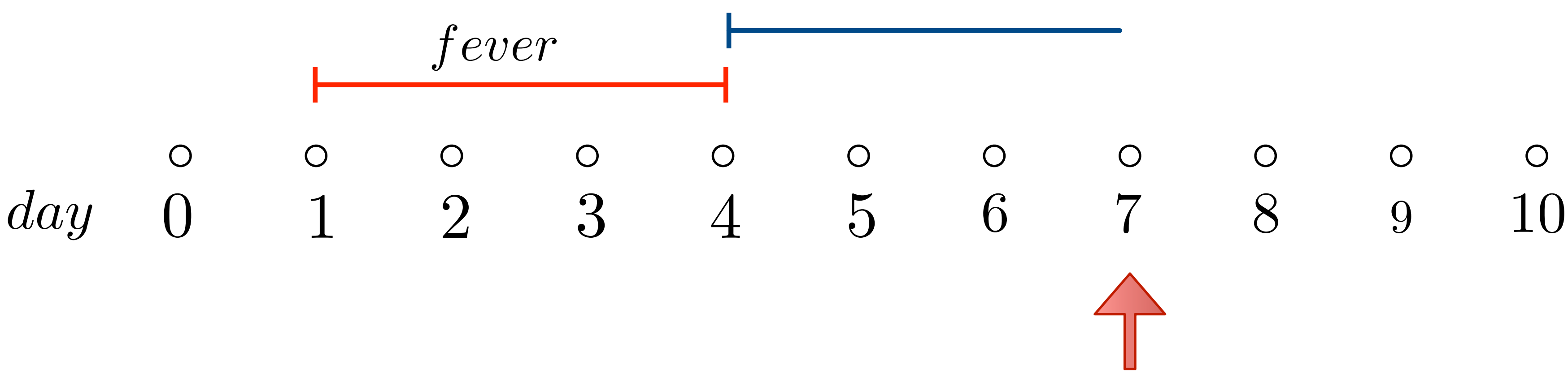
HS Synthesis Expressivity

HS is not just a rewriting of the properties that may be expressed in more classical logics like MSO.

Let us consider the following example taken from the medical domain:

"for some severe illness diagnosed in the Emergency Unit the patient must be kept under observation for at least as many days as it is the days elapsed from the day in which he has contracted the fever"

We assume that each point of our model is a day.



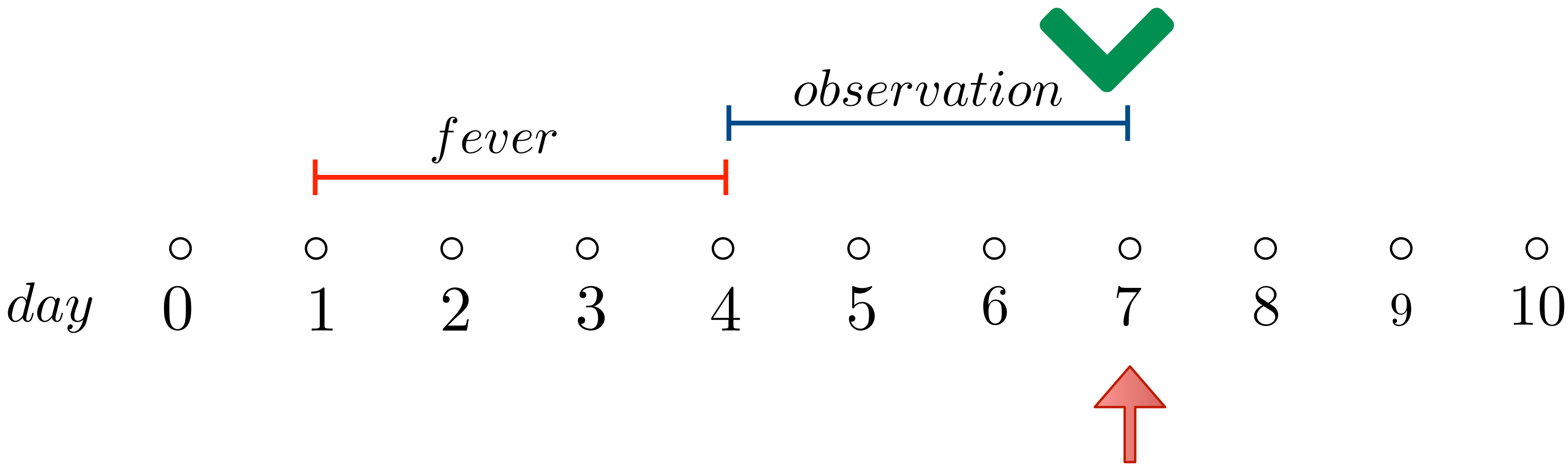
HS Synthesis Expressivity

HS is not just a rewriting of the properties that may be expressed in more classical logics like MSO.

Let us consider the following example taken from the medical domain:

“for some severe illness diagnosed in the Emergency Unit the patient must be kept under observation for at least as many days as it is the days elapsed from the day in which he has contracted the fever”

We assume that each point of our model is a day.



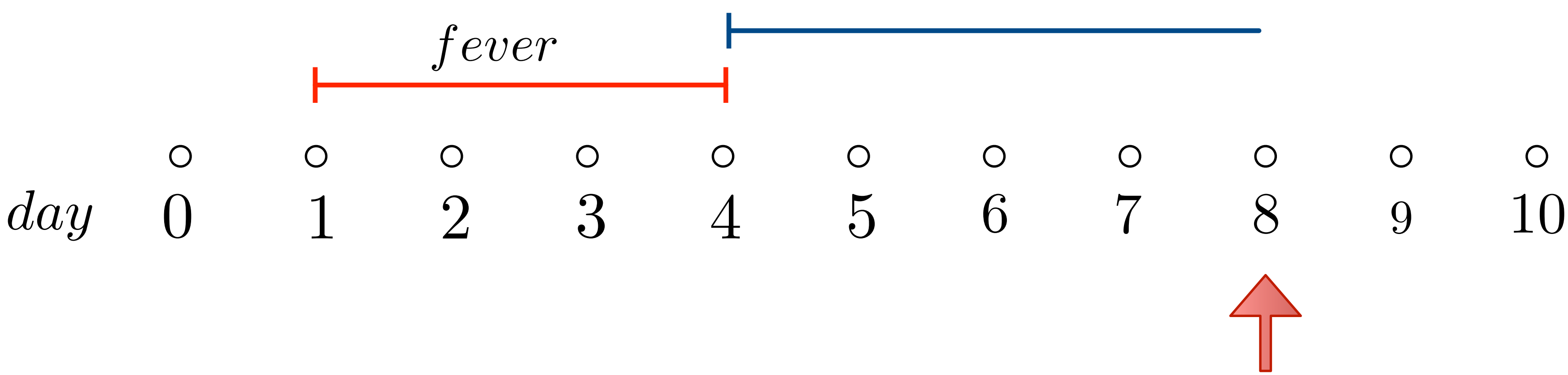
HS Synthesis Expressivity

HS is not just a rewriting of the properties that may be expressed in more classical logics like MSO.

Let us consider the following example taken from the medical domain:

“for some severe illness diagnosed in the Emergency Unit the patient must be kept under observation for at least as many days as it is the days elapsed from the day in which he has contracted the fever”

We assume that each point of our model is a day.



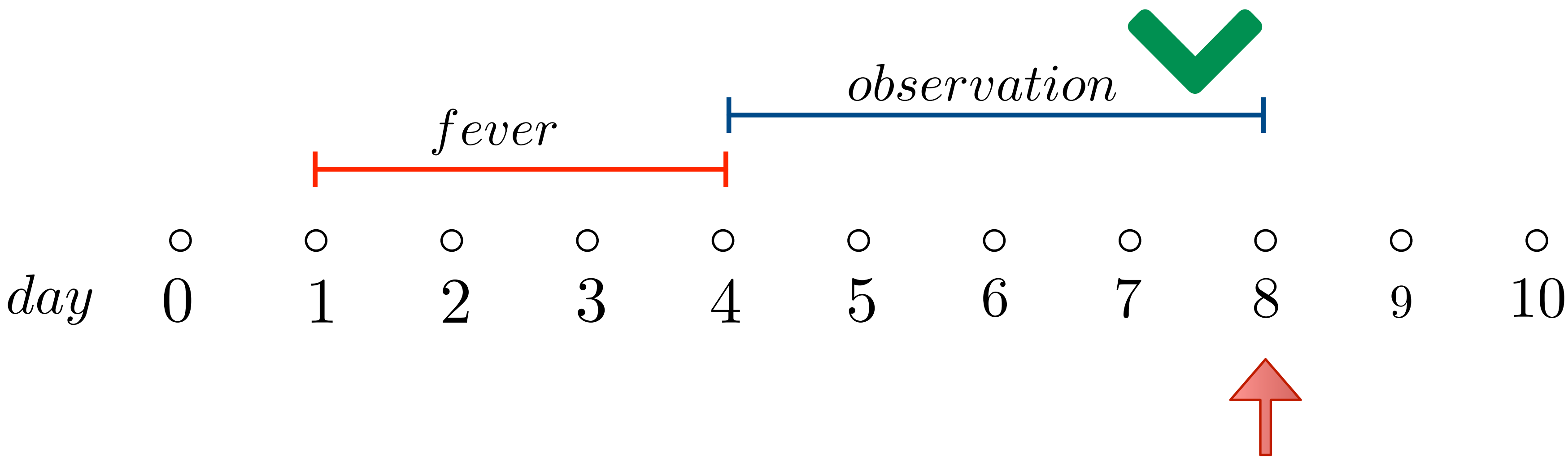
HS Synthesis Expressivity

HS is not just a rewriting of the properties that may be expressed in more classical logics like MSO.

Let us consider the following example taken from the medical domain:

"for some severe illness diagnosed in the Emergency Unit the patient must be kept under observation for at least as many days as it is the days elapsed from the day in which he has contracted the fever"

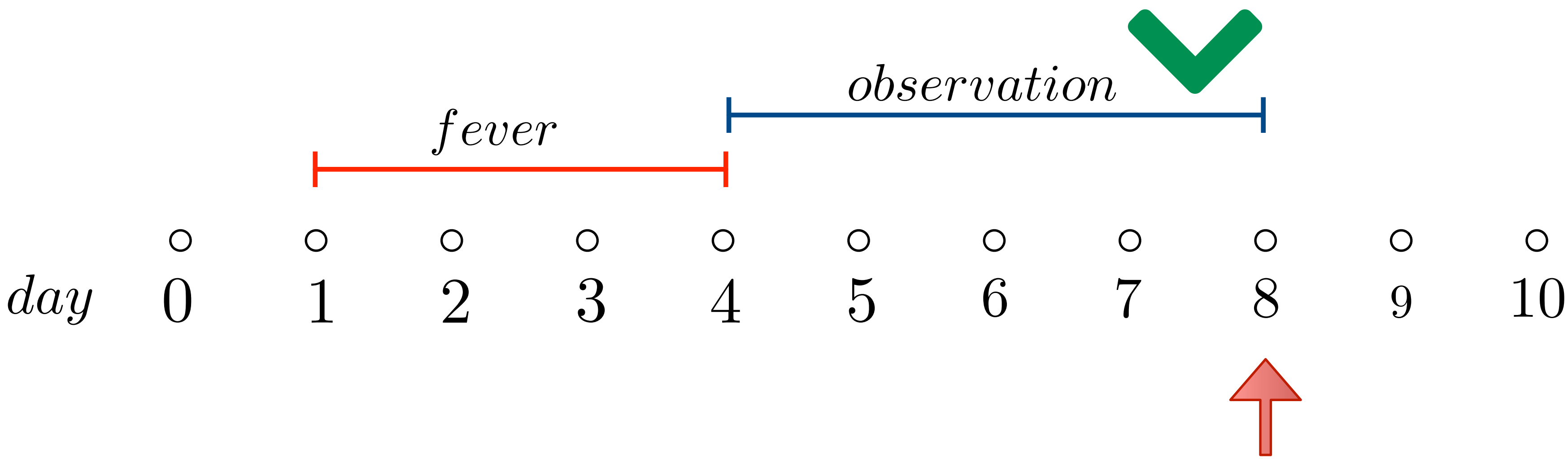
We assume that each point of our model is a day.



HS Synthesis Expressivity

The duration of an event is revealed right at the end of it, we call such property:
Delayed Observation of Durations

The duration of an event depends on the duration of a previous event which is, in principle, NOT bounded:
Arbitrary Dependent Durations



HS Synthesis Expressivity

“for some severe illness diagnosed in the Emergency Unit the patient must be kept under observation for at least as many days as it is the days elapsed from the day in which he has contracted the fever”

HS Synthesis Expressivity

“for some severe illness diagnosed in the Emergency Unit the patient must be kept under observation for at least as many days as it is the days elapsed from the day in which he has contracted the fever”

$$f \in \mathcal{P}_E \qquad o \in \mathcal{P}_S$$

HS Synthesis Expressivity

"for some severe illness diagnosed in the Emergency Unit the patient must be kept under observation for at least as many days as it is the days elapsed from the day in which he has contracted the fever"

$f \in \mathcal{P}_E \quad o \in \mathcal{P}_S$

$$[G]\left(\begin{array}{c} \cdots \\ \wedge \\ \cdots \\ \vee \end{array}\right)$$

$$\langle S \rangle \left(\begin{array}{c} \cdots \\ \vee \\ \cdots \end{array}\right)$$

HS Synthesis Expressivity

“for some severe illness diagnosed in the Emergency Unit the patient must be kept under observation for at least as many days as it is the days elapsed from the day in which he has contracted the fever”

$f \in \mathcal{P}_E \quad o \in \mathcal{P}_S$

$$[G]\left(\begin{array}{c} o \rightarrow [B][A]\neg o \\ \wedge \\ \ldots \end{array} \right)$$

$$\vee$$

$$\langle S \rangle \left(\begin{array}{c} \ldots \\ \vee \\ \ldots \end{array} \right)$$

“observations do not intersect”

HS Synthesis Expressivity

"for some severe illness diagnosed in the Emergency Unit the patient must be kept under observation for at least as many days as it is the days elapsed from the day in which he has contracted the fever"

$f \in \mathcal{P}_E \qquad o \in \mathcal{P}_S$

$$[G]\left(\begin{array}{c} o \rightarrow [B][A]\neg o \\ \wedge \\ \ldots \end{array}\right)$$

$$\vee$$

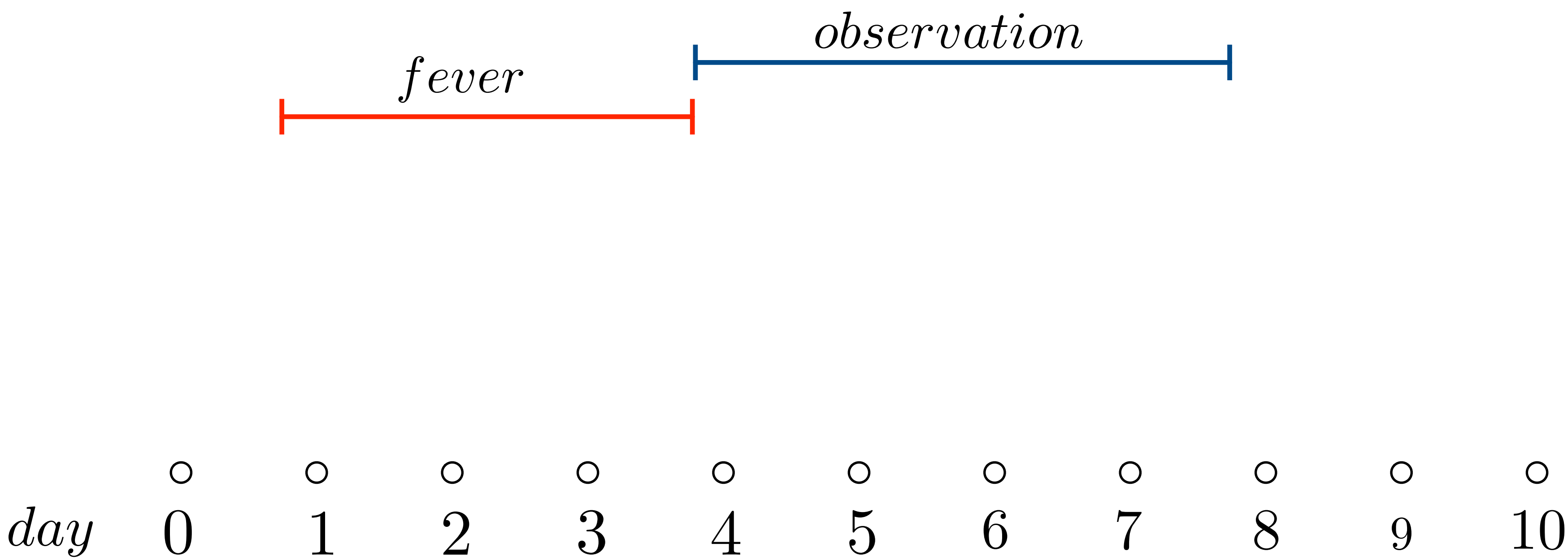
$$\langle S \rangle\left(\begin{array}{c} f \wedge \langle B \rangle\langle A \rangle f \\ \vee \\ \ldots \end{array}\right)$$

"fever periods do not intersect"

HS Synthesis Expressivity

The duration of an event depends on the duration of a previous event which is, in principle, NOT bounded:

Arbitrary Dependent Durations



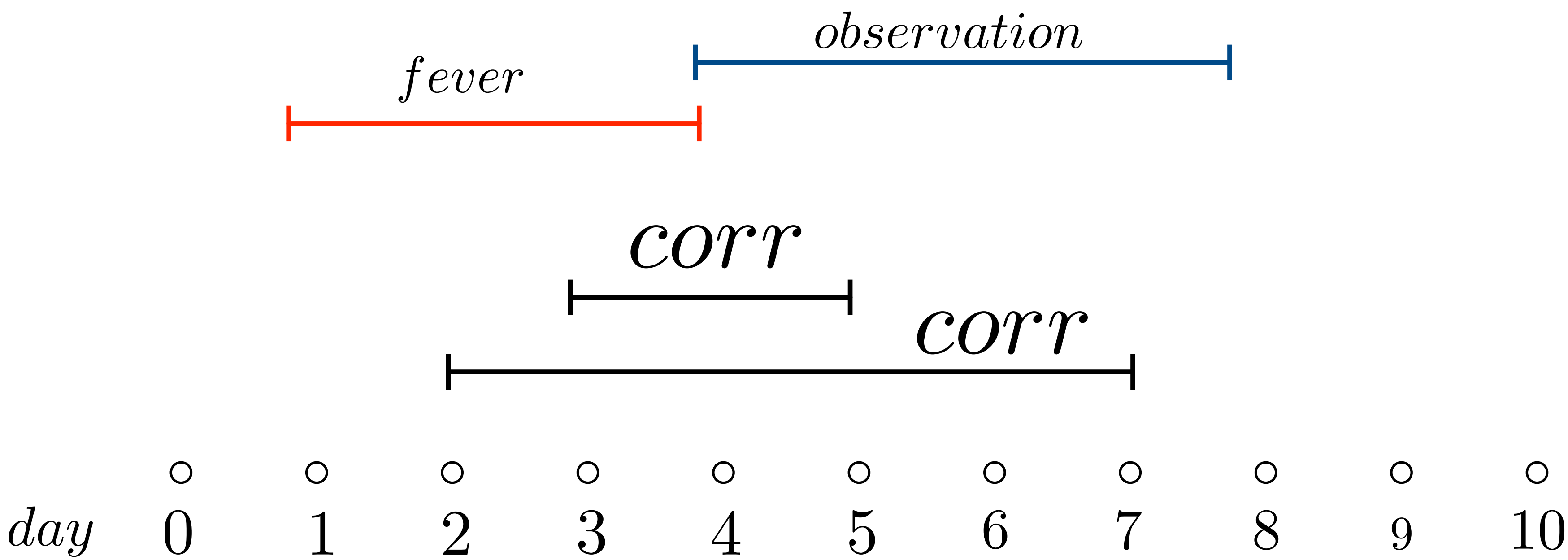
HS Synthesis Expressivity

The duration of an event depends on the duration of a previous event which is, in principle, NOT bounded:

Arbitrary Dependent Durations

corr must encode a total **injective** function between the interior of a fever interval and the interior of its adjacent observation interval

$$corr \in \mathcal{P}_S$$

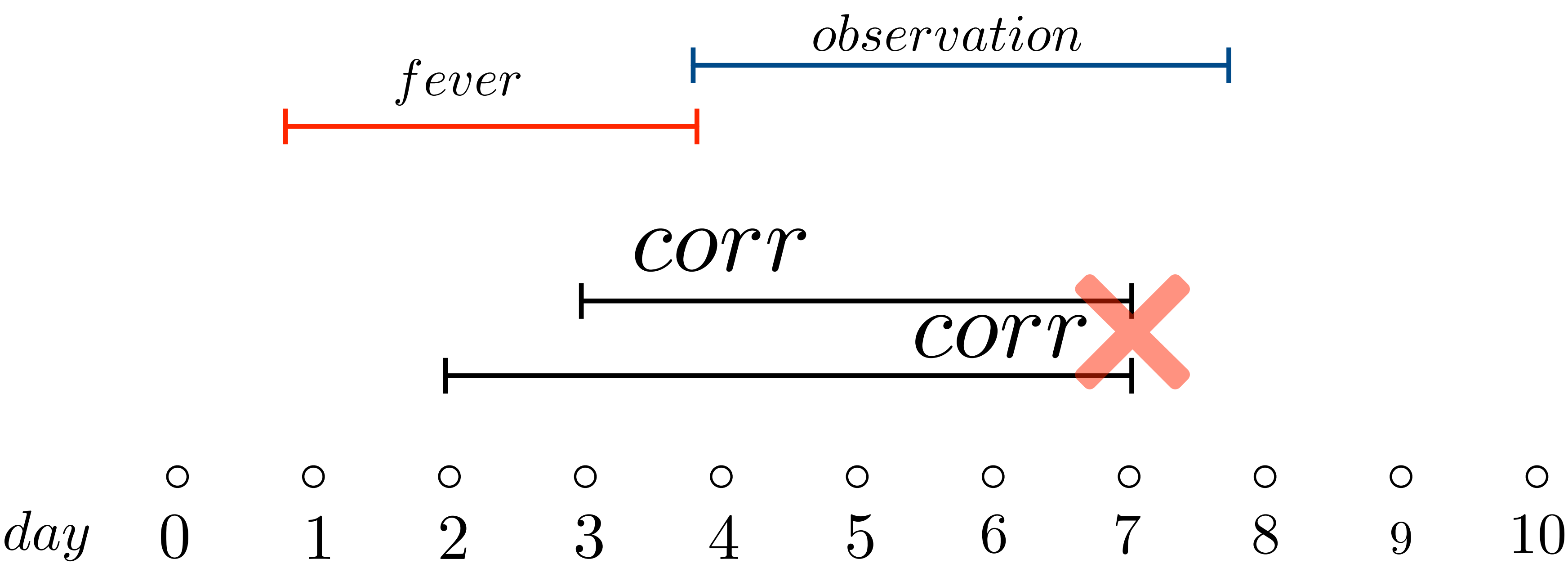


HS Synthesis Expressivity

The duration of an event depends on the duration of a previous event which is, in principle, NOT bounded:

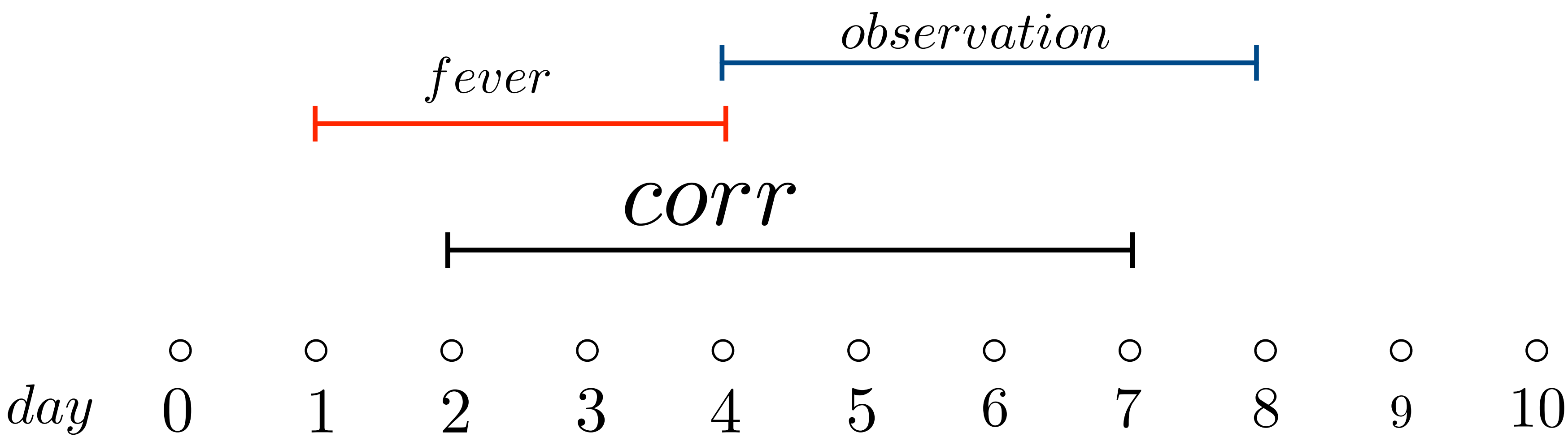
Arbitrary Dependent Durations

core must encode a total **injective** function between the interior of a fever interval and the interior of its adjacent observation interval



HS Synthesis Expressivity

core must encode a total **injective** function between the interior of a fever interval and the interior of its adjacent observation interval

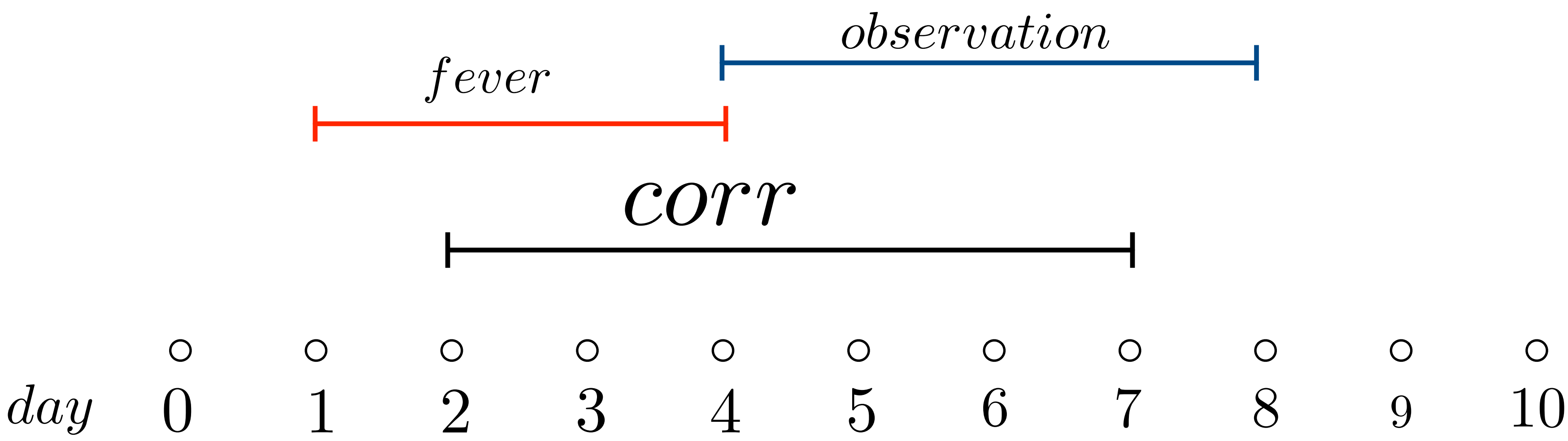


$$[G]\left(corr \rightarrow \langle B\rangle\langle A\rangle o \wedge [B](\langle A\rangle o \rightarrow [B][A]\neg o) \right)$$

"each corr interval crosses exactly one observation"

HS Synthesis Expressivity

core must encode a total **injective** function between the interior of a fever interval and the interior of its adjacent observation interval

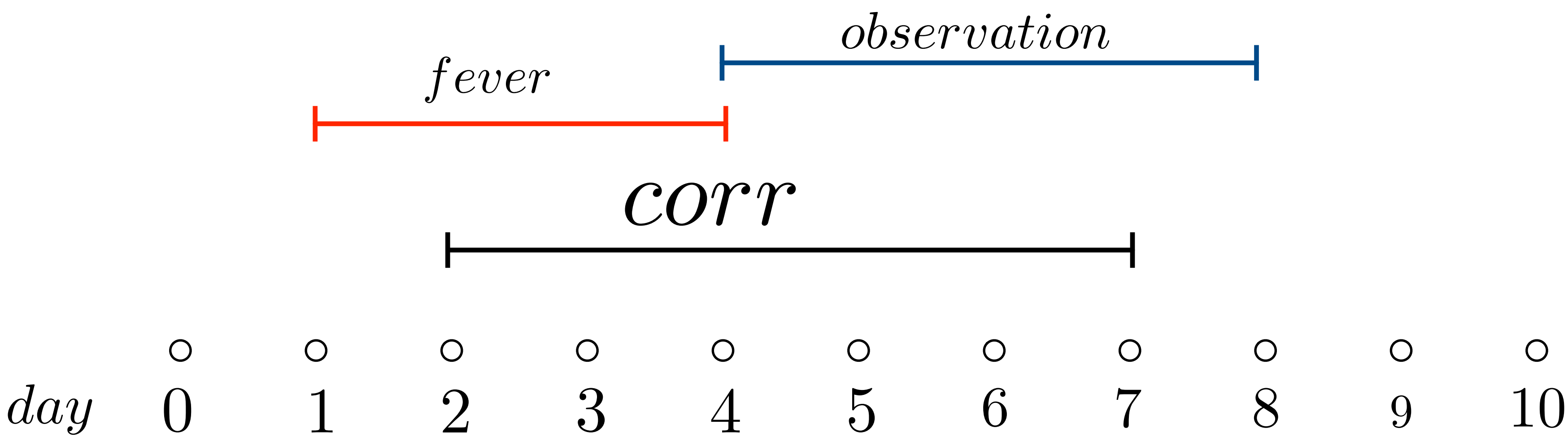


$$[G]\left(\begin{array}{c} corr \rightarrow [B]\neg corr \\ fever \rightarrow [B](\neg len_0 \rightarrow \langle A\rangle corr) \end{array}\right)$$

“from each fever interior point departs exactly one corr interval”

HS Synthesis Expressivity

core must encode a total **injective** function between the interior of a fever interval and the interior of its adjacent observation interval

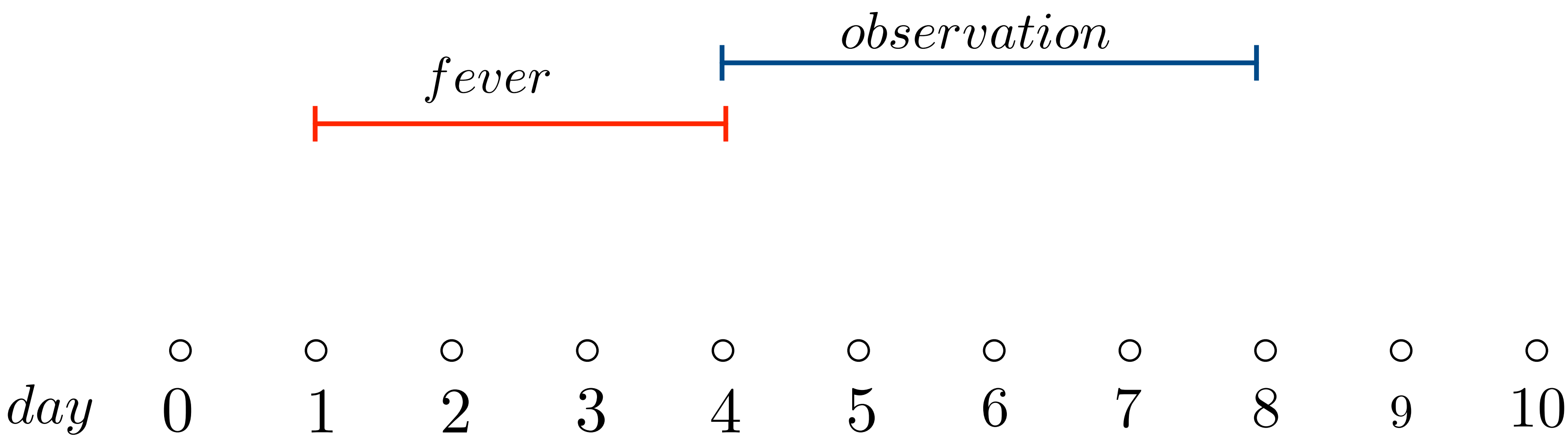


$$[G]\left(\quad corr \rightarrow \langle B\rangle\langle A\rangle o \wedge [B](\langle A\rangle o \rightarrow [B][A]\neg o) \right)$$

“each corr interval crosses exactly one observation”

HS Synthesis Expressivity

core must encode a total **injective** function between the interior of a fever interval and the interior of its adjacent observation interval



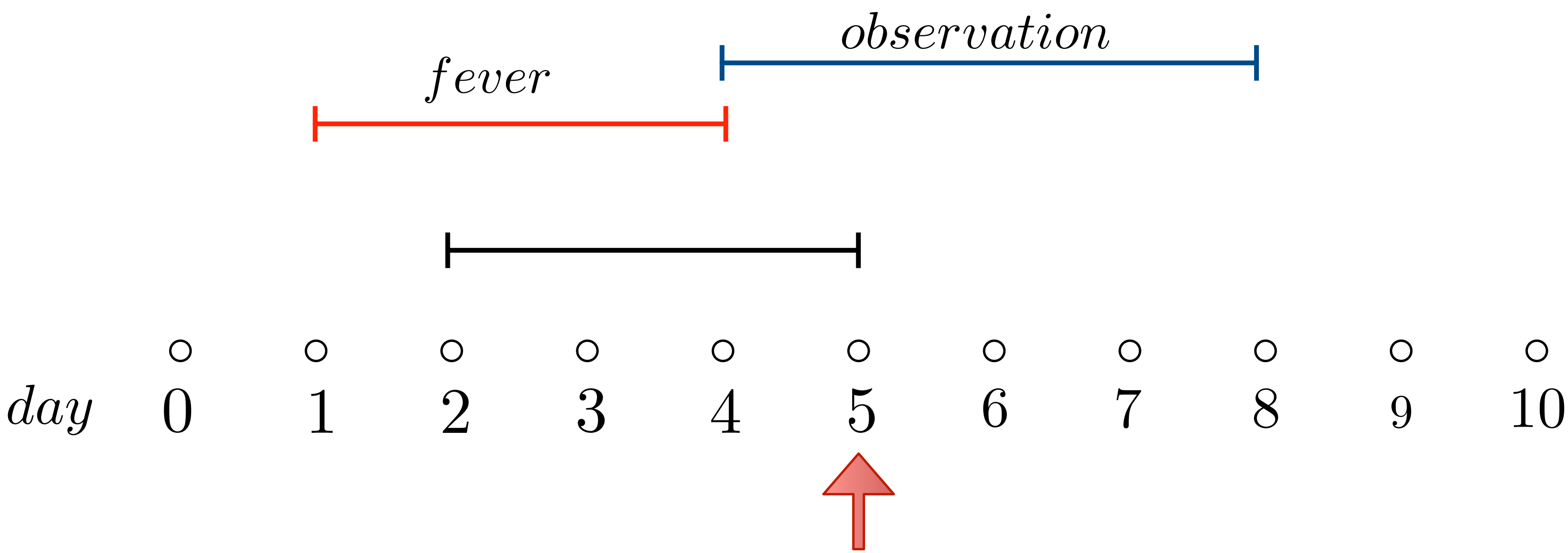
$$[G]\ \begin{pmatrix} h \wedge \langle B\rangle corr \wedge [B][B]\neg corr \rightarrow [A]k \\ \neg h \wedge \langle B\rangle corr \wedge [B][B]\neg corr \rightarrow [A]\neg k \end{pmatrix}$$

$$h \in \mathcal{P}_E, k \in \mathcal{P}_S$$

"injectivity"

HS Synthesis Expressivity

core must encode a total **injective** function between the interior of a fever interval and the interior of its adjacent observation interval



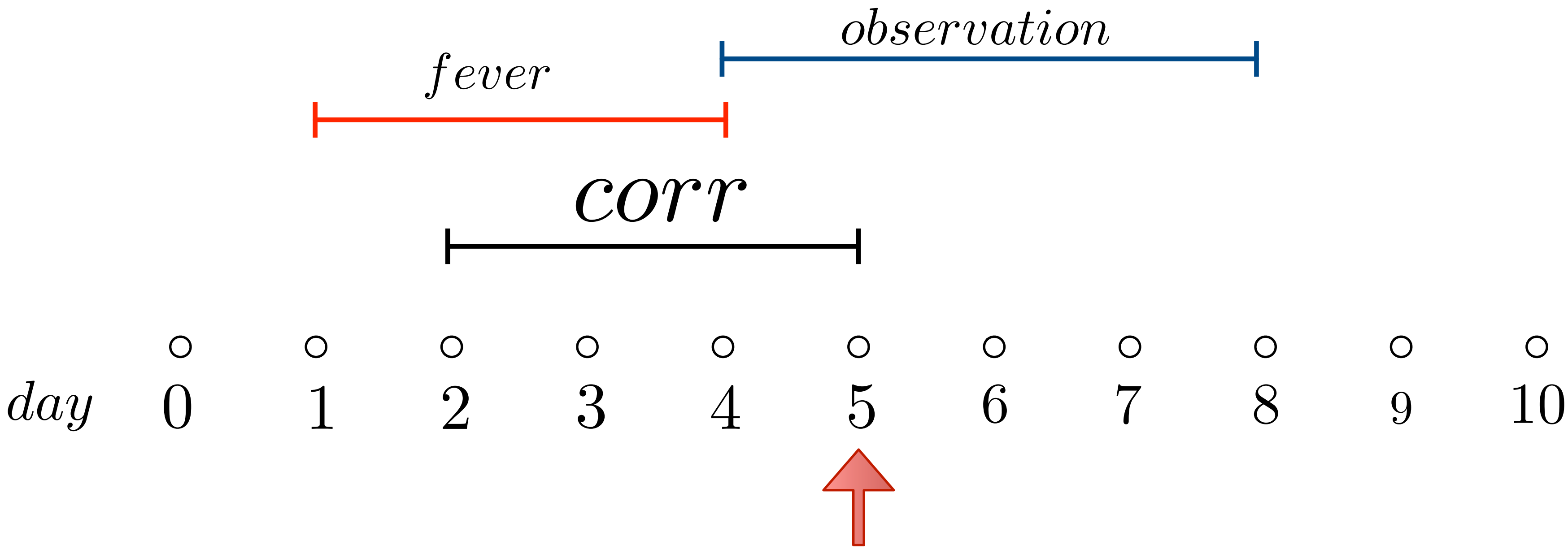
$$[G]\begin{pmatrix} h \wedge \langle B\rangle corr \wedge [B][B]\neg corr \rightarrow [A]k \\ \neg h \wedge \langle B\rangle corr \wedge [B][B]\neg corr \rightarrow [A]\neg k \end{pmatrix}$$

$$h \in \mathcal{P}_E, k \in \mathcal{P}_S$$

"injectivity"

HS Synthesis Expressivity

core must encode a total **injective** function between the interior of a fever interval and the interior of its adjacent observation interval



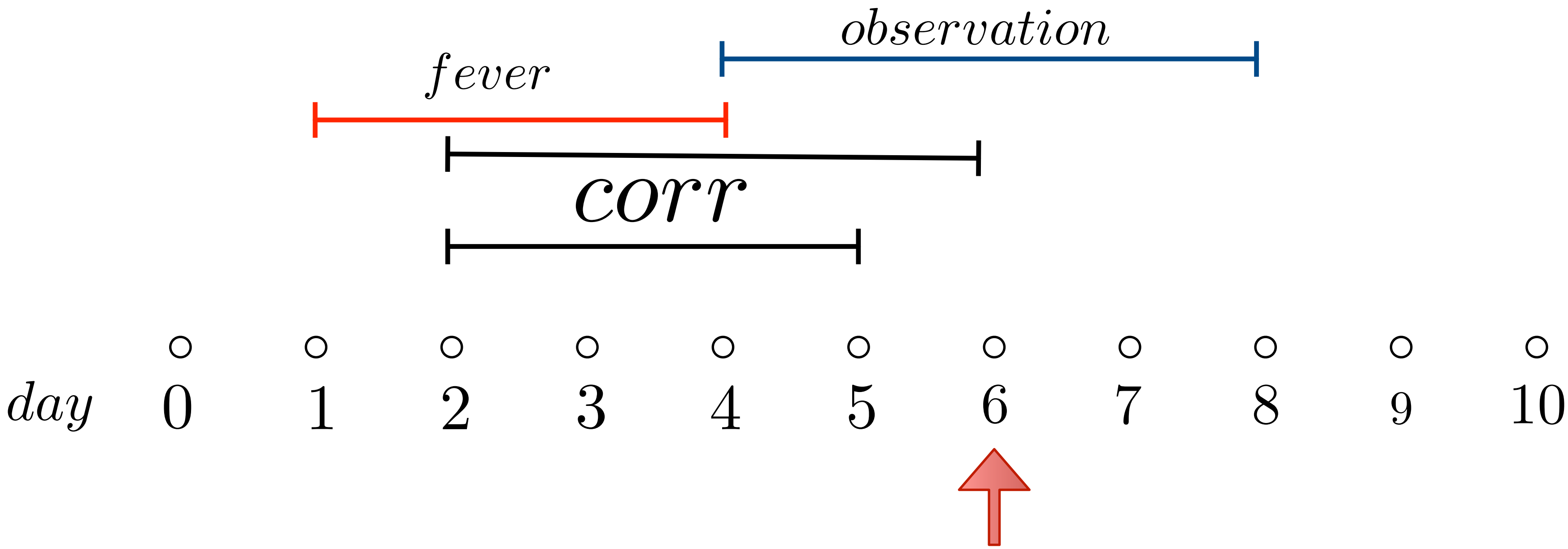
$$[G]\begin{pmatrix} h \wedge \langle B\rangle corr \wedge [B][B]\neg corr \rightarrow [A]k \\ \neg h \wedge \langle B\rangle corr \wedge [B][B]\neg corr \rightarrow [A]\neg k \end{pmatrix}$$

$$h \in \mathcal{P}_E, k \in \mathcal{P}_S$$

“injectivity”

HS Synthesis Expressivity

core must encode a total **injective** function between the interior of a fever interval and the interior of its adjacent observation interval



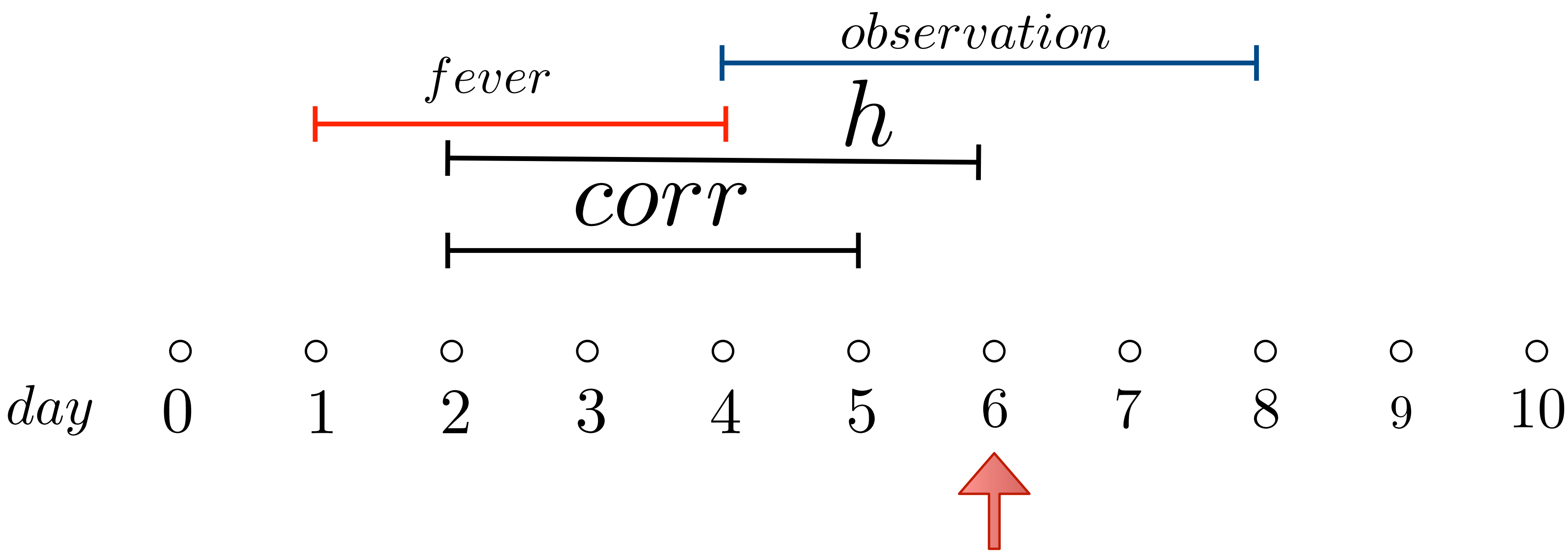
$$[G]\begin{pmatrix} h \wedge \langle B\rangle corr \wedge [B][B]\neg corr \rightarrow [A]k \\ \neg h \wedge \langle B\rangle corr \wedge [B][B]\neg corr \rightarrow [A]\neg k \end{pmatrix}$$

$$h \in \mathcal{P}_E, k \in \mathcal{P}_S$$

"injectivity"

HS Synthesis Expressivity

core must encode a total **injective** function between the interior of a fever interval and the interior of its adjacent observation interval



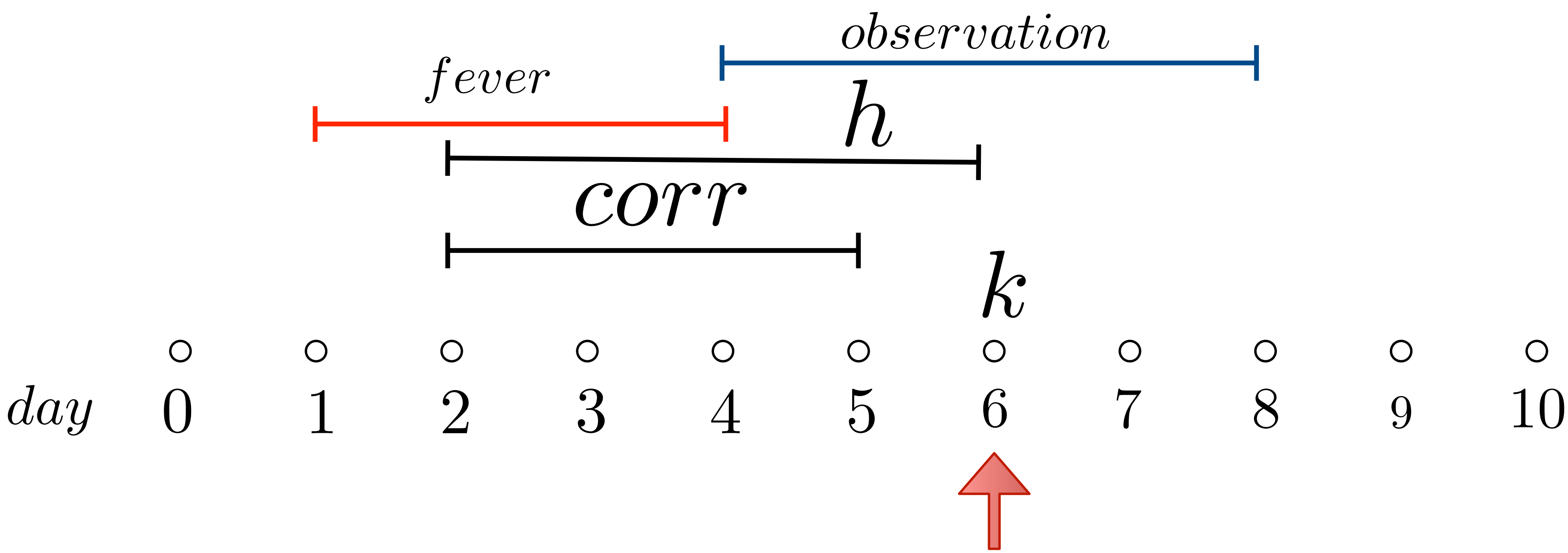
$$[G]\begin{pmatrix} h \wedge \langle B\rangle corr \wedge [B][B]\neg corr \rightarrow [A]k \\ \neg h \wedge \langle B\rangle corr \wedge [B][B]\neg corr \rightarrow [A]\neg k \end{pmatrix}$$

$$h \in \mathcal{P}_E, k \in \mathcal{P}_S$$

"injectivity"

HS Synthesis Expressivity

core must encode a total **injective** function between the interior of a fever interval and the interior of its adjacent observation interval



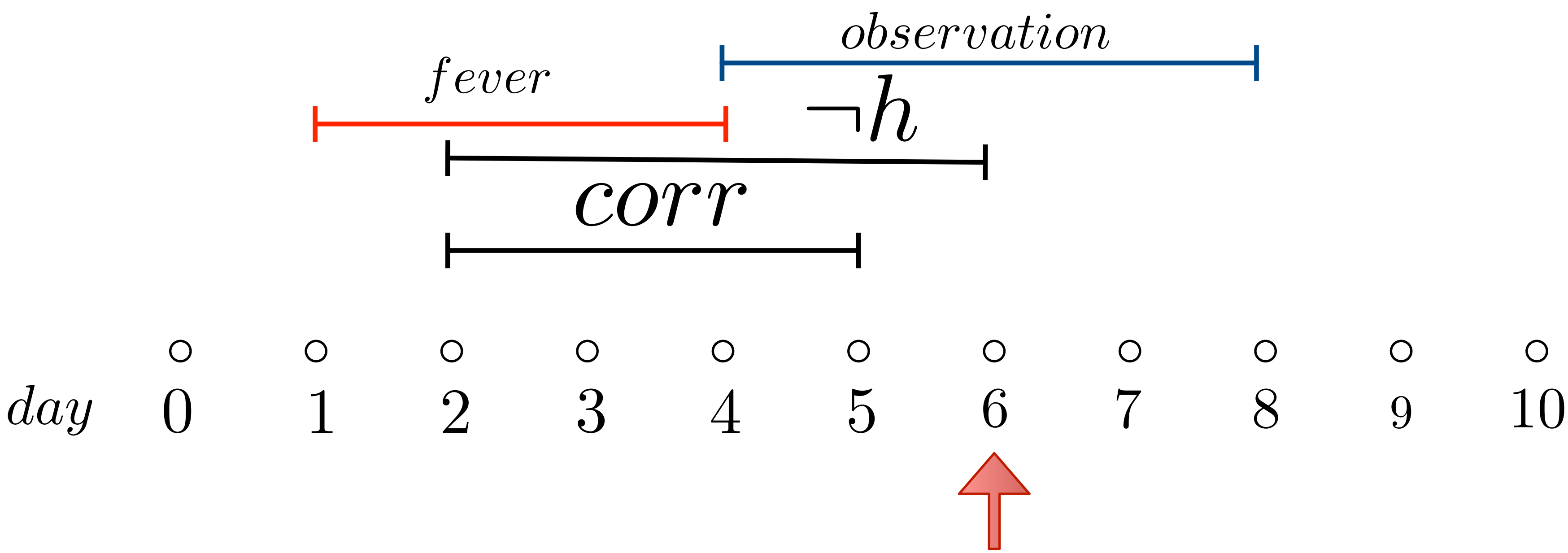
$$[G]\begin{pmatrix} h \wedge \langle B\rangle corr \wedge [B][B]\neg corr \rightarrow [A]k \\ \neg h \wedge \langle B\rangle corr \wedge [B][B]\neg corr \rightarrow [A]\neg k \end{pmatrix}$$

$$h \in \mathcal{P}_E, k \in \mathcal{P}_S$$

"injectivity"

HS Synthesis Expressivity

core must encode a total **injective** function between the interior of a fever interval and the interior of its adjacent observation interval



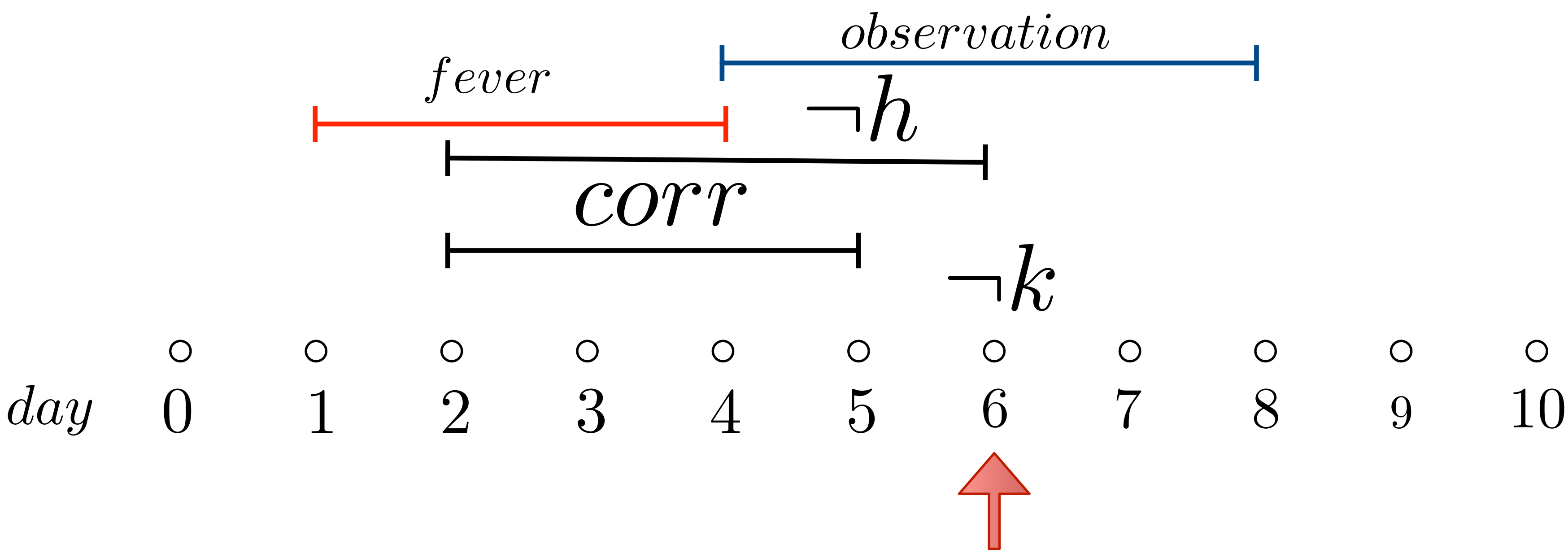
$$[G]\begin{pmatrix} h \wedge \langle B\rangle corr \wedge [B][B]\neg corr \rightarrow [A]k \\ \neg h \wedge \langle B\rangle corr \wedge [B][B]\neg corr \rightarrow [A]\neg k \end{pmatrix}$$

$$h \in \mathcal{P}_E, k \in \mathcal{P}_S$$

"injectivity"

HS Synthesis Expressivity

core must encode a total **injective** function between the interior of a fever interval and the interior of its adjacent observation interval



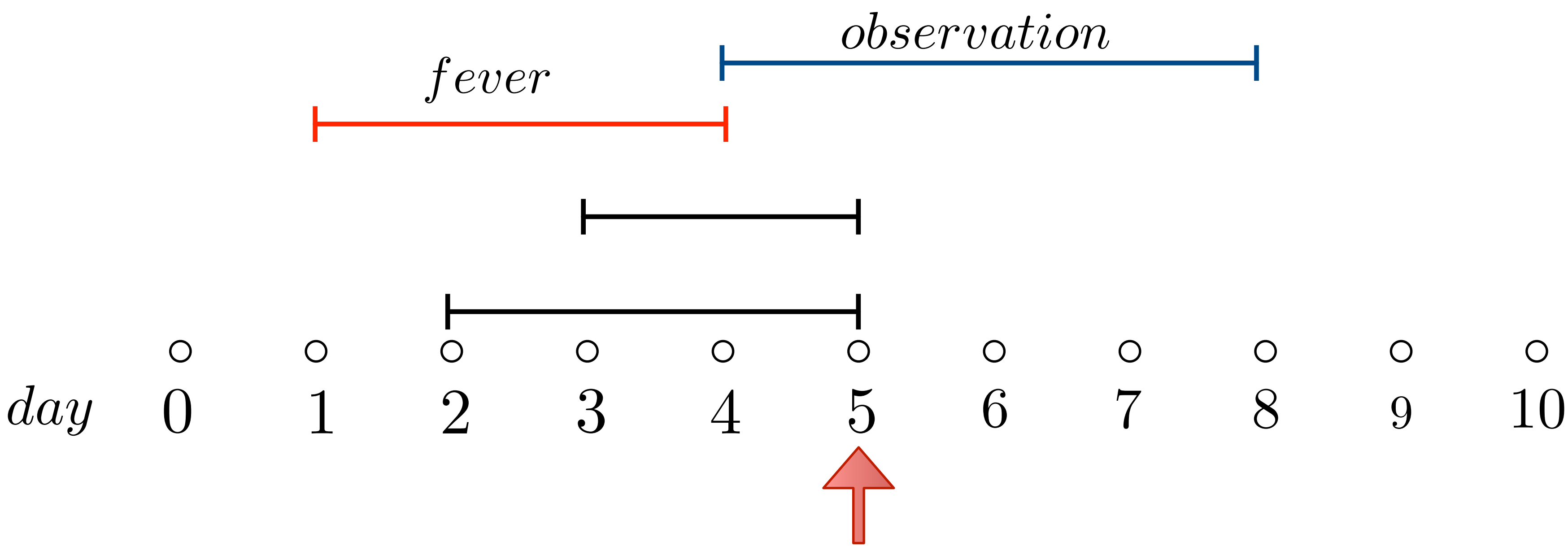
$$[G]\begin{pmatrix} h \wedge \langle B\rangle corr \wedge [B][B]\neg corr \rightarrow [A]k \\ \neg h \wedge \langle B\rangle corr \wedge [B][B]\neg corr \rightarrow [A]\neg k \end{pmatrix}$$

$$h \in \mathcal{P}_E, k \in \mathcal{P}_S$$

"injectivity"

HS Synthesis Expressivity

core must encode a total **injective** function between the interior of a fever interval and the interior of its adjacent observation interval



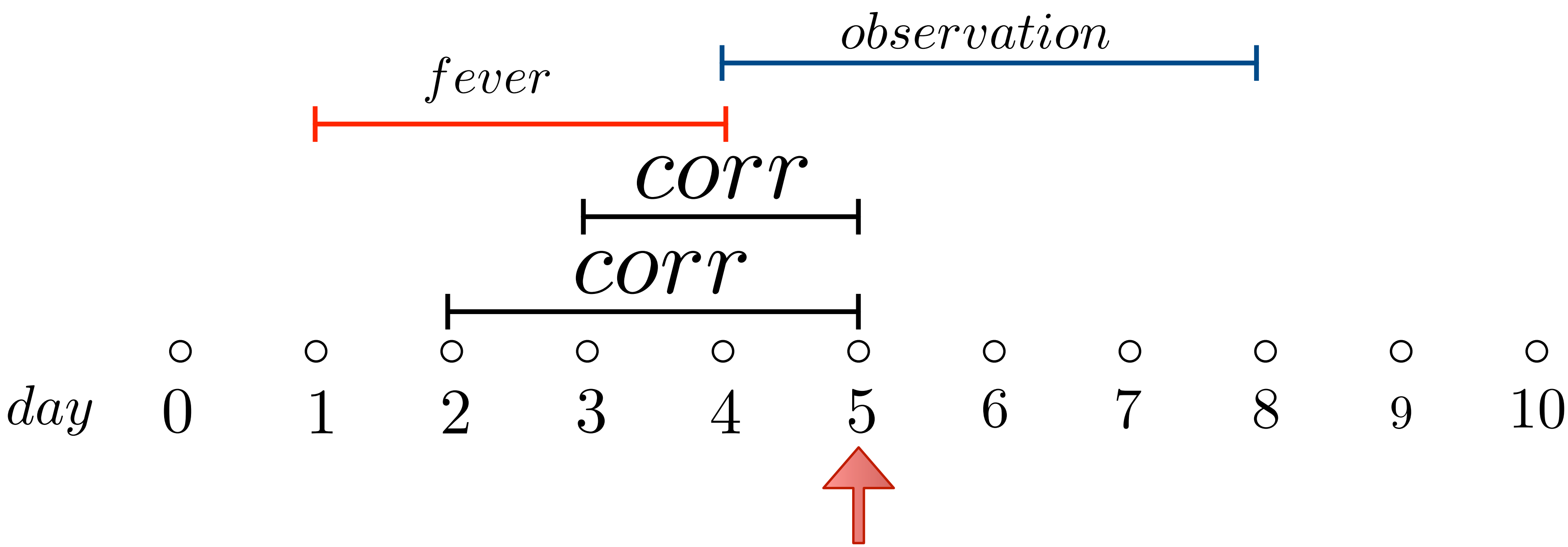
$$[G]\begin{pmatrix} h \wedge \langle B\rangle corr \wedge [B][B]\neg corr \rightarrow [A]k \\ \neg h \wedge \langle B\rangle corr \wedge [B][B]\neg corr \rightarrow [A]\neg k \end{pmatrix}$$

$$h \in \mathcal{P}_E, k \in \mathcal{P}_S$$

"injectivity"

HS Synthesis Expressivity

core must encode a total **injective** function between the interior of a fever interval and the interior of its adjacent observation interval



$$[G]\begin{pmatrix} h \wedge \langle B\rangle corr \wedge [B][B]\neg corr \rightarrow [A]k \\ \neg h \wedge \langle B\rangle corr \wedge [B][B]\neg corr \rightarrow [A]\neg k \end{pmatrix}$$

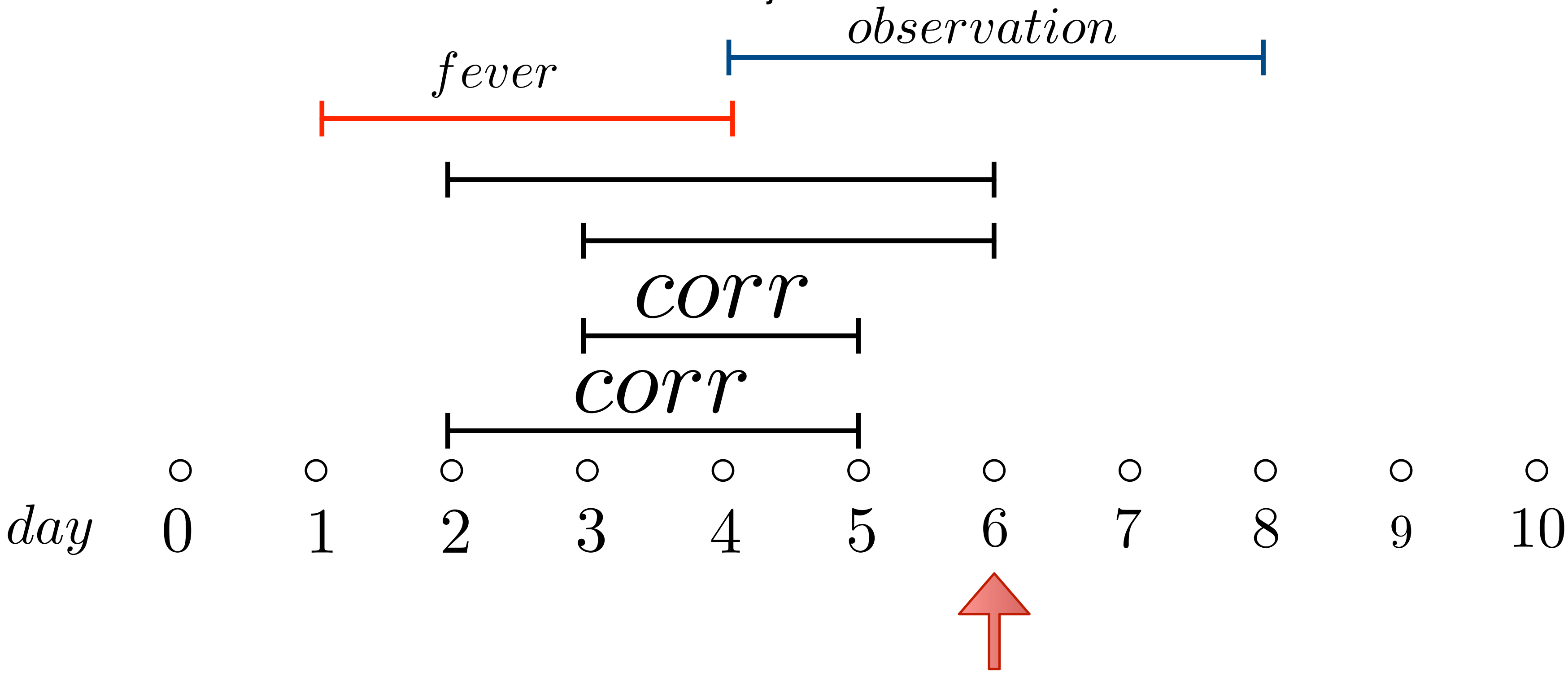
$$h \in \mathcal{P}_E, k \in \mathcal{P}_S$$

"injectivity"

Interval Based Synthesis

HS Synthesis Expressivity

core must encode a total injective function between the interior of a fever interval and the interior of its adjacent observation interval



$$[G]\begin{pmatrix} h \wedge \langle B\rangle corr \wedge [B][B]\neg corr \rightarrow [A]k \\ \neg h \wedge \langle B\rangle corr \wedge [B][B]\neg corr \rightarrow [A]\neg k \end{pmatrix}$$

$$h \in \mathcal{P}_E, k \in \mathcal{P}_S$$

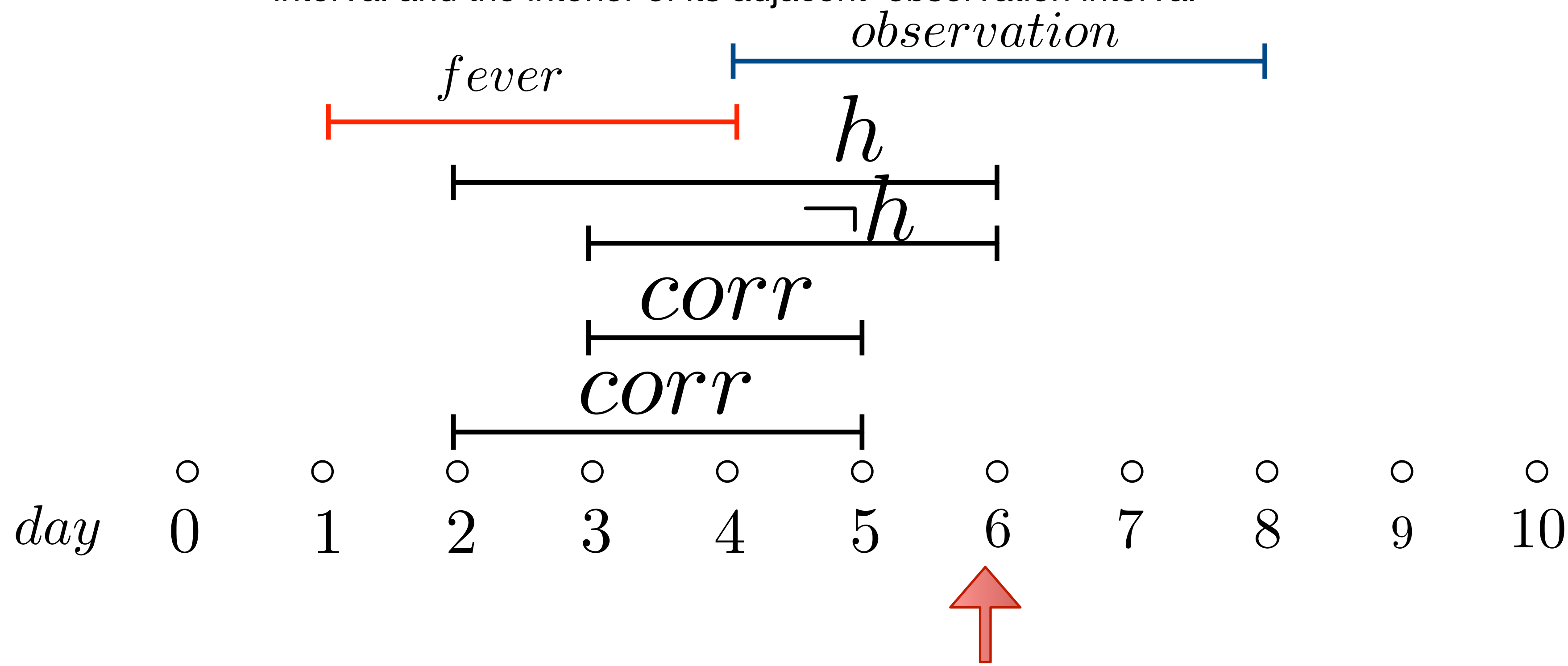
"injectivity"

Pietro Sala Ph.D.

Interval Based Synthesis

HS Synthesis Expressivity

core must encode a total **injective** function between the interior of a fever interval and the interior of its adjacent observation interval



$$[G]\begin{pmatrix} h \wedge \langle B\rangle corr \wedge [B][B]\neg corr \rightarrow [A]k \\ \neg h \wedge \langle B\rangle corr \wedge [B][B]\neg corr \rightarrow [A]\neg k \end{pmatrix}$$

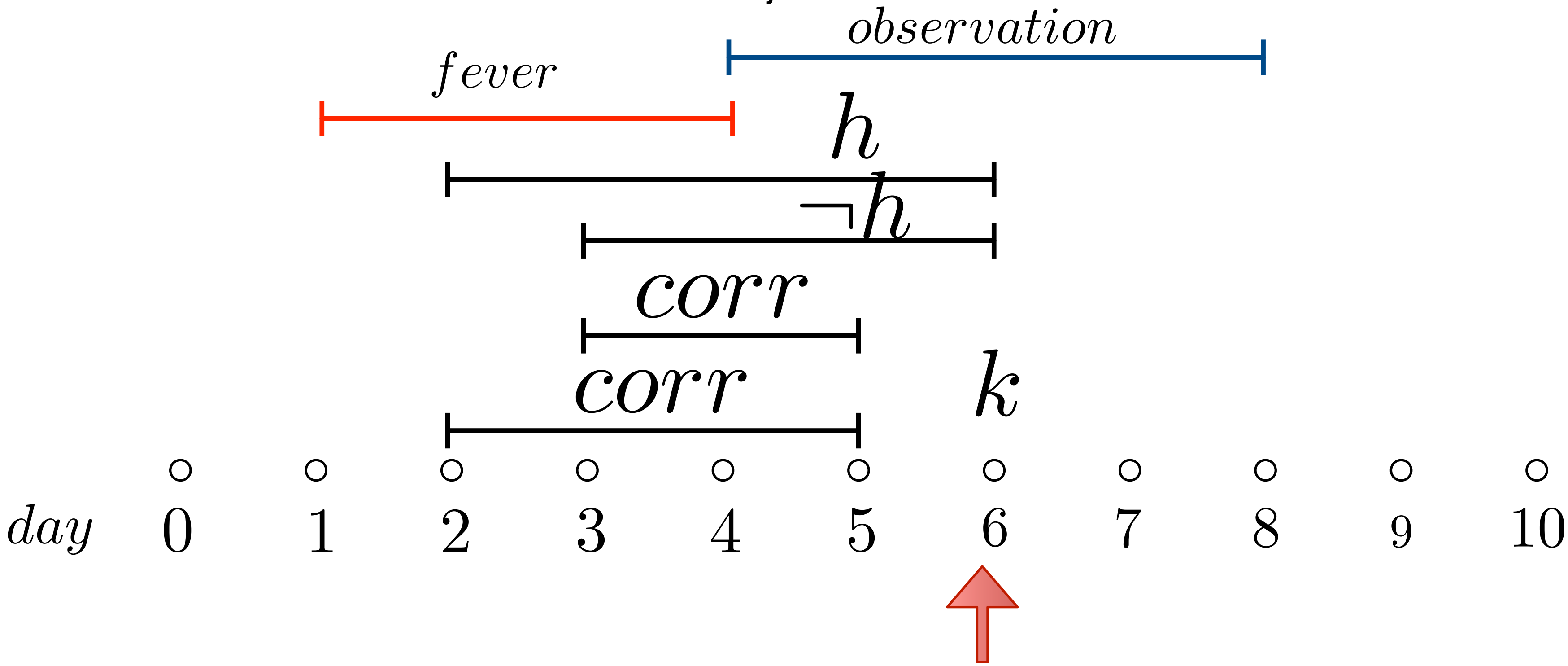
$$h \in \mathcal{P}_E, k \in \mathcal{P}_S$$

"injectivity"

Pietro Sala Ph.D.

HS Synthesis Expressivity

core must encode a total **injective** function between the interior of a fever interval and the interior of its adjacent observation interval



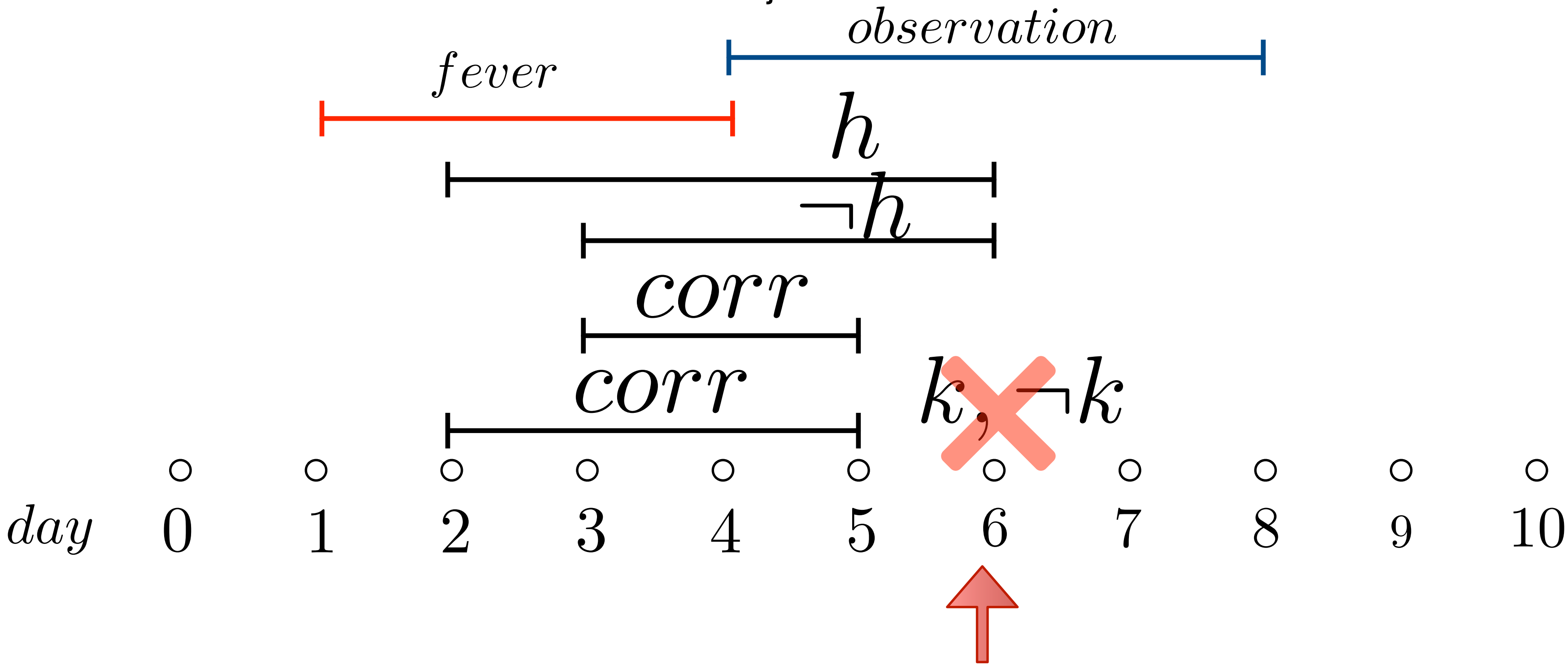
$$[G]\begin{pmatrix} h \wedge \langle B\rangle corr \wedge [B][B]\neg corr \rightarrow [A]k \\ \neg h \wedge \langle B\rangle corr \wedge [B][B]\neg corr \rightarrow [A]\neg k \end{pmatrix}$$

$$h \in \mathcal{P}_E, k \in \mathcal{P}_S$$

"injectivity"

HS Synthesis Expressivity

core must encode a total **injective** function between the interior of a fever interval and the interior of its adjacent observation interval



$$[G]\begin{pmatrix} h \wedge \langle B\rangle corr \wedge [B][B]\neg corr \rightarrow [A]k \\ \neg h \wedge \langle B\rangle corr \wedge [B][B]\neg corr \rightarrow [A]\neg k \end{pmatrix}$$

$$h \in \mathcal{P}_E, k \in \mathcal{P}_S$$

"injectivity"

Adding an Equivalence Relation

Let us extend HS with an equivalence relation:

$$HS\sim \quad \varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle * \rangle\varphi \mid \sim$$

Adding an Equivalence Relation

Let us extend HS with an equivalence relation:

$$HS\sim \quad \varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle * \rangle \varphi \mid \boxed{\sim}$$

$$\sim \in \mathcal{P}$$

Adding an Equivalence Relation

Let us extend HS with an equivalence relation:

$$HS\sim \quad \varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle * \rangle\varphi \mid \boxed{\sim}$$

$$\sim \in \mathcal{P}$$

$$\forall x \left(\begin{array}{l} \sim \in \mathcal{V}([x,x]) \\ \forall y \forall z \left(\begin{array}{l} \sim \in \mathcal{V}([x,y]) \wedge \sim \in \mathcal{V}([y,z]) \rightarrow \sim \in \mathcal{V}([x,z]) \\ \sim \in \mathcal{V}([x,y]) \wedge \sim \in \mathcal{V}([x,z]) \rightarrow \sim \in \mathcal{V}([y,z]) \\ \sim \in \mathcal{V}([x,z]) \wedge \sim \in \mathcal{V}([y,z]) \rightarrow \sim \in \mathcal{V}([x,y]) \end{array} \right) \end{array} \right)$$

Adding an Equivalence Relation

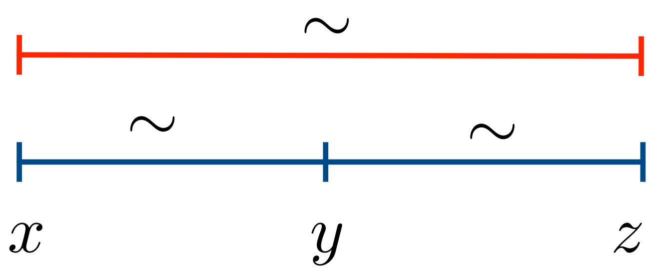
Let us extend HS with an equivalence relation:

$$HS\sim \qquad \varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle * \rangle \varphi \mid \boxed{\sim}$$

$$\sim \in \mathcal{P}$$

Semantics constrain:

$$\forall x \left(\begin{array}{l} \sim \in \mathcal{V}([x,x]) \\ \forall y \forall z \left(\begin{array}{l} \sim \in \mathcal{V}([x,y]) \wedge \sim \in \mathcal{V}([y,z]) \rightarrow \sim \in \mathcal{V}([x,z]) \\ \sim \in \mathcal{V}([x,y]) \wedge \sim \in \mathcal{V}([x,z]) \rightarrow \sim \in \mathcal{V}([y,z]) \\ \sim \in \mathcal{V}([x,z]) \wedge \sim \in \mathcal{V}([y,z]) \rightarrow \sim \in \mathcal{V}([x,y]) \end{array} \right) \end{array} \right)$$



Adding an Equivalence Relation

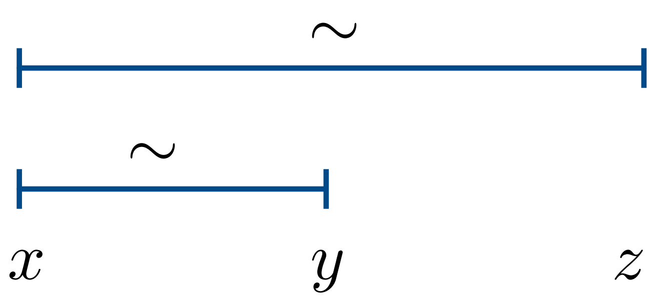
Let us extend HS with an equivalence relation:

$$HS\sim \qquad \varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle * \rangle \varphi \mid \boxed{\sim}$$

$$\sim \in \mathcal{P}$$

Semantics constrain:

$$\forall x \left(\begin{array}{l} \sim\in \mathcal{V}([x,x]) \\ \forall y \forall z \left(\begin{array}{l} \sim\in \mathcal{V}([x,y]) \wedge \sim\in \mathcal{V}([y,z]) \rightarrow \sim\in \mathcal{V}([x,z]) \\ \sim\in \mathcal{V}([x,y]) \wedge \sim\in \mathcal{V}([x,z]) \rightarrow \sim\in \mathcal{V}([y,z]) \\ \sim\in \mathcal{V}([x,z]) \wedge \sim\in \mathcal{V}([y,z]) \rightarrow \sim\in \mathcal{V}([x,y]) \end{array} \right) \end{array} \right)$$



Adding an Equivalence Relation

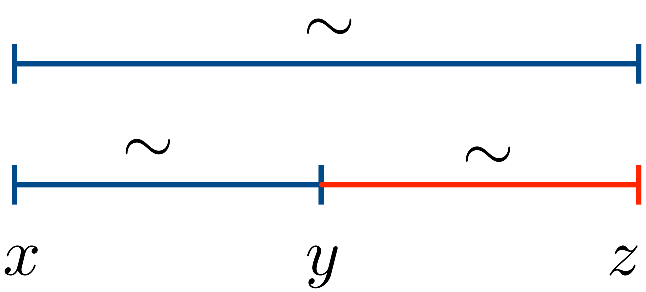
Let us extend HS with an equivalence relation:

$$HS\sim \quad \varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle * \rangle \varphi \mid \boxed{\sim}$$

$$\sim \in \mathcal{P}$$

Semantics constrain:

$$\forall x \left(\begin{array}{l} \sim \in \mathcal{V}([x,x]) \\ \forall y \forall z \left(\begin{array}{l} \sim \in \mathcal{V}([x,y]) \wedge \sim \in \mathcal{V}([y,z]) \rightarrow \sim \in \mathcal{V}([x,z]) \\ \sim \in \mathcal{V}([x,y]) \wedge \sim \in \mathcal{V}([x,z]) \rightarrow \sim \in \mathcal{V}([y,z]) \\ \sim \in \mathcal{V}([x,z]) \wedge \sim \in \mathcal{V}([y,z]) \rightarrow \sim \in \mathcal{V}([x,y]) \end{array} \right) \end{array} \right)$$



Adding an Equivalence Relation

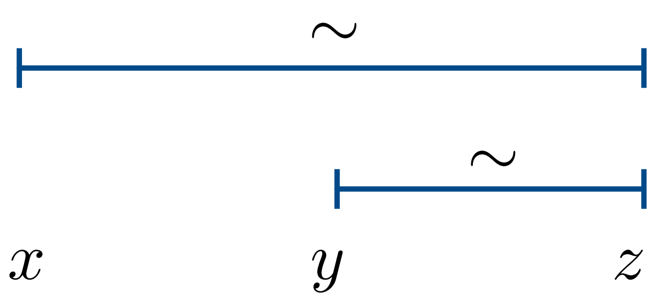
Let us extend HS with an equivalence relation:

$$HS\sim \qquad \varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle * \rangle \varphi \mid \boxed{\sim}$$

$$\sim \in \mathcal{P}$$

Semantics constrain:

$$\forall x \left(\begin{array}{l} \sim\in \mathcal{V}([x,x]) \\ \forall y \forall z \left(\begin{array}{l} \sim\in \mathcal{V}([x,y]) \wedge \sim\in \mathcal{V}([y,z]) \rightarrow \sim\in \mathcal{V}([x,z]) \\ \sim\in \mathcal{V}([x,y]) \wedge \sim\in \mathcal{V}([x,z]) \rightarrow \sim\in \mathcal{V}([y,z]) \\ \sim\in \mathcal{V}([x,z]) \wedge \sim\in \mathcal{V}([y,z]) \rightarrow \sim\in \mathcal{V}([x,y]) \end{array} \right) \end{array} \right)$$



Adding an Equivalence Relation

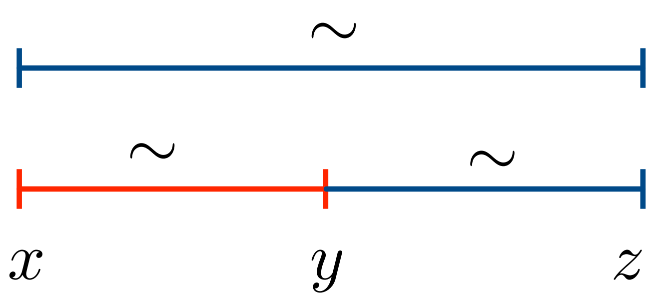
Let us extend HS with an equivalence relation:

$$HS\sim \qquad \varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle * \rangle \varphi \mid \boxed{\sim}$$

$$\sim \in \mathcal{P}$$

Semantics constrain:

$$\forall x \left(\begin{array}{l} \sim \in \mathcal{V}([x,x]) \\ \forall y \forall z \left(\begin{array}{l} \sim \in \mathcal{V}([x,y]) \wedge \sim \in \mathcal{V}([y,z]) \rightarrow \sim \in \mathcal{V}([x,z]) \\ \sim \in \mathcal{V}([x,y]) \wedge \sim \in \mathcal{V}([x,z]) \rightarrow \sim \in \mathcal{V}([y,z]) \\ \sim \in \mathcal{V}([x,z]) \wedge \sim \in \mathcal{V}([y,z]) \rightarrow \sim \in \mathcal{V}([x,y]) \end{array} \right) \end{array} \right)$$



Adding an Equivalence Relation

In the realm of first order logics is not uncommon to extend a logic with one or more equivalence relations.

Mikolaj Bojanczyk, Anca Muscholl, Thomas Schwentick, Luc Segoufin, Claire David:
Two-Variable Logic on Words with Data. LICS 2006: 7-16

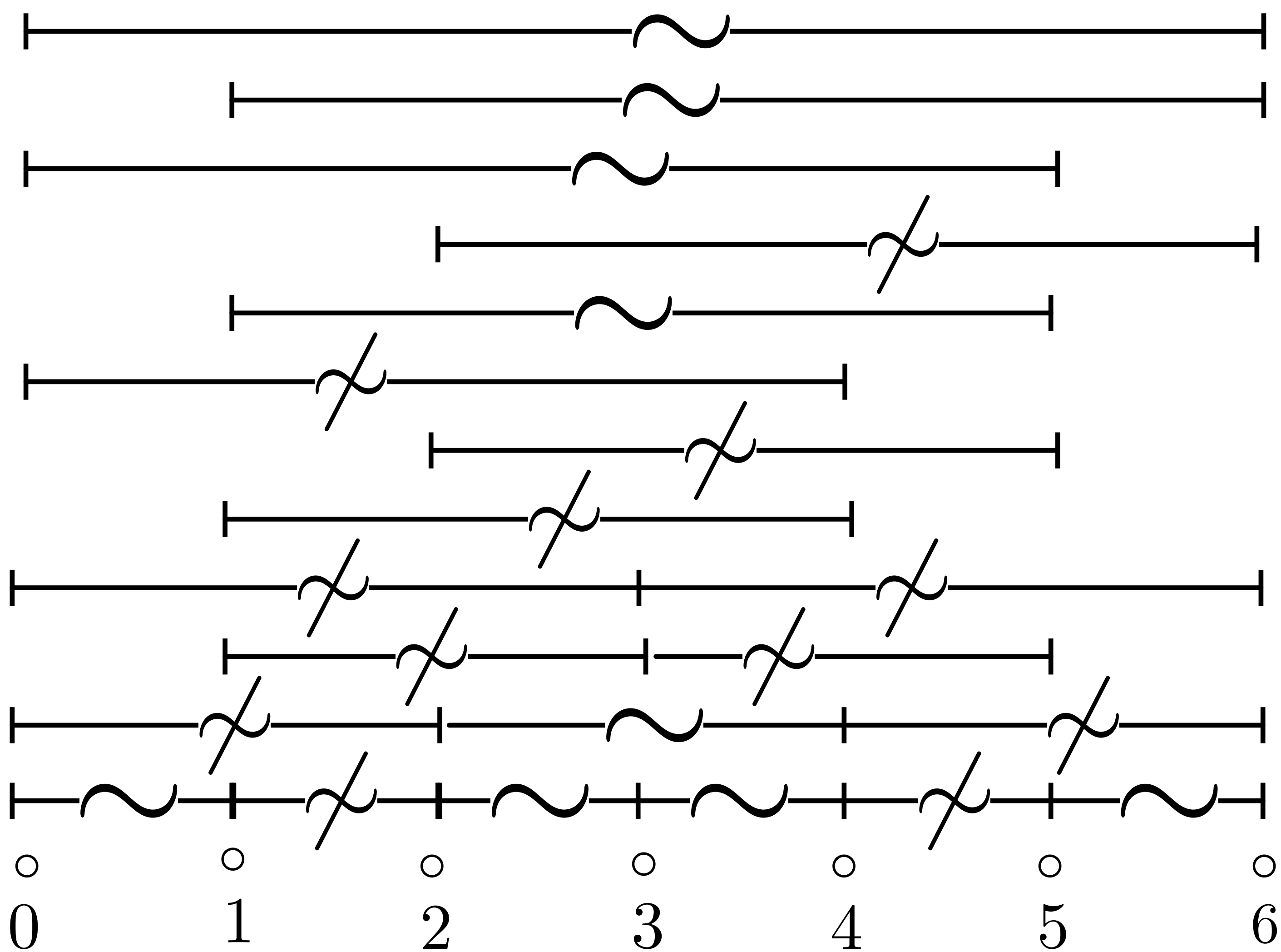
Emanuel Kieronski, Jakub Michaliszyn, Ian Pratt-Hartmann, Lidia Tendera:
Two-Variable First-Order Logic with Equivalence Closure. SIAM J. Comput. 43(3): 1012-1063 (2014)

Emanuel Kieronski, Lidia Tendera:
On Finite Satisfiability of Two-Variable First-Order Logic with Equivalence Relations. LICS 2009: 123-132

Interval Based Synthesis

Adding an Equivalence Relation

In the realm of interval temporal logic it makes even more sense since every $\sim$-equivalence class induces a sub-structure which is itself an interval structure.

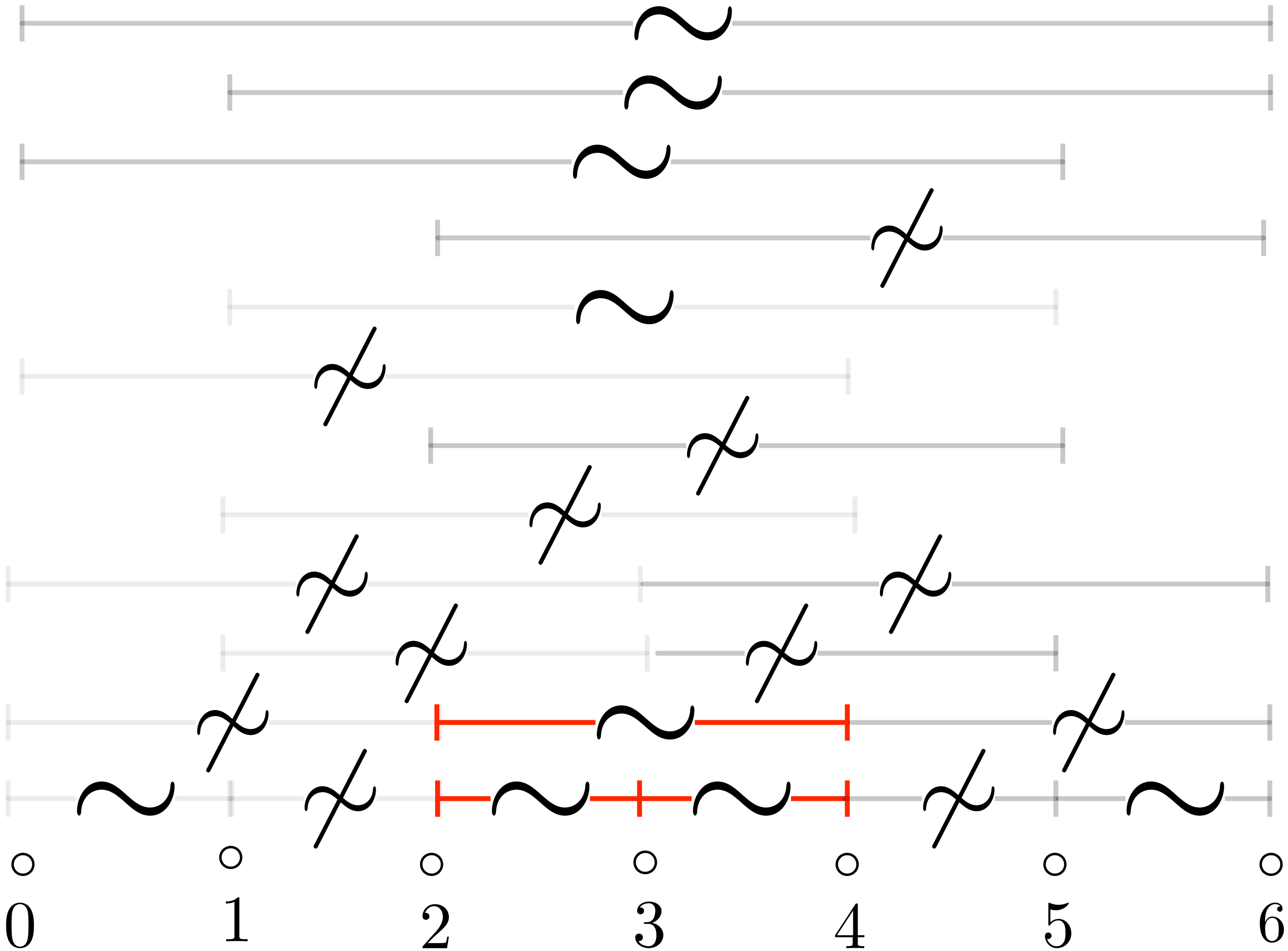


Pietro Sala Ph.D.

Interval Based Synthesis

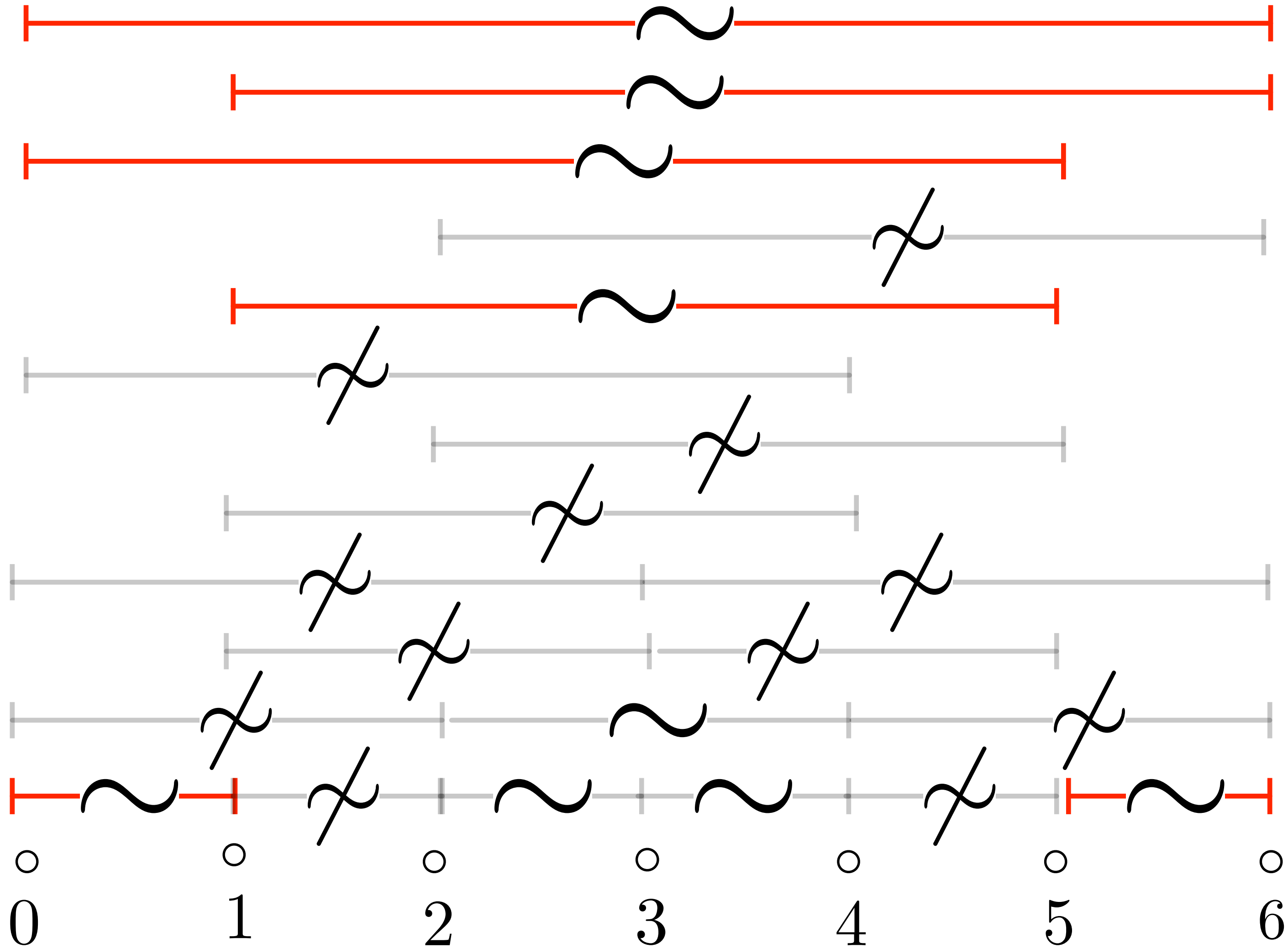
Adding an Equivalence Relation

In the realm of interval temporal logic it makes even more sense since every $\sim$-equivalence class induces a sub-structure which is itself an interval structure.



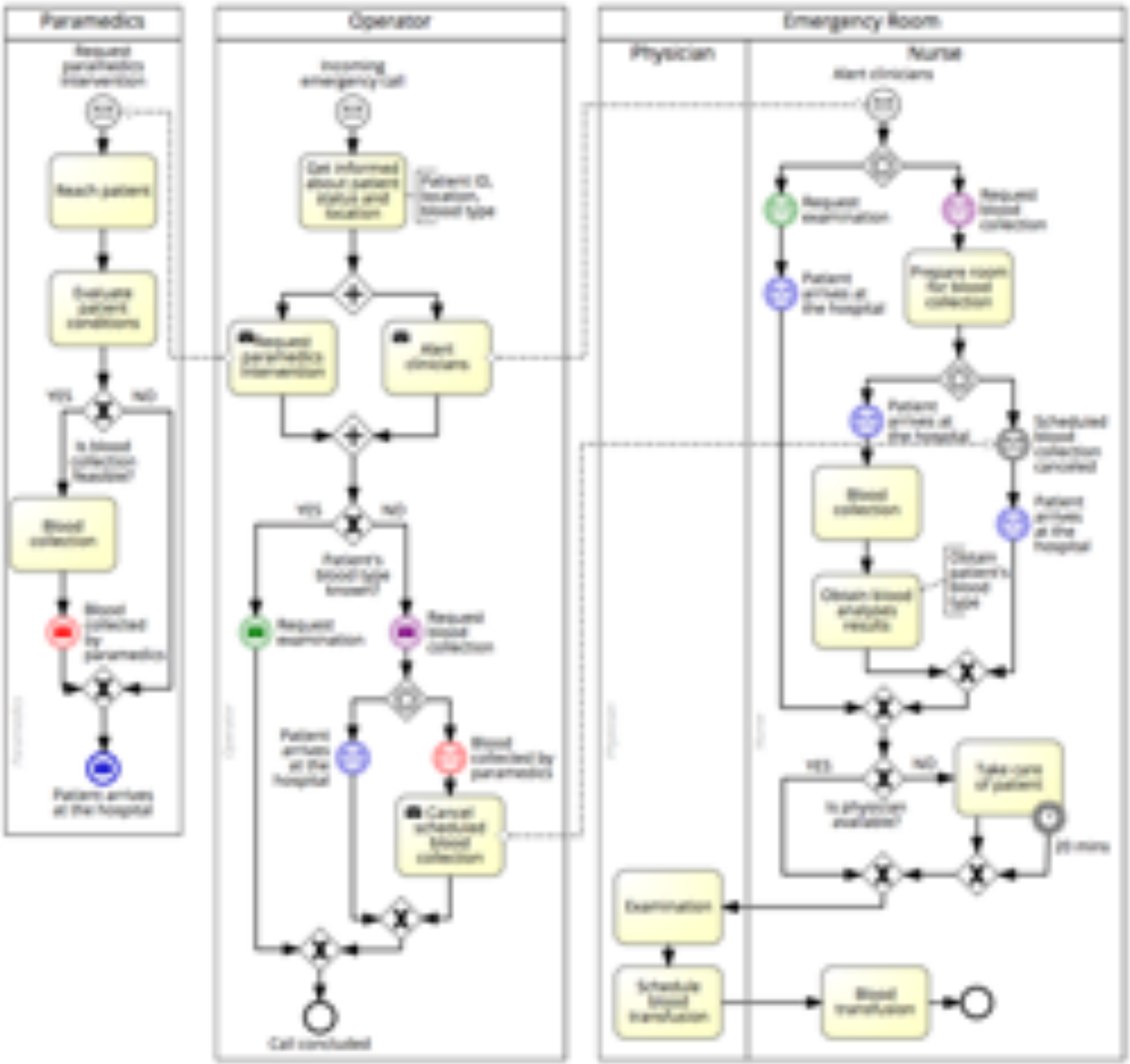
Adding an Equivalence Relation

In the realm of interval temporal logic it makes even more sense since every $\sim$-equivalence class induces a sub-structure which is itself an interval structure.



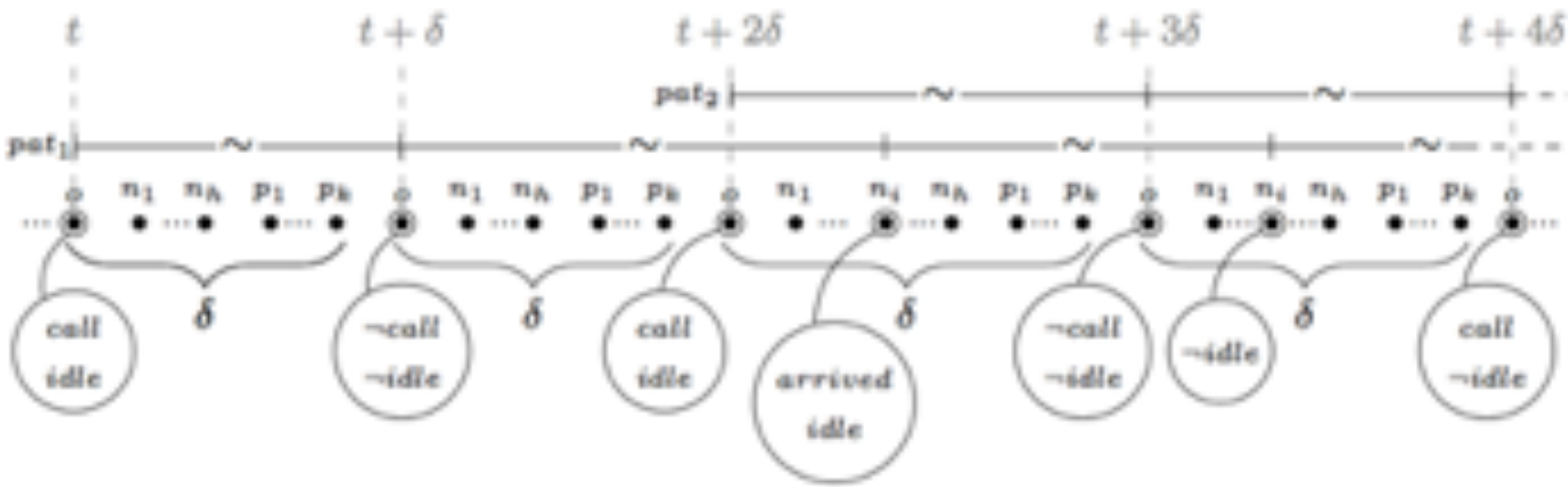
Interval Based Synthesis

Managing Multiple Emergency Room Process



Pietro Sala Ph.D.

Managing Multiple Emergency Room Process



We want to manage multiple instances (patients) of such process at the same time.
We are in presence of a limited amount of resources.

Our points are partitioned in consecutive windows containing exactly M points each of them labelled with a different resource/clinicians. Two clinicians cannot label the same point. Point inside the same window are considered simultaneous. Patients are simply equivalence relations that are passed between resources if a resource point in a window belongs to an equivalence it means that the relative clinicians is taking care of the specific patient

While the number of resources is a priori fixed there is no limit on the numeber of patients that may be created.

Decidability Status of Interval-Based Synthesis

In general if a logic features a decidable synthesis problem then its satisfiability problem is decidable.
The converse it is not true, however this holds for monadic second order theories.

Church synthesis problem for P is computable if and only if the monadic theory of $\langle \mathrm{Nat}, <, P \rangle$ is decidable.

Rabinovich, Alexander - The Church Synthesis Problem with Parameters
lmcs:1233 - Logical Methods in Computer Science, November 14, 2007, Volume 3, Issue 4

It does not hold for HS fragments:

Logic	Linear Order	Satisfiability	Synthesis
$AB\bar{B}$	Finite	Decidable (EXPSPACE-complete)	Decidable (**NonPrimitiveRecursive-hard**)
$AB\bar{B}$	N	Decidable (EXPSPACE-complete)	**Undecidable**
$AB\bar{B}\sim$	Finite	Decidable (NonPrimitiveRecursive-hard)	**Decidable** (NonPrimitiveRecursive-hard)
$AB\bar{B}\sim$	N	Undecidable	Undecidable
$A\bar{A}B\bar{B}$	Finite	Decidable (NonPrimitiveRecursive-hard)	**Undecidable**
$A\bar{A}B\bar{B}$	N	Undecidable	Undecidable
$A\bar{A}B\bar{B}\sim$	Finite	Undecidable	Undecidable
$A\bar{A}B\bar{B}\sim$	N	Undecidable	Undecidable

Conclusion

currently my research in interval based synthesis is focused on:

find suitable models for executing strategies without affecting the expressivity;

find mappings that turns strategies to human-readable formalisms (like BPMN);

provide decidability/undecidability results for the synthesis problem of the remaining fragments of HS;

study the HS synthesis problem on other kinds of linear orders (e.g. real numbers).