Maximally permissive supervisor control of timed discrete-event systems under partial observation

Ziteng Yang & Xiang Yin & Shaoyuan Li

Department of Computer Science, Shanghai Jiao Tong University

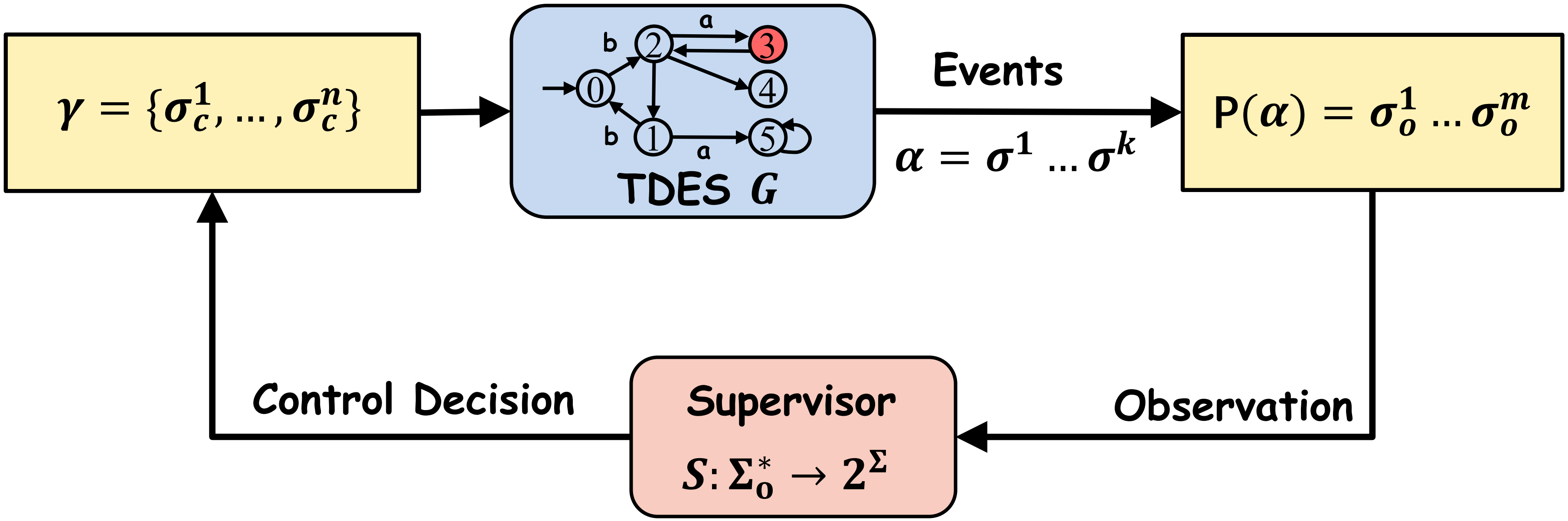
youngster@sjtu.edu.cn

21st IFAC World Congress

July 12-17, 2020, Berlin, Germany

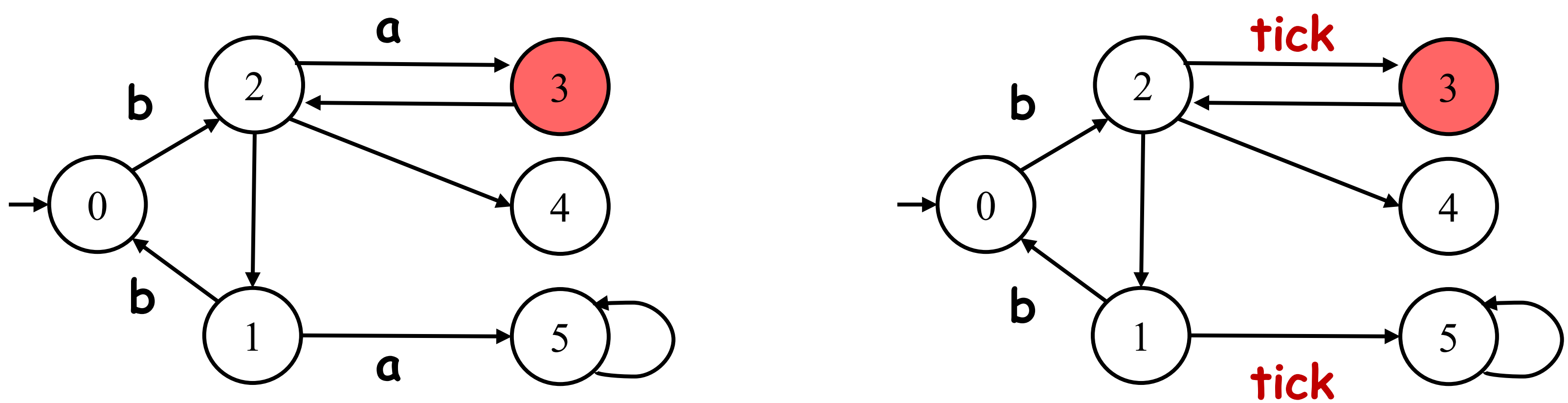
上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

Supervisory Control of DES



- **System:** DES $G = (X, \Sigma, \delta, x_0)$
- **Observable:** $\Sigma = \Sigma_o \dot{\cup} \Sigma_{uo}$ and $P: \Sigma^* \to \Sigma_o^*$
- **Controllable:** $\Sigma = \Sigma_c \dot{\cup} \Sigma_{uc}$
- **Supervisor:** $S: \Sigma_o^* \to 2^\Sigma$ satisfying $\Sigma_{uc} \subseteq S(\alpha)$ for any $\alpha \in \Sigma_o^*$

System Model of TDES

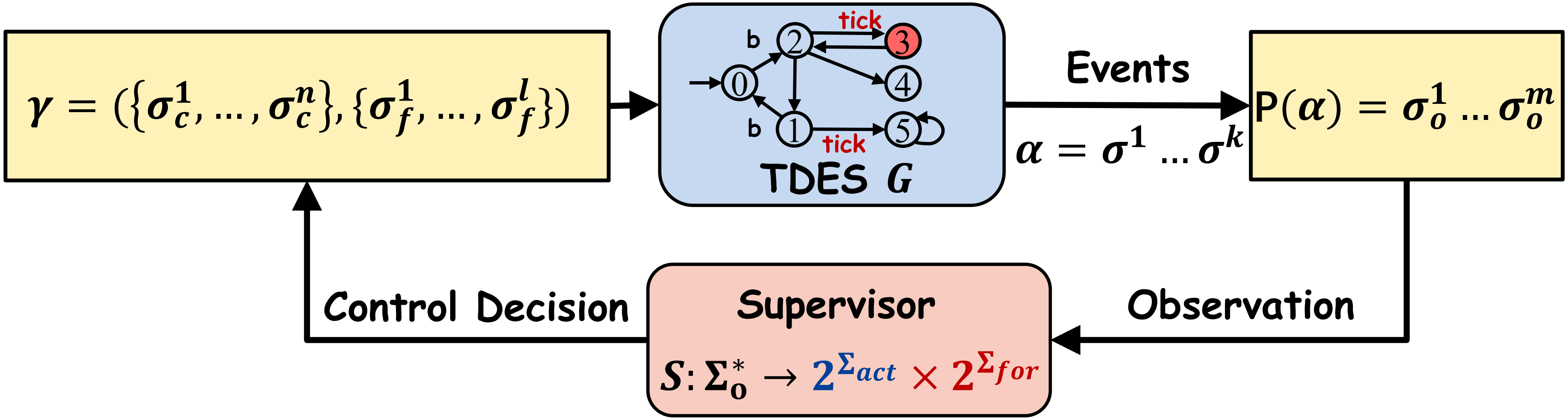


DES G'

TDES G

- **System:** TDES $G = (X, \Sigma, \delta, x_0)$
- **Event:** $\Sigma = \Sigma_{act} \dot{\cup} \{tick\}$
- **Language:** L(G), generated by G under supervisor S

Supervisory Control of TDES



- **System:** TDES $G = (X, \Sigma, \delta, x_0)$
- **Event:** $\Sigma = \Sigma_{act} \dot\cup \{tick\}$
- **Observable:** $\Sigma = \Sigma_o \dot\cup \Sigma_{uo}$ and $P: \Sigma^* \rightarrow \Sigma_o^*$
- **Controllable:** $\Sigma_{act} = \Sigma_c \dot\cup \Sigma_{uc}$, $\Sigma_{for} \subseteq \Sigma_{act}$
- **Supervisor:** $S: \Sigma_o^* \rightarrow 2^{\Sigma_{act}} \times 2^{\Sigma_{for}}$ satisfying $\Sigma_{uc} \subseteq S(\alpha) = (S_a(\alpha), S_f(\alpha))$

  and $S_f(\alpha) \in S_a(\alpha) \cap \Sigma_{for}$ for any $\alpha \in \Sigma_o^*$
- **Language:** L(S/G), generated by G under supervisor S

Literature Review

Timed DES and Supervisory Control

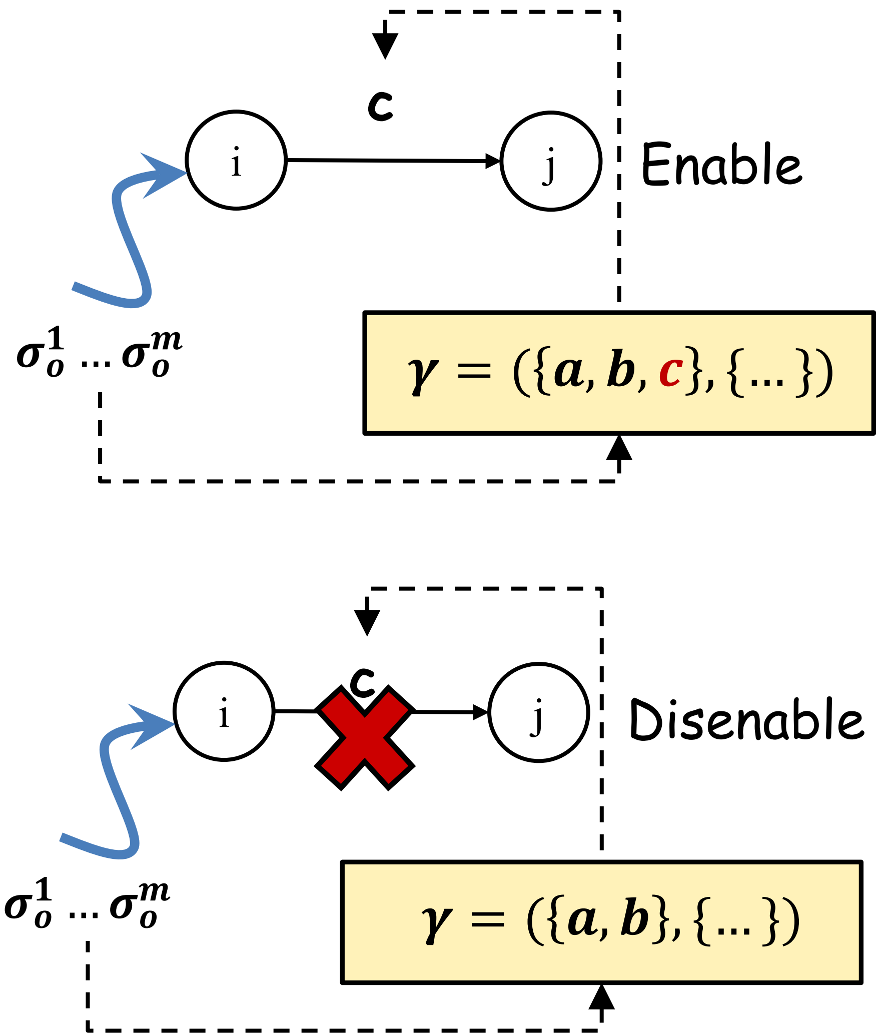
- Brandin, B. and Wonham, W. (1994). Supervisory control of timed discrete-event systems. *IEEE Trans. Automatic Control*, 39(2), 329–342.
- Takai, S. and Ushio, T. (2006). A new class of supervisors for timed discrete event systems under partial observation. *Discrete Event Dynamic Systems*, 16(2), 257–278.

Supervisor Synthesis

- Yin, X. and Lafortune, S. (2016a). Synthesis of maximally permissive supervisors for partially observed discrete event systems. IEEE Trans. Automatic Control, 61(5), 1239–1254.
- Yin, X. and Lafortune, S. (2016b). A uniform approach for synthesizing property-enforcing supervisors for partially-observed discrete-event systems. IEEE Trans. Automatic Control, 61(8), 2140–2154.

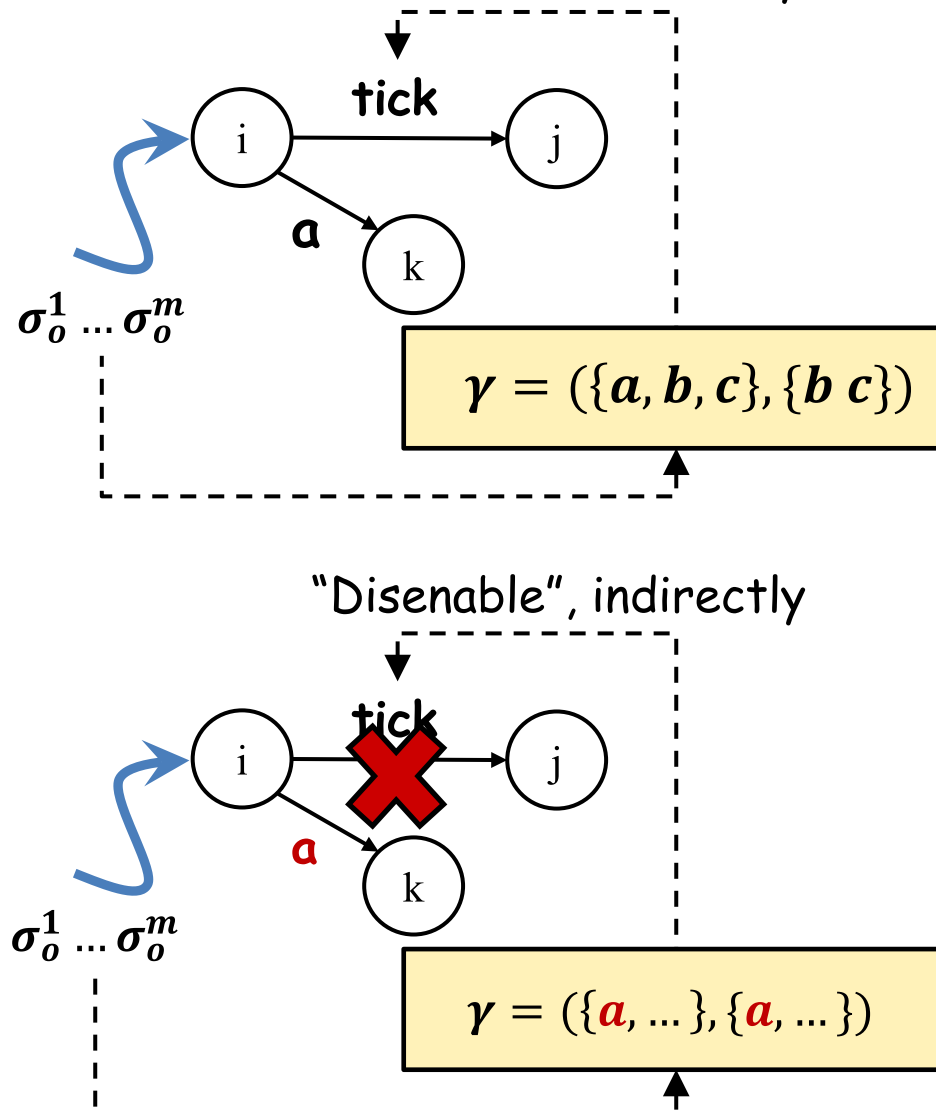
Supervisory Control of TDES

Standard Event

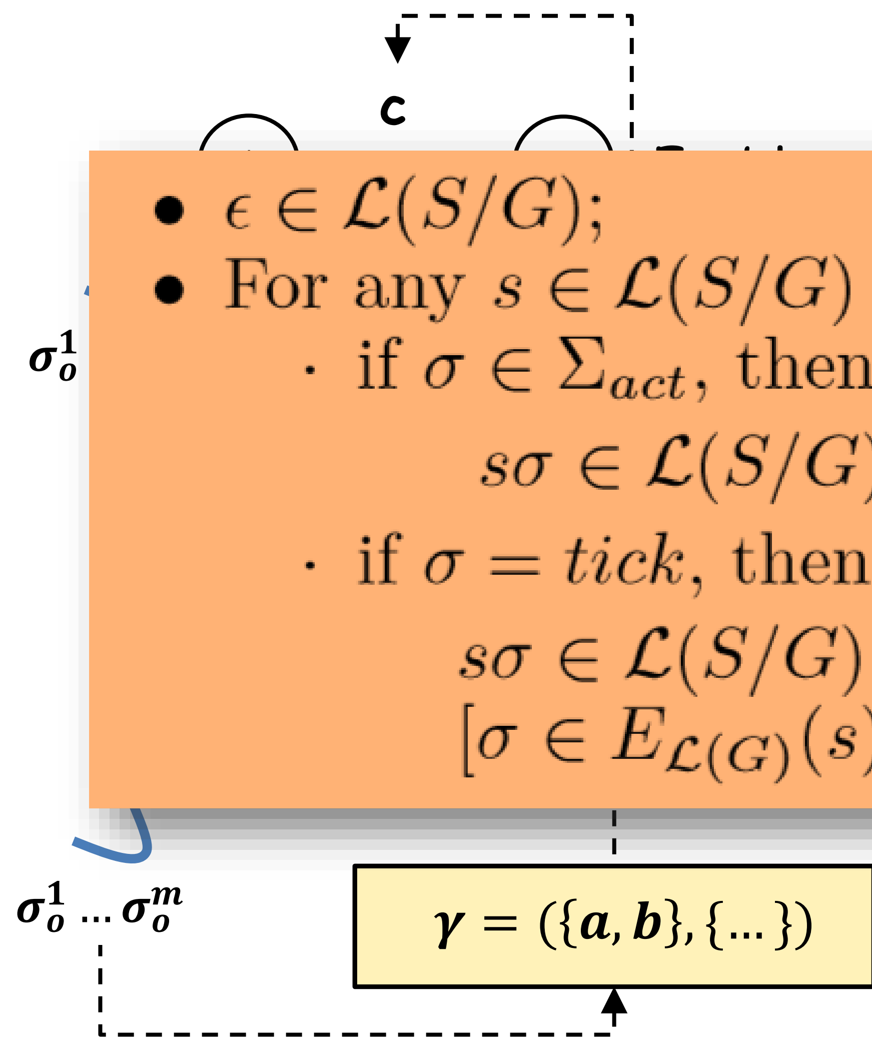


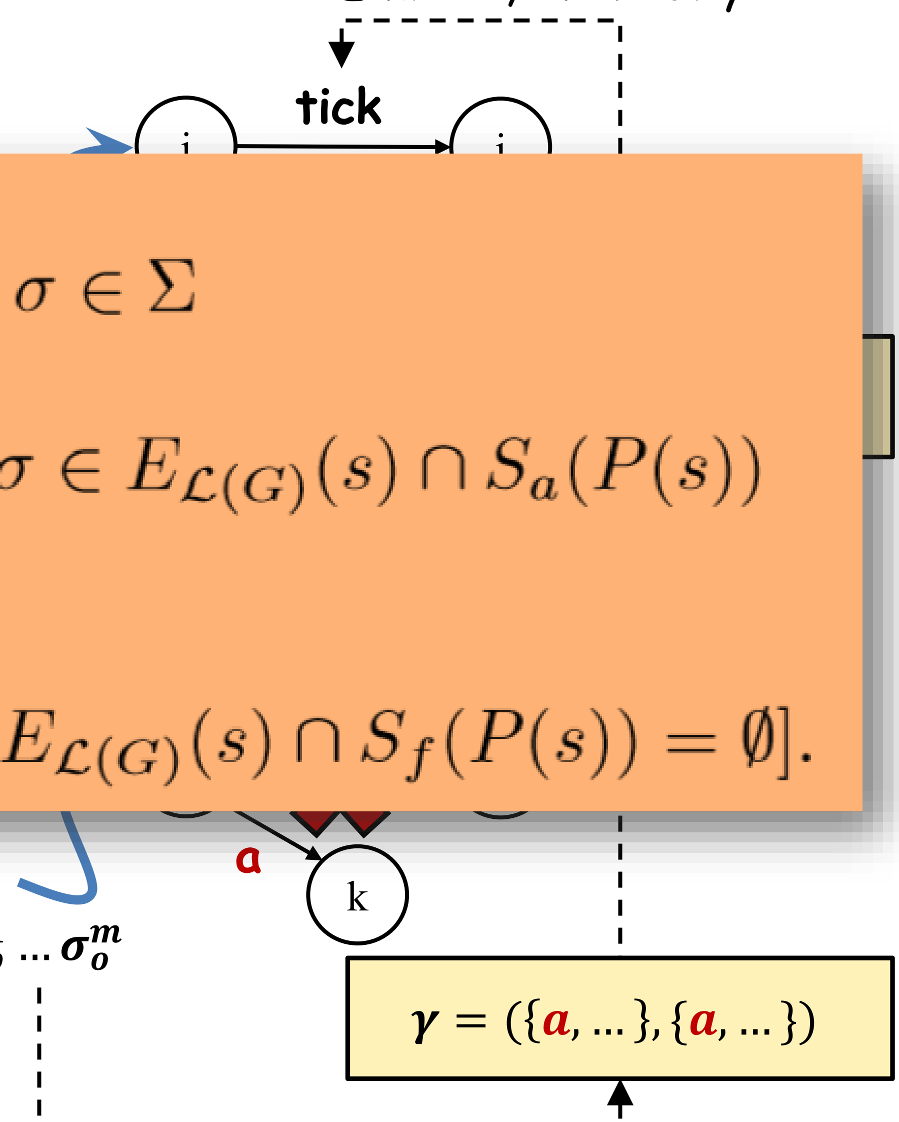
Tick Event

“Enable”, indirectly



Supervisory Control of TDES





- $\epsilon \in \mathcal{L}(S/G)$;
- For any $s \in \mathcal{L}(S/G)$ and $\sigma \in \Sigma$
  - if $\sigma \in \Sigma_{act}$, then

    $$s\sigma \in \mathcal{L}(S/G) \Leftrightarrow \sigma \in E_{\mathcal{L}(G)}(s) \cap S_a(P(s))$$

  - if $\sigma = tick$, then

    $$s\sigma \in \mathcal{L}(S/G) \Leftrightarrow$$
    $$[\sigma \in E_{\mathcal{L}(G)}(s)] \wedge [E_{\mathcal{L}(G)}(s) \cap S_f(P(s)) = \emptyset].$$

Supervisor Synthesis Problem

Problem 1. Given a TDES G and a safety specification $\boldsymbol{K} \subseteq \boldsymbol{L}(\boldsymbol{G})$, find a partial-observation supervisor $\boldsymbol{S}: \boldsymbol{P}(\boldsymbol{L}(\boldsymbol{G})) \rightarrow \mathbf{2}^{\boldsymbol{\Sigma}_{act}} \times \mathbf{2}^{\boldsymbol{\Sigma}_{for}}$ such that

- S is **safe**, i.e., L(S/G) ⊆ K; and
- S is **maximally-permissive**, i.e., for any S' that is safe, we have $L(S/G) \nsubseteq L(S'/G)$.

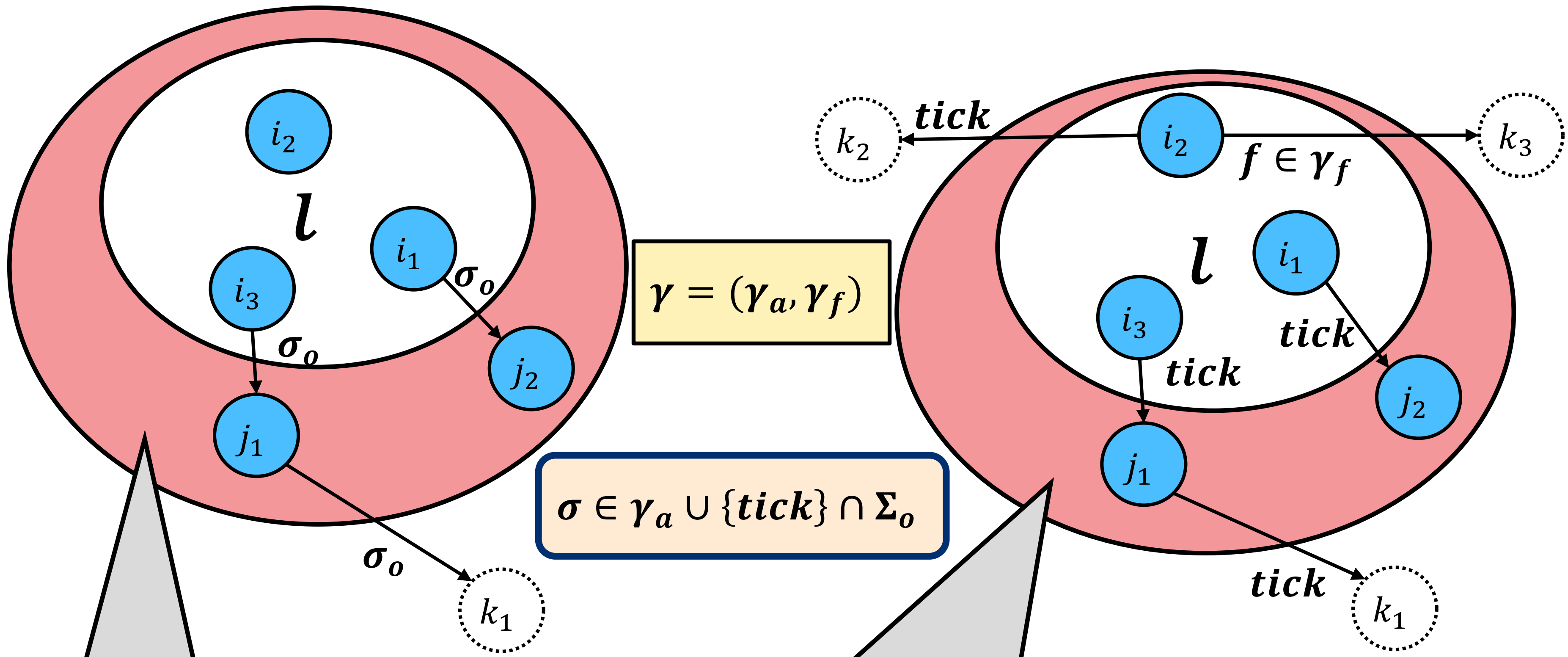
Consider a prefix closed sub-language $\boldsymbol{K} = \overline{\boldsymbol{K}} \subseteq \boldsymbol{L}(\boldsymbol{G})$
Assume that $\boldsymbol{K}$ is recognized by a strict sub-automaton $\boldsymbol{H} = (\boldsymbol{X}_{\boldsymbol{H}}, \boldsymbol{\Sigma}, \boldsymbol{\delta}_{\boldsymbol{H}}, \boldsymbol{x}_{\mathbf{0}})$ of $\boldsymbol{G}$, i.e. $L(H) = K$

Observable Reach

Possible states after seeing σ

* Assume $tick \in \Sigma_o$ here.

σ_o is standard event   **σ_o is *tick* event**



$\gamma = (\gamma_a, \gamma_f)$

$\sigma \in \gamma_a \cup \{tick\} \cap \Sigma_o$

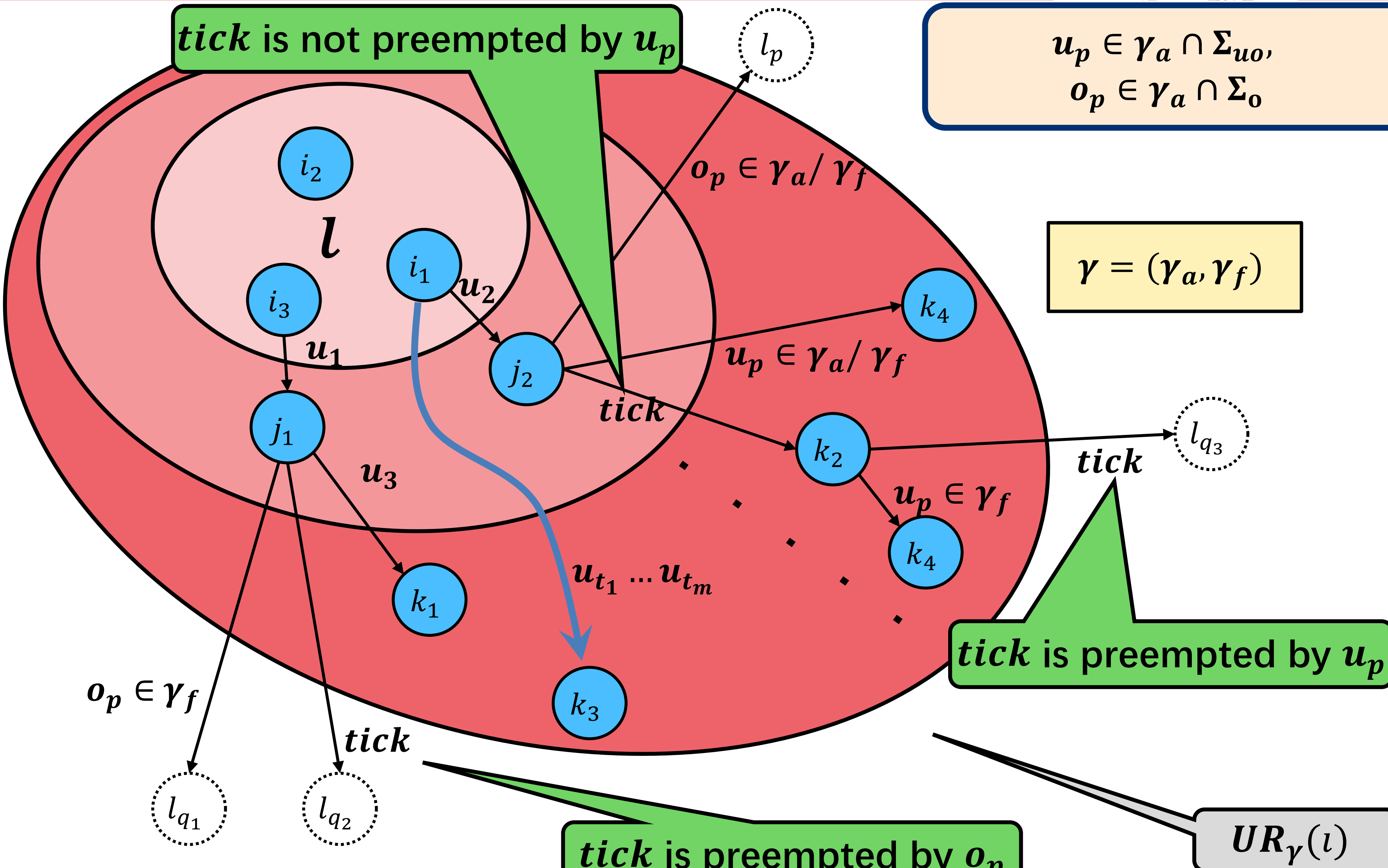
$OR_\sigma(\iota|\gamma) := \{\delta(x,\sigma) \in X: x \in \iota\}$

$OR_{tick}(\iota|\gamma) := \{\delta(x,\sigma) \in X: x \in \iota \wedge E_G(x) \cap \gamma_f = \emptyset\}$

Unobservable Reach

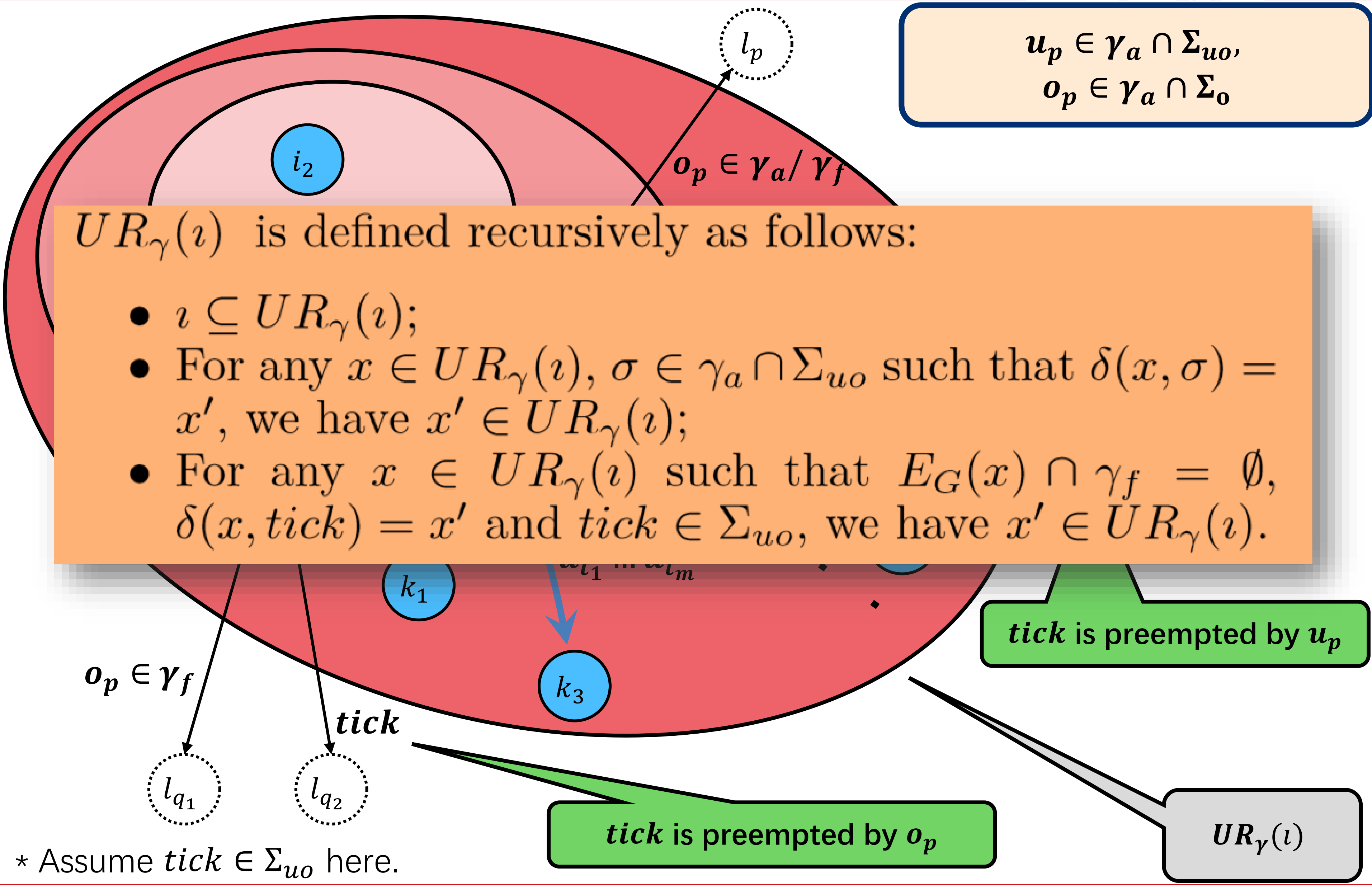
$u_p \in \gamma_a \cap \Sigma_{uo}$,
$o_p \in \gamma_a \cap \Sigma_o$

$\gamma = (\gamma_a, \gamma_f)$



* Assume $tick \in \Sigma_{uo}$ here.

Unobservable Reach



$UR_\gamma(\imath)$ is defined recursively as follows:

- $\imath \subseteq UR_\gamma(\imath)$;
- For any $x \in UR_\gamma(\imath)$, $\sigma \in \gamma_a \cap \Sigma_{uo}$ such that $\delta(x, \sigma) = x'$, we have $x' \in UR_\gamma(\imath)$;
- For any $x \in UR_\gamma(\imath)$ such that $E_G(x) \cap \gamma_f = \emptyset$, $\delta(x, tick) = x'$ and $tick \in \Sigma_{uo}$, we have $x' \in UR_\gamma(\imath)$.

* Assume $tick \in \Sigma_{uo}$ here.

Properties of Unobservable Reach

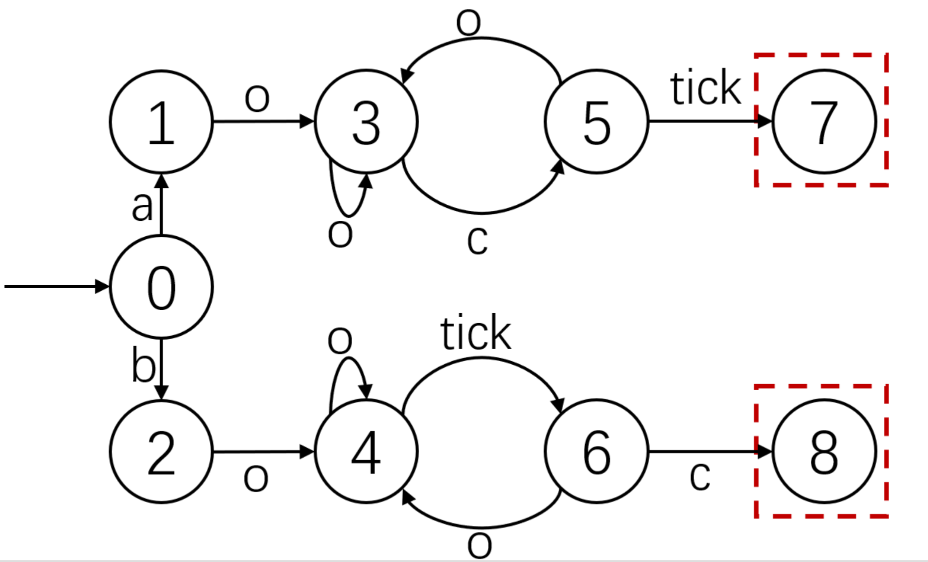
Lemma 1. $\forall \gamma, \iota, UR_\gamma\left(UR_\gamma(\iota)\right) = UR_\gamma(\iota)$.

Intuitively: The unobservable reach defined indeed yields a reachability closure.

Lemma 2. For any γ, ι and state $x \in UR_\gamma(\iota)$, there exists a state $x' \in \iota$ and a sequence of unobservable events $u_1 u_2 \dots u_m \in \Sigma_{uo}$ such that $x = \delta(x', u_1 \dots u_m)$ and $\forall 0 \leq i \leq m, \delta(x', u_1 \dots u_m) \in UR_\gamma(\iota)$.

Intuitively: there is a "construction path" from the initial set ι.

Example: Observable/Unobservable Reach



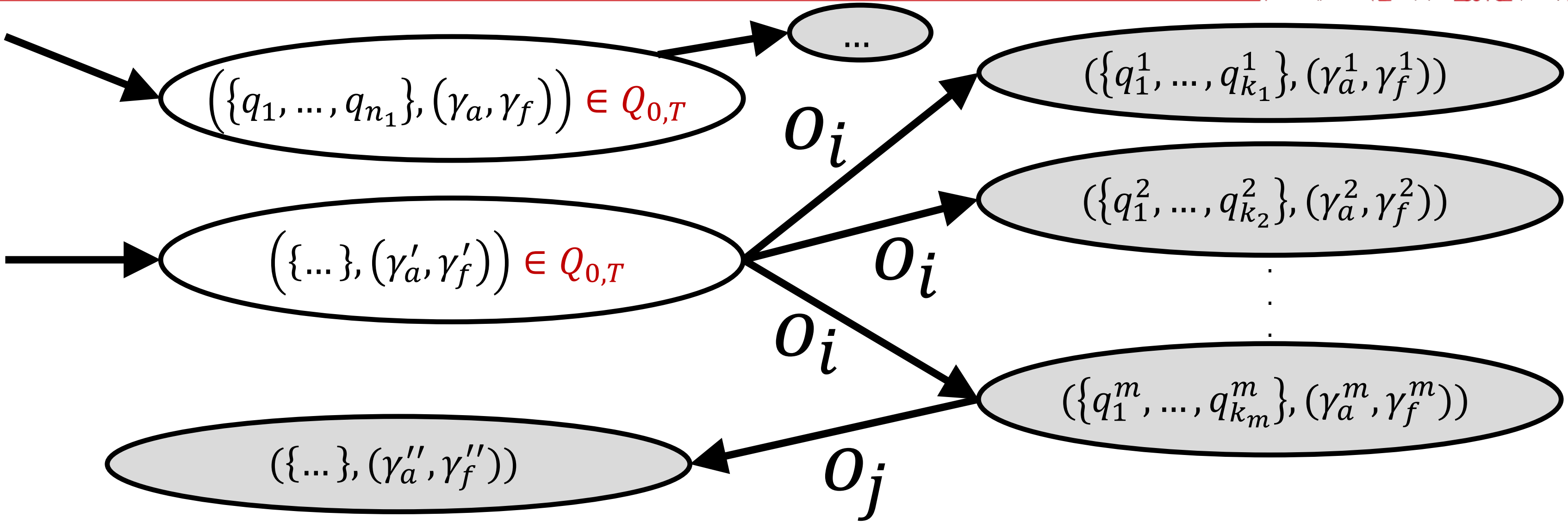
$\Sigma_o = \{o\}$
$\Sigma_c = \{c\}$
$\Sigma_f = \{o\}$

Let $\boldsymbol{\iota_0} = \{\mathbf{0}, \mathbf{1}, \mathbf{2}\}, \boldsymbol{\gamma_0} = (\boldsymbol{\Sigma_{uc}}, \emptyset)$, we have $\iota_1 = OR_o(\iota_0|\gamma_0) = \{3, 4\}$;

Then let $\boldsymbol{\gamma_1} = (\boldsymbol{\Sigma_{uc}} \cup \{\boldsymbol{c}\}, \{\boldsymbol{o}\})$, then $UR_{\gamma_1}(\iota_1) = \{3, 4, 5\}$, pay attention that tick was preempted at state “4” and “5” by forcing event o.

But if we pick control decision $\boldsymbol{\gamma_2} = (\boldsymbol{\Sigma_{uc}} \cup \{\boldsymbol{c}\}, \emptyset)$, then we have $UR_{\gamma_2}(\iota_1) = \{3, 4, 5, 6, 7, 8\}$.

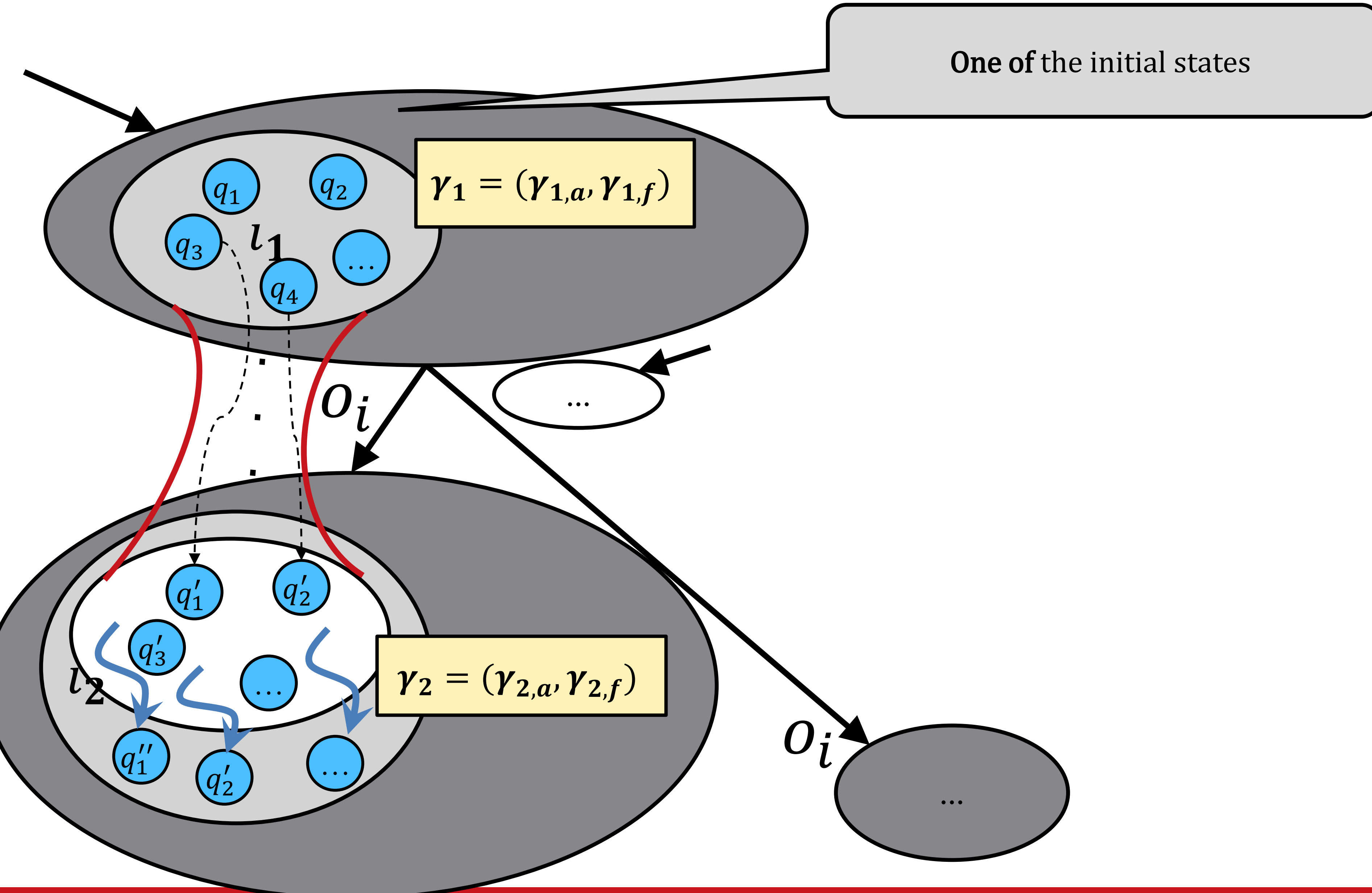
Inclusive Controller: Definition



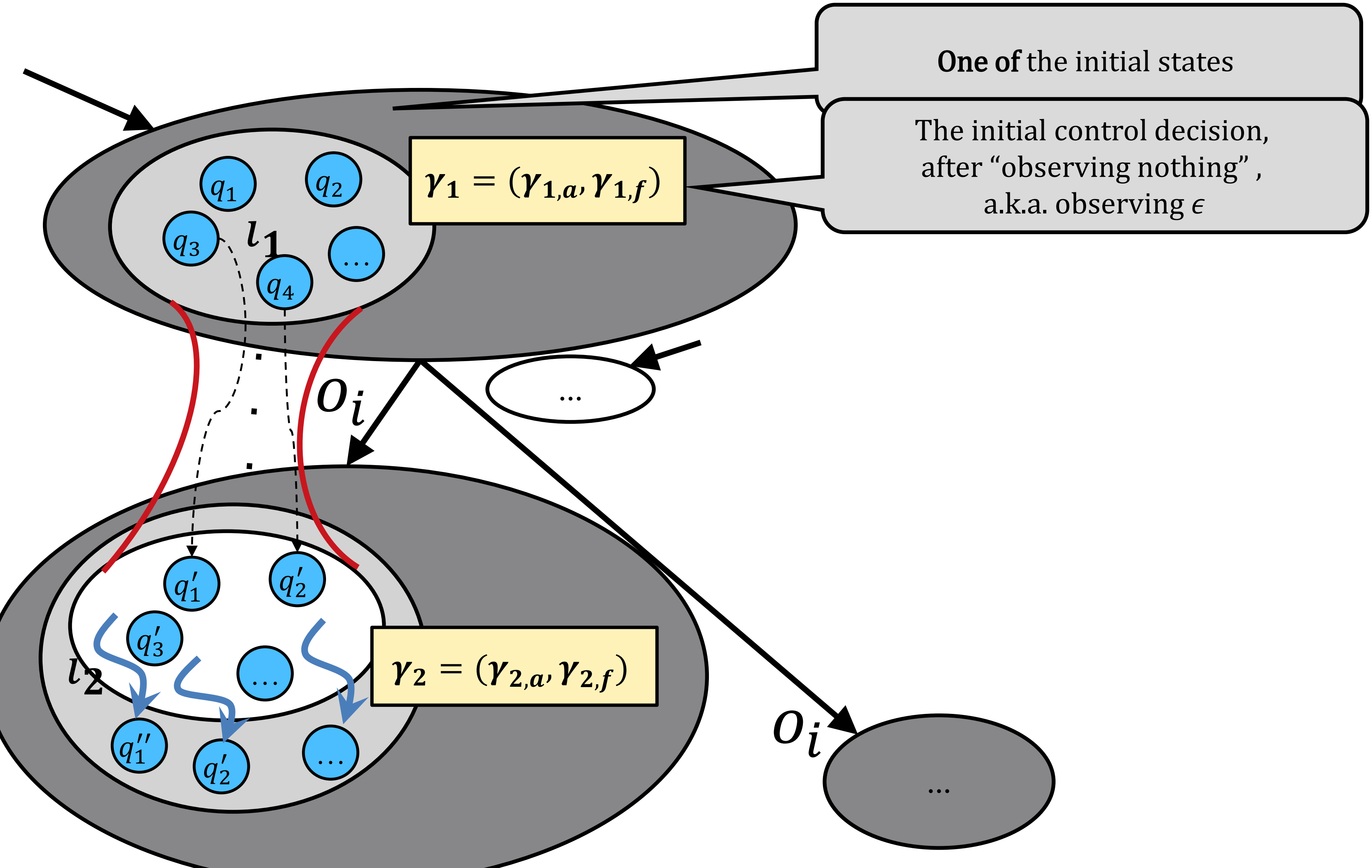
- **System:** $\mathrm{T} = (Q_T, \Sigma_o, \Gamma, h_T, Q_{0,T})$, w.r.t $G = (X, \Sigma, \delta, x_0)$
- **States:** $Q_T \subseteq 2^X \times \Gamma$
- **Initial states:** $Q_{0,T} \subseteq Q_T$
- **Decisions:** $\Gamma \subseteq 2^{\Sigma_{act}} \times 2^{\Sigma_{for}}$
- **Transition:** $h_T\colon Q_T \times \Sigma_o \to 2^{Q_T}$, non-deterministic
- **Supervisor:** $S\colon \Sigma_o^* \to 2^{\Sigma}$ satisfying $\Sigma_{uc} \subseteq S(\alpha)$ for any $\alpha \in \Sigma_o^*$

Use notation $(I(q),(C_a(q),C_f(q)))$ to denote components of $q \in Q_T$

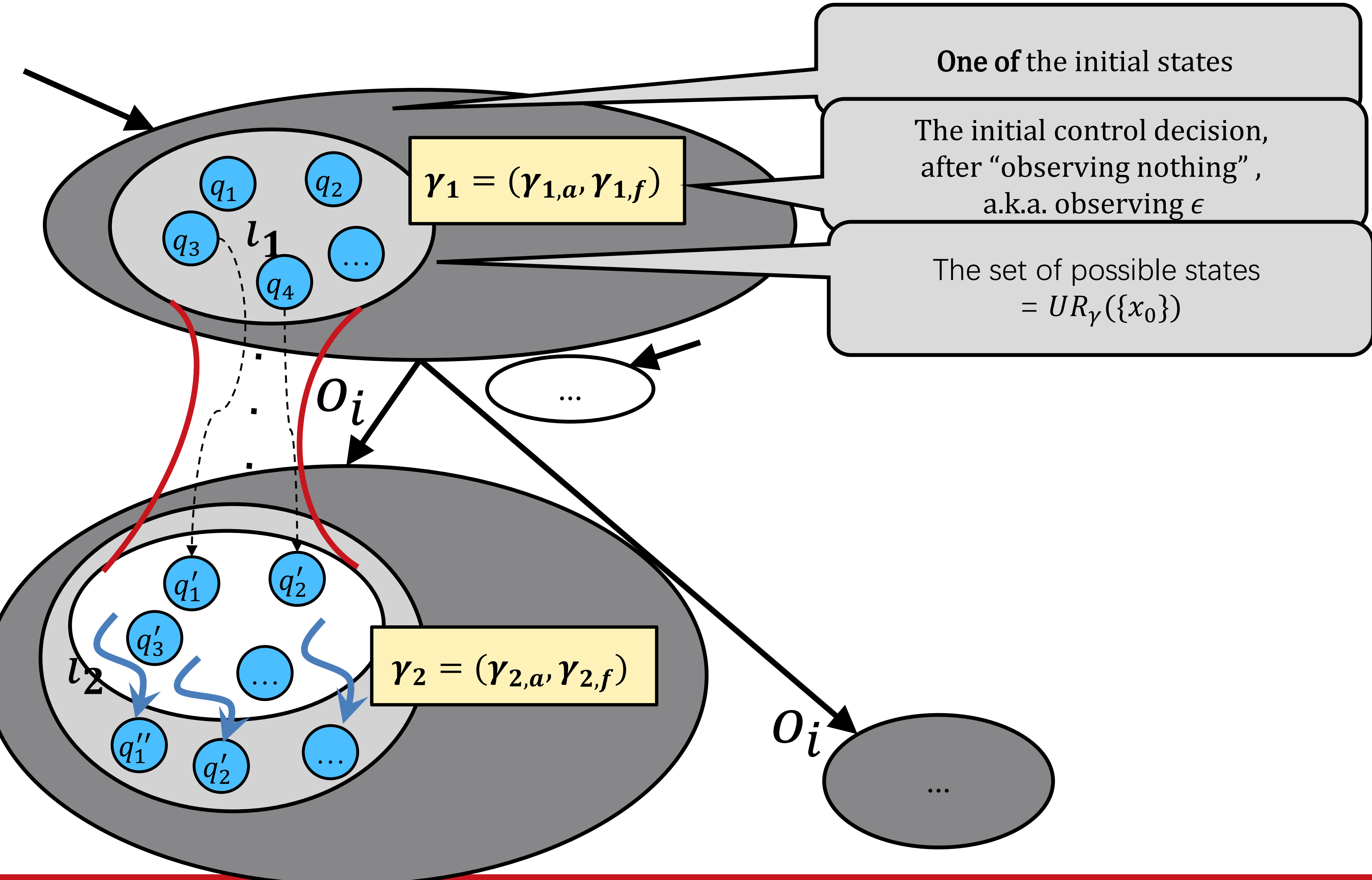
Inclusive Controller: Explanation



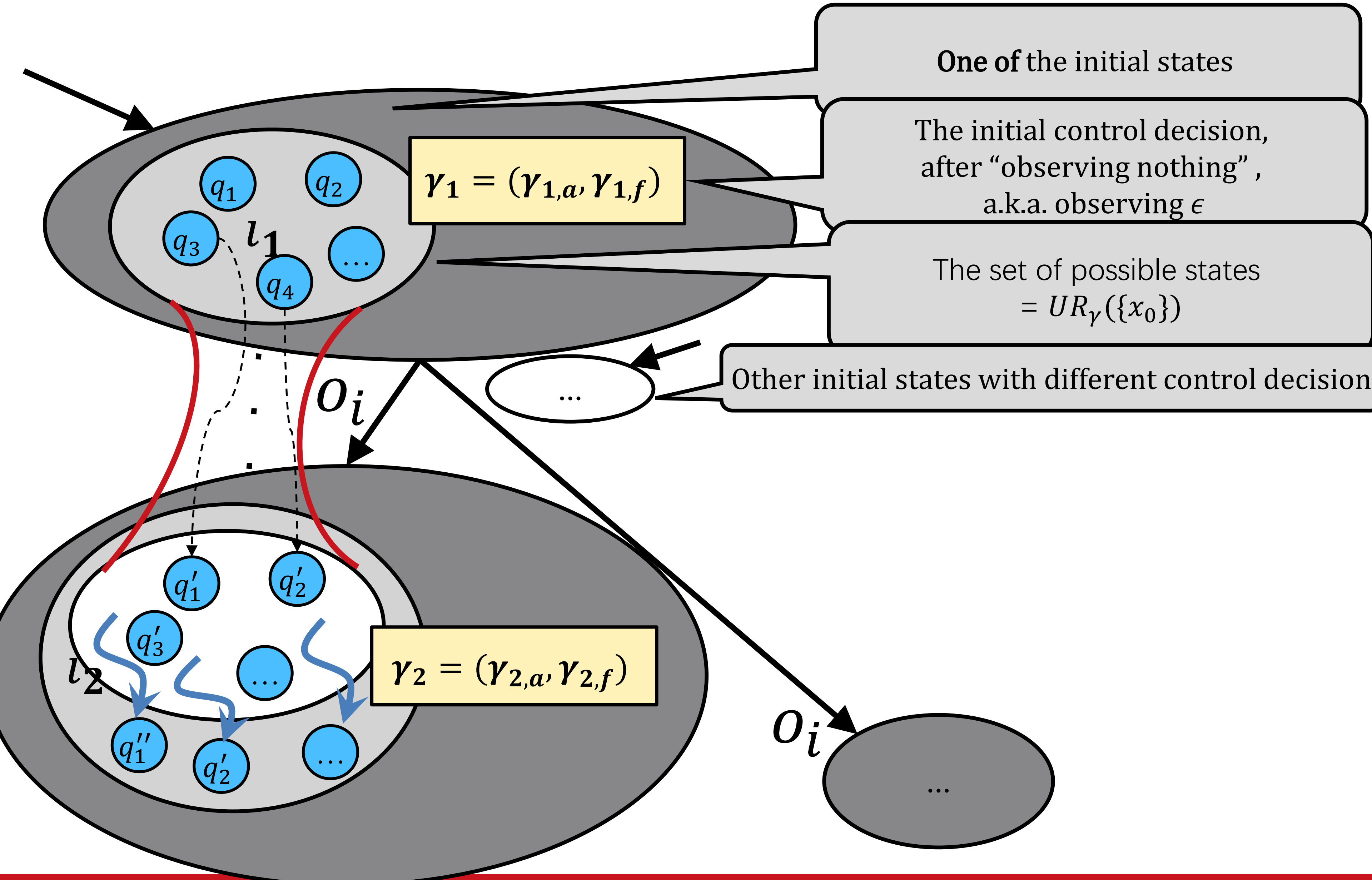
Inclusive Controller: Explanation



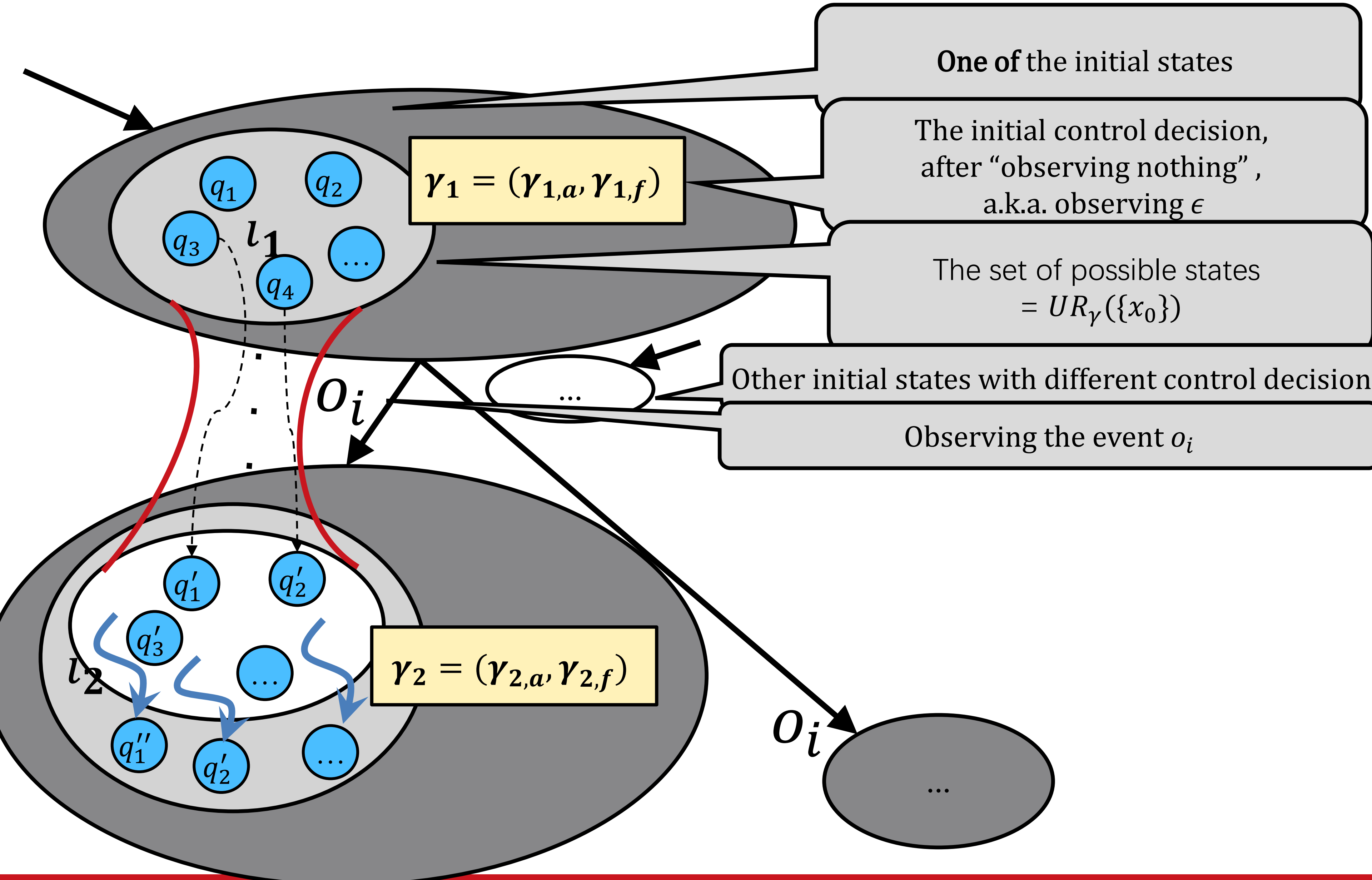
Inclusive Controller: Explanation



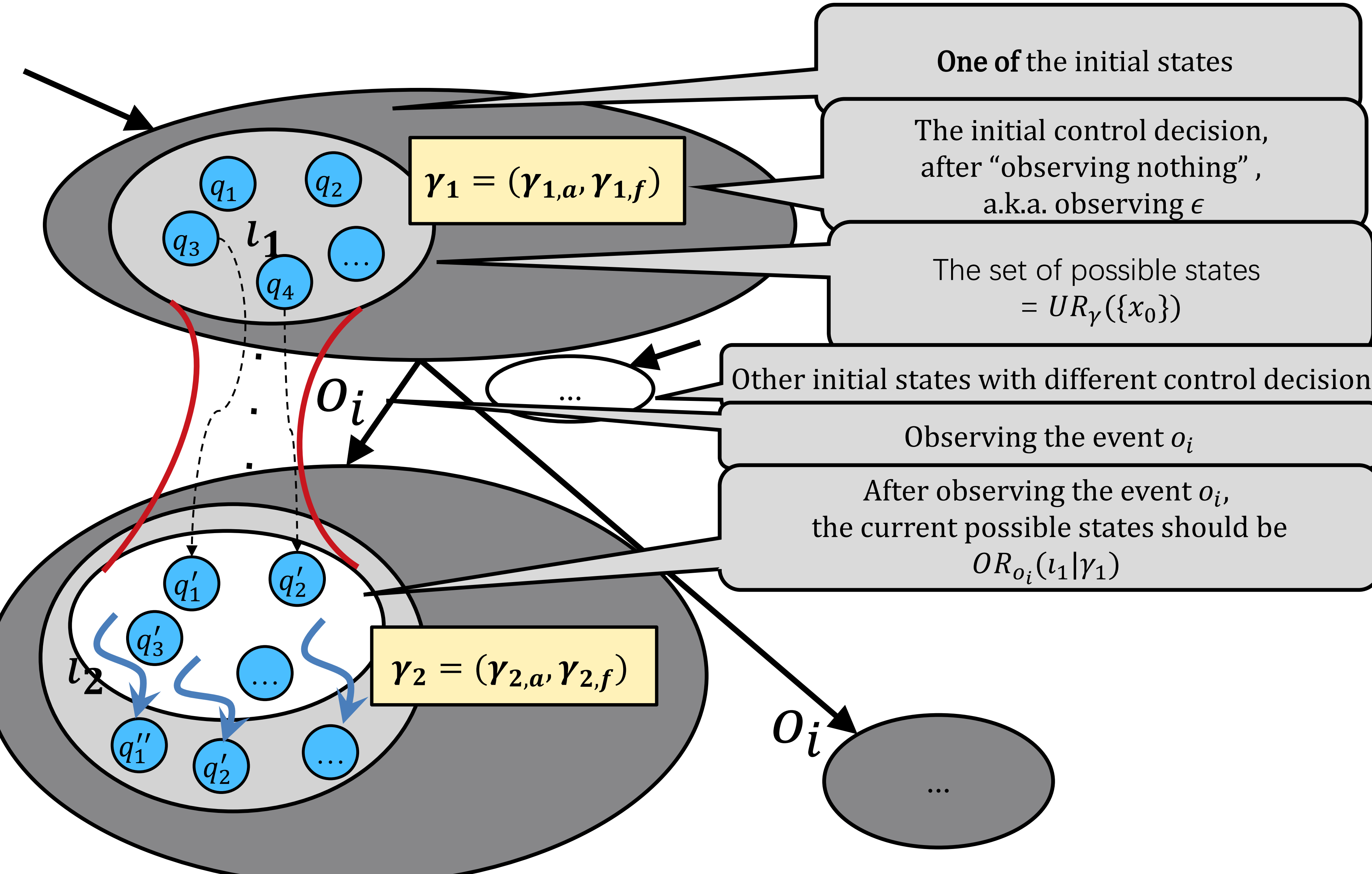
Inclusive Controller: Explanation



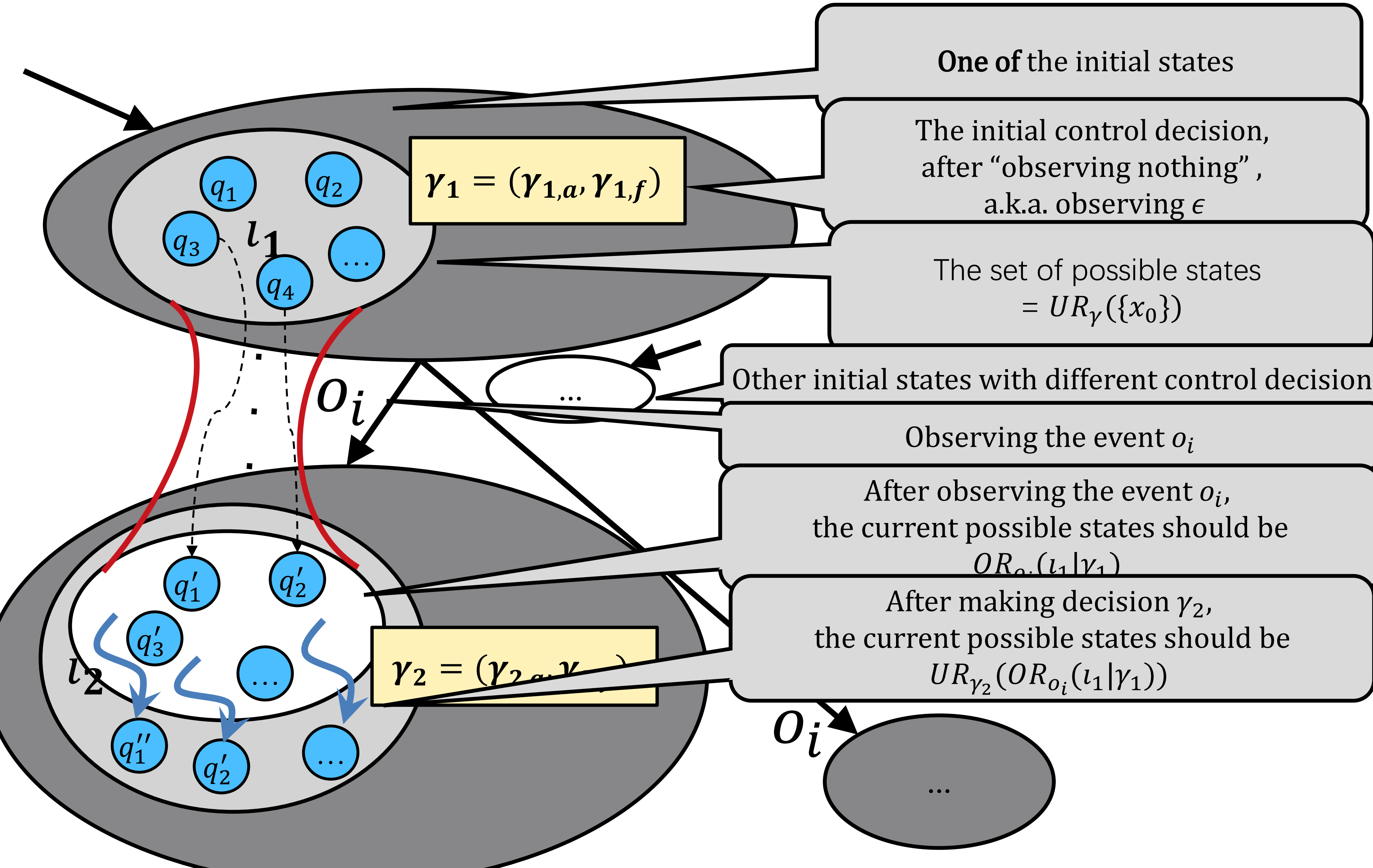
Inclusive Controller: Explanation



Inclusive Controller: Explanation



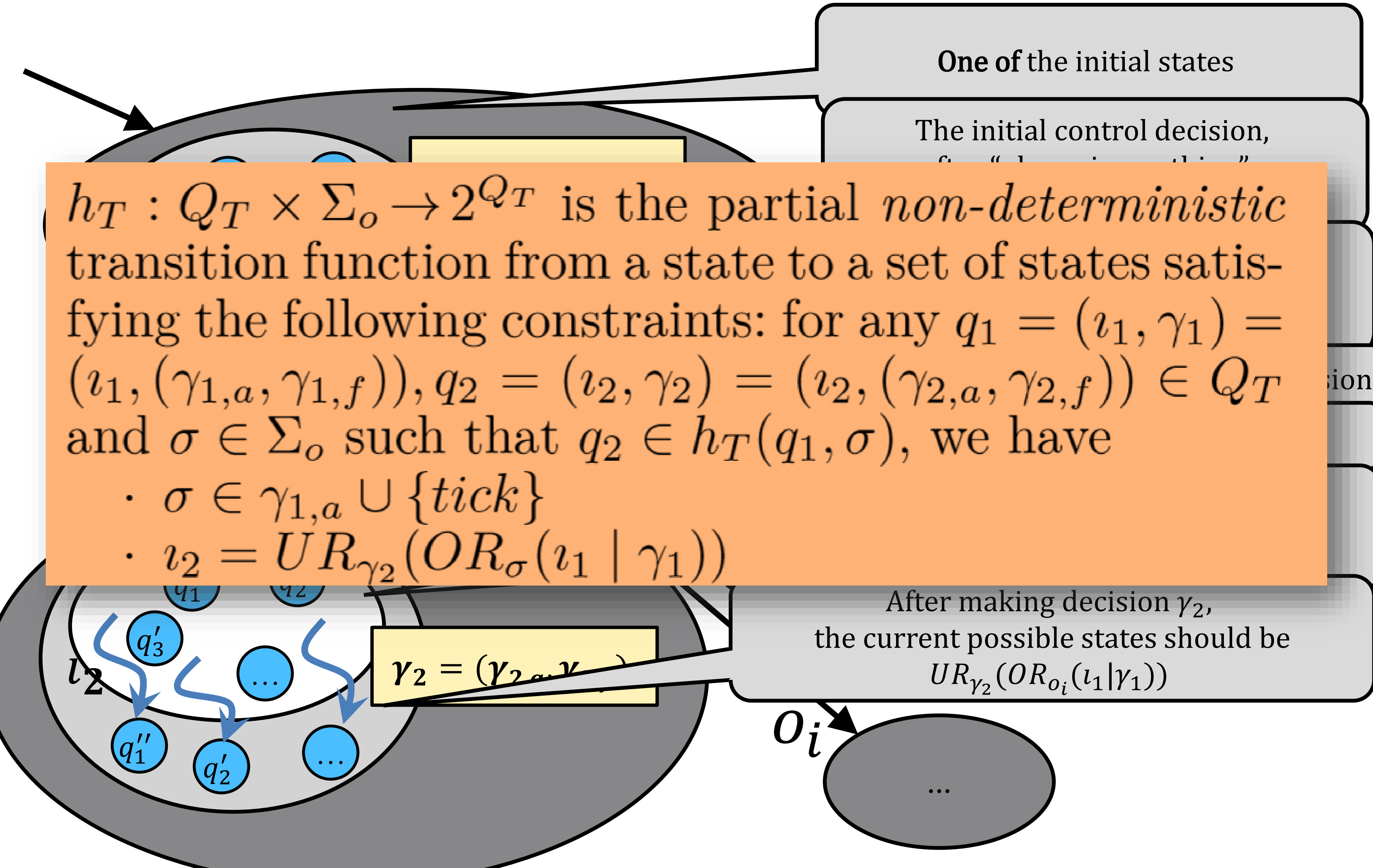
Inclusive Controller: Explanation



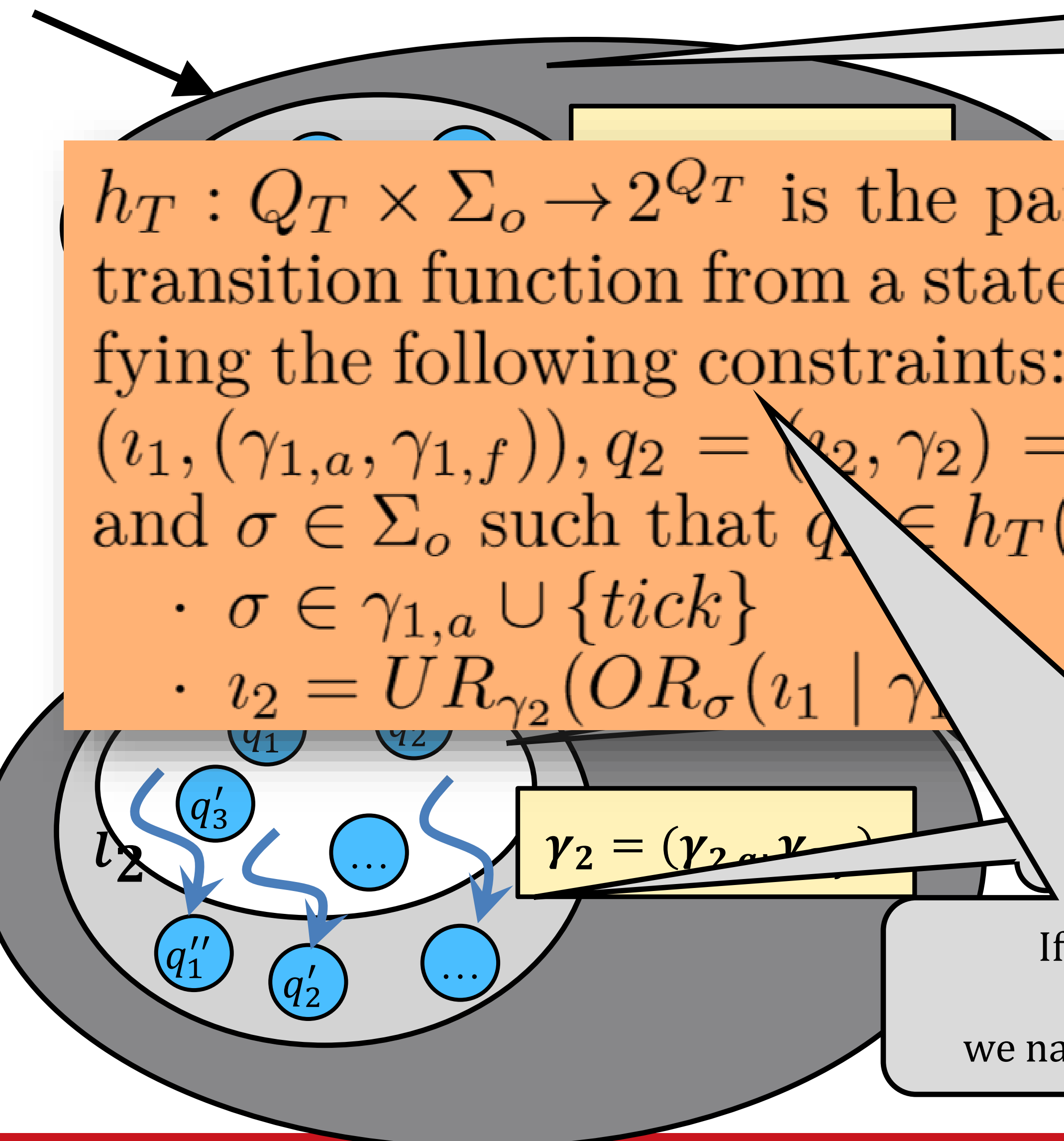
Inclusive Controller: Explanation

$h_T : Q_T \times \Sigma_o \rightarrow 2^{Q_T}$ is the partial *non-deterministic* transition function from a state to a set of states satisfying the following constraints: for any $q_1 = (\imath_1, \gamma_1) = (\imath_1, (\gamma_{1,a}, \gamma_{1,f})), q_2 = (\imath_2, \gamma_2) = (\imath_2, (\gamma_{2,a}, \gamma_{2,f})) \in Q_T$ and $\sigma \in \Sigma_o$ such that $q_2 \in h_T(q_1, \sigma)$, we have

- $\sigma \in \gamma_{1,a} \cup \{tick\}$
- $\imath_2 = UR_{\gamma_2}(OR_\sigma(\imath_1 \mid \gamma_1))$



Inclusive Controller: Explanation



$h_T : Q_T \times \Sigma_o \to 2^{Q_T}$ is the partial *non-deterministic* transition function from a state to a set of states satisfying the following constraints: for any $q_1 = (\imath_1, \gamma_1) = (\imath_1, (\gamma_{1,a}, \gamma_{1,f})), q_2 = (\imath_2, \gamma_2) = (\imath_2, (\gamma_{2,a}, \gamma_{2,f})) \in Q_T$ and $\sigma \in \Sigma_o$ such that $q_2 \in h_T(q_1, \sigma)$, we have

- $\sigma \in \gamma_{1,a} \cup \{tick\}$
- $\imath_2 = UR_{\gamma_2}(OR_\sigma(\imath_1 \mid \gamma_1$

One of the initial states

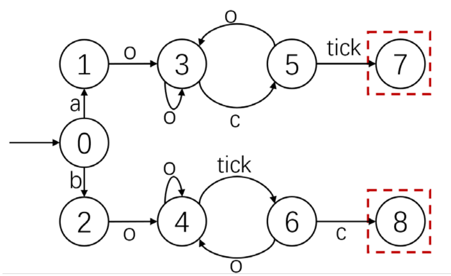
The initial control decision,

ion

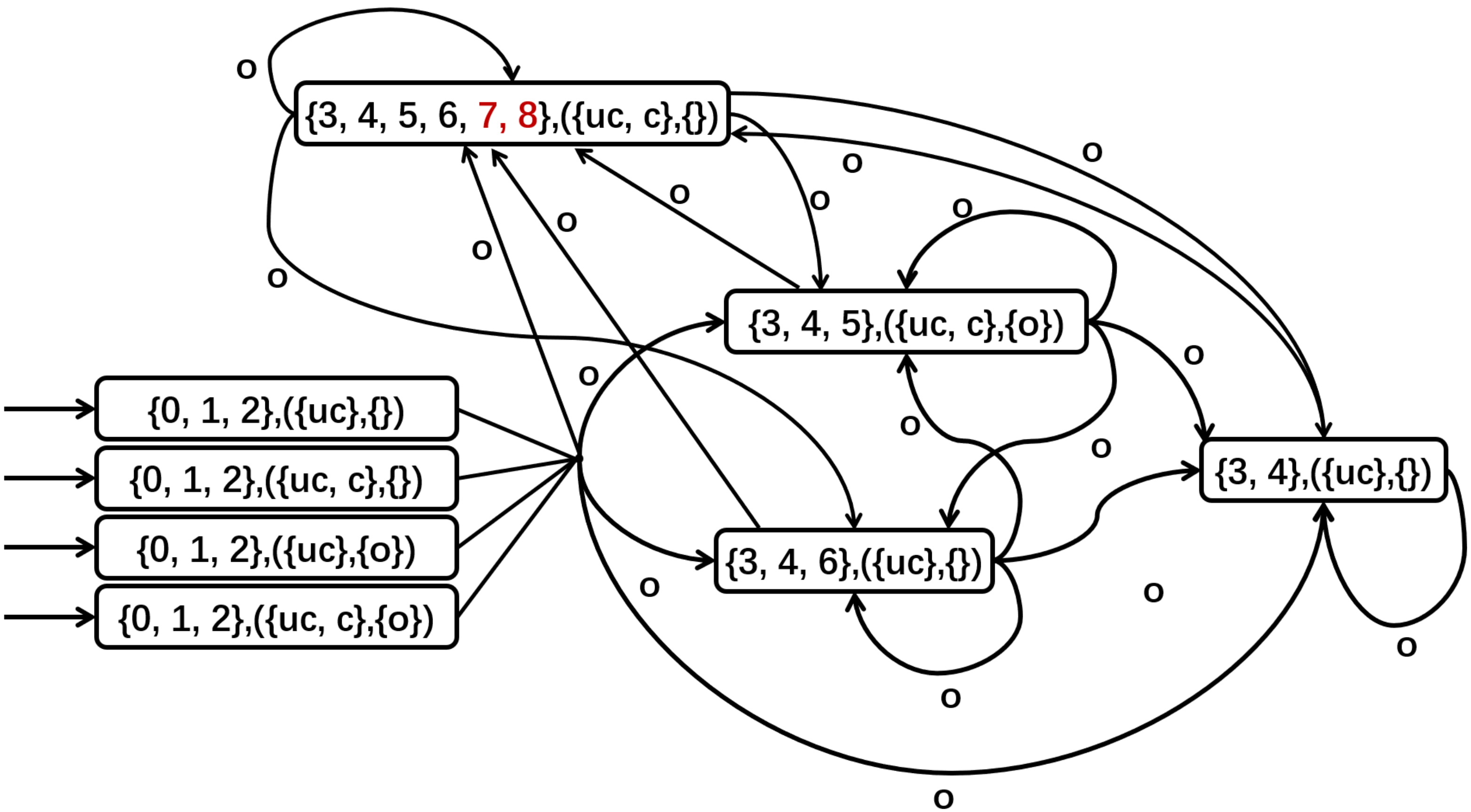
After making decision γ_2, current possible states should be $UR_{\gamma_2}(OR_{o_i}(\imath_1|\gamma_1))$

If any state that satisfies this constraint was included in any $h_T(q, \gamma)$, we name T as a ***Total Controller*** w.r.t. G, $(Tol(G))$

Example: Total Inclusive Controller

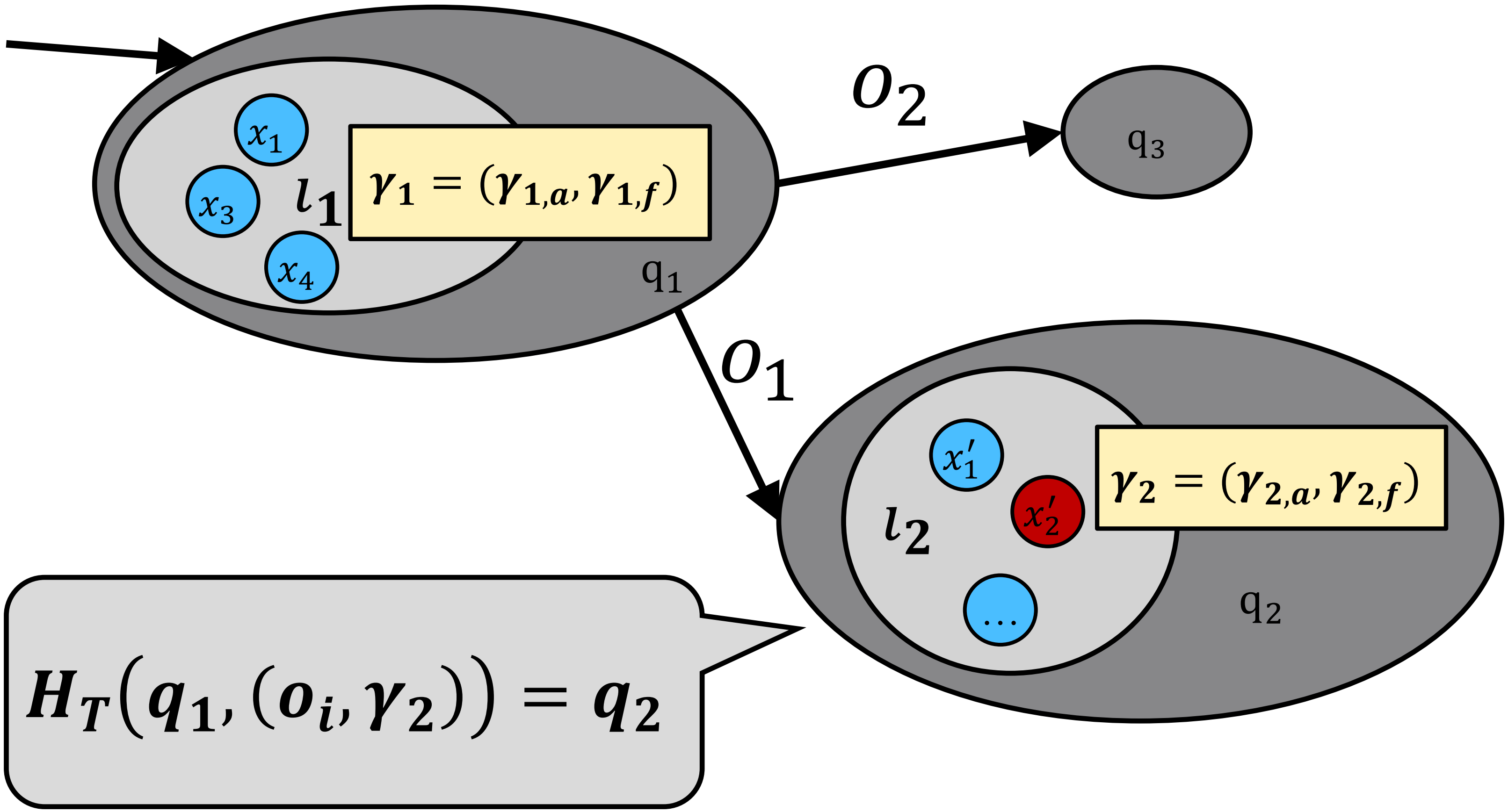


$$\Sigma_o = \{o\}$$
$$\Sigma_c = \{c\}$$
$$\Sigma_f = \{o\}$$



Inclusive Controller: Deterministic Transition

Deterministic Transition $H_T: Q \times (\Sigma_o \times \Gamma) \to Q$ s.t. $H_T(q, (\sigma, \gamma)) = q' = (\iota, \gamma')$
if $q' \in h_T(q, \sigma)$ and $\gamma = \gamma'$.
Extend H_T to $H_T: Q \times (\Sigma_o \times \Gamma)^* \to Q$
Given a supervisor S, we get control decision whenever $\sigma \in \Sigma_o$ occurs.



Inclusive Controller: Deterministic Transition

Deterministic Transition $H_T: Q \times (\Sigma_o \times \Gamma) \rightarrow Q$ s.t. $H_T\big(q, (\sigma, \gamma)\big) = q' = (\iota, \gamma')$ if $q' \in h_T(q, \sigma)$ *and* $\gamma = \gamma'$.

Extend H_T to $H_T: Q \times (\Sigma_o \times \Gamma)^* \rightarrow Q$

Given a supervisor S, we get control decision whenever $\sigma \in \Sigma_o$ occurs.

Given a supervisor S, we get control decision whenever $\sigma \in \Sigma_o$ occurs.

$$\xi_{\alpha,S} := (\sigma_1, S(\sigma_1))(\sigma_2, S(\sigma_1\sigma_2)) \ldots (\sigma_n, S(\alpha)) \in (\Sigma_o \times \Gamma)^*$$

Inclusive Controller: Deterministic Transition

Deterministic Transition $H_T: Q \times (\Sigma_o \times \Gamma) \rightarrow Q$ s.t. $H_T\big(q, (\sigma, \gamma)\big) = q' = (\iota, \gamma')$ if $q' \in h_T(q, \sigma)$ *and* $\gamma = \gamma'$.

Extend H_T to $H_T: Q \times (\Sigma_o \times \Gamma)^* \rightarrow Q$

Given a supervisor S, we get control decision whenever $\sigma \in \Sigma_o$ occurs.

Given a supervisor S, we get control decision whenever $\sigma \in \Sigma_o$ occurs.

$$\xi_{\alpha,S} := (\sigma_1, S(\sigma_1))(\sigma_2, S(\sigma_1\sigma_2))\ldots(\sigma_n, S(\alpha)) \in (\Sigma_o \times \Gamma)^*$$

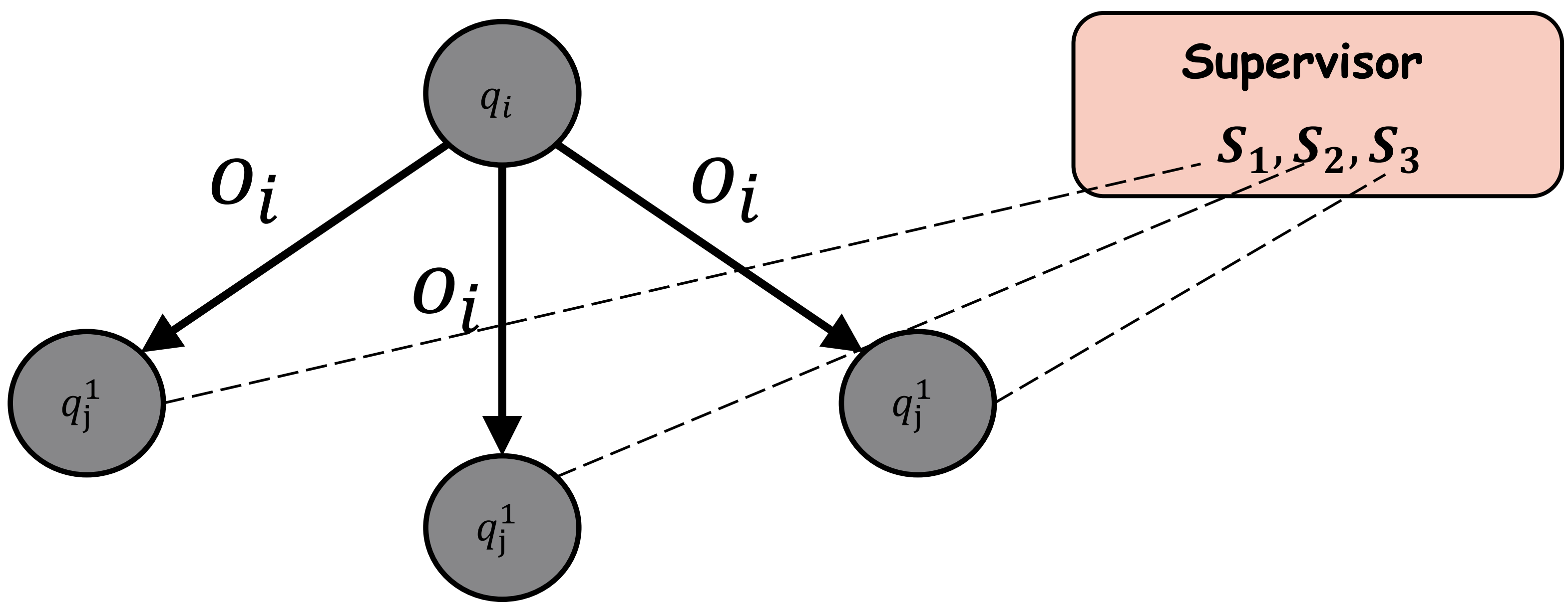
We get an estimation of possible states in $\boldsymbol{Tol(G)}$:

$$\boldsymbol{q_{\alpha,S}} := \boldsymbol{H_T(q_{0,S}, \xi_{\alpha,S})}, \text{ where } q_{0,S} := \big(UR_{S(\epsilon)}(\{x_0\}), S(\epsilon)\big).$$

Inclusive Controller: Properties

Theorem 1. Given a supervisor S for G, and $\alpha \in P(L(S/G))$, we have $\mathrm{I}(q_{\alpha,s}) = \{\delta(x_0, s) \in X: s \in L(S/G) \wedge P(s) = \alpha\}$.

Intuitively: The theorem indicates that all supervisors was "*embedded*" in the Total Inclusive Controller, i.e. we can synthesis a certain supervisor by choosing a certain transition at each $q \in Q_{Tol(G)}$ upon observable event o, which eliminate the non-determinism of the controller.



Inclusive Controller: Properties

Theorem 1. Given a supervisor S for G and $\alpha \in P(L(S/G))$, we have $\mathrm{I}(q_{\alpha,S}) = \{\delta(x_0, s) \in X: s \in L(S/G) \wedge P(s) = \alpha\}$.

Intuitively: The theorem indicates that all supervisors was "embedded" in the Total Inclusive Controller, i.e. we can synthesis a certain supervisor by choosing a certain transition at each $q \in Q_{Tol(G)}$ upon observable event o, which eliminate the non-determinism of the controller.

Theorem 2. Supervisor S is safe if and only is $\forall \alpha \in P(L(S/G))$, $I(q_{\alpha,S}) \subseteq X_H$.

Inclusive Controller: Properties

Theorem 1. Given a supervisor S for G and $\alpha \in P(L(S/G))$, we have $\mathrm{I}(q_{\alpha,S}) = \{\delta(x_0, s) \in X : s \in L(S/G) \wedge P(s) = \alpha\}$.

Intuitively: The theorem indicates that all supervisors was "embedded" in the Total Inclusive Controller, i.e. we can synthesis a certain supervisor by choosing a certain transition at each $q \in Q_{Tol(G)}$ upon observable event o, which eliminate the non-determinism of the controller.

Theorem 2. Supervisor S is safe if and only is $\forall \alpha \in P(L(S\,/G))$, $I(q_{\alpha,s}) \subseteq X_H$.

Suggests an approach for synthesizing a safe controller.
Now we say T is safe if for any $q \in Q_T, I(q) \subseteq X_H$.

All Inclusive Controller for Safety

Definition. (AIC-Safe) Given TDES $\boldsymbol{G}$, A(G) $= (\boldsymbol{Q_A}, \boldsymbol{\Sigma_o}, \boldsymbol{\Gamma}, \boldsymbol{h_A}, \boldsymbol{Q_{0,A}})$ is a **safe** and **complete** inclusive controller such that for any complete controller T that is safe, we have $T \sqsubseteq A(G)$**.**

Construction A(G) :

- Start from all possible initial-states;
- Explore the entire space where all states are subsets of X_H;
- Iteratively remove states that violates the completeness requirement (in order to guarantee safety, some states might make the AIC incomplete) , until the resulting subsystem is complete;

AIC-Safety : Properties

Theorem 3. Use $\boldsymbol{S}(T)$ to denote all supervisors ***included*** in T, then a supervisor S is safe **iff** $S \in \boldsymbol{S}(A(G))$.

Suggests that the AIC-Safe includes all safe supervisors .

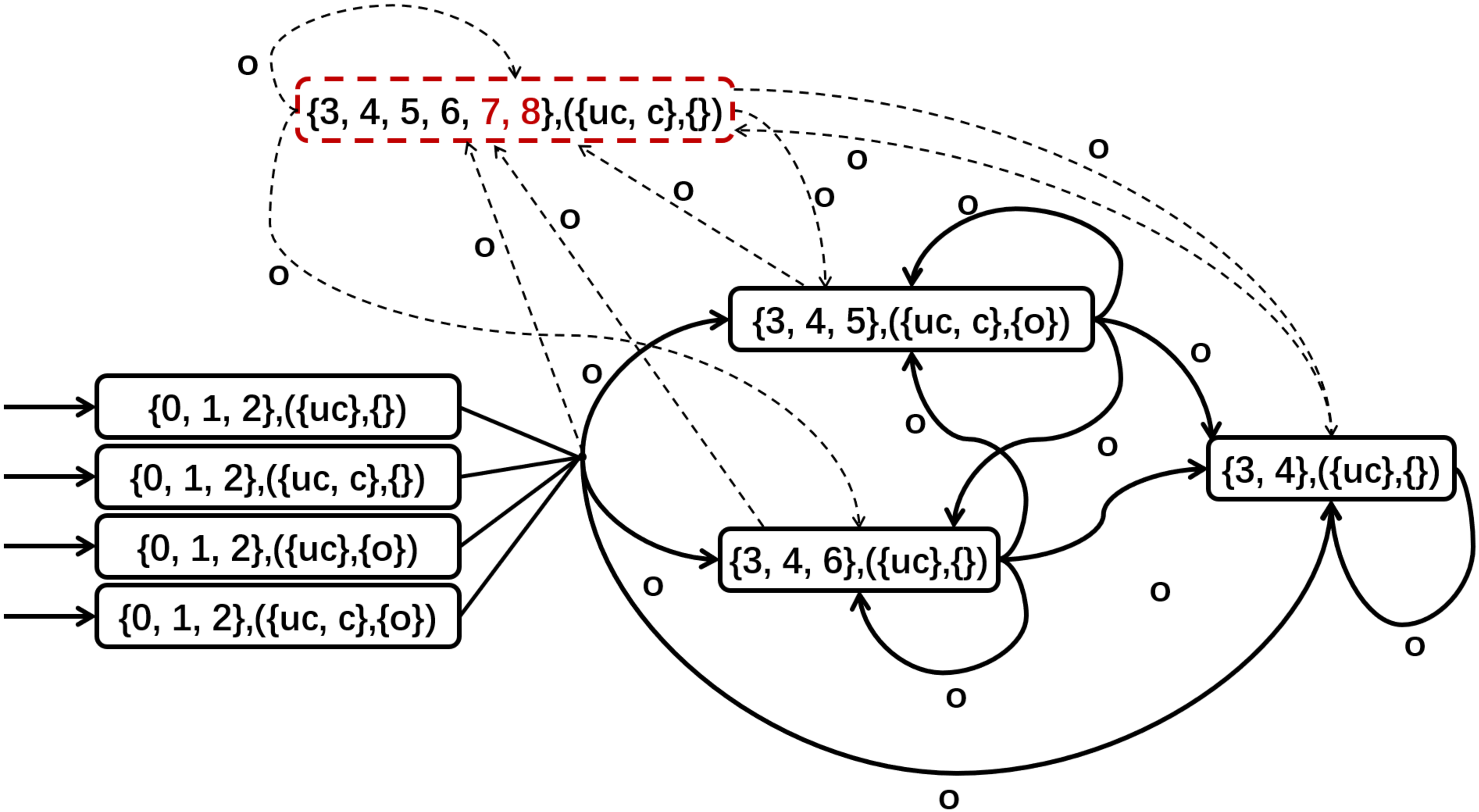
AIC-Safety : Properties

Theorem 3. Use $\boldsymbol{S}(T)$ to denote all supervisors ***included*** in T, then a supervisor S is safe **iff** $S \in \boldsymbol{S}(T)$.

supervisor S is said to be *included* in T if

- $q_{0,S} \in Q_{0,T}$; and
- for any $\alpha\sigma \in P(\mathcal{L}(S/G))$, where $\alpha \in \Sigma_o^*$ and $\sigma \in \Sigma_o$, we have $S(\alpha\sigma) \in C_T(H_T(q_{0,S}, \xi_{\alpha,S}), \sigma)$.

Example: AIC-Safety



Extract a Supervisor from AIC-Safety

$$\textsc{Feas}(q) = \left\{\sigma \in \Sigma : x \in \imath \wedge \delta(x,\sigma)! \wedge \left[\begin{array}{c}(\sigma \in \Sigma_{act} \wedge \sigma \in \gamma_a) \text{ or} \\ (\sigma = tick \wedge E_G(x) \cap \gamma_f = \emptyset)\end{array}\right]\right\}$$

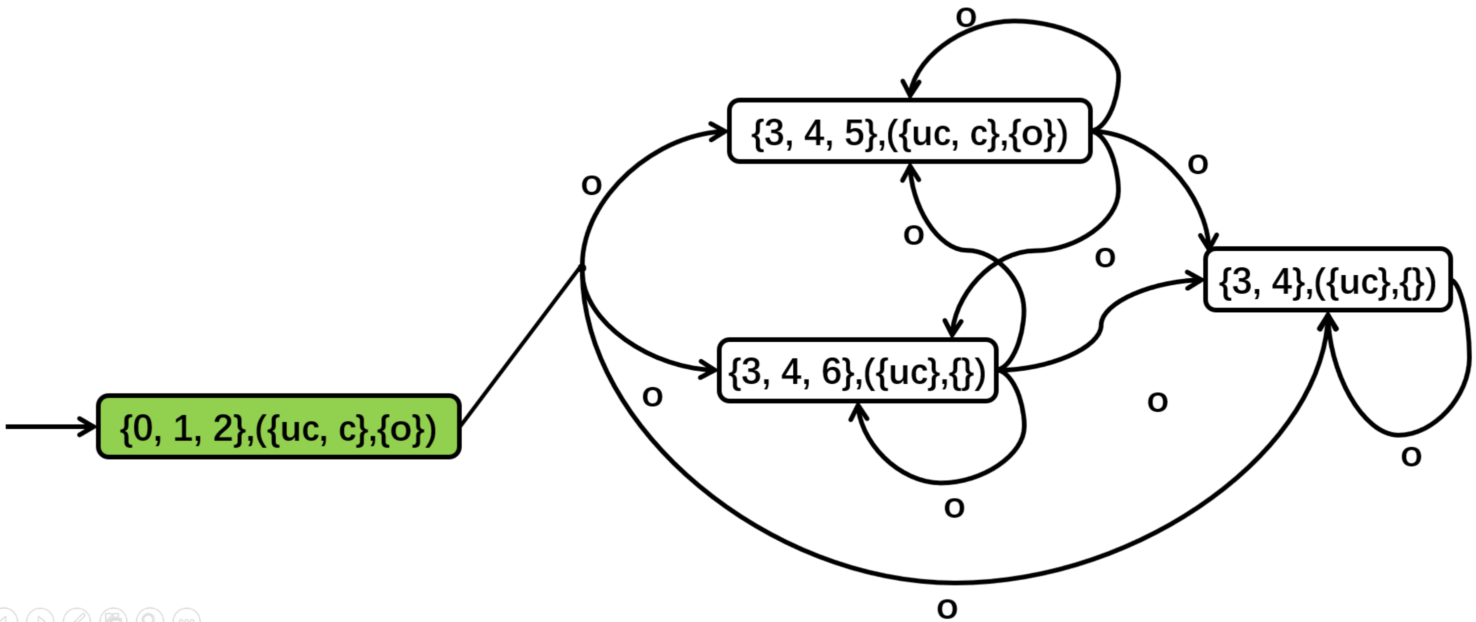
$$q = (\imath, (\gamma_a, \gamma_f))$$

We want to extract a subsystem that
enables as many events as possible at each instant.

Extract a Supervisor from AIC-Safety

Initially, we choose an initial state $q_0 \in Q_{0,A}$ that contains the maximum number of feasible events among all initial states, i.e.,

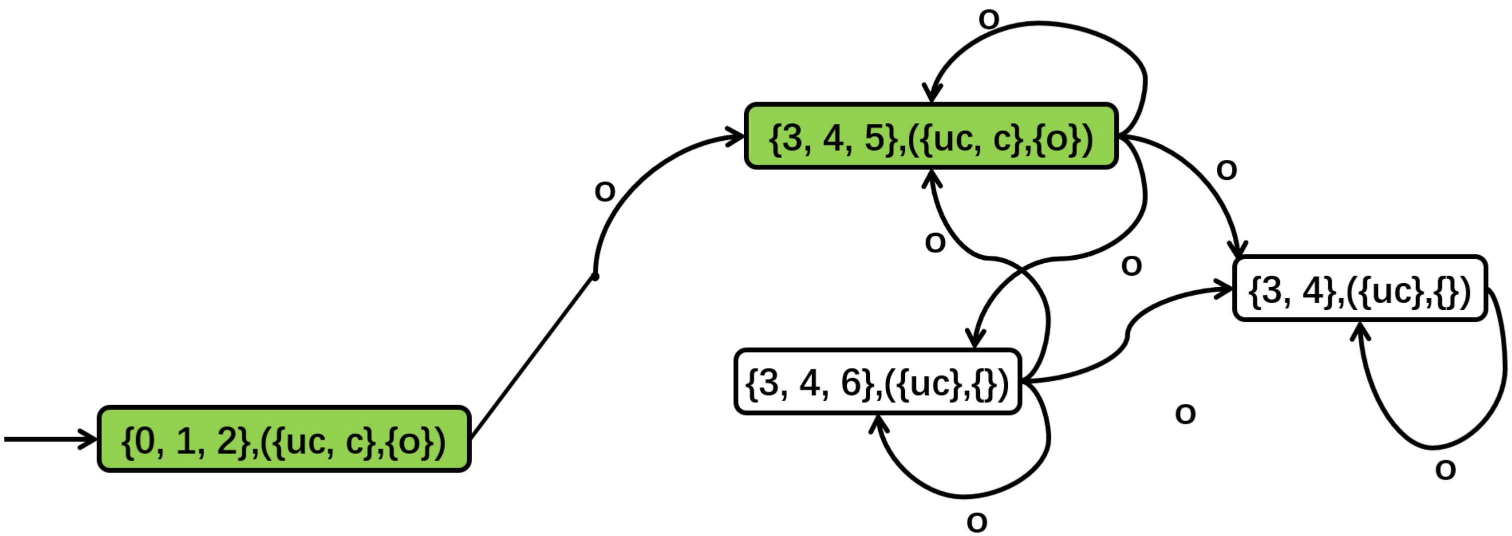
$$\forall q_0' \in Q_{0,A} : |\textsc{Feas}(q_0')| \leq |\textsc{Feas}(q_0)|.$$



Extract a Supervisor from AIC-Safety

At each state q reached, upon the occurrence of observable event $\sigma \in \Sigma_o$, we choose a successor state $q' \in h_A(q, \sigma)$ that contains the maximum number of feasible events among all successor states,

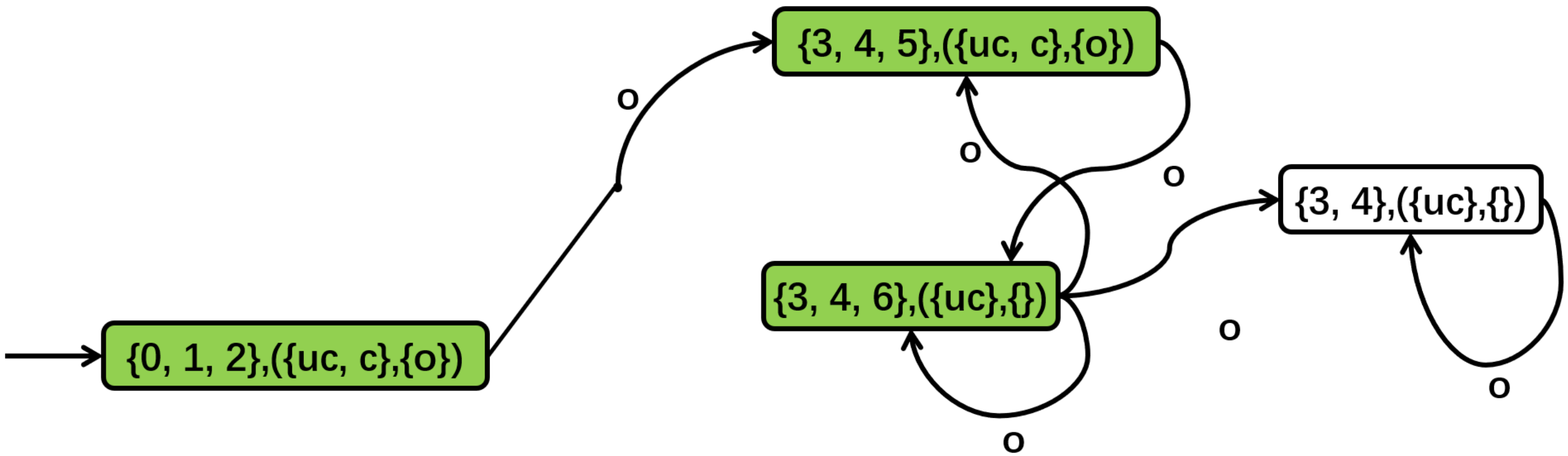
$$\forall q'' \in h_A(q, \sigma) : |\textsc{Feas}(q'')| \leq |\textsc{Feas}(q')|.$$



Extract a Supervisor from AIC-Safety

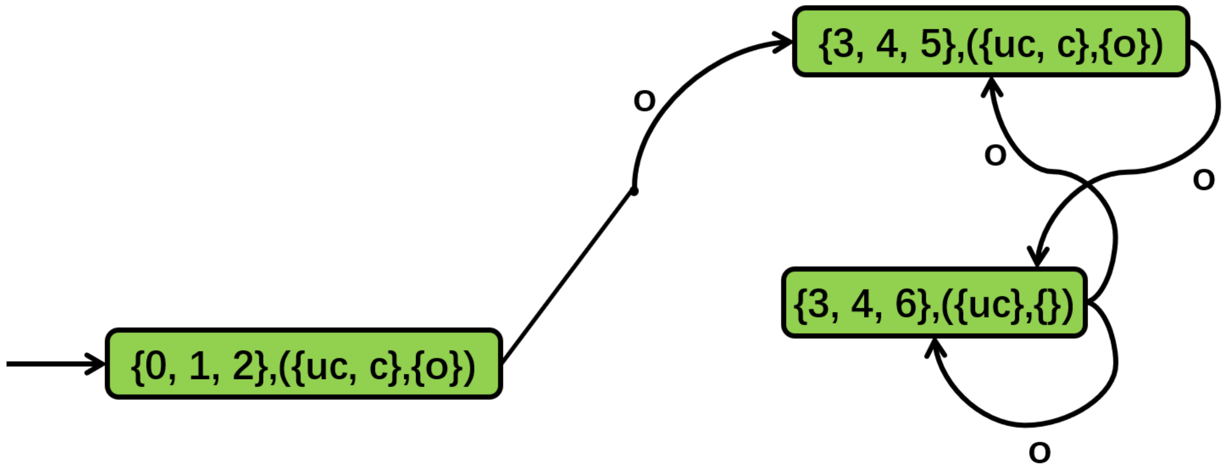
At each state q reached, upon the occurrence of observable event $\sigma \in \Sigma_o$, we choose a successor state $q' \in h_A(q, \sigma)$ that contains the maximum number of feasible events among all successor states,

$$\forall q'' \in h_A(q, \sigma) : |\textsc{Feas}(q'')| \leq |\textsc{Feas}(q')|.$$



Extract a Supervisor from AIC-Safety

We repeat the above procedure until all reachable states are visited (either by a depth-first search or a breath-first search) and denote by $T^* \sqsubseteq \mathcal{A}(G)$ the resulting inclusive controller.



Theorem 4. $\boldsymbol{T}^*$ includes a unique supervisor S^*, which solves **Problem 1**.

Conclusion

Contributions:

- Supervisor control of Timed Discrete Event System
- Solved synthesizing problem of safe supervisors for TDES under partial observation
- The solution is maximally permissive
- Generalize previous synthesis techniques from the untimed setting to the timed setting

Conclusion

Contributions:

- Supervisor control of Timed Discrete Event System
- Solved synthesizing problem of safe supervisors for TDES under partial observation
- The solution is maximally permissive
- Generalize previous synthesis techniques from the untimed setting to the timed setting

Future Direction:

- Investigate the non-blocking control problem for TDES

Conclusion

Contributions:

- Supervisor control of Timed Discrete Event System
- Solved synthesizing problem of safe supervisors for TDES under partial observation
- The solution is maximally permissive
- Generalize previous synthesis techniques from the untimed setting to the timed setting

Future Direction:

- Investigate the non-blocking control problem for TDES

Thank You!