Automatic Assumption Synthesis from Timed Automata for Compositional Model Checking

FIT 2008

Bernd Finkbeiner Hans-Jörg Peter Sven Schewe

Universität des Saarlandes

April 5th, 2008

Compositional Model Checking

Compositional verification rule

$$\frac{\begin{array}{lccc} (1) & E \| A & \models & P \\ (2) & M & \models & A \end{array}}{\begin{array}{ccc} & E \| M & \models & P \end{array}}$$

Compute an assumption A such that

1. $E \| A \models P$ iff $E \| M \models P$
2. $|A| \ll |M|$

Compositional Model Checking

Compositional verification rule

$$\frac{\begin{array}{lccc} (1) & E \| A & \models & P \\ (2) & M & \models & A \end{array}}{\begin{array}{ccc} E \| M & \models & P \end{array}}$$

Compute an assumption A such that

1. $E \| A \models P$ iff $E \| M \models P$
2. $|A| \ll |M|$

Real-Time Setup

Input

- Timed automaton M representing the model
- Timed automaton E representing the environment
- Monitor automaton P representing a safety property

Output

- Timed automaton A representing the assumption

Talk Outline

Assumptions as equivalences

- Forward equivalence
- Backward equivalence

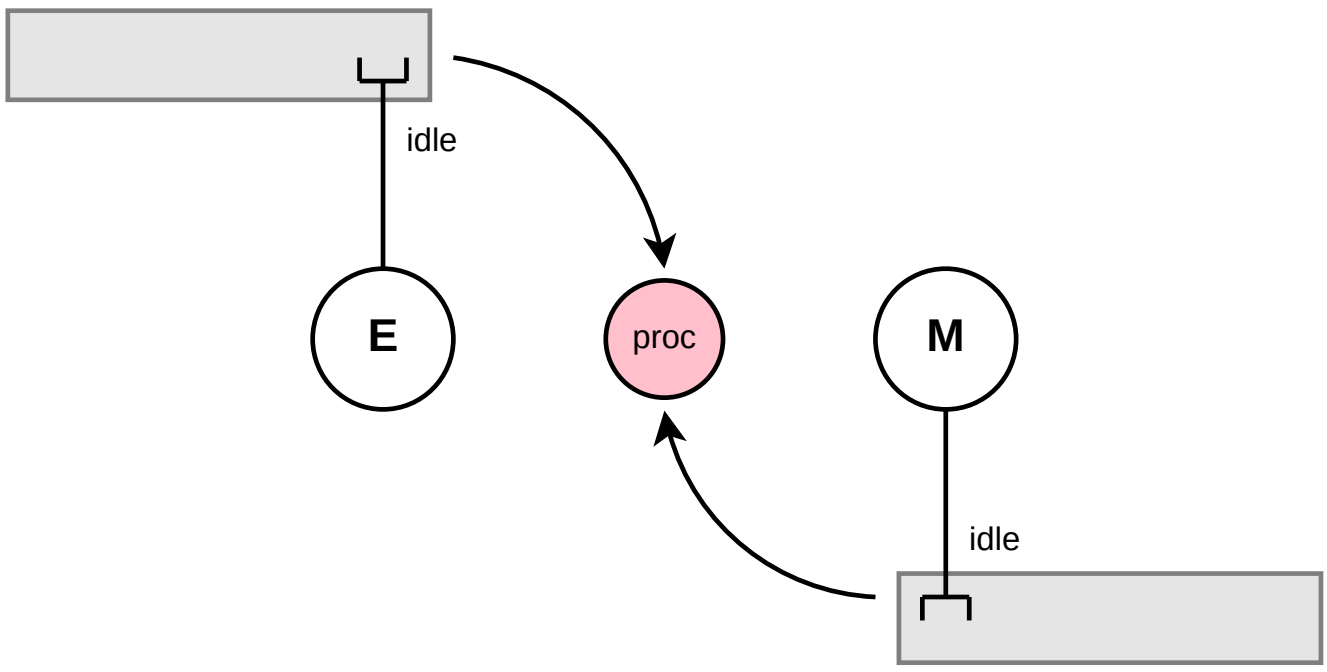
Assumptions from abstractions

- Modal transition systems
- Abstraction refinement

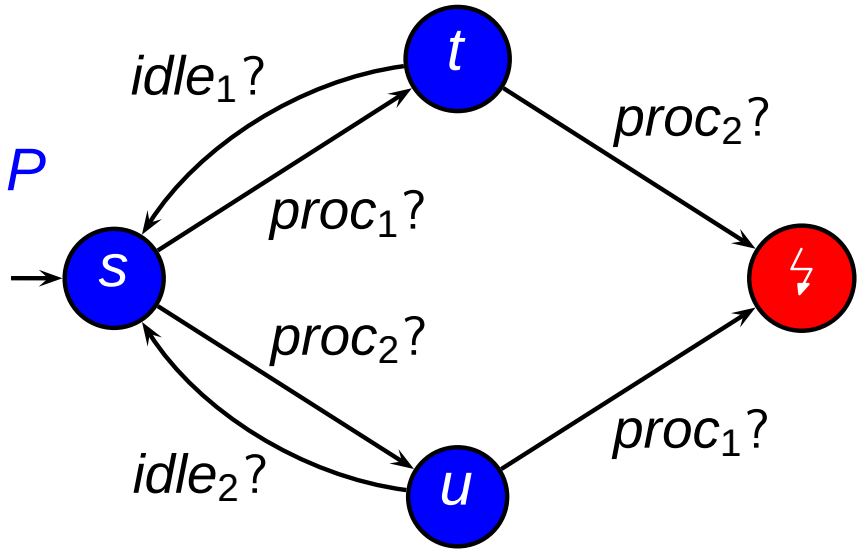
Conclusion

- RESY
- Summary and future work

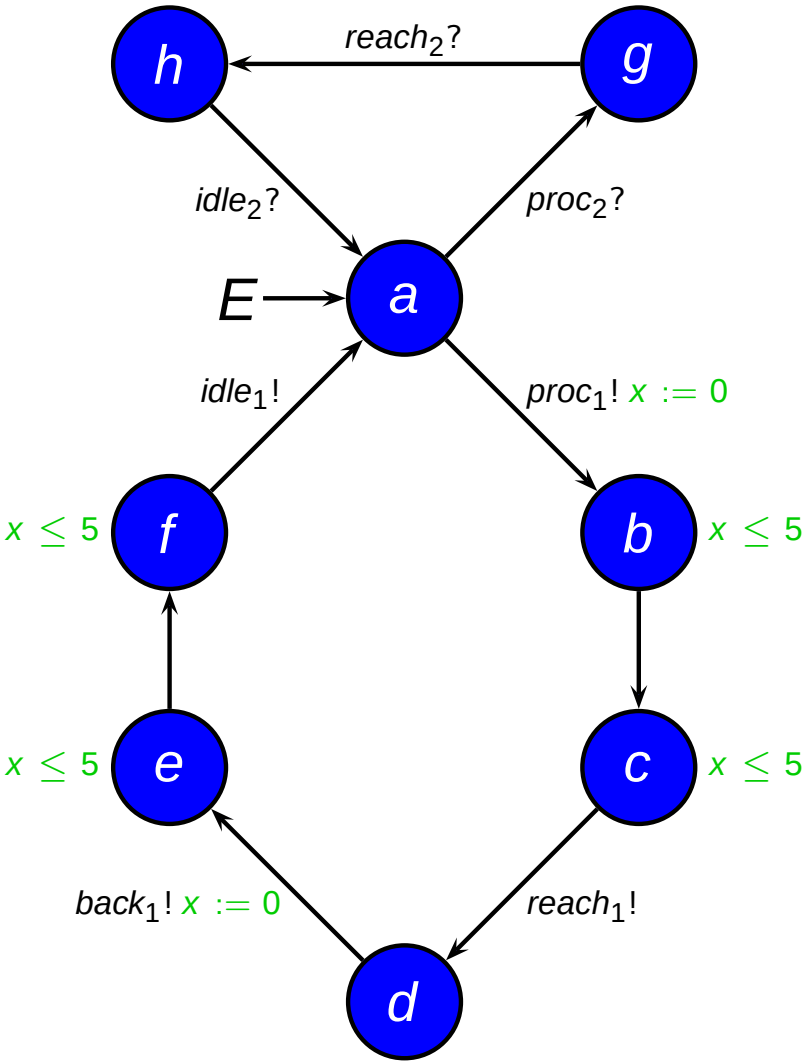
Example: Production Cell

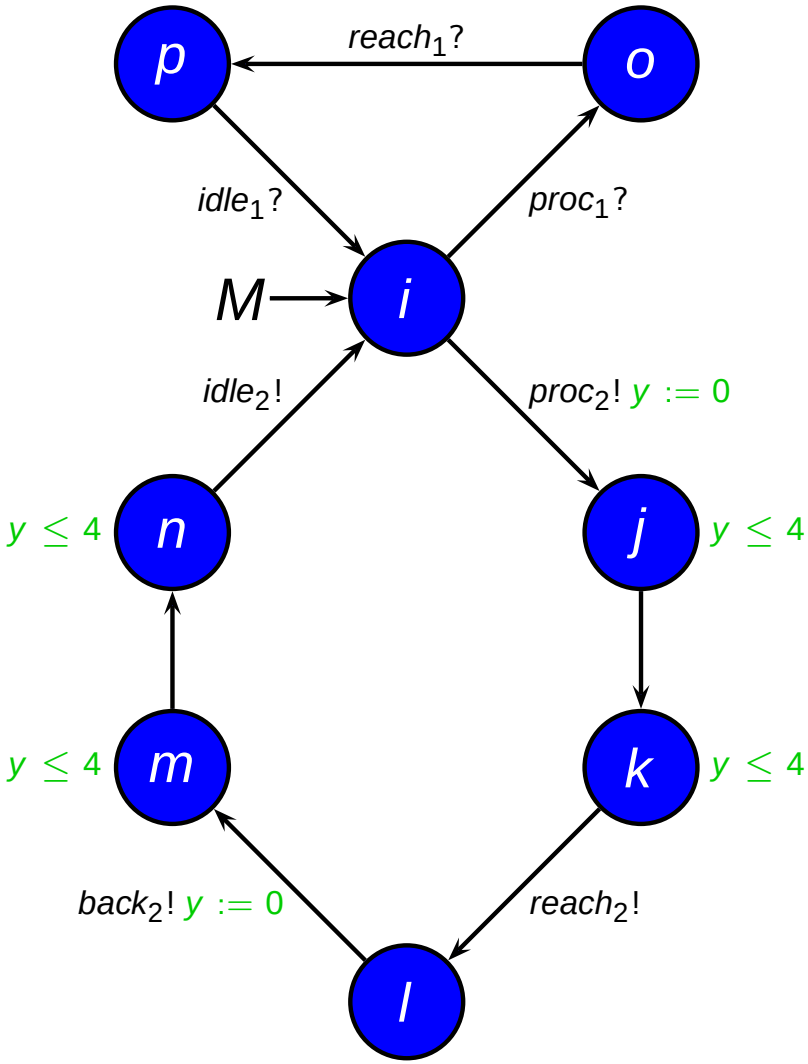


Production Cell: Property

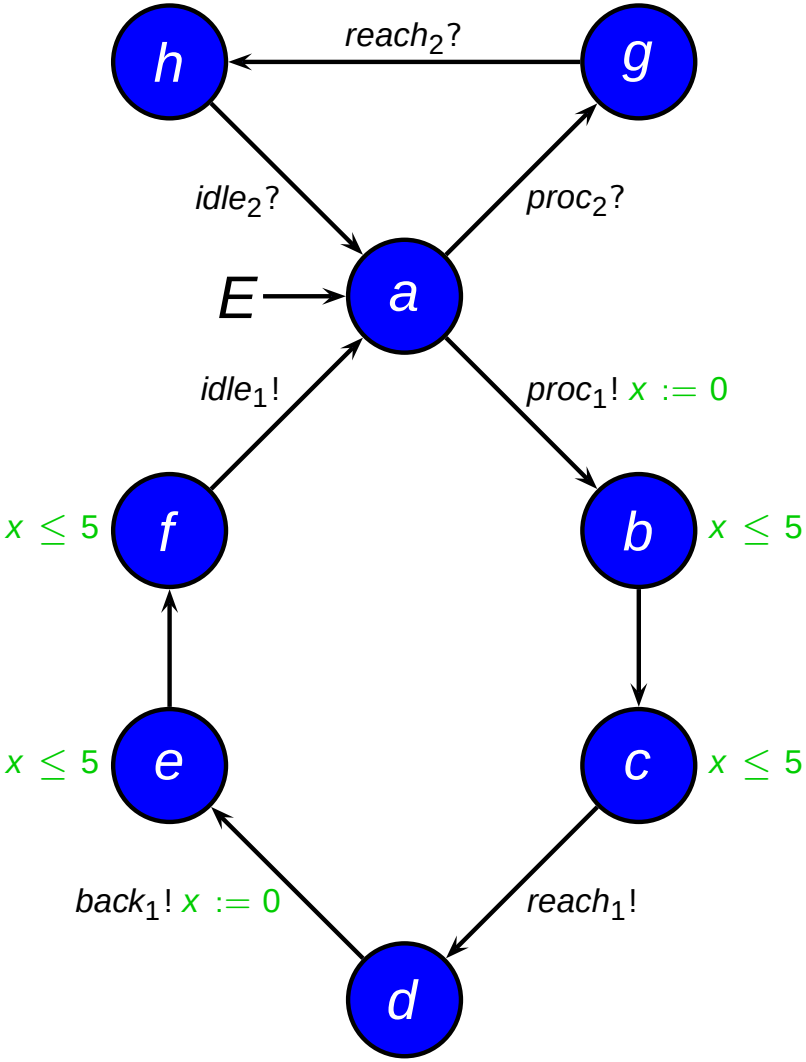


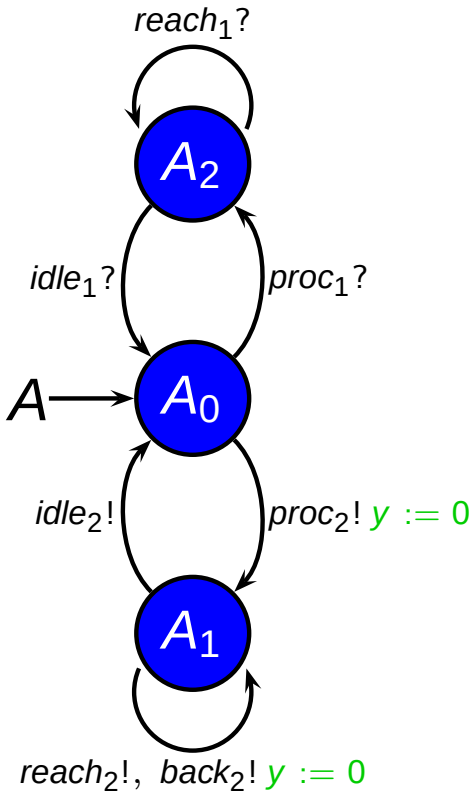
Production Cell: Implementation



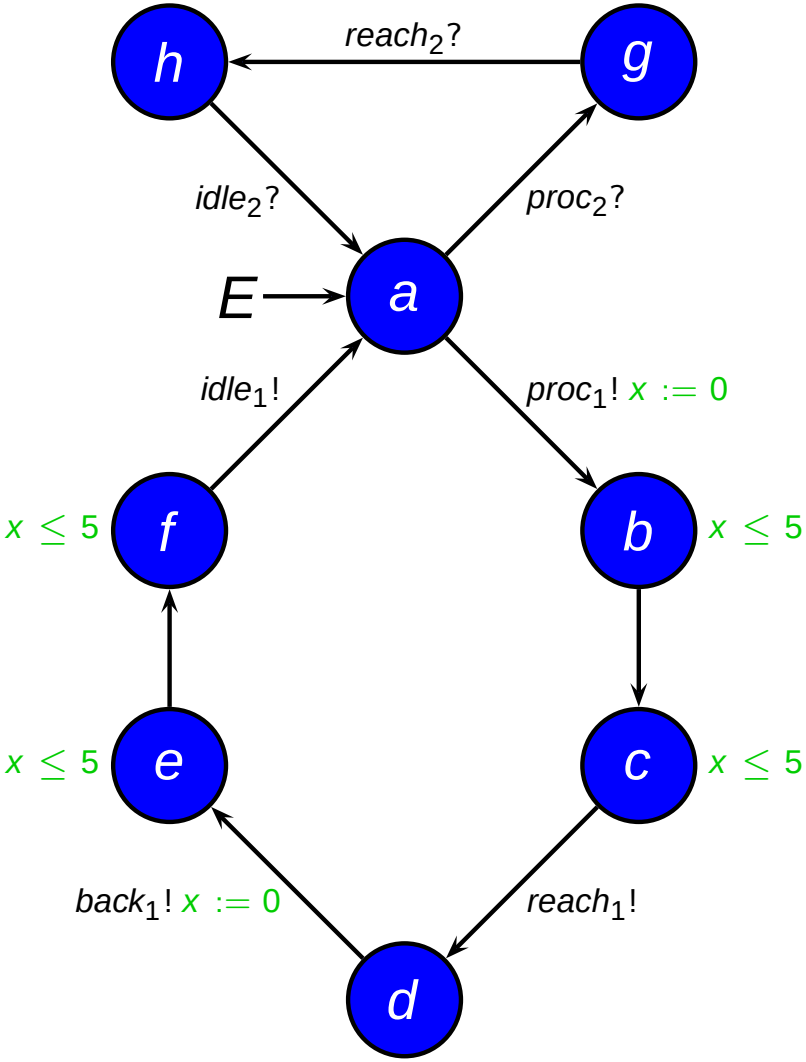


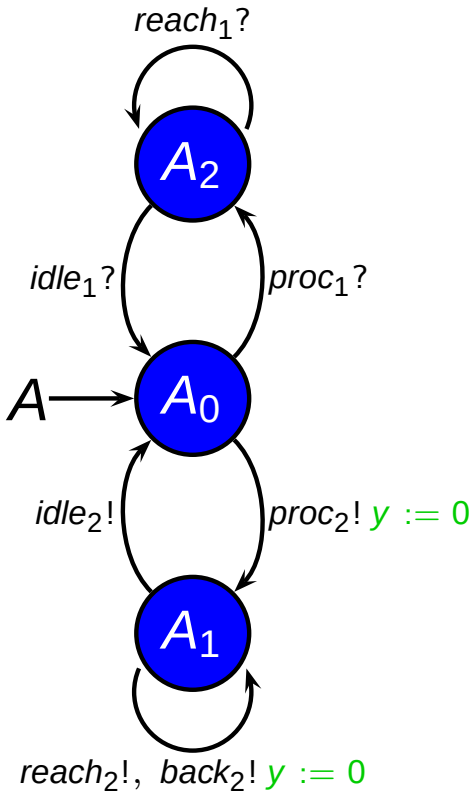
Production Cell: Minimal Assumption





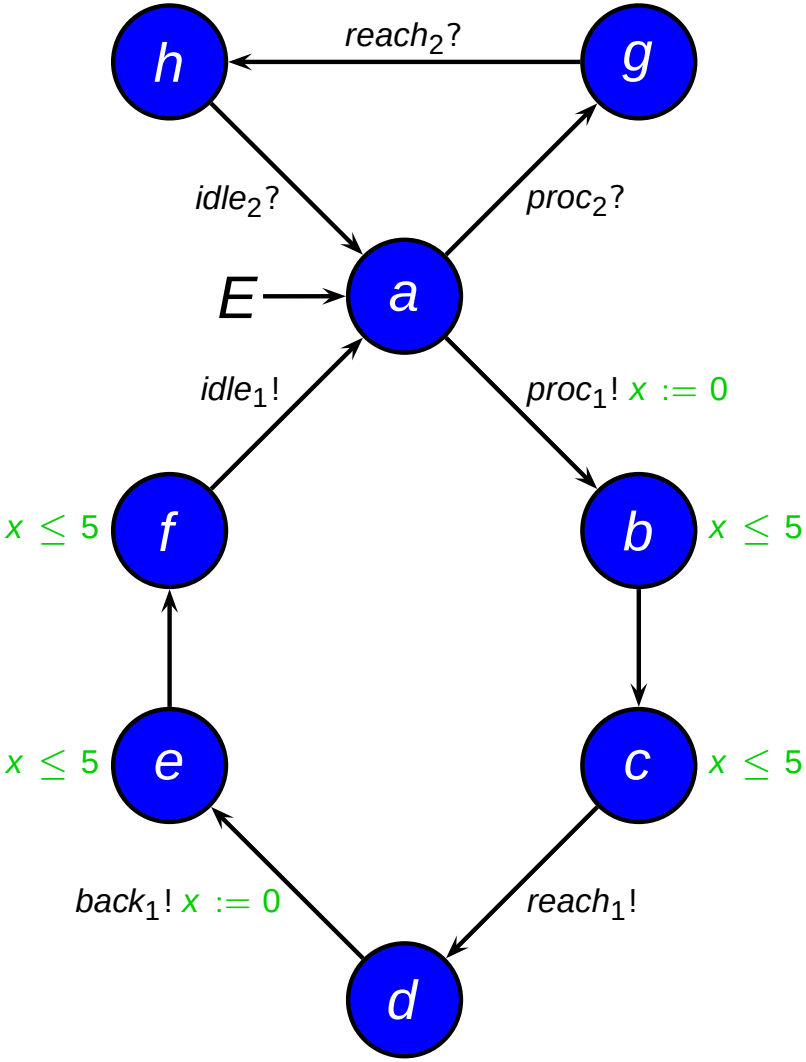
Production Cell: Minimal Assumption

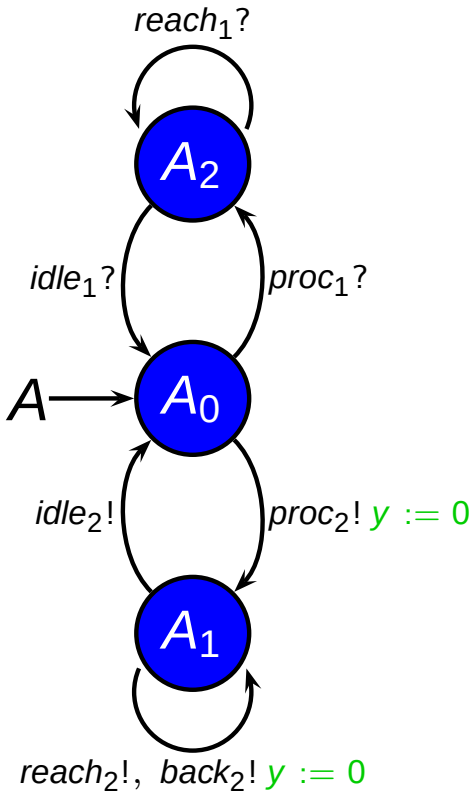




sufficient for revalidation
in the context of E and P

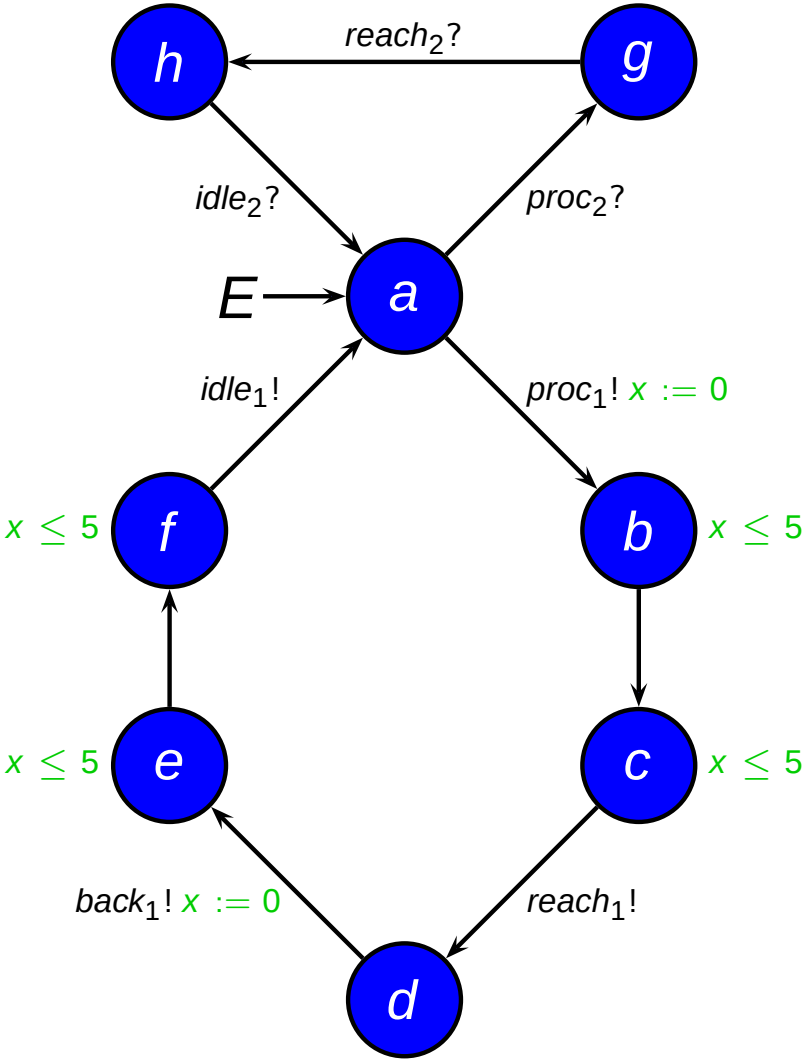
Production Cell: Minimal Assumption

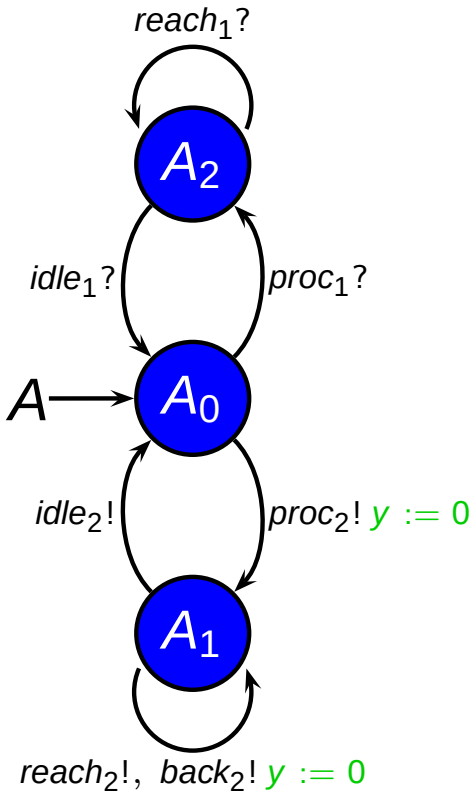




$$E \| A \models P \iff E \| M \models P$$

Production Cell: Minimal Assumption





for any M' with $L(M') \subseteq L(A)$:

$$E||M \models P \implies E||M' \models P$$

Forward Assumption

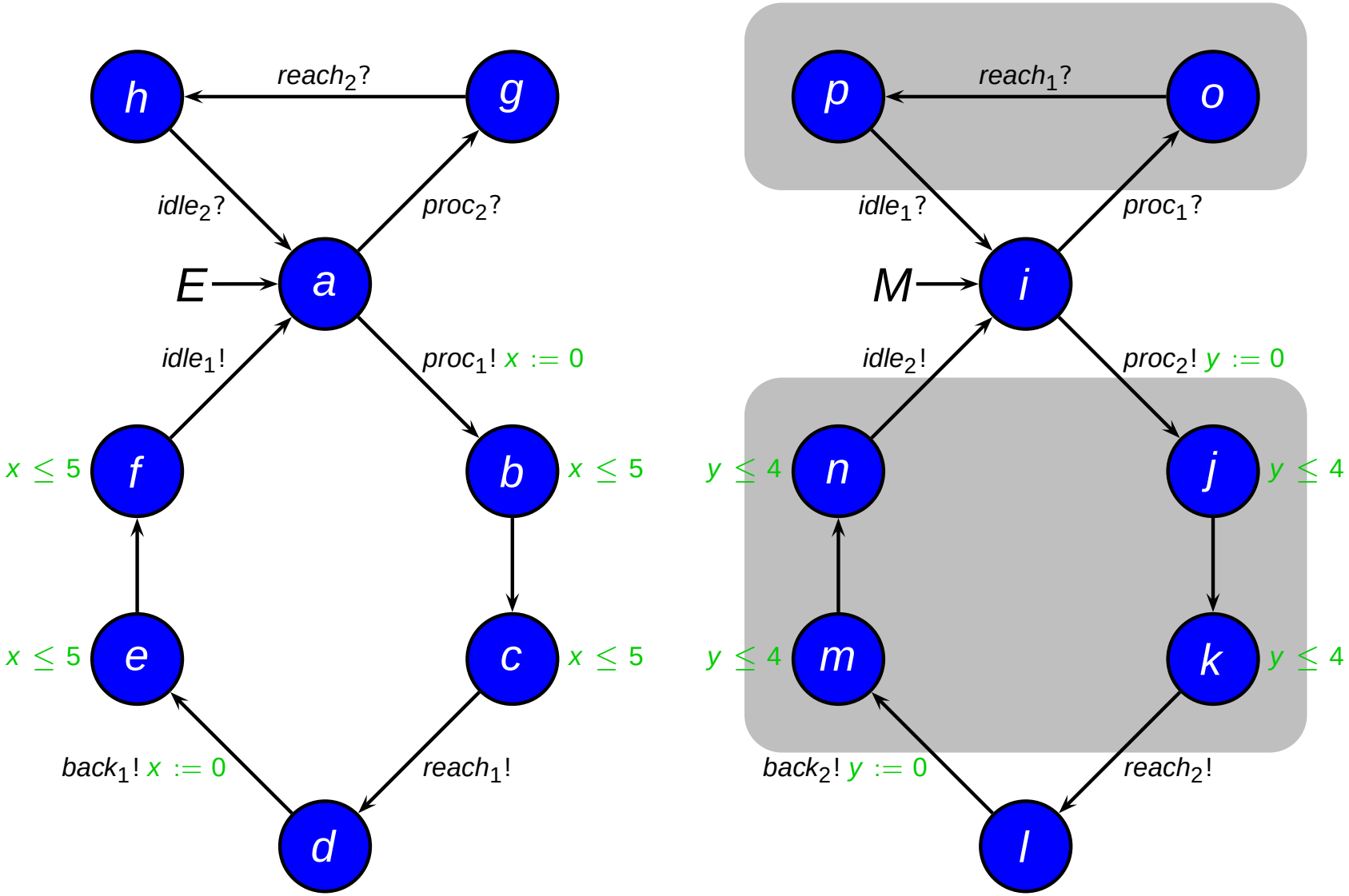
Forward equivalence

- $m_1 \sim_F m_2$ iff for all e of E, p of P, and $\vec{t} \in \mathbb{R}^X_{\geq 0}$,

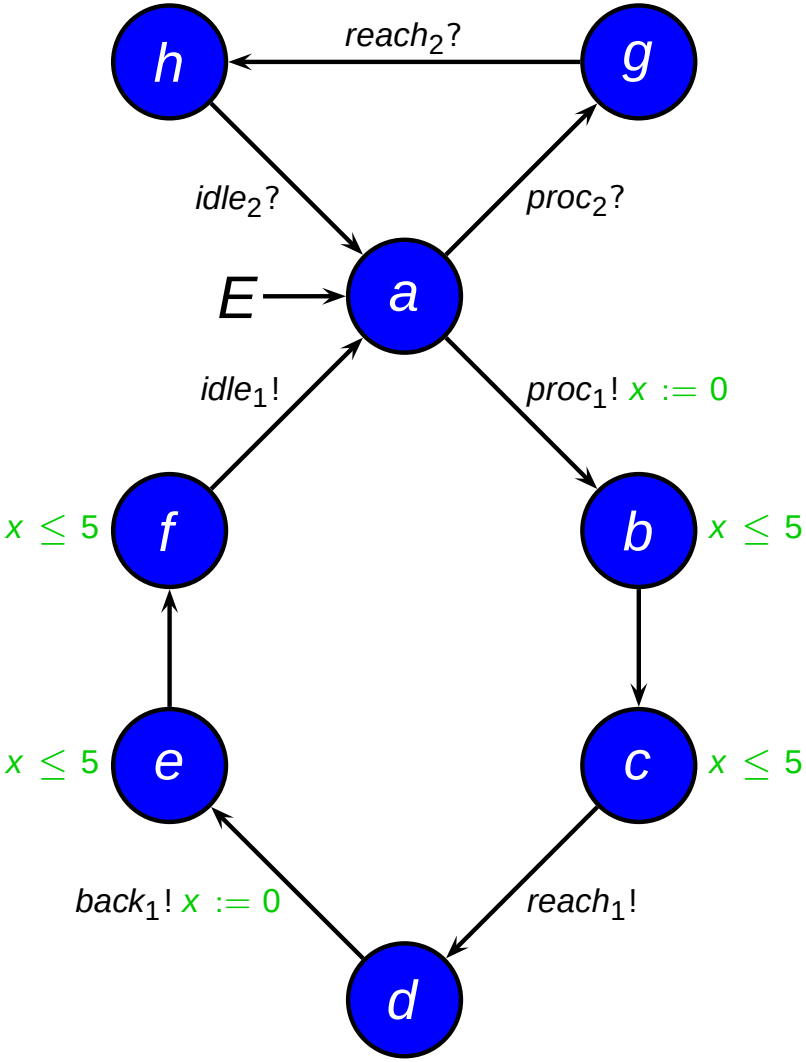
$$(e, m_1, p, \vec{t}) \text{ is reachable from } \mathit{init}$$
$$\Longleftrightarrow$$
$$(e, m_2, p, \vec{t}) \text{ is reachable from } \mathit{init}$$

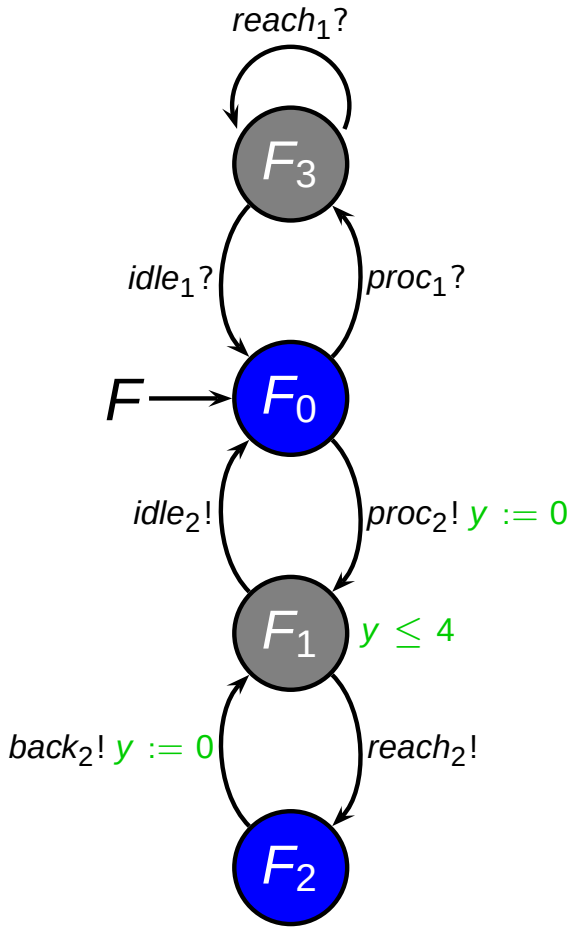
$$\Longrightarrow F = M/\sim_F$$

Production Cell: Forward Assumption



Production Cell: Forward Assumption





Backward Assumption

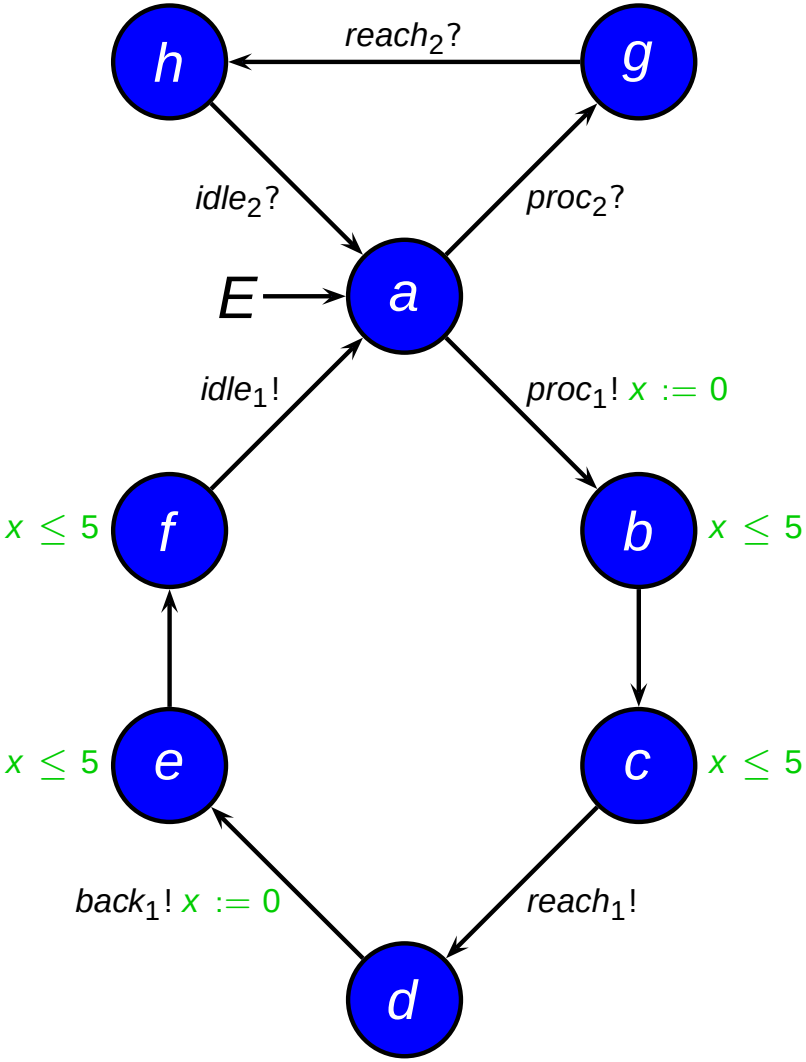
Backward equivalence

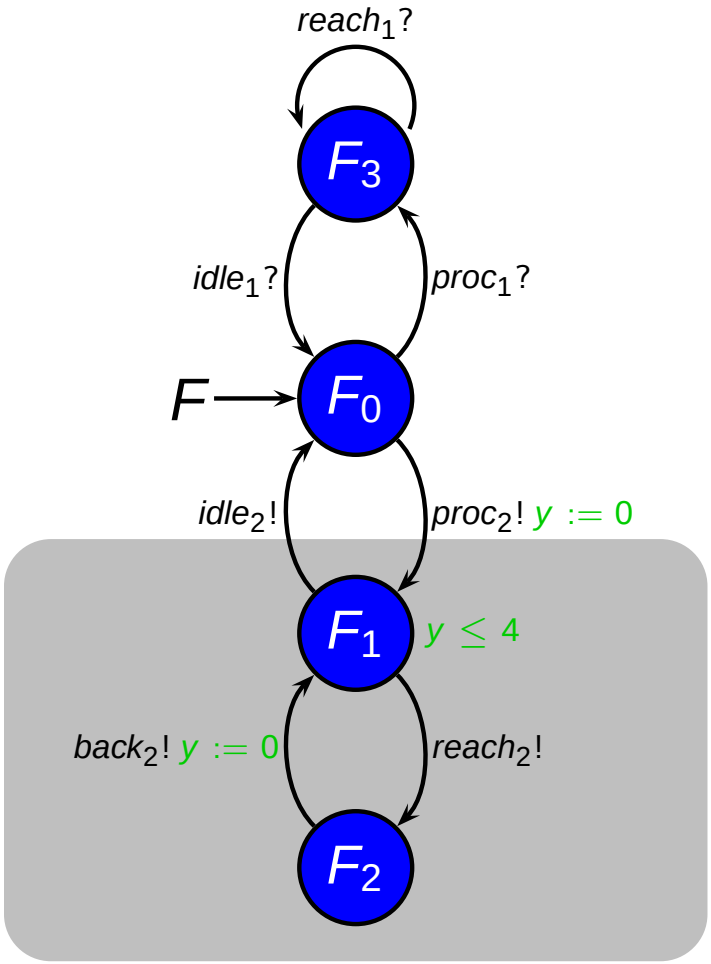
- $m_1 \sim_B m_2$ iff for all e of E, p of P, and $\vec{t} \in \mathbb{R}^X_{\geq 0}$,

$$\textit{error} \text{ is reachable from } (e, m_1, p, \vec{t})$$
$$\iff$$
$$\textit{error} \text{ is reachable from } (e, m_2, p, \vec{t})$$

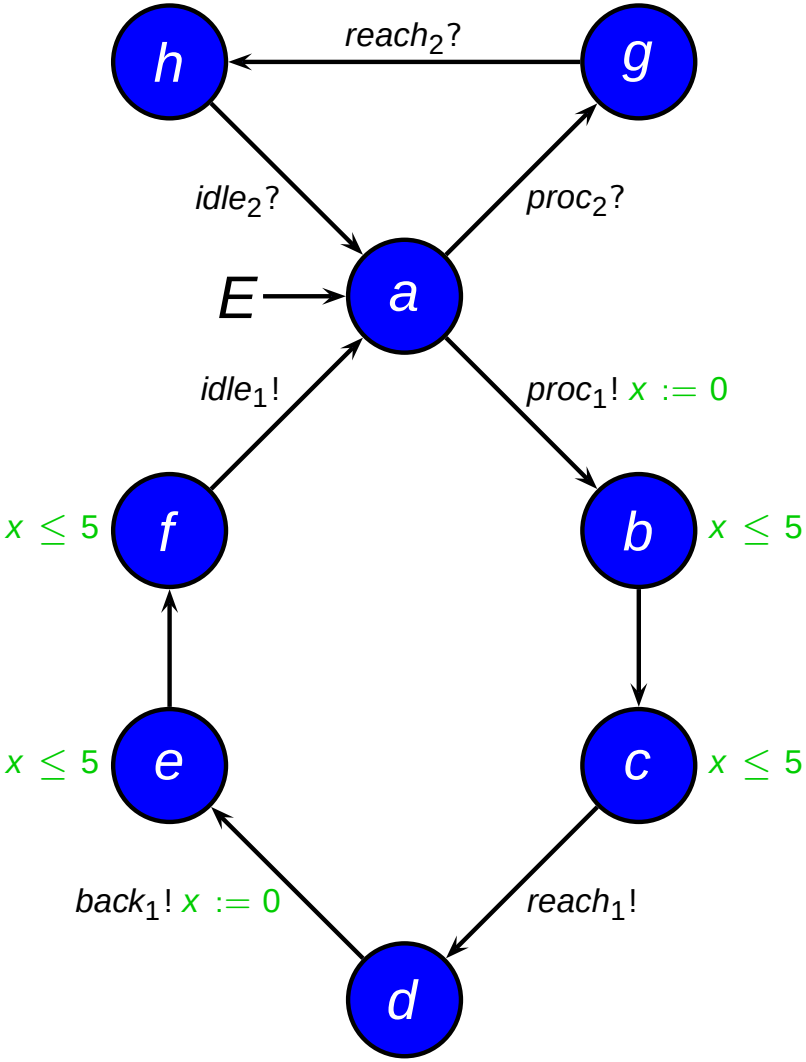
$$\Longrightarrow A = F / \sim_B$$

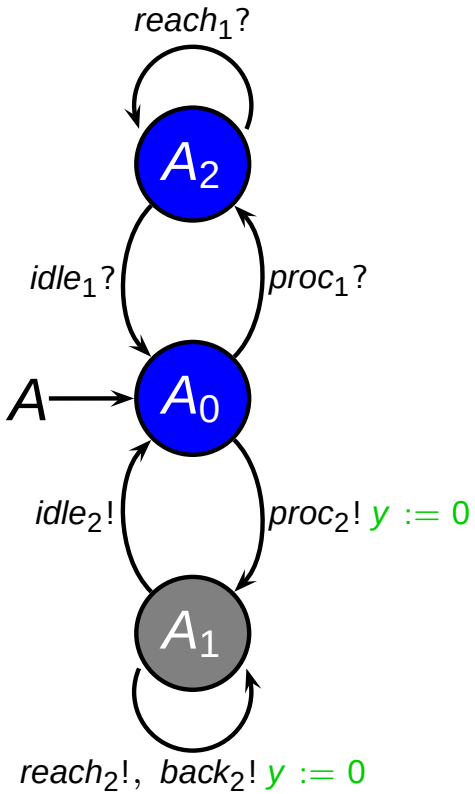
Production Cell: Backward Assumption



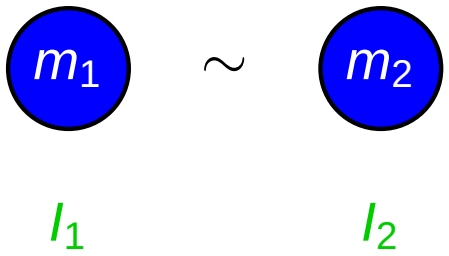


Production Cell: Backward Assumption





Assumption Location Invariants



Location invariants must be convex

- Invariant of $m_1 \sim m_2 = l_1 \cup l_2$
- If $l_1 \cup l_2$ is convex
  $\rightarrow$ no problem
- If $l_1 \cup l_2$ is *not* convex
  $\rightarrow$ partial merge

Assumptions from Abstractions

New setting

- *Modal abstraction* $\mathcal{E}$ of E
- Abstract states subsuming concrete states
- May / must transitions

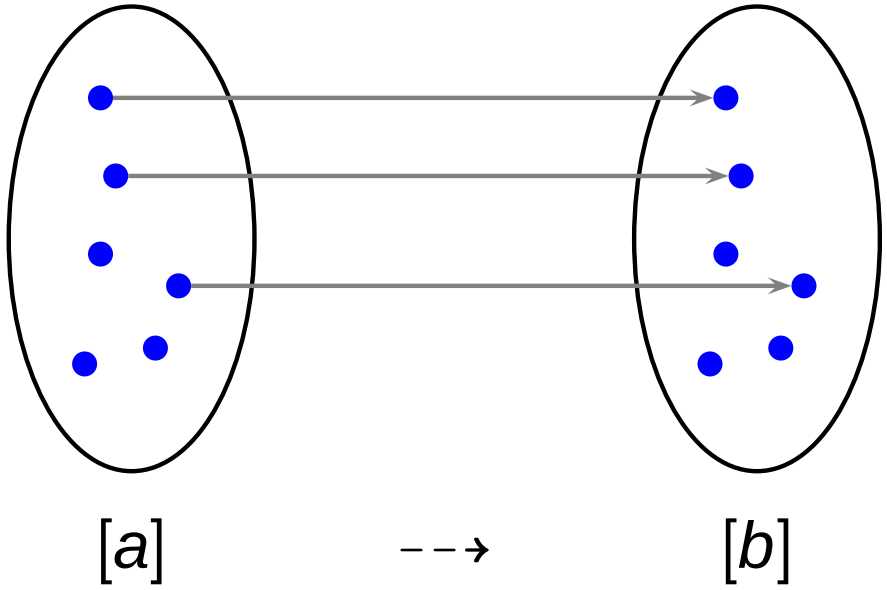
$\longrightarrow$ Modal equivalence relation

Symbolic representation

- Represent $\mathcal{E}$ as a zone graph:

$$S_{\mathcal{E}} \subseteq 2^E \times Z$$

Abstract Transitions

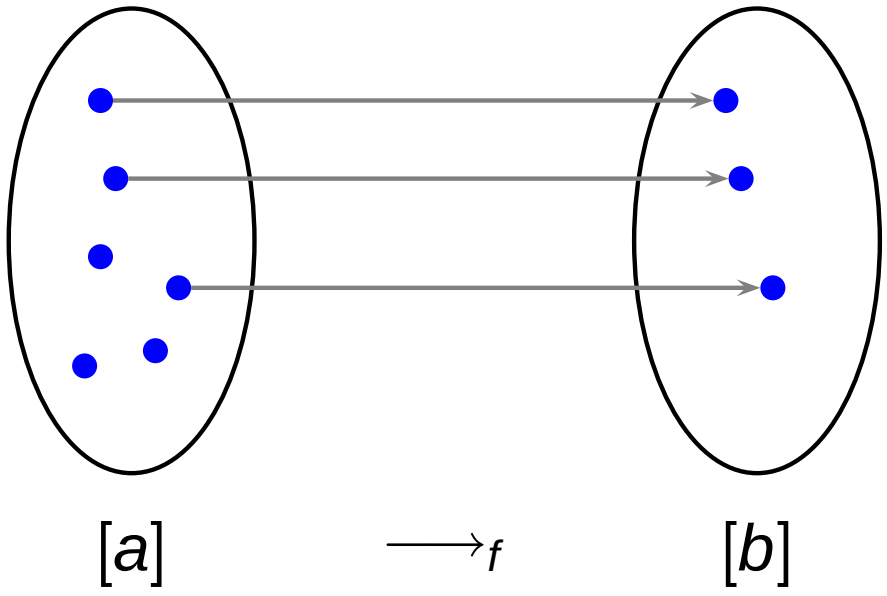


May transitions

- There is a may transition α from [a] to [b] in $\mathcal{E}$ iff

$$[b] \cap Post([a],\ \alpha) \neq \emptyset$$

Forward Must Transitions

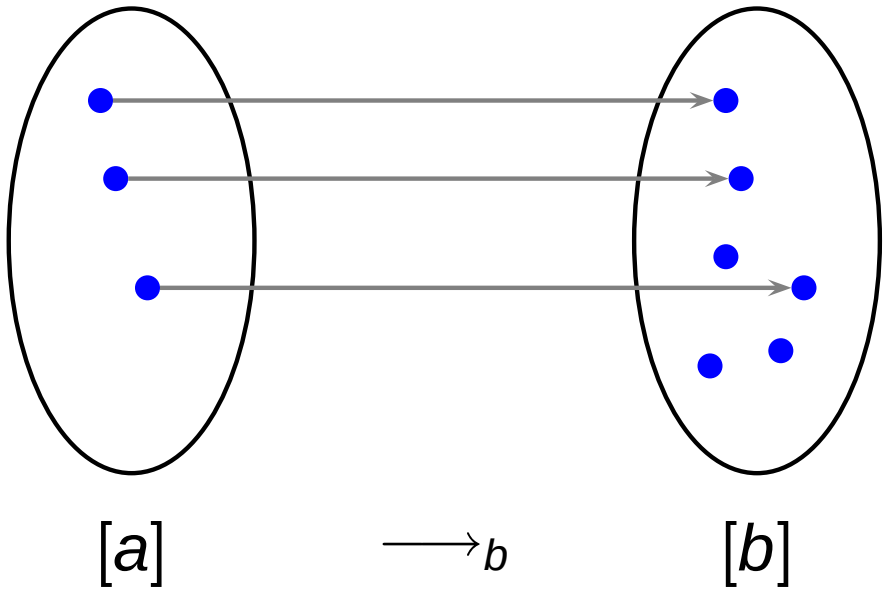


Forward must

- A transition α from $[a]$ to $[b]$ in $\mathcal{E}$ is called *forward must* iff

$$[b] \subseteq \mathit{Post}([a],\ \alpha)$$

Backward Must Transitions



Backward must

- A transition α from $[a]$ to $[b]$ in $\mathcal{E}$ is called *backward must* iff

$$[a] \subseteq Pre([b], \alpha)$$

Recap: Forward Equivalence

Forward equivalence

- Let $(e, p, \vec{t}) \in (E \times P \times \mathbb{R}^X_{\geq 0})$
- $m_1 \sim_F m_2$ iff for all $s_1 = (e, m_1, p, \vec{t})$ and $s_2 = (e, m_2, p, \vec{t})$

$$init \longrightarrow s_1 \iff init \longrightarrow s_2$$

Recap: Forward Equivalence

Forward equivalence

- Let $(e, p, \vec{t}) \in (E \times P \times \mathbb{R}^X_{\geq 0})$
- $m_1 \sim_F m_2$ iff for all $s_1 = (e, m_1, p, \vec{t})$ and $s_2 = (e, m_2, p, \vec{t})$

$$\begin{aligned} & init \longrightarrow s_1 \ \wedge \ init \longrightarrow s_2 \\ \vee \quad & \neg(init \longrightarrow s_1) \ \wedge \ \neg(init \longrightarrow s_2) \end{aligned}$$

Modal Forward Equivalence

Forward equivalence

- Let $(e, p, \vec{t}) \in (\mathcal{E} \times P \times \mathbb{R}^X_{\geq 0})$
- $m_1 \sim_F m_2$ iff for all $s_1 = (e, m_1, p, \vec{t})$ and $s_2 = (e, m_2, p, \vec{t})$

$$\begin{aligned} & init \longrightarrow_f s_1 \ \wedge \ init \longrightarrow_f s_2 \\ \vee \quad & \neg(init \dashrightarrow s_1) \ \wedge \ \neg(init \dashrightarrow s_2) \end{aligned}$$

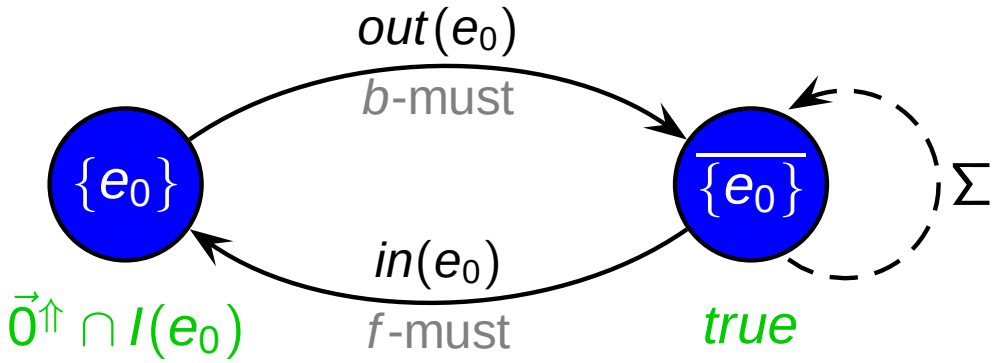
Modal Backward Equivalence

Backward equivalence

- Let $(e, p, \vec{t}) \in (\mathcal{E} \times P \times \mathbb{R}^X_{\geq 0})$
- $m_1 \sim_B m_2$ iff for all $s_1 = (e, m_1, p, \vec{t})$ and $s_2 = (e, m_2, p, \vec{t})$

$$s_1 \longrightarrow_b \mathit{error} \ \wedge \ s_2 \longrightarrow_b \mathit{error}$$
$$\vee \quad \neg(s_1 \dashrightarrow \mathit{error}) \ \wedge \ \neg(s_2 \dashrightarrow \mathit{error})$$

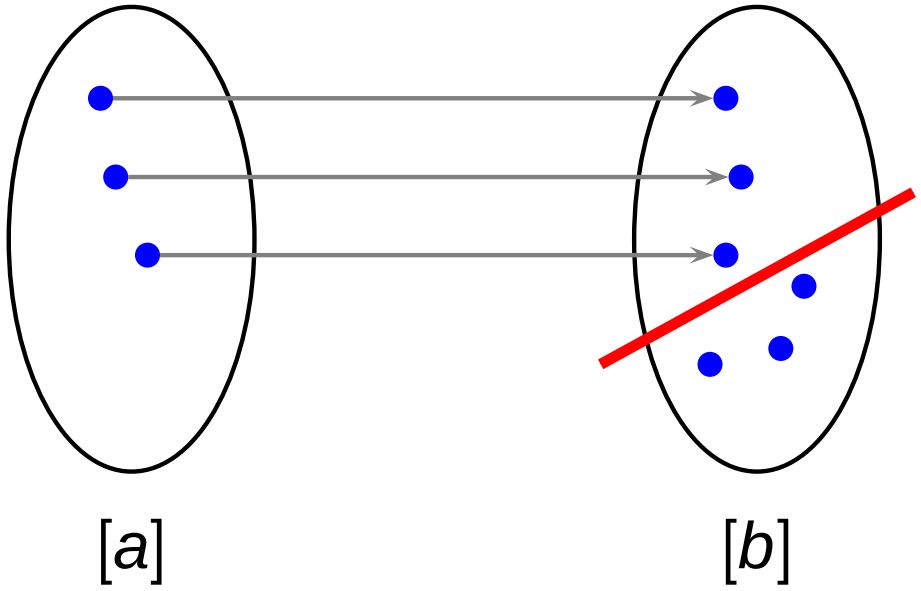
Abstraction Refinement



Refinement algorithm

- Start with an initial abstraction
- As long as refinement is possible...
  - pick may transition t
  - refine t

Abstraction Refinement



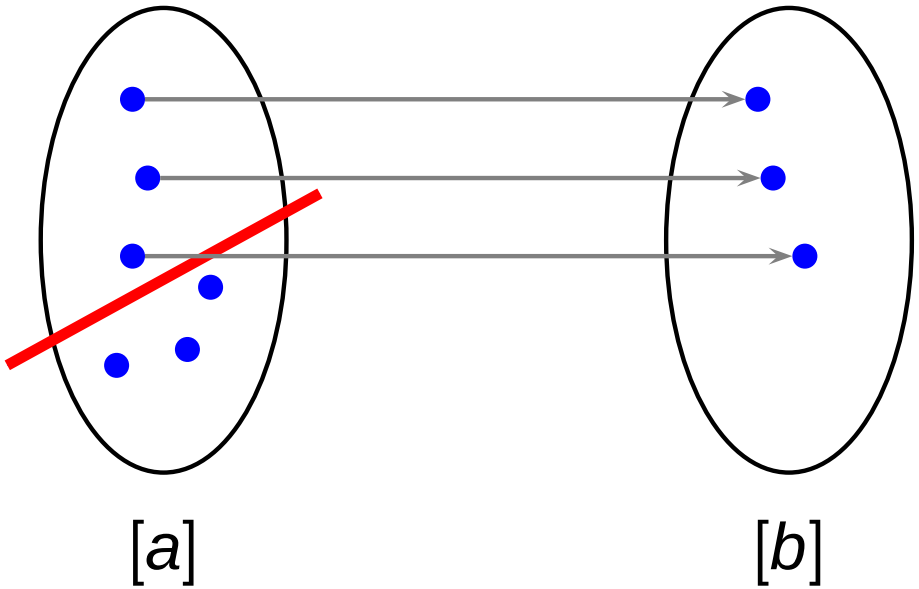
Forward refinement of $\alpha = [a] \rightarrow [b]$

- Split $[b]$ into $[b_1]$ and $[b_2]$ such that

$$[b_1] = [b] \cap Post([a],\ \alpha)$$

$$[b_2] = [b] \setminus Post([a],\ \alpha)$$

Abstraction Refinement



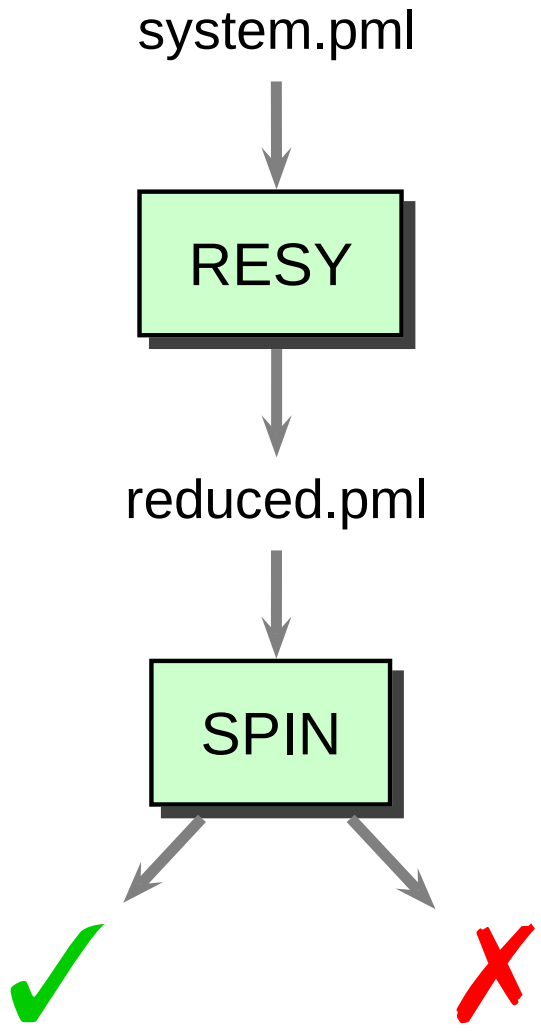
Backward refinement of $\alpha = [a] \rightarrow [b]$

- Split $[a]$ into $[a_1]$ and $[a_2]$ such that

$$[a_1] = [a] \cap Pre([b], \alpha)$$

$$[a_2] = [a] \setminus Pre([b], \alpha)$$

RESY: Requirement Synthesis for CMC



Summary and Future Work

Synthesis of assumption

- Equivalence based approach
- Also applicable for timed systems
- Symbolic modal abstraction
- Automatic abstraction refinement

Future work

- Refinement strategies
- Multiple processes

Summary and Future Work

Synthesis of assumption

- Equivalence based approach
- Also applicable for timed systems
- Symbolic modal abstraction
- Automatic abstraction refinement

Future work

- Refinement strategies
- Multiple processes

Thank you for your attention!