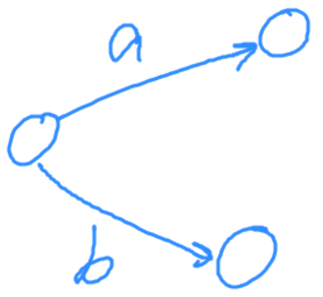
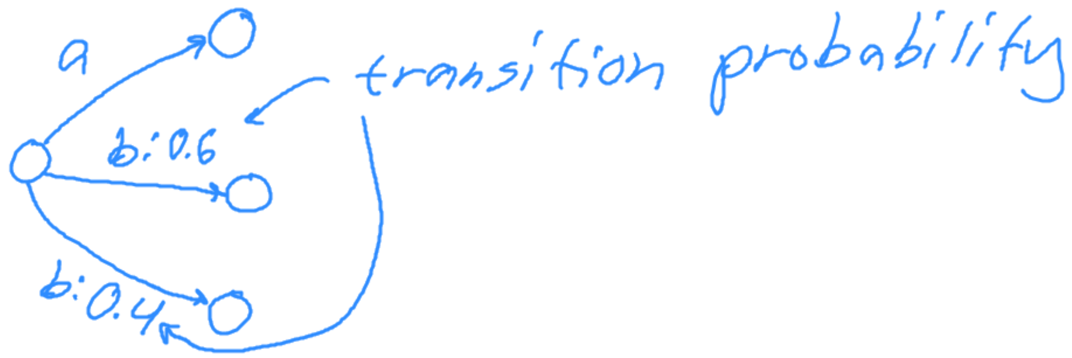
Markov Decision Processes

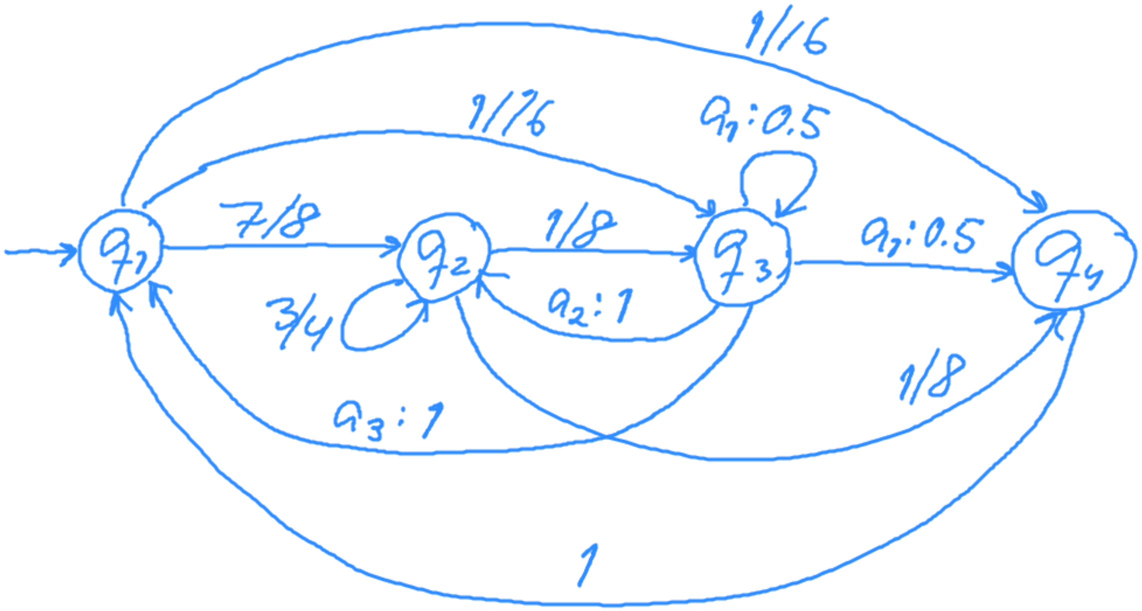
Automata



Markov decision process



Ex Repair or replace a machine



q_1 = Good as new

q_2 = Operable – minor deterioation $\Rightarrow$ minor loss

q_3 = Operable – major deterioation $\Rightarrow$ major loss

q_4 = Inoperable $\Rightarrow$ replace machine

a_1 = continue to use existing machine

a_2 = repair machine

a_3 = replace machine

$Q = \{q_1, q_2, q_3, q_4\}$

A = set of actions $= \{a_1, a_2, a_3\}$

$$\mathbb{P} = \begin{bmatrix} 0 & 7/8 & 1/16 & 1/16 \\ 0 & 3/4 & 1/8 & 1/8 \\ p_{31} & p_{32} & p_{33} & p_{34} \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$a_1 \Rightarrow \quad p_{31} = p_{32} = 0 \quad p_{33} = 0.5 \quad p_{34} = 0.5$

$a_2 \Rightarrow \quad p_{32} = 1, \quad p_{31} = p_{33} = p_{34} = 0$

$a_3 \Rightarrow \quad p_{31} = 1, \quad p_{32} = p_{33} = p_{34} = 0$

state probabilities are now given by

$$\bar{p} = \bar{p}\,\mathbb{P}$$

Evaluate this equation for $a = 2 \Rightarrow$

$$\mathbb{P} = \begin{bmatrix} 0 & 7/8 & 1/16 & 1/16 \\ 0 & 3/4 & 1/8 & 1/8 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$\underbrace{[p_1 \;\; p_2 \;\; p_3 \;\; p_4]}_{\bar{p}} = [p_1 \;\; p_2 \;\; p_3 \;\; p_4] \cdot \begin{bmatrix} 0 & 7/8 & 1/16 & 1/16 \\ 0 & 3/4 & 1/8 & 1/8 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} =$$

$$= \Big(\underbrace{p_4}_{p_1} \quad \underbrace{\tfrac{7}{8}p_1 + \tfrac{3}{4}p_2 + p_3}_{p_2} \quad \underbrace{\tfrac{p_1}{16} + \tfrac{p_2}{8}}_{p_3} \quad \underbrace{\tfrac{p_1}{16} + \tfrac{p_2}{8}}_{p_4}\Big]$$

$p_1 = p_4 \overset{def}{=} p$

$p_3 = p_4 = p$

$p_1 + p_2 + p_3 + p_4 = 1$

$p_2 = 1 - p_1 - p_3 - p_4 =$

$1 - 3p$

$$\frac{7}{8}p + \frac{3}{4}(1-3p) + p = p_2 = 1-3p$$

$$\frac{7}{8}p - \frac{9}{4}p + p + 3p = 1 - \frac{3}{4} = \frac{1}{4}$$

$$(7 - 18 + 8 + 24)p = \frac{8}{4} = 2$$

$$p = \frac{2}{21} = p_1 = p_3 = p_4$$

$$p_2 = 1 - 3p = \frac{21-6}{21} = \frac{15}{21}$$

$p(a_2)$ = state probability for action 2 $= [p_1 \; p_2 \; p_3 \; p_4] = \frac{1}{21}[2 \; 15 \; 2 \; 2]$

(*) values decided by the user.

In the same way

$$p(a_1) = \frac{1}{13}(2 \; 7 \; 2 \; 2)$$

$$p(a_3) = \frac{1}{11}(2 \; 7 \; 1 \; 1)$$

Cost

$$c(a_1) = (0 \; 1 \; 3 \; 6)$$

$$c(a_2) = (0 \; 1 \; 4 \; 6)$$

$$c(a_3) = (0 \; 1 \; 6 \; 6)$$

1 $\leftrightarrow$ minor loss (*)

3 $\leftrightarrow$ major loss

4 $\leftrightarrow$ repair cost

6 $\leftrightarrow$ replacement cost

cost function

$$J(a_1) = p(a_1)\,c^T(a_1) = \frac{7\cdot 1 + 2\cdot 3 + 2\cdot 6}{13} = \frac{25}{13} \approx 1.92$$

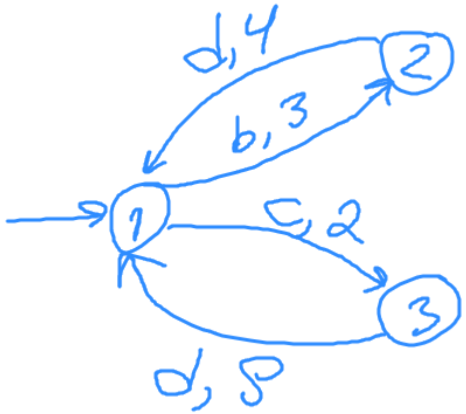
$$J(a_2) = p(a_2)\,c^T(a_2) = \frac{15\cdot 1 + 2\cdot 4 + 2\cdot 6}{21} = \frac{35}{21} \approx 1.67$$

$$J(a_3) = p(a_3)\,c^T(a_3) = \frac{7\cdot 1 + 1\cdot 6 + 1\cdot 6}{11} = \frac{19}{11} \approx 1.73$$

$$\min_{a \in \{a_1, a_2, a_3\}} J(a) = J(a_2) = 1.67$$

$\therefore$ Policy a_2 gives the lowest cost including minor loss in q_2, repair in state q_3, and replacement in state q_4.

Reinforcement learning example



For which ρ is the action c better than action b in state 1?

$$J^*(x) = \max_{a \in \Sigma(x)} \{ \rho(x,a) + \gamma J^*(\delta(x,a)) \}$$

$$J^*(1) = \max \{ 3 + \gamma J^*(2), 2 + \gamma J^*(3) \}$$

$$J^*(2) = 4 + \gamma J^*(1)$$

$$J^*(3) = \rho + \gamma J^*(1)$$

$$J^*(1) = \max \{ \underbrace{3 + \gamma(4 + \gamma J^*(1))}_{b}, \underbrace{2 + \gamma(\rho + \gamma J^*(1))}_{c} \} =$$

$$= \max \{ 3 + 4\gamma + \gamma^2 J^*(1), 2 + \rho\gamma + \gamma^2 J^*(1) \}$$

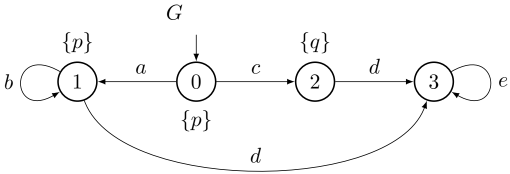
$$Q(1,c) > Q(1,b)$$

$$2 + \rho\gamma > 3 + 4\gamma$$

$$\rho > \frac{1 + 4\gamma}{\gamma} \qquad \gamma = 0.9$$

$\rho > 5.11$ to take action c.

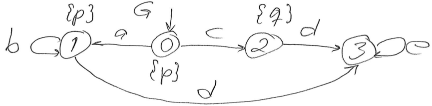
Extra μ-calculus task



Show by μ-calculus that $[\![\exists\Diamond q]\!] = \{0, 2\}$, $[\![\exists\Box p]\!] = \{0, 1\}$, $\exists p\mathcal{U}q = \{0, 2\}$ for the transition system G.

Extra μ-calculus task: Solution

6.



$[\![\exists\Diamond q]\!] = \mu Y.\ [\![q]\!] \cup Pre^{\exists}(Y)$

$Y_0 = \emptyset,\quad Y_1 = [\![q]\!] = \{2\},\quad Y_2 = \{2\} \cup Pre^{\exists}(\{2\})$

$= \{2\} \cup \{0\} = \{0,2\},\quad Y_3 = Y_2 = Y_\infty$

$[\![\exists\Diamond q]\!] = Y_\infty = \{0,2\}$

$(*)\ Y_3 = Y_2 = Y_\infty$

$[\![\exists p\,\mathcal{U}\,q]\!] = \{0,2\}$

$[\![\exists\Box p]\!] = \nu Z.\ [\![p]\!] \cap Pre^{\exists}(Z)$

$Z_0 = \{0,1,2,3\},\quad Z_1 = [\![p]\!] \cap Pre^{\exists}(Z) =$

$= \{0,1\} \cap \{0,1,2,3\} = \{0,1\},\quad Z_2 = \{0,1\} \cap \{0,1\}$

$= Z_1 = Z_\infty \qquad [\![\exists\Box p]\!] = Z_\infty = \{0,1\}$

$[\![\exists p\,\mathcal{U}\,q]\!] = \mu Y.\ [\![q]\!] \cup ([\![p]\!] \cap Pre^{\exists}(Y)) \qquad Y_0 = \emptyset$

$Y_1 = \{2\} \cup (\{0,1\} \cap \emptyset) = \{2\} \qquad Y_2 = \{2\} \cup (\{0,1\} \cap \{0\}) = \{0,2\}\ (*)$

DES Course Summary

- Discrete Math: Prove logical statements by equivalences, implications, and contradictions. Applied to propositional logic, predicate logic and set expressions.
- Automata, formal languages and Petri nets: Transformations between the different models including the important synchronization operator.
- Specification of DESs: Ch 4 shows typical examples of modeling features for DESs.
- Implementation: See the end of Ch 1.

DES Course Summary

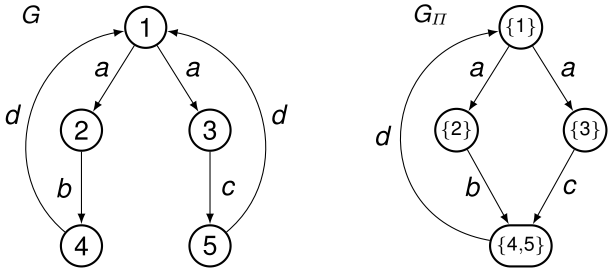
- Verification and synthesis: Nonblocking and controllability.
- Automata and Petri nets with shared variables, timed and hybrid automata.
- Temporal logic verified by μ-calculus.
- Markov processes, queuing theory, Markov decision processes.
- Reinforcement learning: Dynamic programming and Q-learning.

DES Lecture 15
Bisimulation & Model Reduction

Bengt Lennartson

Division of Systems and Control
Chalmers University of Technology

Reduction by state partitioning



State partition: $\Pi = \{\{1\}, \{2\}, \{3\}, \{4, 5\}\}$

Block transitions: $T_\Pi(1) = \{\{1\} \xrightarrow{a} \{2\}, \{1\} \xrightarrow{a} \{3\}\},$

$$T_\Pi(2) = \{\{2\} \xrightarrow{b} \{4, 5\}\},\ T_\Pi(3) = \{\{3\} \xrightarrow{b} \{4, 5\}\},$$

$$T_\Pi(4) = T_\Pi(5) = \{\{4, 5\} \xrightarrow{b} \{1\}\}$$

Bisimulation Partition

Given a transition system $G = \langle X, \Sigma, T, I, AP, \lambda \rangle$, a partition Π is a *bisimulation partition* if, for all $x \in X$,

$$\Pi(x) = \{ y \in X \mid \Pi \preceq \Pi_\lambda \wedge T_\Pi(x) = T_\Pi(y) \},$$

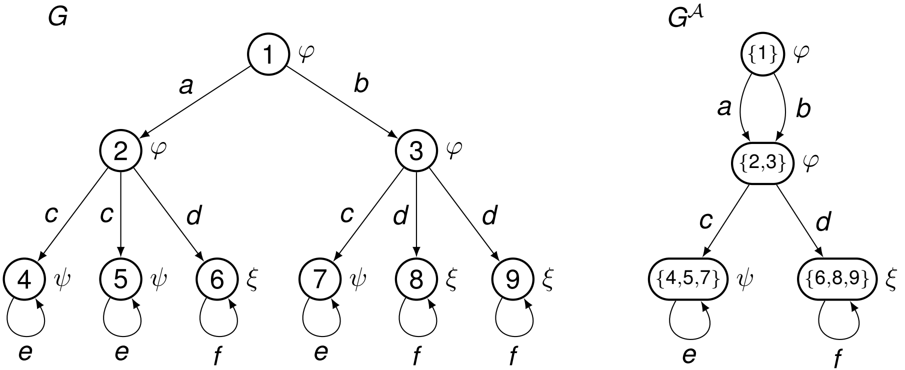
where $\Pi_\lambda(x) = \{ y \in X \mid \lambda(x) = \lambda(y) \}$ is the state label partition, and the set of block transitions from state x

$$T_\Pi(x) = \{ \Pi(x) \overset{a}{\to} \Pi(x') \mid x \overset{a}{\to} x' \}.$$

Reduced automaton

$G^{\mathcal{A}}$ = abstracted automaton based on state blocks and block transitions

Bisimulation Reduction: Example



Standard bisimulation is restricted to either state label models Kripke structures or transition label models (events or actions). Here both state and event labels are included.

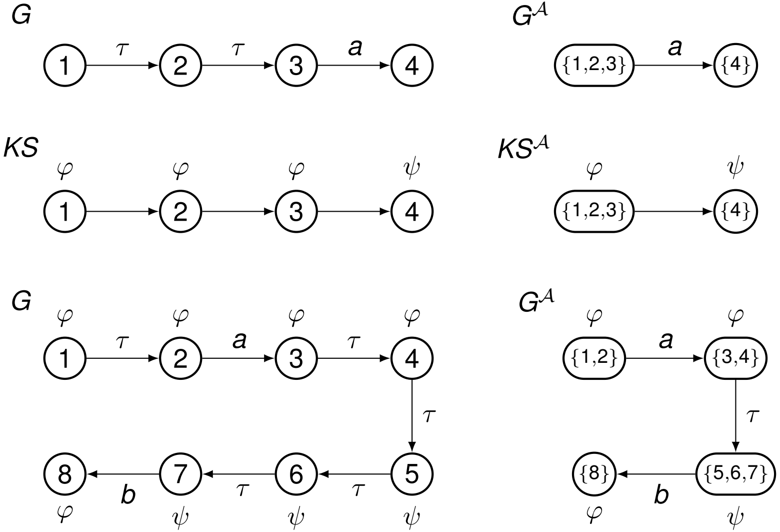
Bisimulation Reduction: Example

The state label partition $\Pi_\lambda = \{\{1,2,3\},\{4,5,7\},\{6,8,9\}\}$ and the set of block transitions

$$\begin{aligned}
T_\Pi(1) &= \{\{1\} \xrightarrow{a} \{2,3\}, \{1\} \xrightarrow{b} \{2,3\}\},\\
T_\Pi(2) = T_\Pi(3) &= \{\{2,3\} \xrightarrow{c} \{4,5,7\}, \{2,3\} \xrightarrow{d} \{6,8,9\}\},\\
T_\Pi(4) = T_\Pi(5) = T_\Pi(7) &= \{\{4,5,7\} \xrightarrow{e} \{4,5,7\}\},\\
T_\Pi(6) = T_\Pi(8) = T_\Pi(9) &= \{\{6,8,9\} \xrightarrow{f} \{6,8,9\}\}.
\end{aligned}$$

generate the partition $\Pi = \{\{1\},\{2,3\},\{4,5,7\},\{6,8,9\}\}$.

Branching Bisimulation: Example 1



Branching Bisimulation

A path

$$x \xrightarrow{\tau} x_1 \xrightarrow{\tau} \cdots \xrightarrow{\tau} x_n \xrightarrow{a} x',$$

is a *stuttering-visible transition*, denoted

$$x \underset{\Pi}{\overset{a}{\twoheadrightarrow}} x'$$

if $\Pi(x) = \Pi(x_1) = \ldots = \Pi(x_n)$, and $a \neq \tau$ or $\Pi(x_n) \neq \Pi(x')$, meaning that the first n transitions are invisible, while the last one is visible.

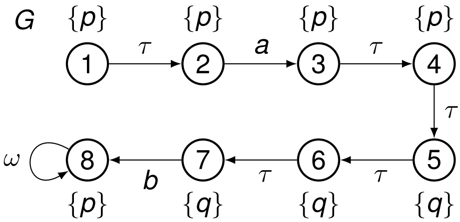
A *branching bisimulation* is obtained if, for all $x \in X$,

$$\Pi(x) = \{y \in X \,|\, \Pi \preceq \Pi_\lambda \wedge T_\Pi(x) = T_\Pi(y)\},$$

where $\Pi_\lambda(x) = \{y \in X \,|\, \lambda(x) = \lambda(y)\}$ is the state label partition, and the set of block transitions from state x

$$T_\Pi(x) = \{\Pi(x) \xrightarrow{a} \Pi(x') \,|\, x \underset{\Pi}{\overset{a}{\twoheadrightarrow}} x'\}.$$

Branching Bisimulation: Example 1



State label partition $\Pi_\lambda = \{\{1,2,3,4\}, \{5,6,7\}\}$. All states $y \in \Pi(x)$ have the same set of block transitions, i.e. $T_\Pi(y) = T_\Pi(x)$.

$$T_\Pi(1) = T_\Pi(2) = \{\{1,2\} \xrightarrow{a} \{3,4\}\},$$

$$T_\Pi(3) = T_\Pi(4) = \{\{3,4\} \xrightarrow{\tau} \{5,6,7\}\},$$

$$T_\Pi(5) = T_\Pi(6) = T_\Pi(7) = \{\{5,6,7\} \xrightarrow{b} \{8\}\},$$

$$T_\Pi(8) = \{\{8\} \xrightarrow{\omega} \{8\}\}.$$

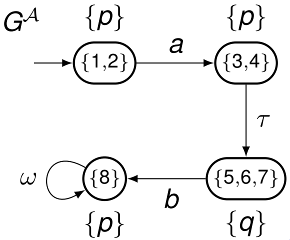
Branching Bisimulation: Example 1

$$T_{\Pi}(1) = T_{\Pi}(2) = \{\{1,2\} \stackrel{a}{\rightarrow} \{3,4\}\},$$

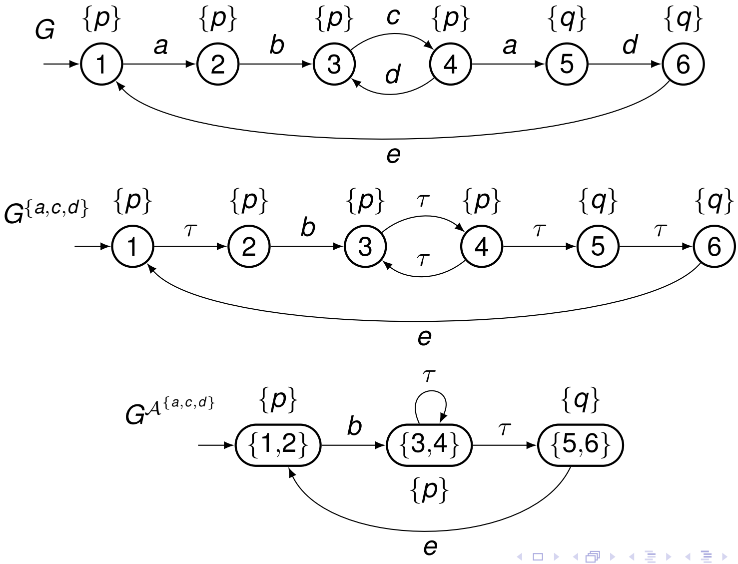
$$T_{\Pi}(3) = T_{\Pi}(4) = \{\{3,4\} \stackrel{\tau}{\rightarrow} \{5,6,7\}\},$$

$$T_{\Pi}(5) = T_{\Pi}(6) = T_{\Pi}(7) = \{\{5,6,7\} \stackrel{b}{\rightarrow} \{8\}\},$$

$$T_{\Pi}(8) = \{\{8\} \stackrel{\omega}{\rightarrow} \{8\}\}.$$



Branching Bisimulation: Example 2



Incremental Abstraction

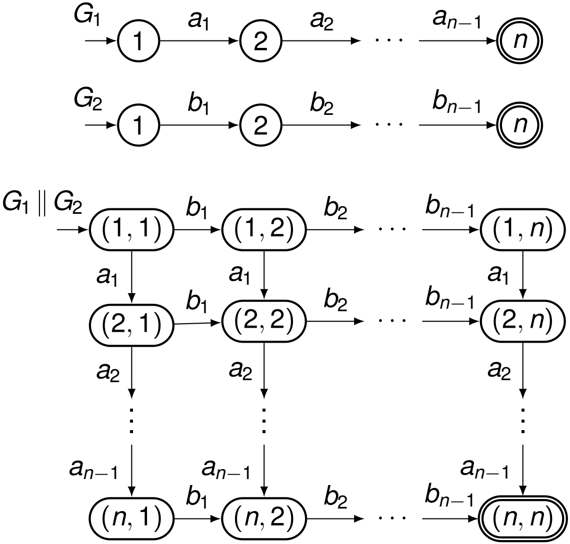
Incremental abstraction of $G = G_1 \| G_2 \| \cdots \| G_n$.

(i) Hide ordinary events by τ events when they become local.

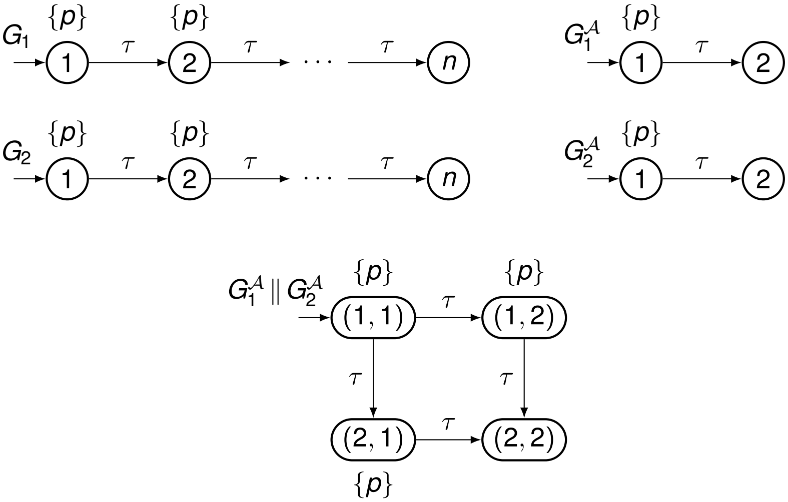
(ii) Apply visible bisimulation abstraction after every synchronization.

(iii) Evaluate any CTL* expression without the next operator on the abstracted model.

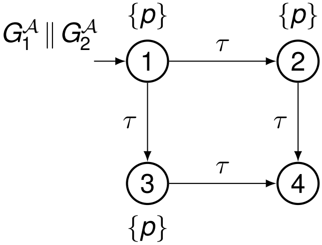
Reachability Analysis: Simple Example

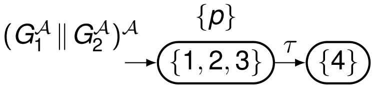


Reachability Analysis with Incremental Abstraction



Reachability Analysis with Incremental Abstraction





Reduction from n^2 to 2 states

State label partition

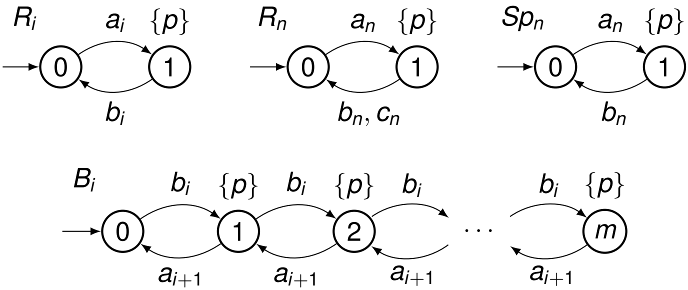
$$\Pi_\lambda = \{\{1,2,3\},\{4\}\}$$

Block transitions

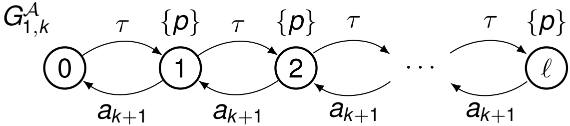
$$T_\Pi(1) = T_\Pi(2) = T_\Pi(3) = \{\{1,2,3\} \xrightarrow{\tau} \{4\}\}$$

Incremental Abstraction: Buffer/Resource Example

$$G = R_1 \,\|\, B_1 \,\|\, R_2 \,\|\, B_2 \,\|\, \cdots \,\|\, R_{n-1} \,\|\, B_{n-1} \,\|\, R_n \,\|\, Sp_n,$$

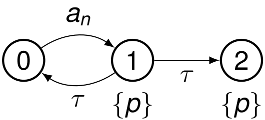


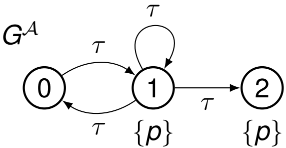
Incremental Abstraction: Buffer/Resource Example



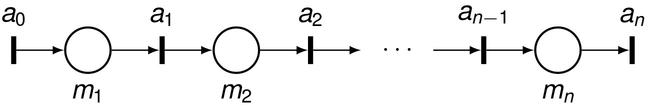
$\ell = k(m+1)$

$G^{\mathcal{A}}_{n,n} = (R^{\mathcal{A}}_n \,\|\, Sp_n)^{\mathcal{A}}$



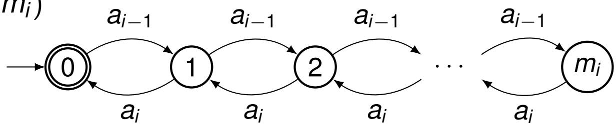


From Petri Net to Transition System



$$\mathrm{N}(a_0, \ldots, a_n, m_1, \ldots, m_n) = \|_{i \in \mathbb{N}_n^+} \mathrm{N}(a_{i-1}, a_i, m_i)$$

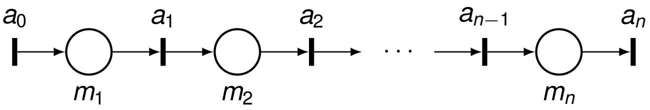
$\mathrm{GB}(a_{i-1}, a_i, m_i)$



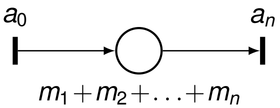
$$\mathrm{TS}\big(\|_{i \in \mathbb{N}_n^+} \mathrm{N}(a_{i-1}, a_i, m_i)\big) = \|_{i \in \mathbb{N}_n^+} \mathrm{GB}(a_{i-1}, a_i, m_i)$$

Every bounded place is replaced by a buffer transition system model!

Analytical Petri Net Reduction



Reduced Petri net



Capacity in each place = m

Reduction from $(1+m)^n$ to $1+nm$ states in the reduced Petri net

Branching Bisimulation: Simplified Rule

A τ transition can be deleted when no alternative transitions are involved in the source state and the source and target states have the same state label.