On Maximal Permissiveness in Partially-Observed Discrete Event Systems: Verification and Synthesis

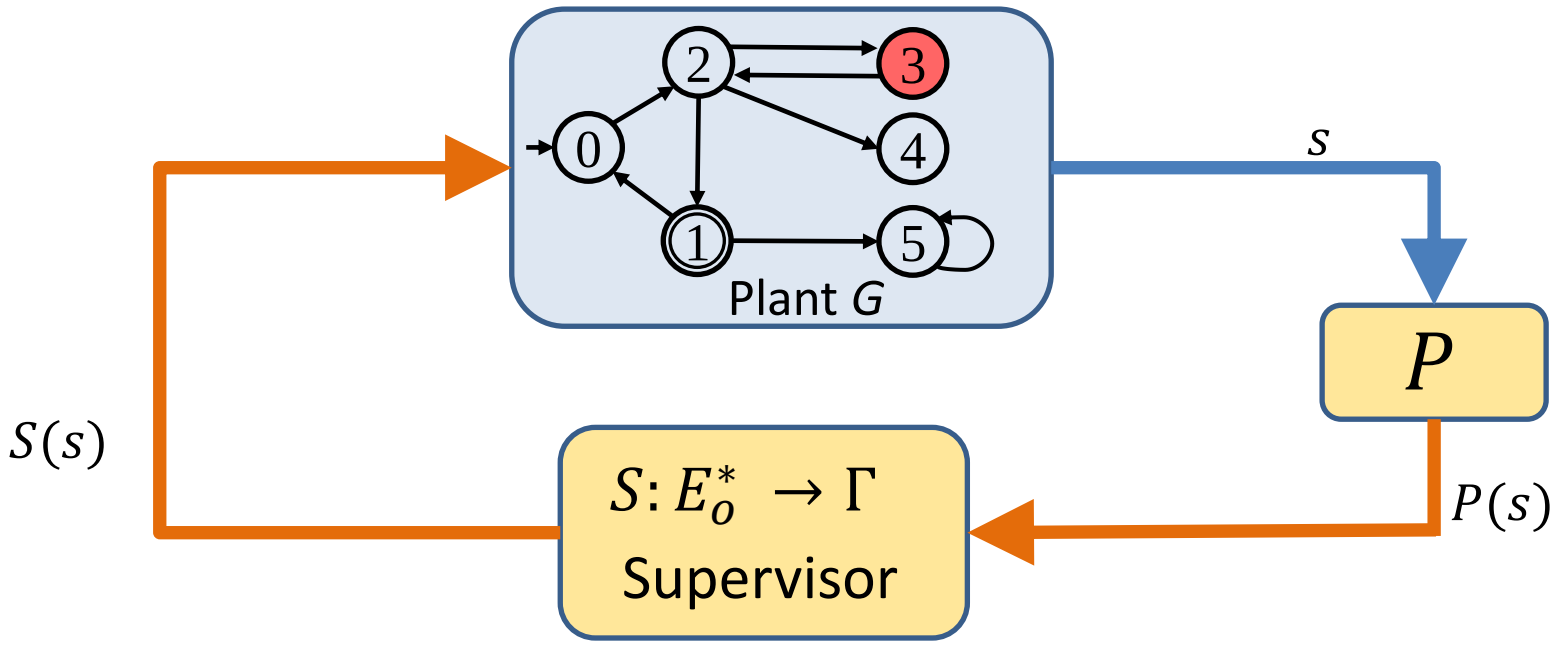
Xiang Yin and Stéphane Lafortune

EECS Department, University of Michigan

13th WODES, May 30-June 1, 2016, Xi'an, China

Introduction

Control Engineering Perspective



- $E = E_c \dot{\cup} E_{uc} = E_o \dot{\cup} E_{uo}$
- Supervisor: $S: E_o^* \rightarrow 2^{\mathrm{E}}$; Disable events in E_c based on its observations
- Closed-loop Behavior: $L(S/G)$

Introduction

- $G = (X, E, f, x_0)$ is a *deterministic* FSA
  - X is the finite set of states
  - E is the finite set of events
  - $f: X \times E \rightarrow X$ is the partial transition function
  - x_0 is the initial state
- Safety specification automaton: $L(H) \subseteq L(G)$

Introduction

- $G = (X, E, f, x_0)$ is a *deterministic* FSA
  - X is the finite set of states
  - E is the finite set of events
  - $f: X \times E \to X$ is the partial transition function
  - x_0 is the initial state
- Safety specification automaton: $L(H) \subseteq L(G)$

We say that a supervisor $S: E_o^* \to 2^E$ is

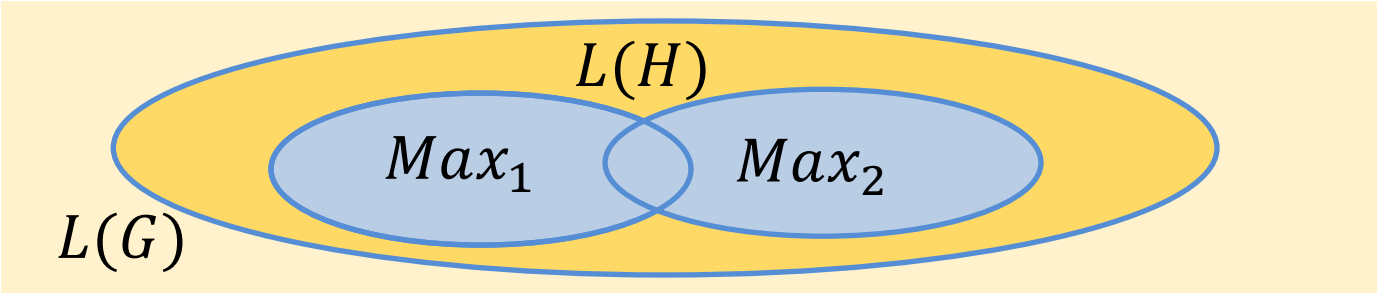
- *Safe*, if $L(S/G) \subseteq L(H)$
- *Maximally Permissive*, if for any safe supervisor S', we have $L(S/G) \not\subset L(S'/G)$.

Introduction

- $G = (X, E, f, x_0)$ is a *deterministic* FSA
  - X is the finite set of states
  - E is the finite set of events
  - $f: X \times E \to X$ is the partial transition function
  - x_0 is the initial state
- Safety specification automaton: $L(H) \subseteq L(G)$

We say that a supervisor $S: E_o^* \to 2^E$ is

- *Safe*, if $L(S/G) \subseteq L(H)$
- *Maximally Permissive*, if for any safe supervisor S', we have $L(S/G) \not\subset L(S'/G)$.



Literature Review

- F. Lin, and W. M. Wonham. "On observability of discrete-event systems." *Inform. Sci.,* 44.3 (1988): 173-198.
- R. Cieslak, et al. "Supervisory control of discrete-event processes with partial observations." *IEEE Transactions on Automatic Control,* 33.3 (1988): 249-260.

- Supremal normal and controllable solution

Literature Review

- F. Lin, and W. M. Wonham. "On observability of discrete-event systems." *Inform. Sci.,* 44.3 (1988): 173-198.
- R. Cieslak, et al. "Supervisory control of discrete-event processes with partial observations." *IEEE Transactions on Automatic Control,* 33.3 (1988): 249-260.
- S. Takai, and T. Ushio. "Effective computation of an $\mathcal{L}_m(G)$-closed, controllable, and observable sublanguage arising in supervisory control." *Sys. Cont. Let.* 49.3 (2003): 191-200.
- K. Cai, R. Zhang, and W. M. Wonham. "Relative observability of discrete-event Systems and its supremal sublanguages." *IEEE Trans. Automatic Control,* 60.3 (2015): 659-670.

  - Supremal normal and controllable solution
  - Solutions larger than supremal normal and controllable solution

Literature Review

- F. Lin, and W. M. Wonham. "On observability of discrete-event systems." *Inform. Sci.,* 44.3 (1988): 173-198.
- R. Cieslak, et al. "Supervisory control of discrete-event processes with partial observations." *IEEE Transactions on Automatic Control,* 33.3 (1988): 249-260.
- S. Takai, and T. Ushio. "Effective computation of an $\mathcal{L}_m(G)$-closed, controllable, and observable sublanguage arising in supervisory control." *Sys. Cont. Let.* 49.3 (2003): 191-200.
- K. Cai, R. Zhang, and W. M. Wonham. "Relative observability of discrete-event Systems and its supremal sublanguages." *IEEE Trans. Automatic Control,* 60.3 (2015): 659-670.

  - Supremal normal and controllable solution
  - Solutions larger than supremal normal and controllable solution
  - These solutions are sound but not complete

Literature Review

- F. Lin, and W. M. Wonham. "On observability of discrete-event systems." *Inform. Sci.,* 44.3 (1988): 173-198.
- R. Cieslak, et al. "Supervisory control of discrete-event processes with partial observations." *IEEE Transactions on Automatic Control,* 33.3 (1988): 249-260.
- S. Takai, and T. Ushio. "Effective computation of an $\mathcal{L}_m(G)$-closed, controllable, and observable sublanguage arising in supervisory control." *Sys. Cont. Let.* 49.3 (2003): 191-200.
- K. Cai, R. Zhang, and W. M. Wonham. "Relative observability of discrete-event Systems and its supremal sublanguages." *IEEE Trans. Automatic Control,* 60.3 (2015): 659-670.

  - Supremal normal and controllable solution
  - Solutions larger than supremal normal and controllable solution
  - These solutions are sound but not complete

- N. Ben Hadj-Alouane, S. Lafortune, and F. Lin. "Centralized and distributed algorithms for on-line synthesis of maximal control policies under partial observation." Discrete Event Dynamic Systems 6.4 (1996): 379-427.
- X. Yin and S. Lafortune. "Synthesis of Maximally Permissive Supervisors for Partially-Observed Discrete-Event Systems." *IEEE Trans. Automatic Control,* 61.5 (2016): 1239-1254.

  - Solutions are both sound and complete
  - **A certain class of maximal policies**

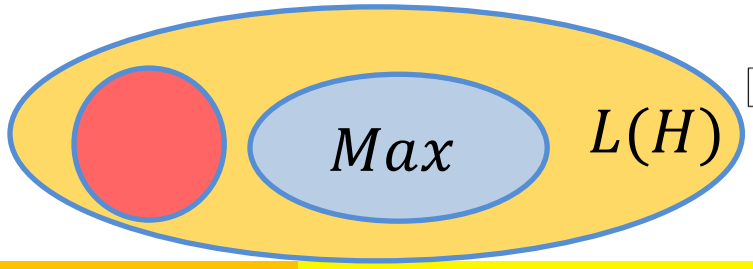
Literature Review

- F. Lin, and W. M. Wonham. "On observability of discrete-event systems." *Inform. Sci.,* 44.3 (1988): 173-198.
- R. Cieslak, et al. "Supervisory control of discrete-event processes with partial observations." *IEEE Transactions on Automatic Control,* 33.3 (1988): 249-260.
- S. Takai, and T. Ushio. "Effective computation of an $\mathcal{L}_m(G)$-closed, controllable, and observable sublanguage arising in supervisory control." *Sys. Cont. Let.* 49.3 (2003): 191-200.
- K. Cai, R. Zhang, and W. M. Wonham. "Relative observability of discrete-event Systems and its supremal sublanguages." *IEEE Trans. Automatic Control,* 60.3 (2015): 659-670.

  - Supremal normal and controllable solution
  - Solutions larger than supremal normal and controllable solution
  - These solutions are sound but not complete

- N. Ben Hadj-Alouane, S. Lafortune, and F. Lin. "Centralized and distributed algorithms for on-line synthesis of maximal control policies under partial observation." Discrete Event Dynamic Systems 6.4 (1996): 379-427.
- X. Yin and S. Lafortune. "Synthesis of Maximally Permissive Supervisors for Partially-Observed Discrete-Event Systems." *IEEE Trans. Automatic Control,* 61.5 (2016): 1239-1254.

  - Solutions are both sound and complete
  - **A certain class of maximal policies**



Problem Formulation

- **Supervisor Verification Problem.**

  Given a safe supervisor $S_R: E_o^* \rightarrow, 2^E$, verify whether or not S_R is maximal.

Problem Formulation

- **Supervisor Verification Problem.**

  Given a safe supervisor $S_R: E_o^* \rightarrow, 2^E$, verify whether or not S_R is maximal.

- **Supervisor Synthesis Problem.**

  Given a non-maximal safe supervisor $S_R: E_o^* \rightarrow 2^E$, find a safe supervisor S such that $L(S_R/G) \subset L(S/G)$.

Problem Formulation

- **Supervisor Verification Problem.**

  Given a safe supervisor $S_R: E_o^* \rightarrow, 2^E$, verify whether or not S_R is maximal.

- **Supervisor Synthesis Problem.**

  Given a non-maximal safe supervisor $S_R: E_o^* \rightarrow 2^E$, find a safe supervisor S such that $L(S_R/G) \subset L(S/G)$.

Motivation:

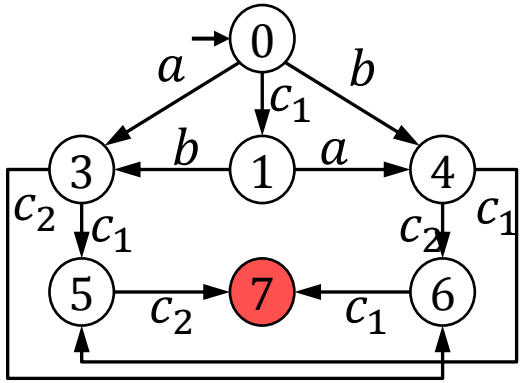
- Lower bound behavior L_r
- $\boldsymbol{L(S_R/G) = L_r{}^{\downarrow CO}}$, the infimal controllable and observable super-language
- Achieve both the lower bound and **permissiveness**

Bipartite Transition System

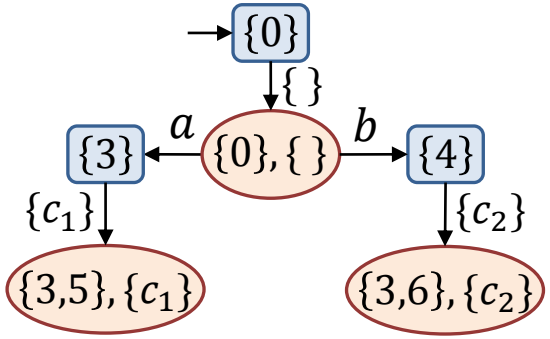
- **Information State**: a set of states, $I := 2^X$
- **BTS:** A bipartite transition system T w.r.t. G is a 7-tuple

$$T = (Q_Y^T, Q_Z^T, h_{YZ}^T, h_{ZY}^T, E, \Gamma, y_0)$$

where
- $Q_Y^T \subseteq I$ is the set of Y-states;
- $Q_Z^T \subseteq I \times \Gamma$ is the set of Z-states so that $z = (I(z), \Gamma(z))$;
- $h_{YZ}^T: Q_Y^T \times \Gamma \to Q_Z^T$ represents the unobservable reach;
- $h_{ZY}^T: Q_Z^T \times E \to Q_Y^T$ represents the observation transition;



$E_c = \{c_1, c_2\}, E_o = \{a, b\}$



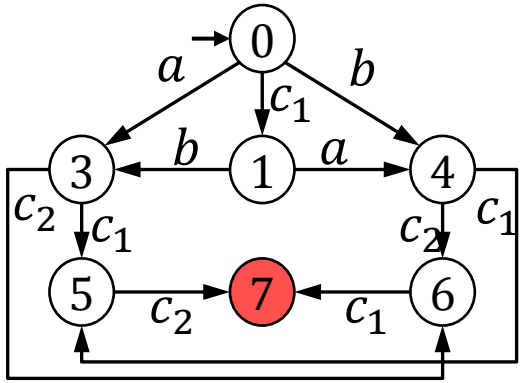
$\boldsymbol{T_R}$

Bipartite Transition System

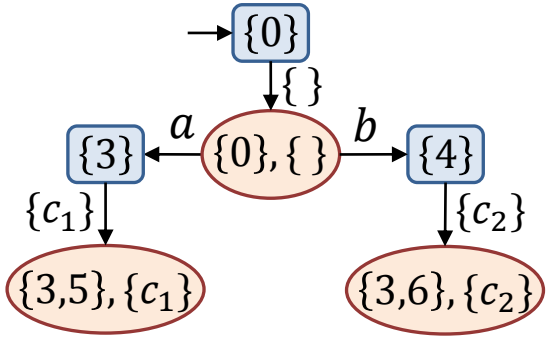
- **Information State**: a set of states, $I := 2^X$
- **BTS:** A bipartite transition system T w.r.t. G is a 7-tuple

$$T = (Q_Y^T, Q_Z^T, h_{YZ}^T, h_{ZY}^T, E, \Gamma, y_0)$$

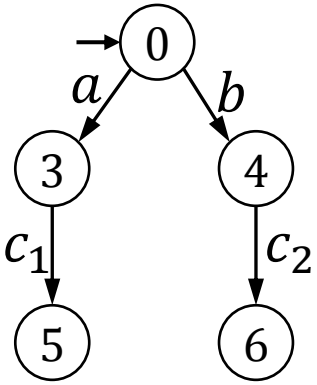
where
- $Q_Y^T \subseteq I$ is the set of Y-states;
- $Q_Z^T \subseteq I \times \Gamma$ is the set of Z-states so that $z = (I(z), \Gamma(z))$;
- $h_{YZ}^T: Q_Y^T \times \Gamma \to Q_Z^T$ represents the unobservable reach;
- $h_{ZY}^T: Q_Z^T \times E \to Q_Y^T$ represents the observation transition;



$E_c = \{c_1, c_2\}, E_o = \{a, b\}$



$\boldsymbol{T_R}$



$\boldsymbol{L(S_R/G)}$

- **The given supervisor S_R can be realized as a BTS T_R**

All Inclusive Controller

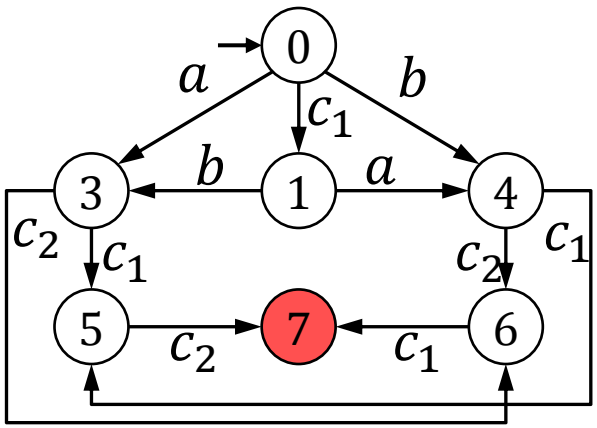
Definition. (AIC).

The All Inclusive Controller

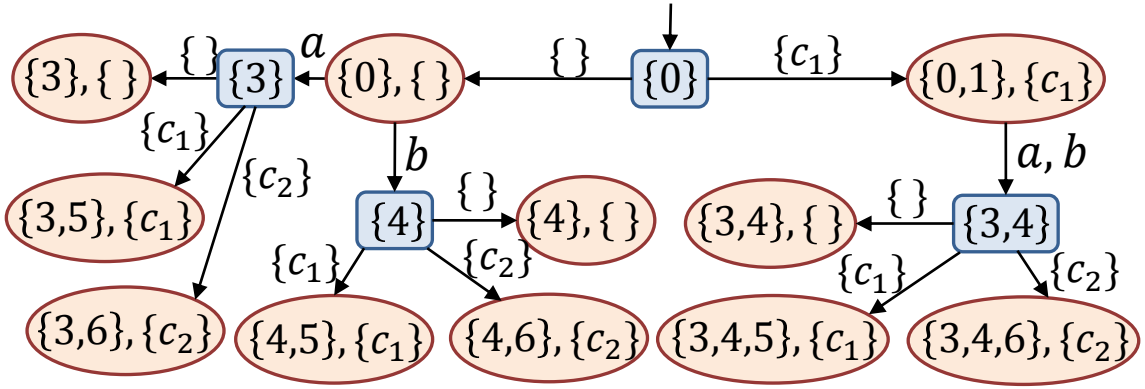
$$\mathcal{AIC}(G) = (Q_Y^{AIC}, Q_Z^{AIC}, h_{YZ}^{AIC}, h_{ZY}^{AIC}, E, \Gamma, y_0),$$

is defined as the *largest* BTS such

1. For any $y \in Q_Y^{AIC}$, there exists at least one control decision
2. For any $z \in Q_Z^{AIC}$, we have
   2.1. all feasible observable events are defined
   2.2. $I(z)$ only contains legal states



$E_c = \{c_1, c_2\}, E_o = \{a, b\}$



$\mathcal{AIC}$

All Inclusive Controller

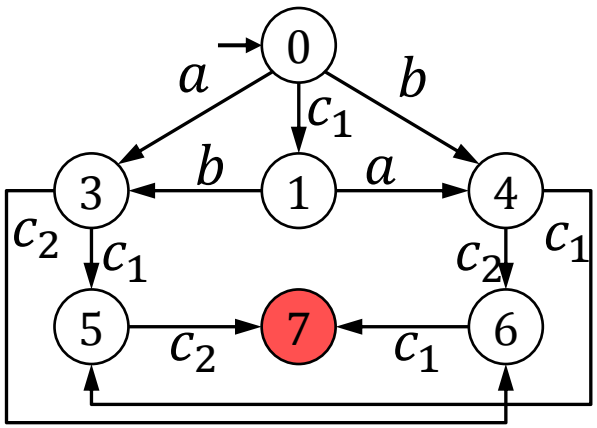
Definition. (AIC).

The All Inclusive Controller

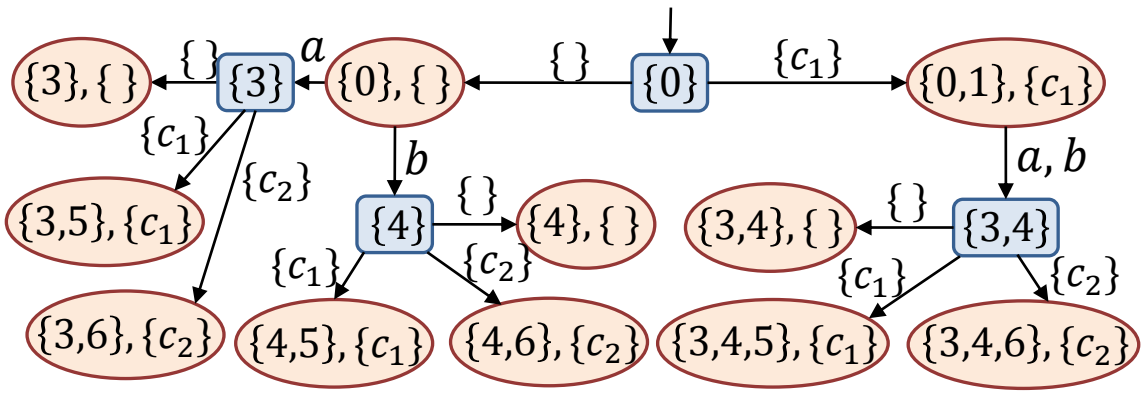
$$\mathcal{AIC}(G) = (Q_Y^{AIC}, Q_Z^{AIC}, h_{YZ}^{AIC}, h_{ZY}^{AIC}, E, \Gamma, y_0),$$

is defined as the *largest* BTS such

1. For any $y \in Q_Y^{AIC}$, there exists at least one control decision
2. For any $z \in Q_Z^{AIC}$, we have
   2.1. all feasible observable events are defined
   2.2. $I(z)$ only contains legal states



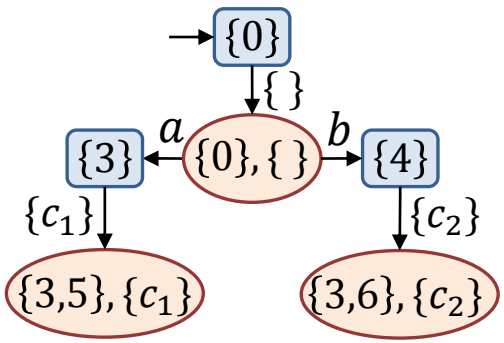
$E_c = \{c_1, c_2\}, E_o = \{a, b\}$



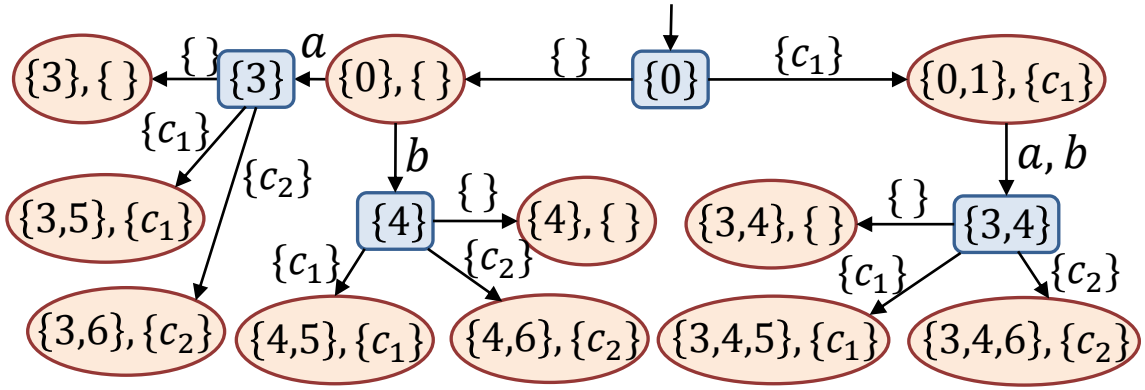
$\mathcal{AIC}$

- **The AIC contains all safe supervisors**

Verification of Maximality: Basic Idea and Difficulties

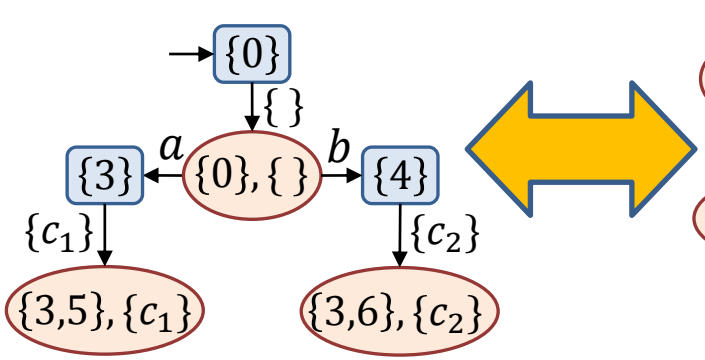


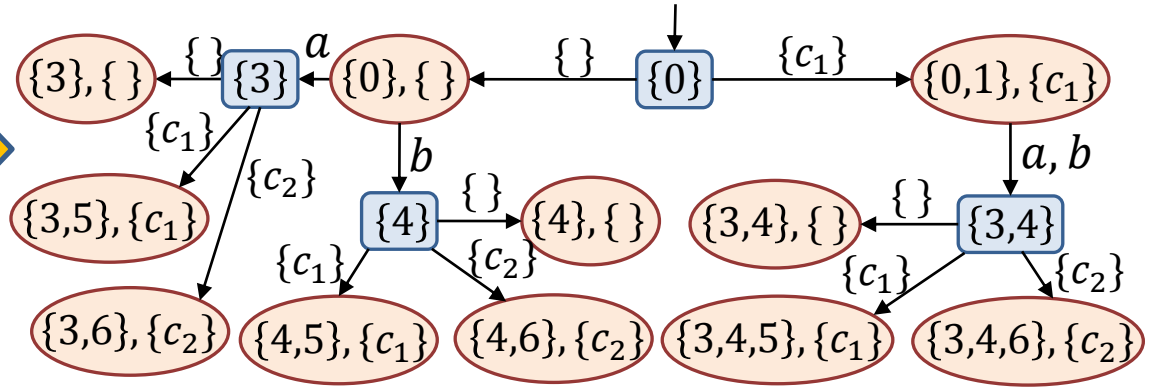
T_R: realizes the given supervisor



$\mathcal{AIC}$: includes all safe supervisors

Verification of Maximality: Basic Idea and Difficulties



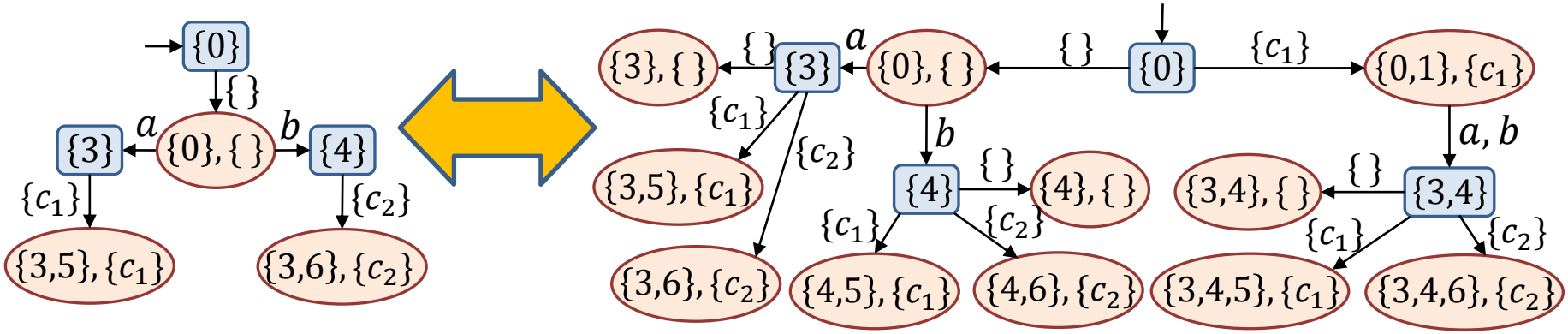


T_R: realizes the given supervisor

$\mathcal{AIC}$: includes all safe supervisors

- "Compare" T_R with the AIC

Verification of Maximality: Basic Idea and Difficulties

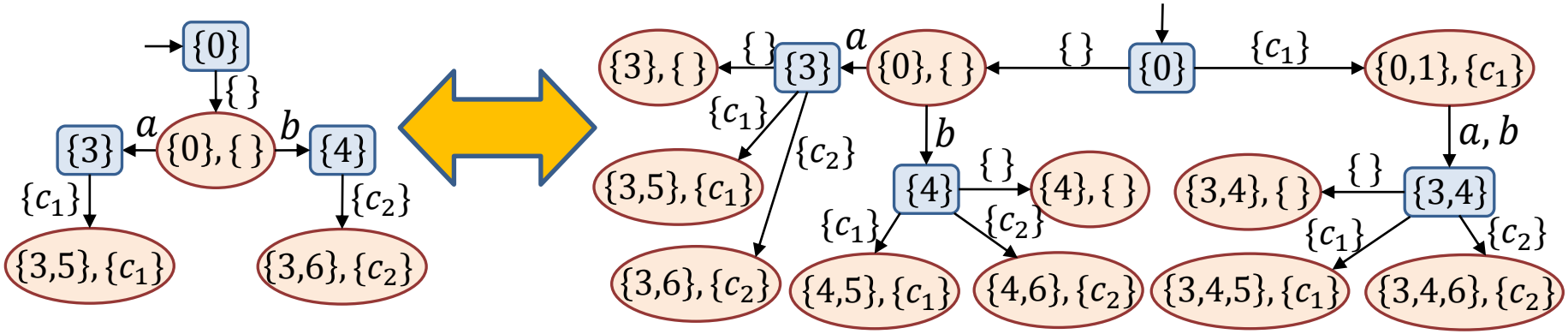


T_R: realizes the given supervisor

$\mathcal{AIC}$: includes all safe supervisors

- "Compare" T_R with the AIC
- How to compare?
- The effect of enabling an event depends on future information

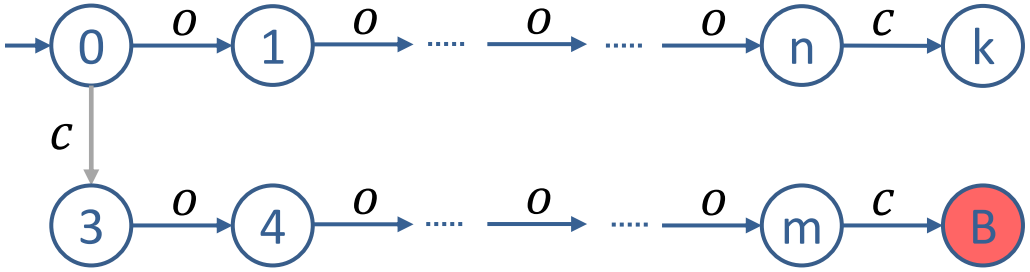
Verification of Maximality: Basic Idea and Difficulties



T_R: realizes the given supervisor

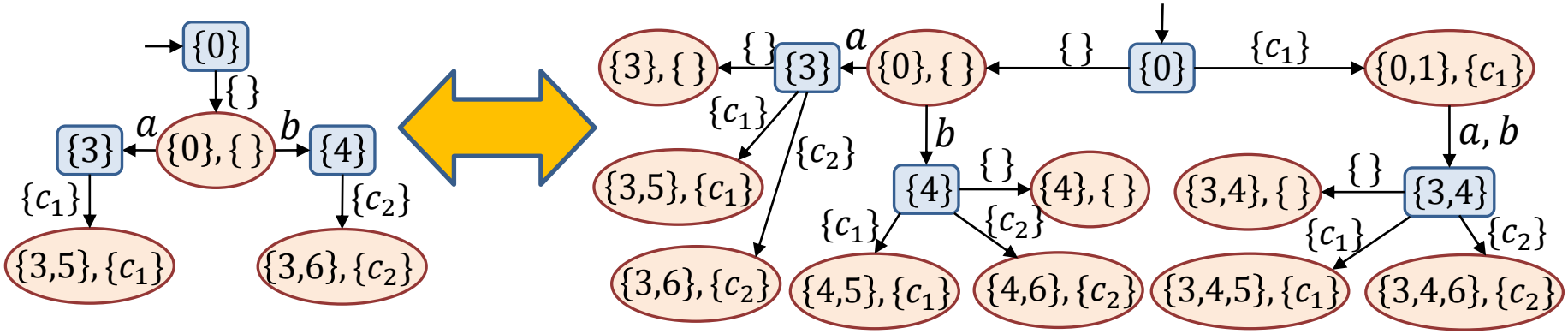
$\mathcal{AIC}$: includes all safe supervisors

- "Compare" T_R with the AIC
- How to compare?
- The effect of enabling an event depends on future information



$E_c = \{c\}, E_o = \{o\}$

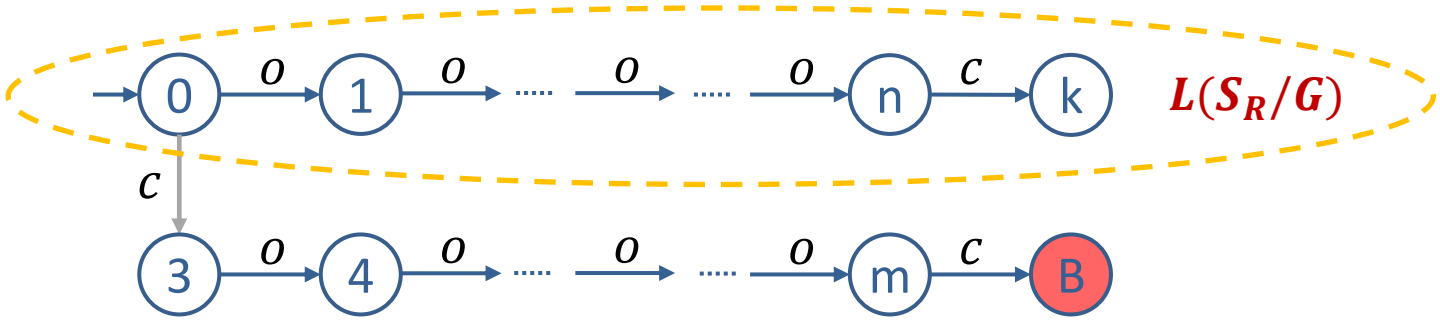
Verification of Maximality: Basic Idea and Difficulties



T_R: realizes the given supervisor

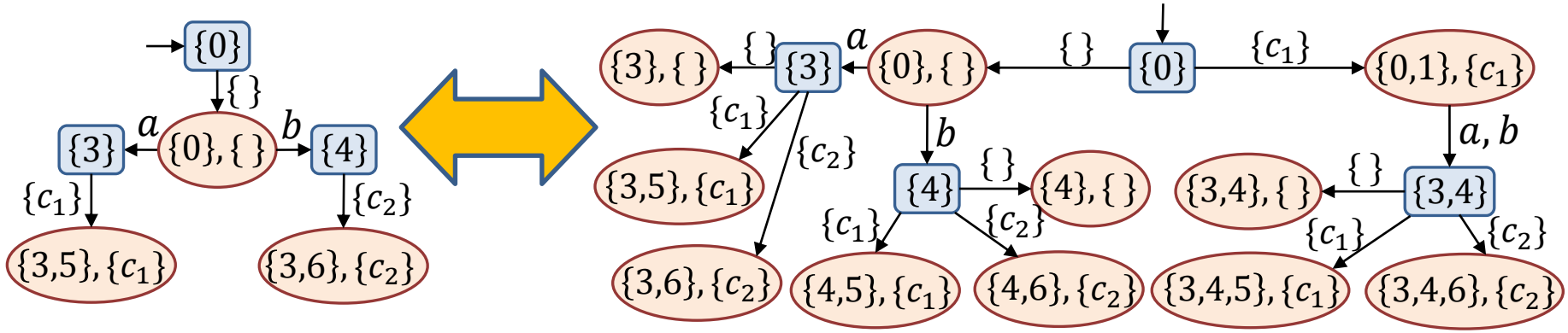
$\mathcal{AIC}$: includes all safe supervisors

- "Compare" T_R with the AIC
- How to compare?
- The effect of enabling an event depends on future information



$E_c = \{c\}, E_o = \{o\}$

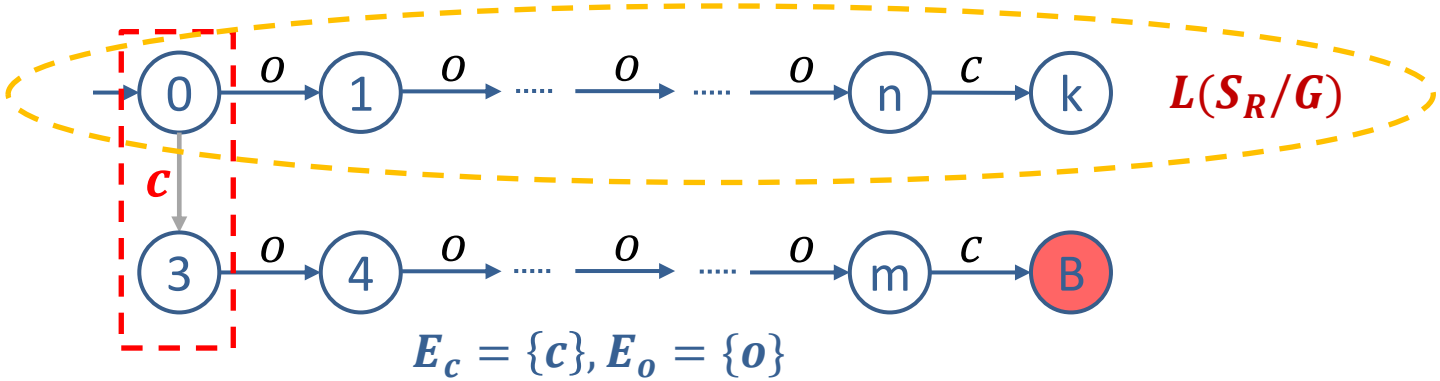
Verification of Maximality: Basic Idea and Difficulties



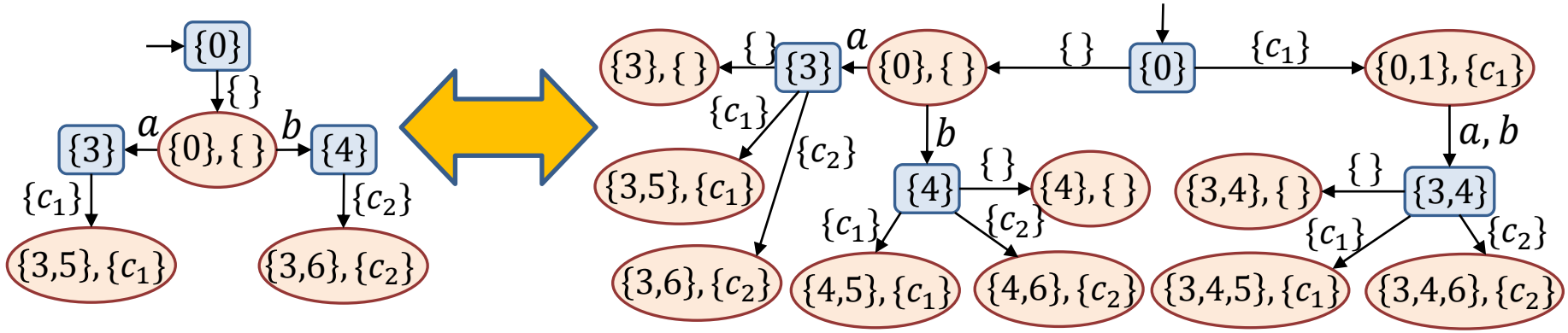
T_R: realizes the given supervisor

$\mathcal{AIC}$: includes all safe supervisors

- "Compare" T_R with the AIC
- How to compare?
- The effect of enabling an event depends on future information



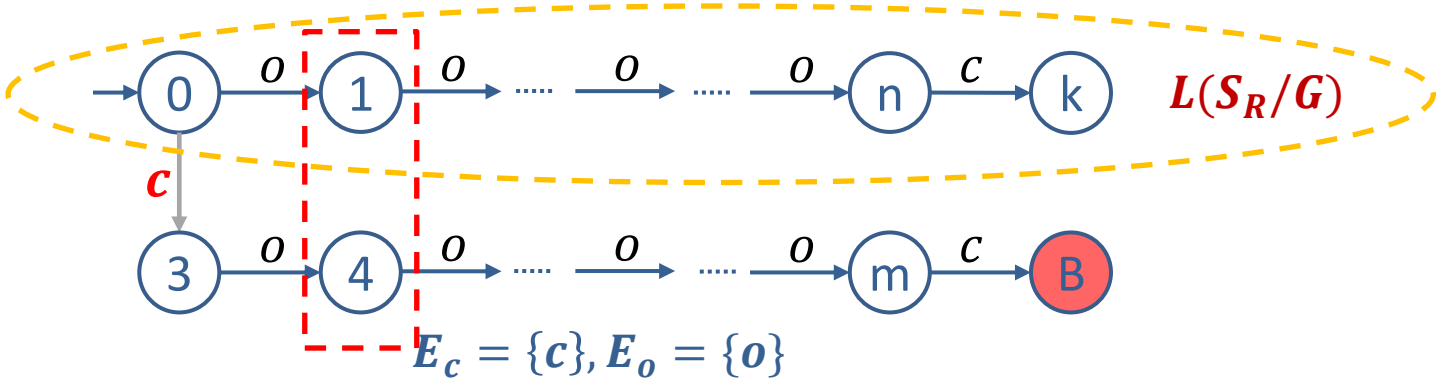
Verification of Maximality: Basic Idea and Difficulties



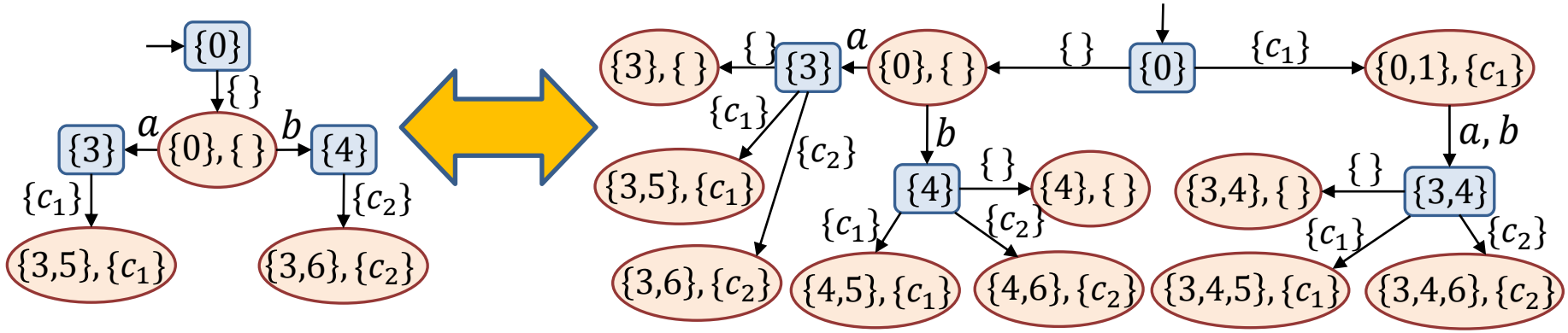
T_R: realizes the given supervisor

$\mathcal{AIC}$: includes all safe supervisors

- "Compare" T_R with the AIC
- How to compare?
- The effect of enabling an event depends on future information



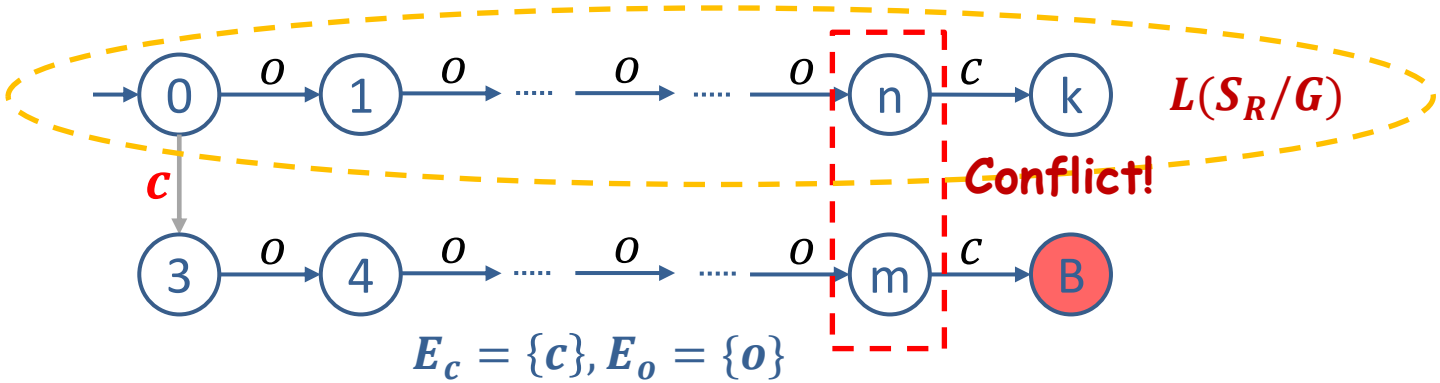
Verification of Maximality: Basic Idea and Difficulties



T_R: realizes the given supervisor

$\mathcal{AIC}$: includes all safe supervisors

- "Compare" T_R with the AIC
- How to compare?
- The effect of enabling an event depends on future information



Control Simulation Relation

Definition. (Control Simulation Relation)

Let T_1 and T_2 between BTSs. A relation $\Phi = \Phi_Y \cup \Phi_Z \subseteq \left(Q_Y^{T_1} \times Q_Y^{T_2}\right) \times \left(Q_Z^{T_1} \times Q_Z^{T_2}\right)$ is said to be a **control** simulation relation from $\boldsymbol{T_1}$ **to** $\boldsymbol{T_2}$ if the following conditions hold:

1. $(y_0, y_0) \in \Phi_Y$;
2. For every $(y_1, y_2) \in \Phi_Y$, we have that: for any $y_1 \xrightarrow{\gamma_1} z_1$ in T_1, there exists $y_2 \xrightarrow{\gamma_2} z_2$ such that $(z_1, z_2) \in \Phi_Z$ and $\boldsymbol{\gamma_1 \subseteq \gamma_2}$.
3. For every $(z_1, z_2) \in \Phi_z$, we have that: for any $z_1 \xrightarrow{\sigma} y_1$ in T_1, there exists $z_2 \xrightarrow{\sigma} y_2$ such that $(y_1, y_2) \in \Phi_Y$.

Control Simulation Relation

Definition. (Control Simulation Relation)

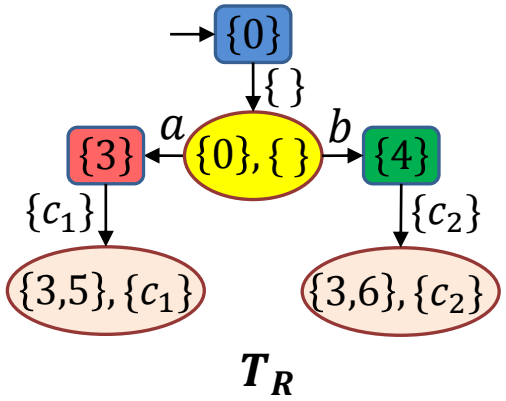
Let T_1 and T_2 between BTSs. A relation $\Phi = \Phi_Y \cup \Phi_Z \subseteq \left(Q_Y^{T_1} \times Q_Y^{T_2}\right) \times \left(Q_Z^{T_1} \times Q_Z^{T_2}\right)$ is said to be a **control** simulation relation from $\boldsymbol{T_1}$ **to** $\boldsymbol{T_2}$ if the following conditions hold:

1. $(y_0, y_0) \in \Phi_Y$;
2. For every $(y_1, y_2) \in \Phi_Y$, we have that: for any $y_1 \xrightarrow{\gamma_1} z_1$ in T_1, there exists $y_2 \xrightarrow{\gamma_2} z_2$ such that $(z_1, z_2) \in \Phi_Z$ and $\boldsymbol{\gamma_1 \subseteq \gamma_2}$.
3. For every $(z_1, z_2) \in \Phi_z$, we have that: for any $z_1 \xrightarrow{\sigma} y_1$ in T_1, there exists $z_2 \xrightarrow{\sigma} y_2$ such that $(y_1, y_2) \in \Phi_Y$.

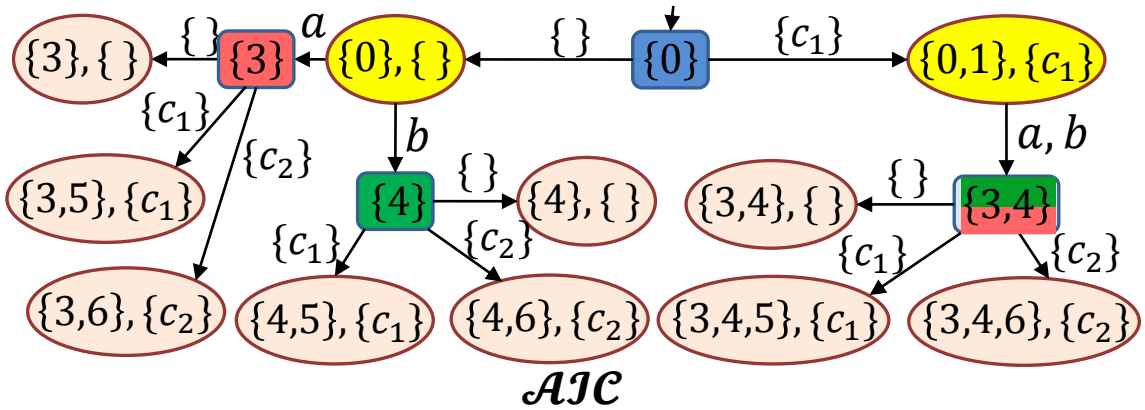
- There exists a unique maximal CSR $\Phi^*(T_1, T_2)$ from T_1 to T_2 if one exists
- $\Phi^*(T_1, T_2)$ can be computed by

$$\Phi^*(T_1, T_2) = \lim_{k \to \infty} F^k\left(\left(Q_Y^{T_1} \times Q_Y^{T_2}\right) \cup \left(Q_Z^{T_1} \times Q_Z^{T_2}\right)\right)$$

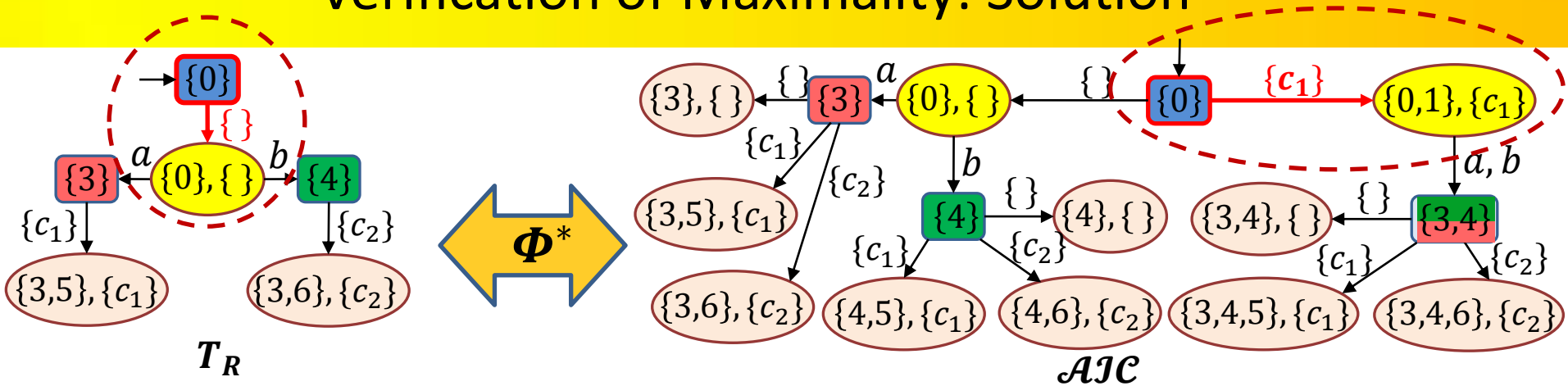
Verification of Maximality: Solution



Φ^*



Verification of Maximality: Solution



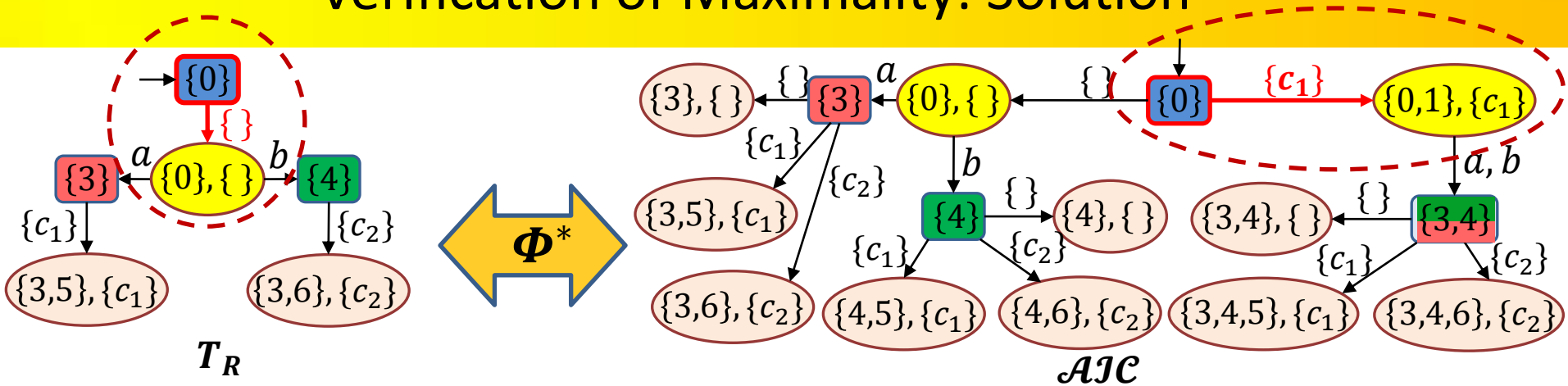
Replacement.

Let y be a Y-state in T_R and $c_{T_R}(y)$ be the control decision defined at y.

We say that control decision γ **replaces** $c_{T_R}(y)$ at y if

1. γ is defined at y in the AIC
2. $c_{T_R}(y) \subset \gamma$
3. $(z, z') \in \Phi^*(T_R, \mathcal{AIC})$, where $y \xrightarrow{c_{T_R}(y)} z$ and $y \xrightarrow{\gamma} z'$

Verification of Maximality: Solution



Replacement.

Let y be a Y-state in T_R and $c_{T_R}(y)$ be the control decision defined at y.

We say that control decision γ **replaces** $c_{T_R}(y)$ at y if

1. γ is defined at y in the AIC
2. $c_{T_R}(y) \subset \gamma$
3. $(z, z') \in \Phi^*(T_R, \mathcal{AIC})$, where $y \xrightarrow{c_{T_R}(y)} z$ and $y \xrightarrow{\gamma} z'$

Theorem.

Supervisor S_R is maximal iff no control decision in T_R can be replaced

Synthesis of Larger Supervisor: Basic Idea and Difficulties

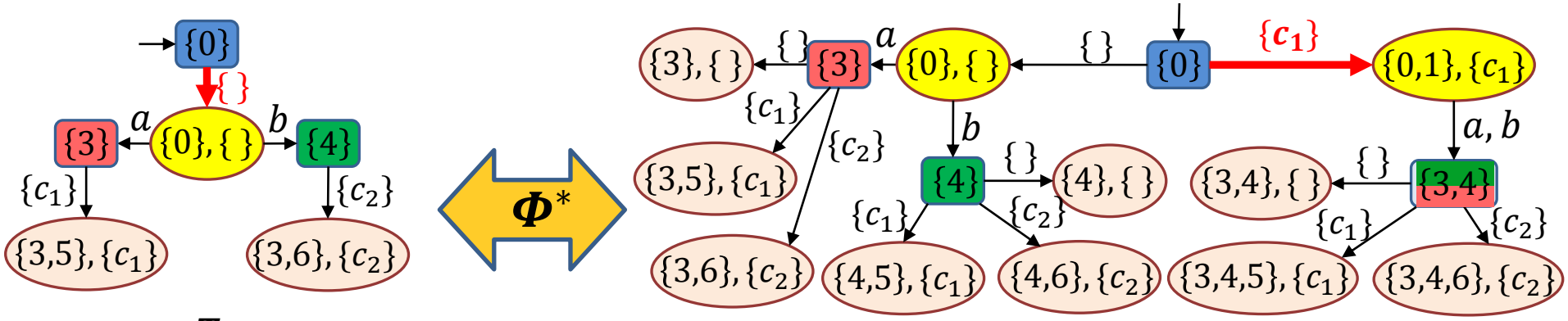
- **Construct a new BTS: replace the original control decision in T_R by a larger one**

Synthesis of Larger Supervisor: Basic Idea and Difficulties

- **Construct a new BTS: replace the original control decision in T_R by a larger one**
- **Such a BTS may not exist!**

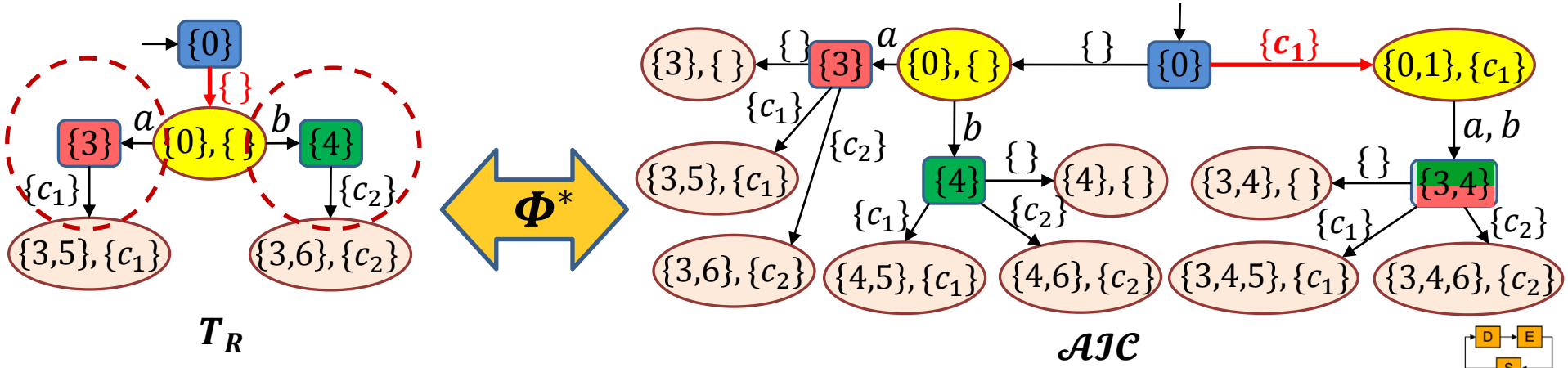
Synthesis of Larger Supervisor: Basic Idea and Difficulties

- **Construct a new BTS: replace the original control decision in T_R by a larger one**
- **Such a BTS may not exist!**



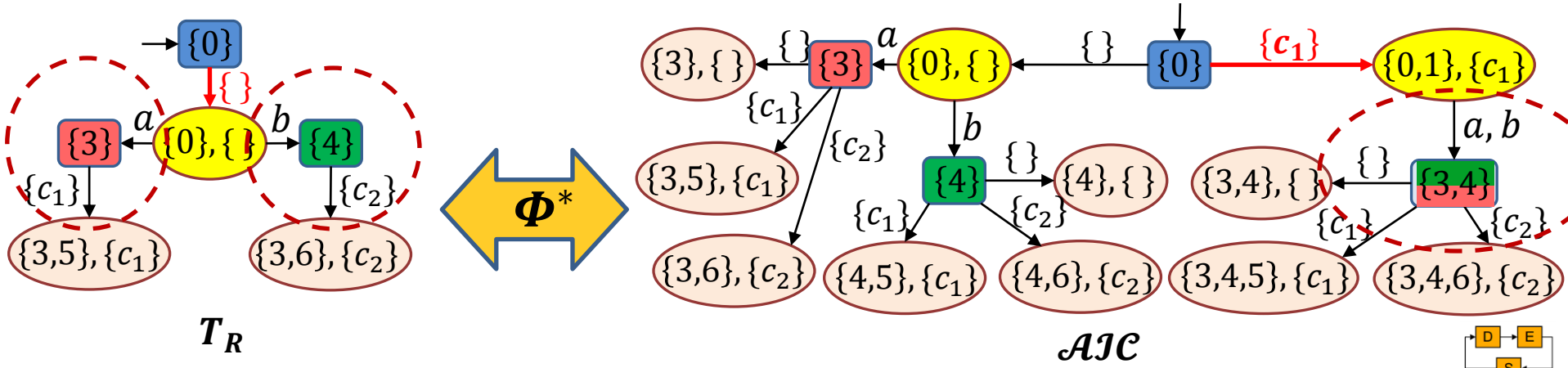
Synthesis of Larger Supervisor: Basic Idea and Difficulties

- **Construct a new BTS: replace the original control decision in T_R by a larger one**
- **Such a BTS may not exist!**



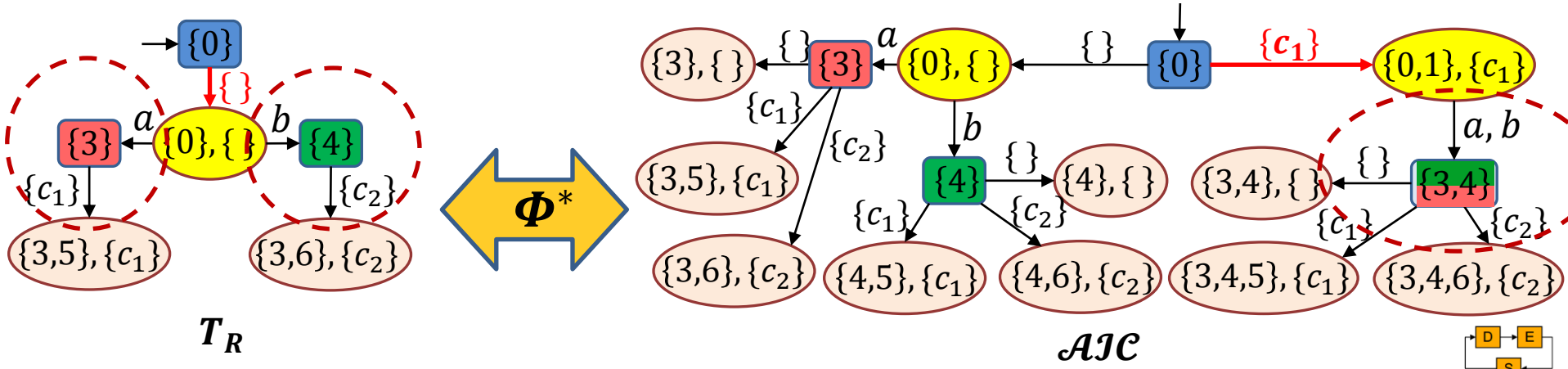
Synthesis of Larger Supervisor: Basic Idea and Difficulties

- **Construct a new BTS: replace the original control decision in T_R by a larger one**
- **Such a BTS may not exist!**



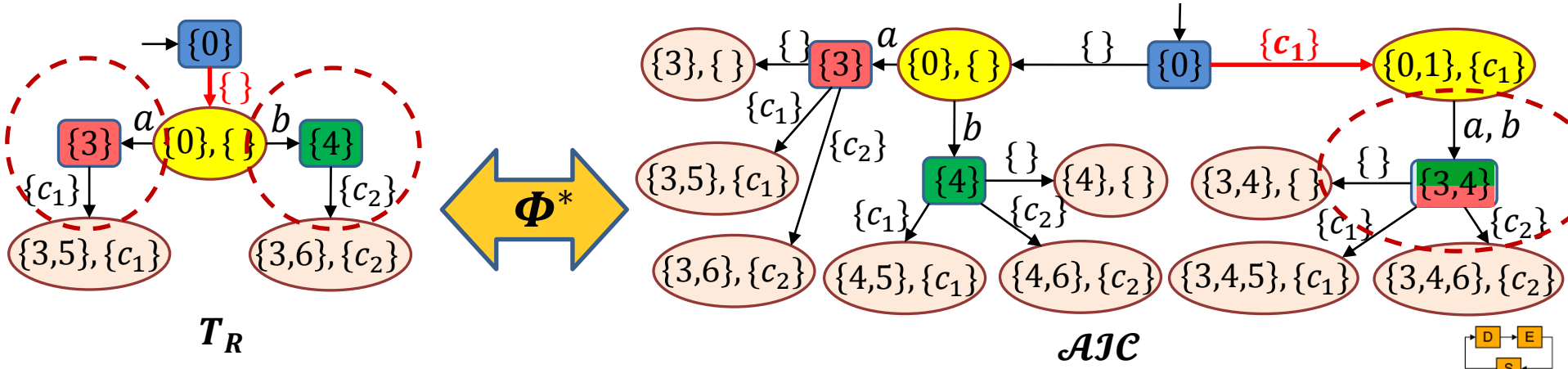
Synthesis of Larger Supervisor: Basic Idea and Difficulties

- **Construct a new BTS: replace the original control decision in T_R by a larger one**
- **Such a BTS may not exist!**
- **Information Merge Phenomenon: Information is lost**



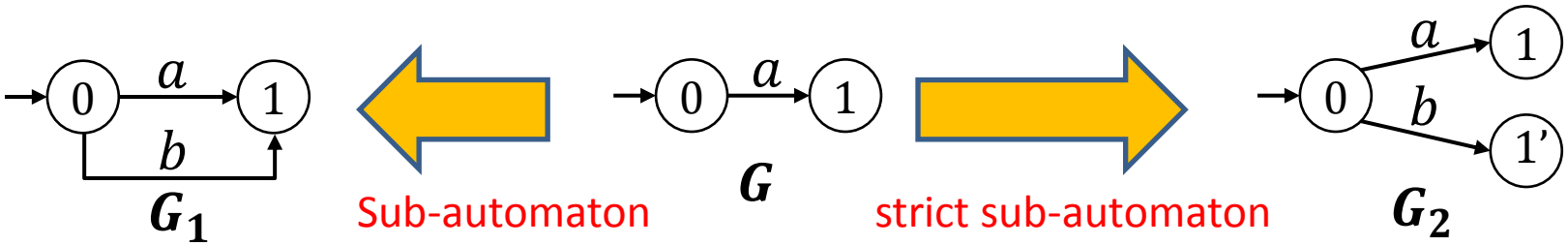
Synthesis of Larger Supervisor: Basic Idea and Difficulties

- **Construct a new BTS: replace the original control decision in T_R by a larger one**
- **Such a BTS may not exist!**
- **Information Merge Phenomenon: Information is lost**
- **2^X may not be sufficient for the synthesis problem!**



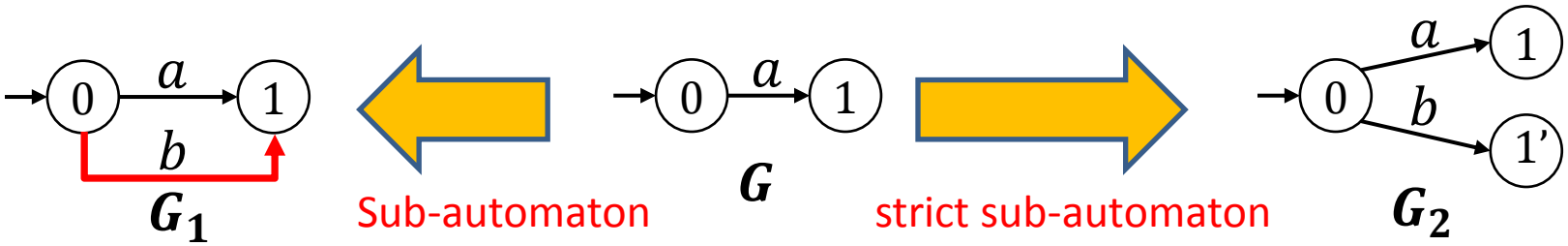
The Role of Strict Sub-automaton

- The information merge phenomenon **will not occur** under the assumption that R is a **strict sub-automaton** of G, where $L(R) = L(S_R/G)$.



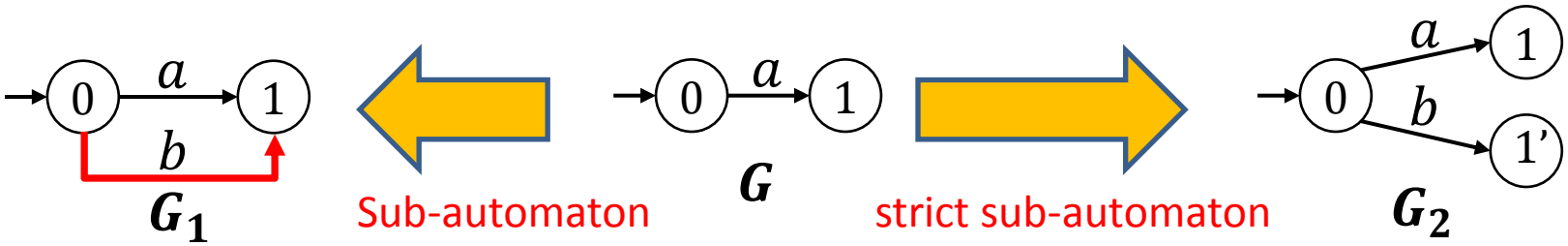
The Role of Strict Sub-automaton

- The information merge phenomenon **will not occur** under the assumption that R is a **strict sub-automaton** of G, where $L(R) = L(S_R/G)$.
- **Strict sub-automaton**: if a string goes outside, then it stays outside forever

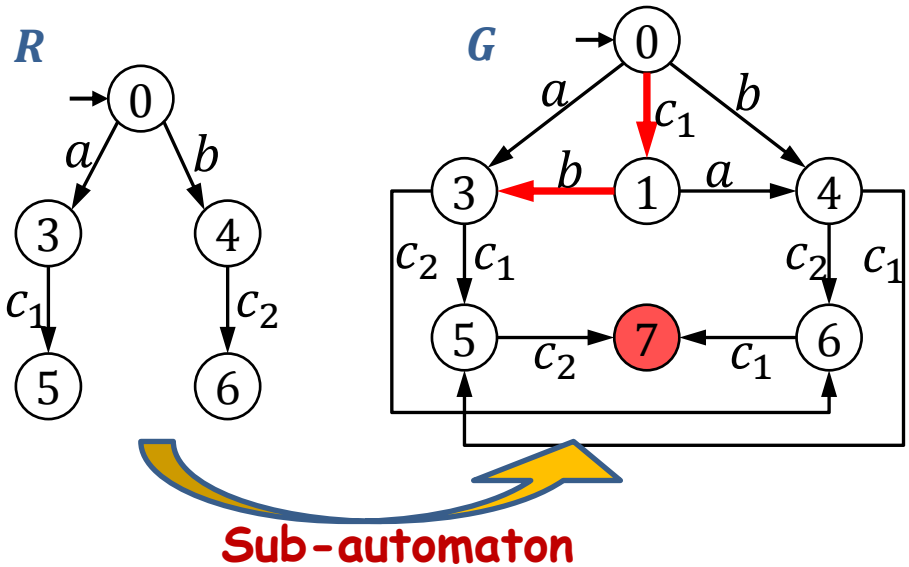


The Role of Strict Sub-automaton

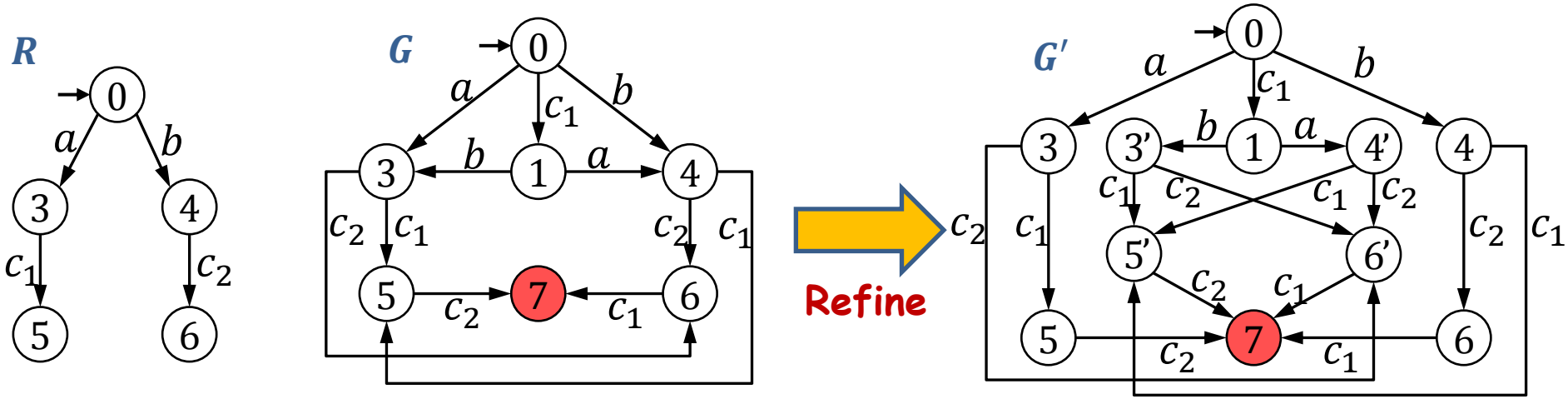
- The information merge phenomenon **will not occur** under the assumption that R is a **strict sub-automaton** of G, where $L(R) = L(S_R/G)$.
- **Strict sub-automaton**: if a string goes outside, then it stays outside forever
- We can always obtain strict sub-automaton [Cho & Marcus, 1989]



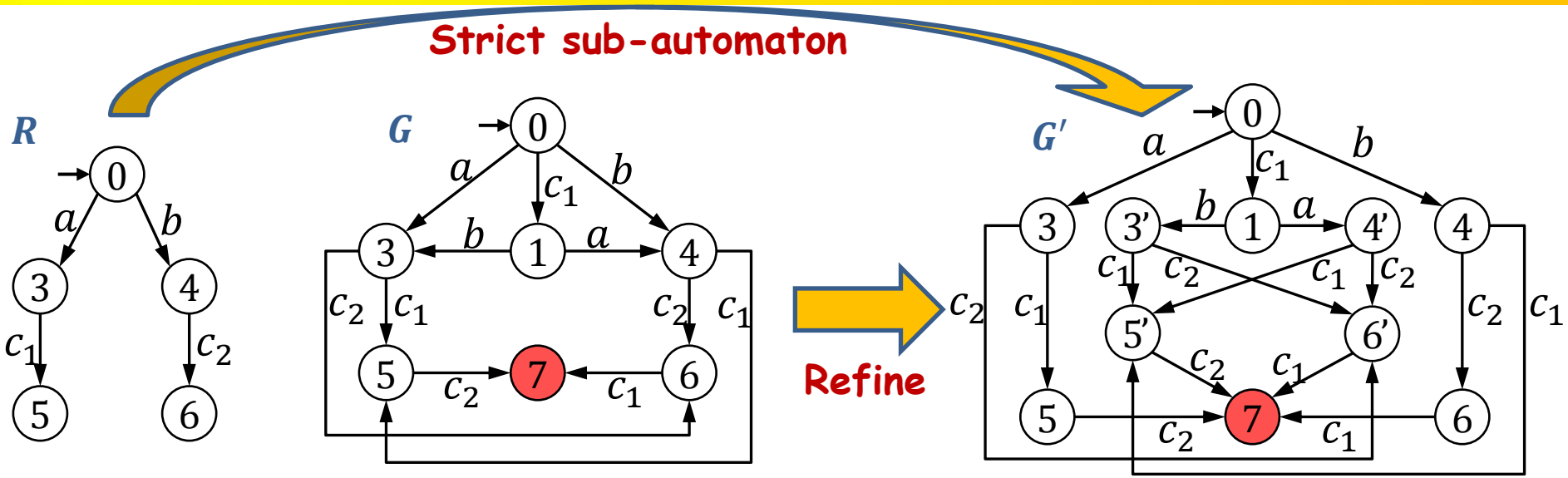
The Role of Strict Sub-automaton



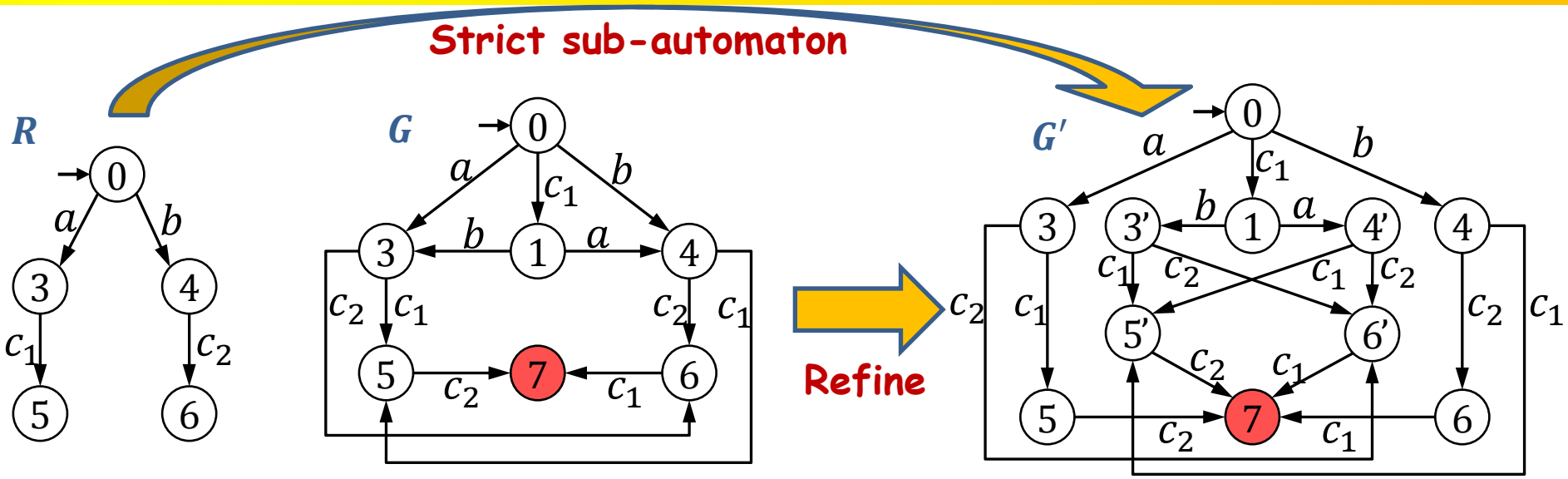
The Role of Strict Sub-automaton

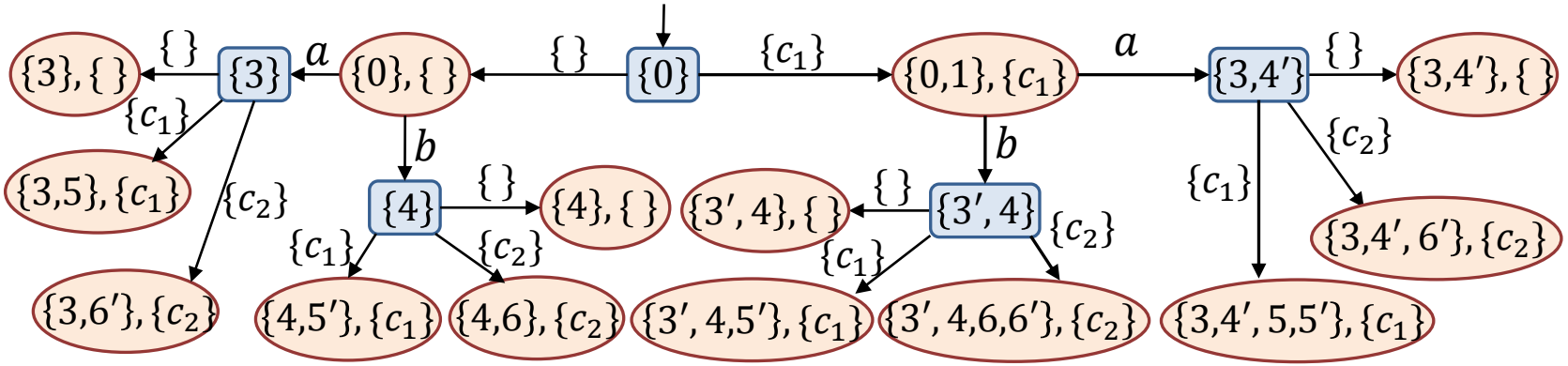


The Role of Strict Sub-automaton



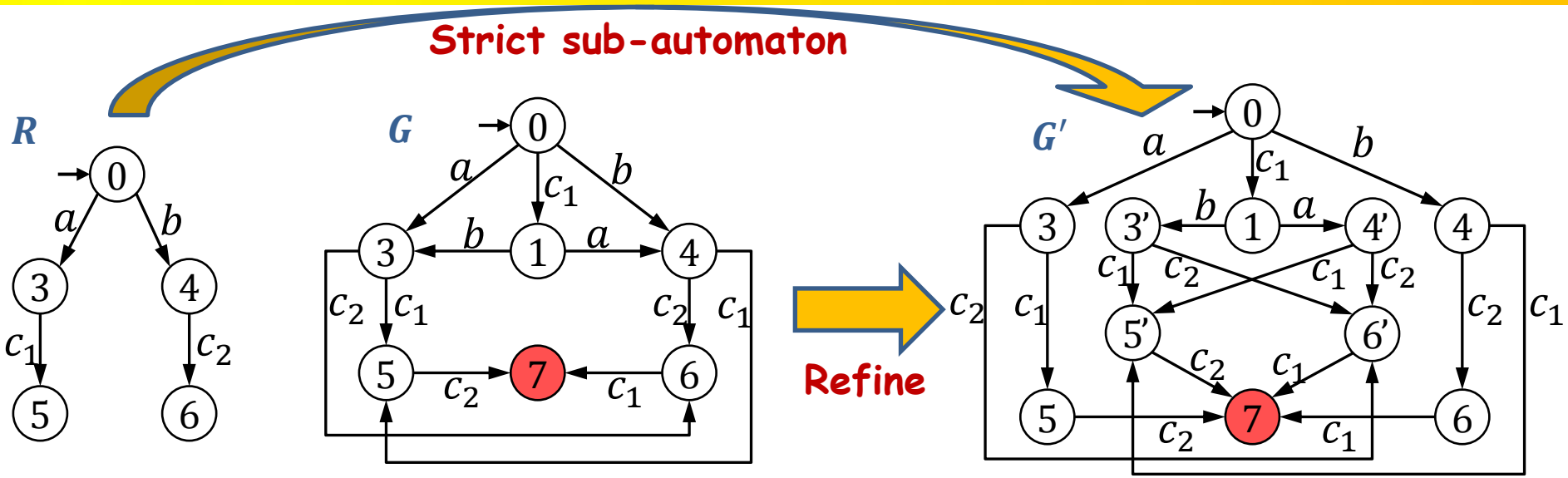
The Role of Strict Sub-automaton

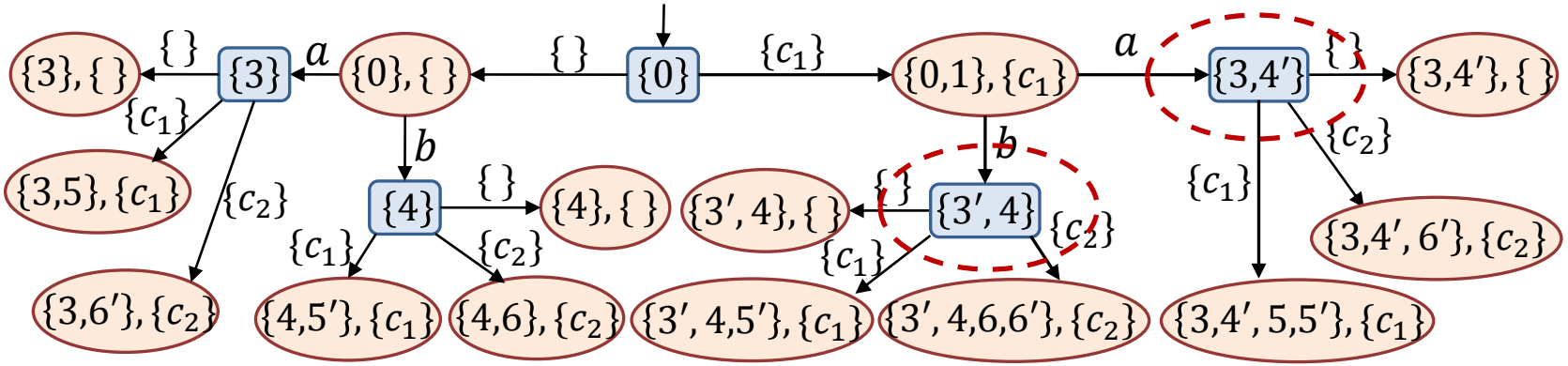




AIC for the refined system

The Role of Strict Sub-automaton





AIC for the refined system

Synthesis Algorithm

Synthesis Steps:

1. Construct BTS T_R and $\mathcal{AIC}(G)$ (make sure R is a strict sub-automaton of G)
2. Compute the maximal CSR Φ^* from T_R to $\mathcal{AIC}(G)$
3. Find a Y-state y in T_R such that its control decision can be replaced by γ
4. Construct BTS T^* by

   - For Y-state y, choose γ which is larger than $c_{T_R}(y)$

   - For other Y-states, choose the same control decisions in T_R

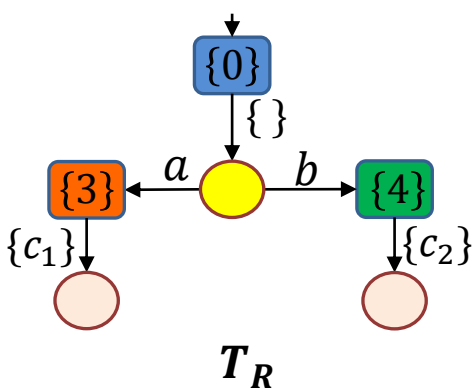
Synthesis Algorithm

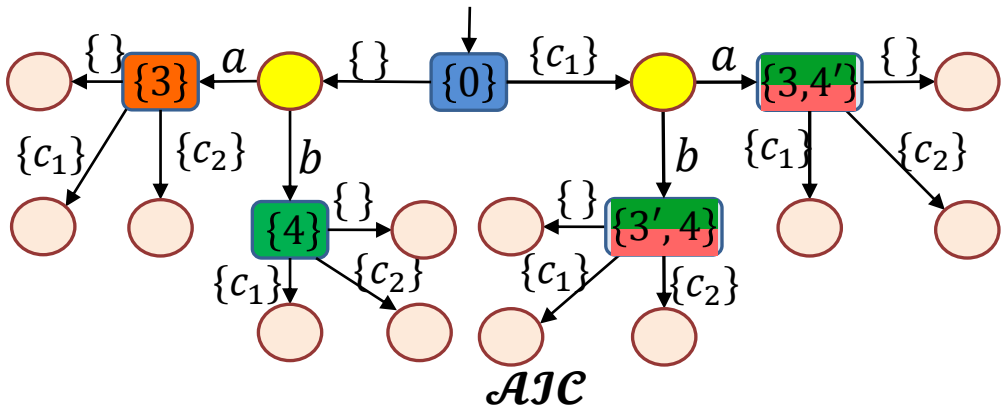
Synthesis Steps:

1. Construct BTS T_R and $\mathcal{AIC}(G)$ (make sure R is a strict sub-automaton of G)
2. Compute the maximal CSR Φ^* from T_R to $\mathcal{AIC}(G)$
3. Find a Y-state y in T_R such that its control decision can be replaced by γ
4. Construct BTS T^* by

   - For Y-state y, choose γ which is larger than $c_{T_R}(y)$

   - For other Y-states, choose the same control decisions in T_R





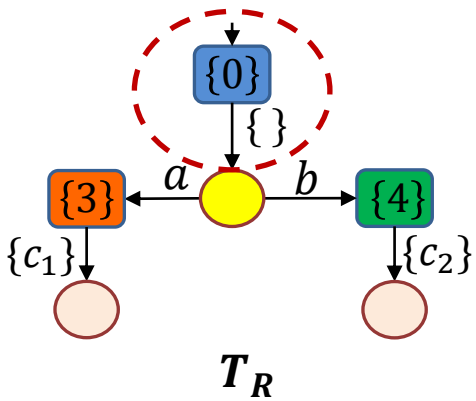
Synthesis Algorithm

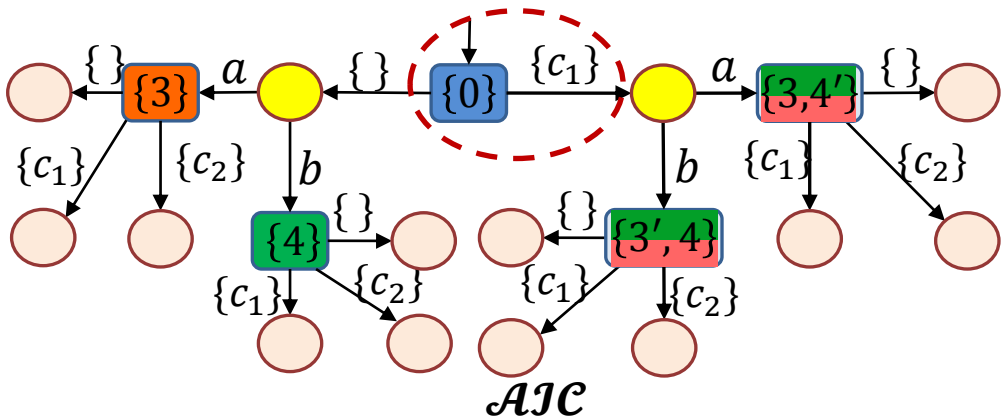
Synthesis Steps:

1. Construct BTS T_R and $\mathcal{AIC}(G)$ (make sure R is a strict sub-automaton of G)
2. Compute the maximal CSR Φ^* from T_R to $\mathcal{AIC}(G)$
3. Find a Y-state y in T_R such that its control decision can be replaced by γ
4. Construct BTS T^* by

   - For Y-state y, choose γ which is larger than $c_{T_R}(y)$

   - For other Y-states, choose the same control decisions in T_R





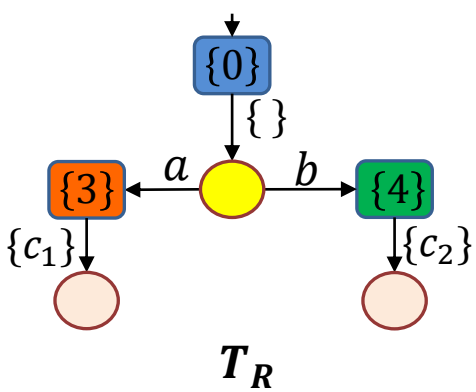
Synthesis Algorithm

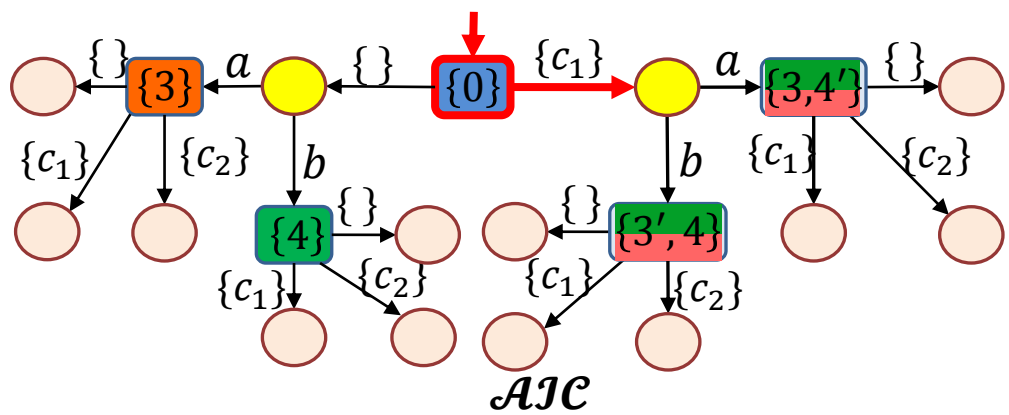
Synthesis Steps:

1. Construct BTS T_R and $\mathcal{AIC}(G)$ (make sure R is a strict sub-automaton of G)
2. Compute the maximal CSR Φ^* from T_R to $\mathcal{AIC}(G)$
3. Find a Y-state y in T_R such that its control decision can be replaced by γ
4. Construct BTS T^* by

   - For Y-state y, choose γ which is larger than $c_{T_R}(y)$

   - For other Y-states, choose the same control decisions in T_R





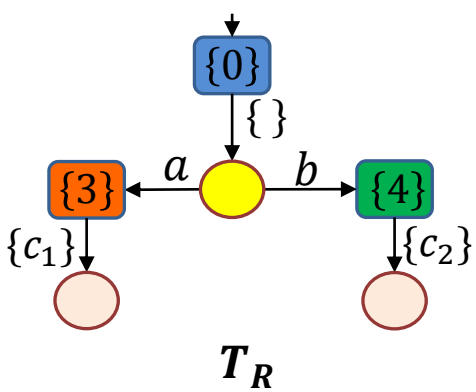
Synthesis Algorithm

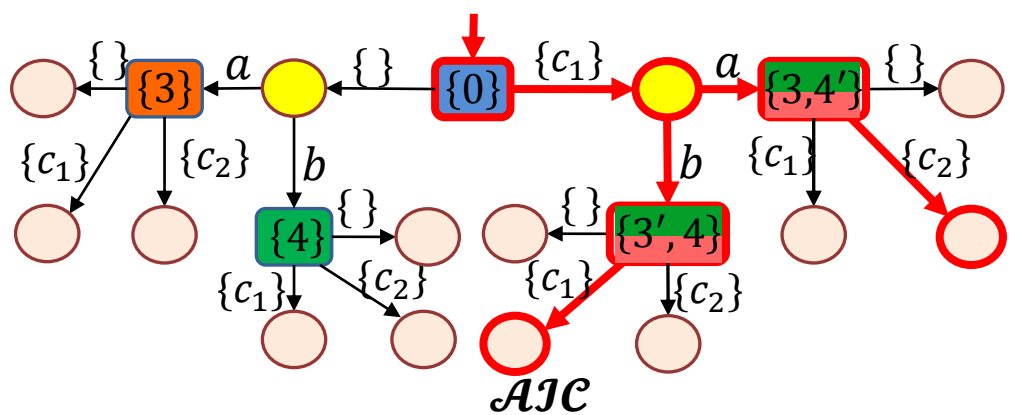
Synthesis Steps:

1. Construct BTS T_R and $\mathcal{AIC}(G)$ (make sure R is a strict sub-automaton of G)
2. Compute the maximal CSR Φ^* from T_R to $\mathcal{AIC}(G)$
3. Find a Y-state y in T_R such that its control decision can be replaced by γ
4. Construct BTS T^* by

   - For Y-state y, choose γ which is larger than $c_{T_R}(y)$

   - For other Y-states, choose the same control decisions in T_R





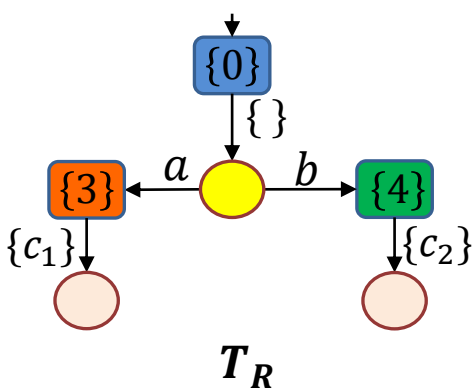
Synthesis Algorithm

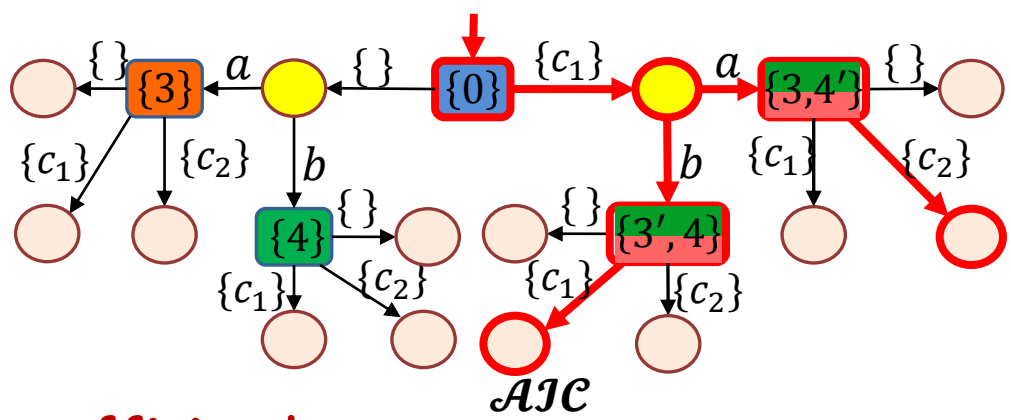
Synthesis Steps:

1. Construct BTS T_R and $\mathcal{AIC}(G)$ (make sure R is a strict sub-automaton of G)
2. Compute the maximal CSR Φ^* from T_R to $\mathcal{AIC}(G)$
3. Find a Y-state y in T_R such that its control decision can be replaced by γ
4. Construct BTS T^* by

   - For Y-state y, choose γ which is larger than $c_{T_R}(y)$

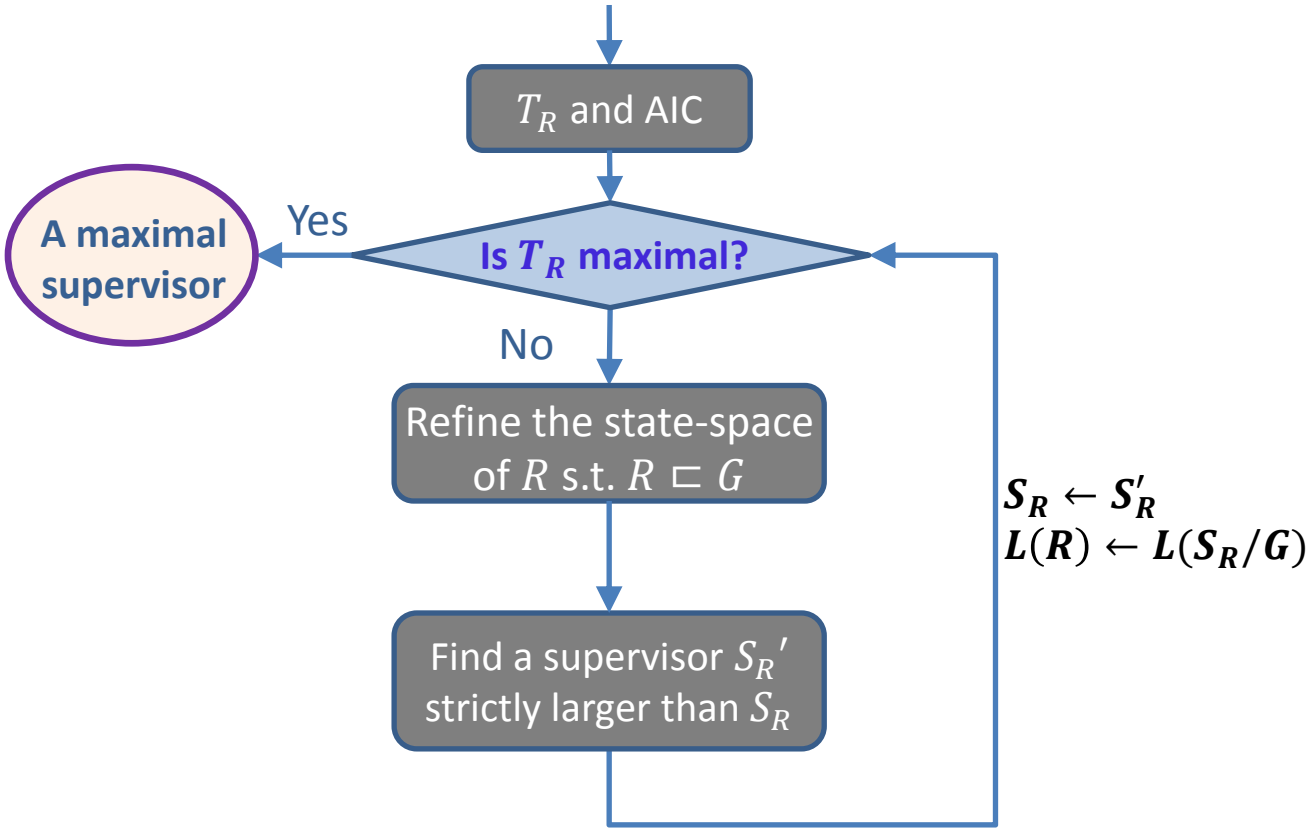
   - For other Y-states, choose the same control decisions in T_R



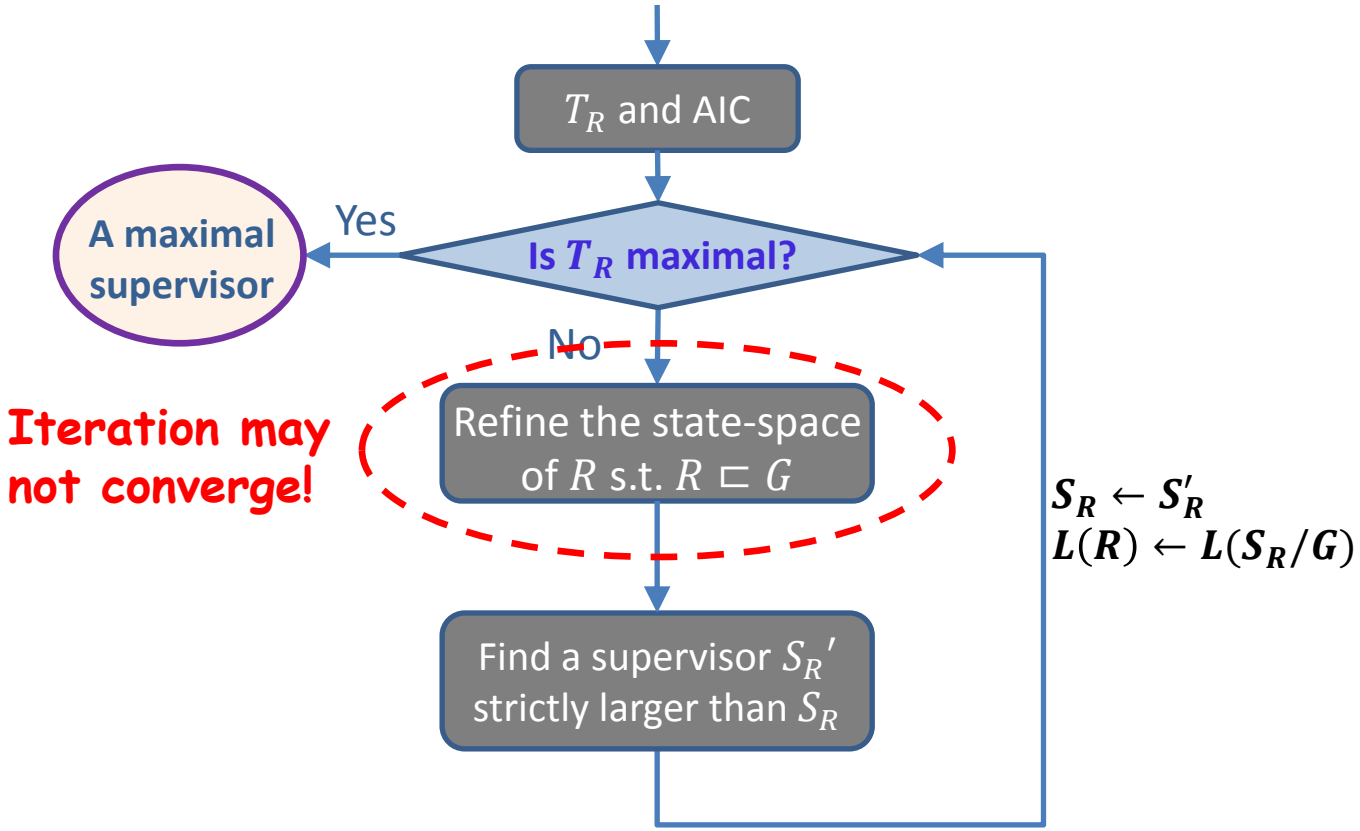


- 2^X is sufficient!

Achieve More Permissiveness



Achieve More Permissiveness



Conclusion

Contribution

- Verify whether or not a given supervisor is maximal
- The notion of control simulation relation
- Synthesis a new supervisor that is strictly more permissive
- Information Merge Phenomenon & Strict sub-automaton

Conclusion

Contribution

- Verify whether or not a given supervisor is maximal
- The notion of control simulation relation
- Synthesis a new supervisor that is strictly more permissive
- Information merge phenomenon & Strict sub-automaton

On Going Work

- Synthesize a maximal supervisor that contains the given one