Abstraction of Nondeterministic Automata

Dr Rong Su
S1-B1b-59, School of EEE
Nanyang Technological University
Tel: +65 6790-6042, Email: rsu@ntu.edu.sg

Outline

- Motivation
- Automaton Abstraction
- Relevant Properties
- Conclusions

Key Concepts of RW Supervisory Control Theory (SCT)

- Controllability
- Observability
- Nonblockingness
  - Checking nonblockingness is computationally intensive
    - Let $L_m(S/G) = L_m(G_1) \,\|\, \ldots \,\|\, L_m(G_n) \,\|\, L_m(S_1) \,\|\, \ldots \,\|\, L_m(S_r)$
    - Let $L(S/G) = L(G_1) \,\|\, \ldots \,\|\, L(G_n) \,\|\, L(S_1) \,\|\, \ldots \,\|\, L(S_r)$
    - Check whether or not $\overline{L_m(S/G)} = L(S/G)$

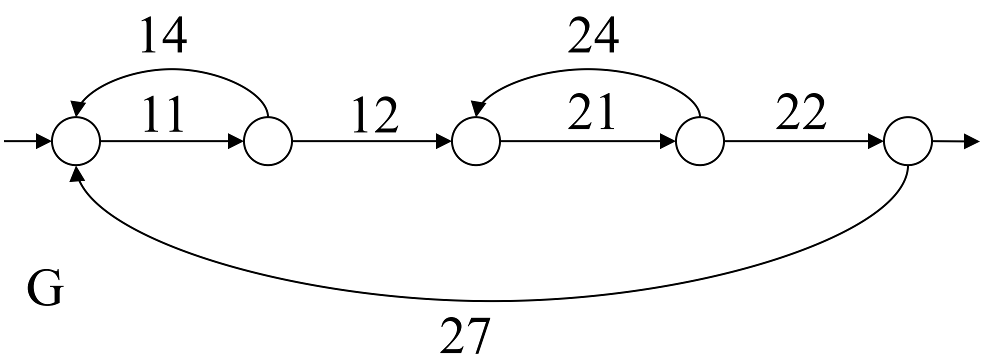
We have the state-space explosion issue here!

A Few Attempts to Deal with Nonblockingness

- State-feedback Control and Symbolic Computation, e.g.
  - supervisory control of state tree structures (STS)
- Abstraction-Based Synthesis, e.g.
  - coordinated modular supervisory control (MSC)
  - hierarchical supervisory control (HSC)
- Synthesis based on Structural Decoupling, e.g.
  - interface-based supervisory control (IBSC)

Problems Associated with These Attempts

- STS is centralized, not suitable for very large systems
- Current hierarchical and modular approaches need observers
  - The observer property is too strong!



- $\Sigma = \{11,12,14,21,22,24,27\}$
- $\Sigma' = \{11,21\}$ and $\Sigma' \subseteq \Sigma''$
- To make $P:\Sigma^* \rightarrow \Sigma''^*$ an $L_m(G)$-observer
  - we need $\Sigma'' = \Sigma$

- Interfaces are very difficult to design

Our Goal

- To define an abstraction κ over (nondeterministic) FSAs,
  - It has the following property similar to what an observer has, namely

    for any G and an S whose alphabet is the same as $\kappa(G)$,
    $G \times S$ is nonblocking if (and only if) $\kappa(G) \times S$ is nonblocking

  - It has no special requirement on a target alphabet as an observer does

Outline

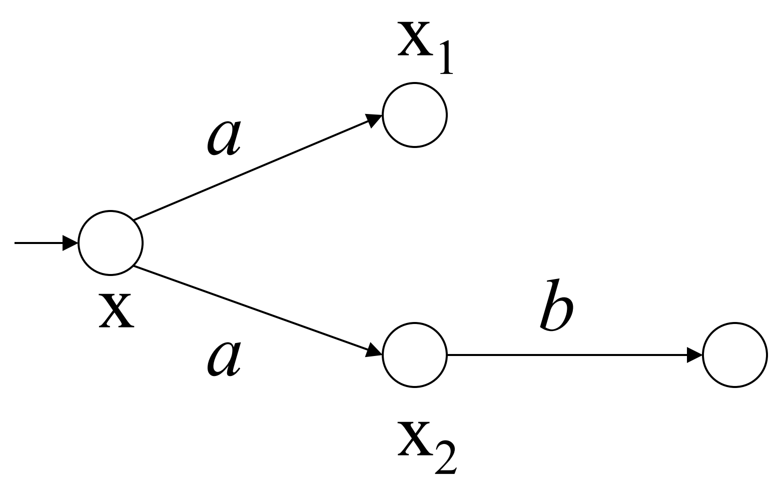
- Motivation
- Automaton Abstraction
- Relevant Properties
- Conclusions

Nondeterministic Finite-State Automaton

- A finite-state automaton $G=(X, \Sigma, \xi, x_0, X_m)$ is *nondeterministic* if

$$\xi: X \times \Sigma \rightarrow 2^X$$

  - i.e a state may have more than one transition with the same event label



- From now on we assume all automata are nondeterministic

Automaton Product

- Let $G_i=(X_i,\Sigma_i,\xi_i,x_{0,i},X_{m,i})\in\phi(\Sigma_i)$ with i=1,2.
- The *product* of G_1 and G_2, written as $G_1\times G_2$, is an automaton

$$G_1\times G_2=(X_1\times X_2,\ \Sigma_1\cup\Sigma_2,\ \xi_1\times\xi_2,\ (x_{0,1},x_{0,2}),X_{m,1}\times X_{m,2})$$

where $\xi_1\times\xi_2:X_1\times X_2\times(\Sigma_1\cup\Sigma_2)\to 2^{X1\times X2}$ is defined as follows,

$$(\xi_1\times\xi_2)((\boldsymbol{x}_1,\boldsymbol{x}_2),\sigma):=\begin{cases}\xi_1(\boldsymbol{x}_1,\sigma)\times\{\boldsymbol{x}_2\} & \text{if } \sigma\in\Sigma_1-\Sigma_2\\ \{\boldsymbol{x}_1\}\times\xi_2(\boldsymbol{x}_2,\sigma) & \text{if } \sigma\in\Sigma_2-\Sigma_1\\ \xi_1(\boldsymbol{x}_1,\sigma)\times\xi_2(\boldsymbol{x}_2,\sigma) & \text{if } \sigma\in\Sigma_1\cap\Sigma_2\end{cases}$$

The Concept of Equivalence Relation

- Given a set X, let R be a *binary relation* on X, namely $R \subseteq X \times X$
  - For any $(x,x) \in R$, we write xRx.

- We say R is an *equivalence relation* on X, if
  - R is *reflexive*, i.e. $(\forall x \in X)\ xRx$
  - R is *symmetric*, i.e. $(\forall x,y \in X)\ xRy \Rightarrow yRx$
  - R is *transitive*, i.e. $(\forall x,y,z \in X)\ xRy \wedge yRz \Rightarrow xRz$

- Let $\mathsf{E}(X)$ be the collection of all equivalence relations on X
  - $\mathsf{E}(X)$ is a complete lattice

The Concept of Marking Weak Bisimilarity

- Given $G=(X,\Sigma,\xi,x_0,X_m)$, let $\Sigma'\subseteq\Sigma$, $R \subseteq X \times X$ be an equivalence relation.
- R is a *marking weak bisimulation* relation over X with respect to Σ' if
  - $R \subseteq X_m \times X_m \cup (X - X_m)\times(X - X_m)$
  - For all $(x,x')\in R$ and $s\in\Sigma^*$, if $\xi(x,s)\neq\varnothing$ then there exists $s'\in\Sigma^*$ such that

$$\xi(x',s')\neq\varnothing \wedge P(s)=P(s') \wedge (\forall y\in\xi(x,s))(\exists y'\in\xi(x',s'))\ (y,y')\in R$$

where $P : \Sigma^* \to \Sigma'^*$ is the natural projection

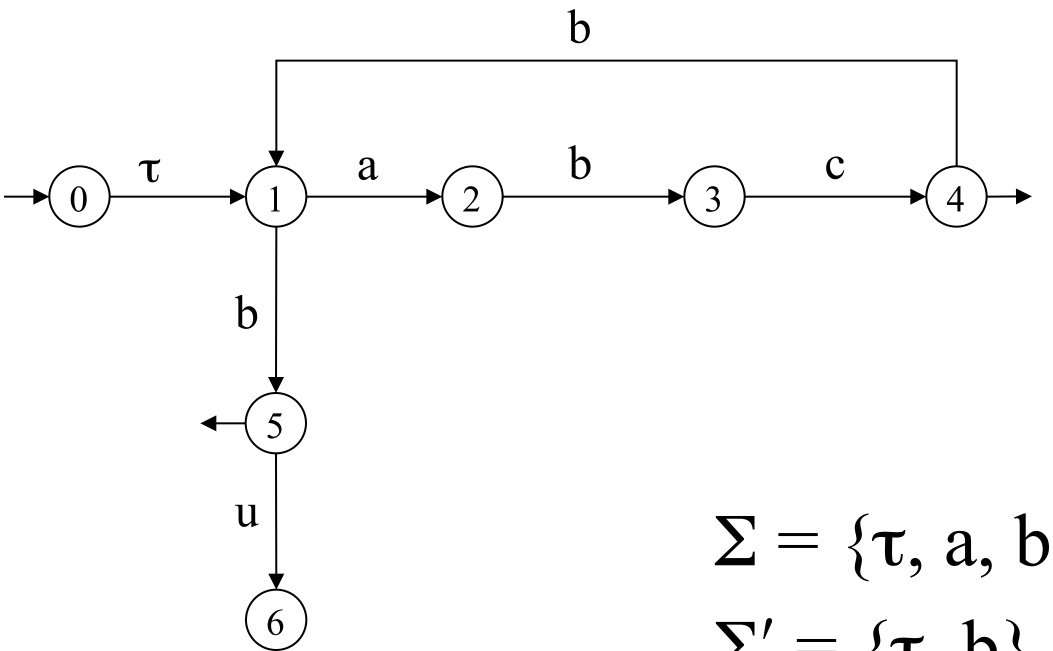
- The largest marking weak bisimulation is *marking weak bisimilarity*, written as $\approx_{\Sigma'}$

Automaton Abstraction

- Let $G=(X,\Sigma,\xi,x_0,X_m)$ and $\Sigma'\subseteq\Sigma$
- For each $x\in X$ let $[x] := \{x'\in X \mid (x,x')\in\approx\Sigma'\}$, and $X/\approx_{\Sigma'} := \{[x] \mid x\in X\}$.
- $G/\approx_{\Sigma'} = (X',\Sigma',\xi',x_0',X_m')$ is an *automaton abstraction* of G w.r.t. $\approx_{\Sigma'}$ if
  - $X' = X/\approx_{\Sigma'}$, $X_m' = \{[x]\in X' \mid [x] \cap X_m \neq \varnothing\}$, $x_0' = [x_0]\in X'$
  - $\xi':X'\times\Sigma' \rightarrow 2^{X'}$, where for any $[x]\in X'$ and $\sigma\in\Sigma'$,

$$\xi'([x],\sigma):=\{[x']\in X'|(\exists y\in[x],y'\in[x'])(\exists u,u'\in(\Sigma-\Sigma')^*)\ y'\in\xi(y,u\sigma u')\}$$

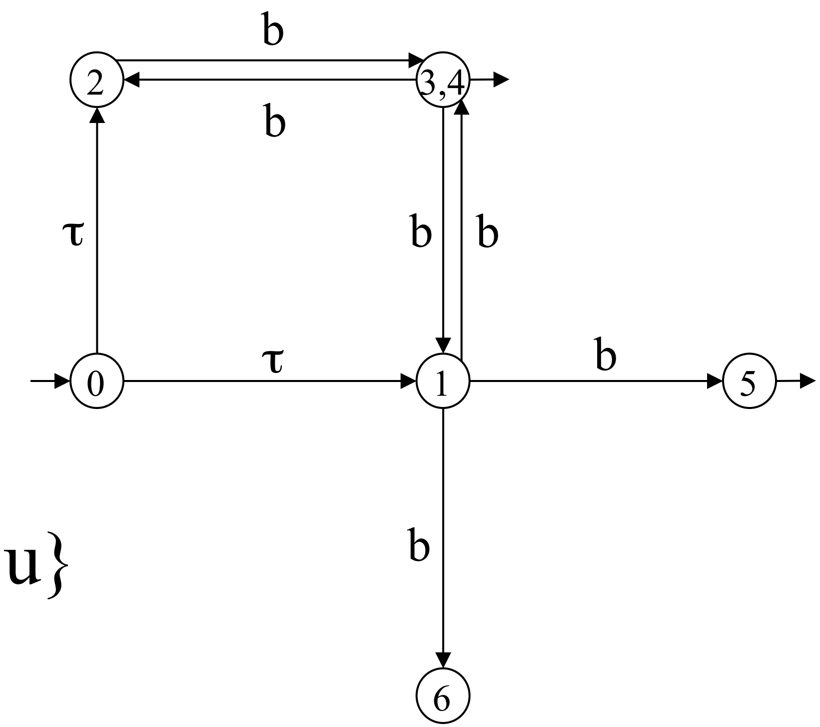
Example



$$\Sigma = \{\tau, a, b, c, u\}$$

$$\Sigma' = \{\tau, b\}$$

G



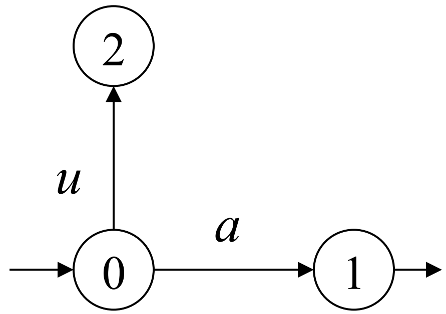
$G/\approx_{\Sigma'}$

Outline

- Motivation
- Automaton Abstraction
- Relevant Properties
- Conclusions

Effect of Silence Paths

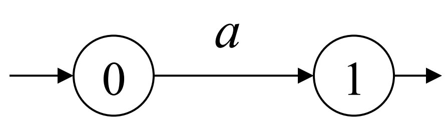
G



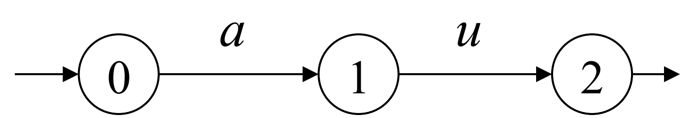
$\Sigma = \{a, b\}$

$\Sigma' = \{a\}$

$G/\approx_{\Sigma'}$



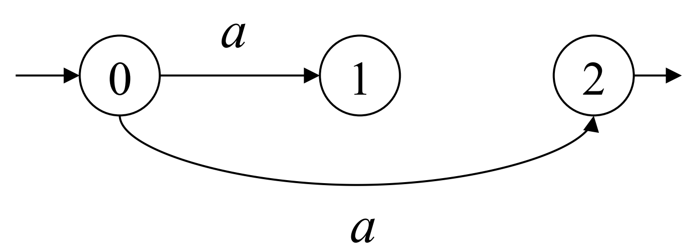
G



$\Sigma = \{a, b\}$

$\Sigma' = \{a\}$

$G/\approx_{\Sigma'}$

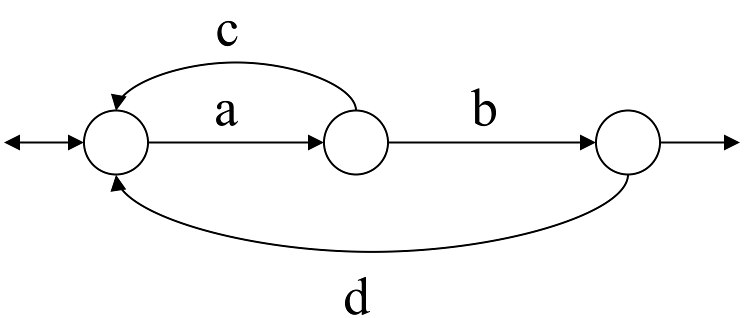


- Abstraction may create unwanted behaviours.
- To avoid this, we introduce the concept of standardized automata.

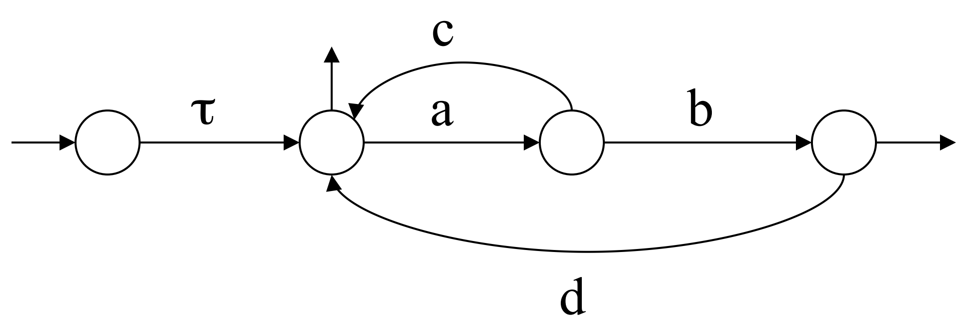
The Standardized Automata

- Suppose $G = (X,\Sigma,\xi,x_0,X_m)$. Bring in a new event symbol τ.
  - τ will be treated as uncontrollable and unobservable.
- An automaton $G = (X,\Sigma\cup\{\tau\},\xi,x_0,X_m)$ is *standardized* if
  - $x_0 \notin X_m$
  - $(\forall x\in X)\ \xi(x,\tau) \neq \varnothing \Leftrightarrow x = x_0$
  - $(\forall \sigma\in\Sigma)\ \xi(x_0,\sigma) = \varnothing$
  - $(\forall x\in X)(\forall \sigma\in\Sigma\cup\{\tau\})\ x_0\notin\xi(x,\sigma)$
- Let $\phi(\Sigma)$ be the collection of all standardized automata over Σ.

Example of a Standardized Automaton



G : before standardization



G : after standardization

Marking Awareness

- $G \in \phi(\Sigma)$ is *marking aware* with respect to $\Sigma' \subseteq \Sigma$, if

$$(\forall x \in X - X_m)(\forall s \in \Sigma^*)\ \xi(x,s) \cap X_m \neq \varnothing \Rightarrow P(s) \neq \varepsilon$$

where $P: \Sigma^* \rightarrow \Sigma'^*$ is the natural projection.

Automaton Abstraction vs Natural Projection

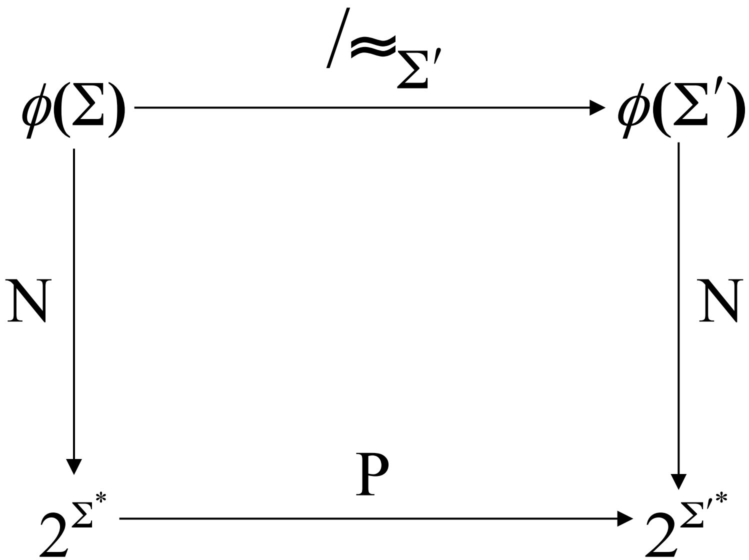
- Let $B(G) = \{s \in \Sigma^* \mid (\exists x \in \xi(x_0,s))(\forall s' \in \Sigma^*)\ \xi(x,s') \cap X_m = \varnothing\}$.
- Let $N_G(x) = \{s \in \Sigma^* \mid \xi(x,s') \cap X_m \neq \varnothing\}$. In particular, $N(G) := N_G(x_0)$.
- **Proposition 1**

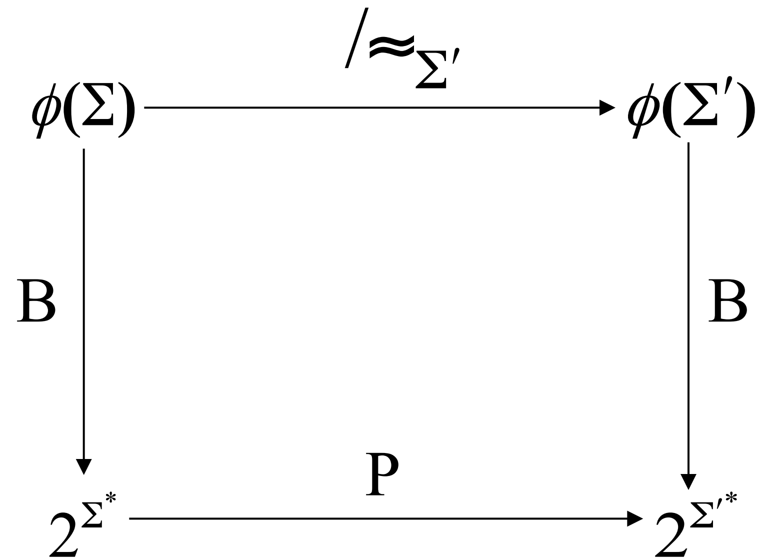
  Let $G \in \phi(\Sigma)$, $\Sigma' \subseteq \Sigma$, and $P: \Sigma^* \to \Sigma'^*$ be the natural projection. Then

  – $P(B(G)) \subseteq B(G/\approx_{\Sigma'})$ and $P(N(G)) = N(G/\approx_{\Sigma'})$

    i.e. automaton abstraction may potentially create more blocking behaviours

  – If G is marking aware with respect to Σ', then $P(B(G)) = B(G/\approx_{\Sigma'})$





When G is marking aware with respect to Σ'

Nonblocking Preservation and Equivalence

- Let $G_1, G_2 \in \phi(\Sigma)$.
- G_1 is *nonblocking preserving* w.r.t. G_2, denoted as $G_1 \sqsubseteq G_2$, if
  - $B(G_1) \subseteq B(G_2)$ and $N(G_1) = N(G_2)$
  - For any $s \in \overline{N(G_1)}$, and $x_1 \in \xi_1(x_{1,0}, s)$, there exists $x_2 \in \xi_2(x_{2,0}, s)$ such that
    - $N_{G2}(x_2) \subseteq N_{G1}(x_1)$
    - $x_1 \in X_{1,m} \Leftrightarrow x_2 \in X_{2,m}$
- G_1 is *nonblocking equivalent* to G_2, denoted as $G_1 \cong G_2$, if
  - $G_1 \sqsubseteq G_2$ and $G_2 \sqsubseteq G_1$

- **Proposition 2 (Nonblocking Invariance under product)**

  For any $\Sigma' \subseteq \Sigma$, $G_1, G_2 \in \phi(\Sigma)$ and $G_3 \in \phi(\Sigma')$,

  – if $G_1 \sqsubseteq G_2$ then $G_1 \times G_3 \sqsubseteq G_2 \times G_3$

  – if $G_1 \cong G_2$ then $G_1 \times G_3 \cong G_2 \times G_3$

- **Proposition 3 (Nonblocking Invariance under abstraction)**

  For any $\Sigma' \subseteq \Sigma$ and $G_1, G_2 \in \phi(\Sigma)$,

  - if $G_1 \sqsubseteq G_2$ then $G_1/\approx_{\Sigma'} \sqsubseteq G_2/\approx_{\Sigma'}$
  - if $G_1 \cong G_2$ then $G_1/\approx_{\Sigma'} \cong G_2/\approx_{\Sigma'}$

- **Proposition 4 (Chain Rule of Automaton Abstraction)**

  Suppose $\Sigma'' \subseteq \Sigma' \subseteq \Sigma$ and $G \in \phi(\Sigma)$. Then $(G/\approx_{\Sigma'})/\approx_{\Sigma''} \cong G/\approx_{\Sigma''}$.

- **Proposition 5 (Distribution of Abstraction over Product)**

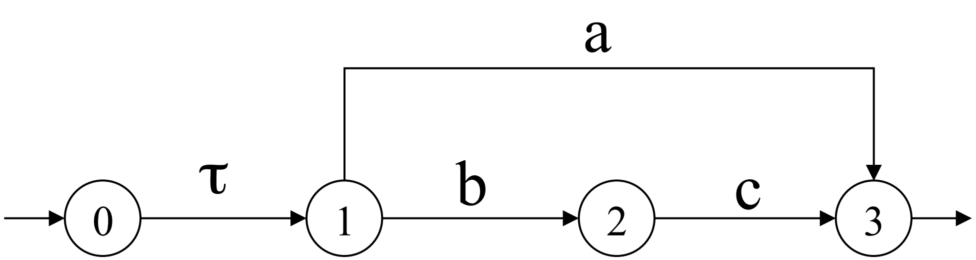
  Let $G_i \in \phi(\Sigma_i)$, where i=1,2, and $\Sigma' \subseteq \Sigma_1 \cup \Sigma_2$.

  – If $\Sigma_1 \cap \Sigma_2 \subseteq \Sigma'$, then $(G_1 \times G_2)/\approx_{\Sigma'} \sqsubseteq (G_1/\approx_{\Sigma 1 \cap \Sigma'}) \times (G_2/\approx_{\Sigma 2 \cap \Sigma'})$.

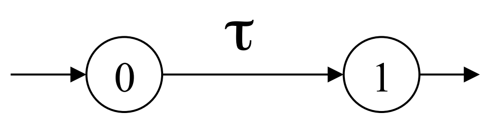
  – If $\Sigma_1 \cap \Sigma_2 \subseteq \Sigma'$ and G_i (i=1,2) is marking aware w.r.t. $\Sigma_i \cap \Sigma'$, then

$$(G_1 \times G_2)/\approx_{\Sigma'} \cong (G_1/\approx_{\Sigma 1 \cap \Sigma'}) \times (G_2/\approx_{\Sigma 2 \cap \Sigma'})$$

Example 1



G_1

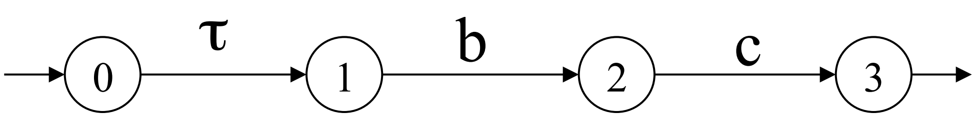


G_2

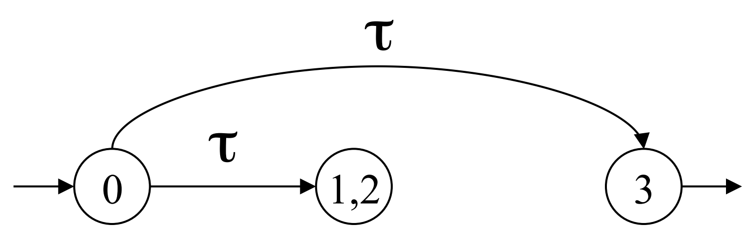
$\Sigma_1 = \{\tau, a, b, c\}$

$\Sigma_2 = \{\tau, a\}$

$\Sigma' = \{\tau, a\}$

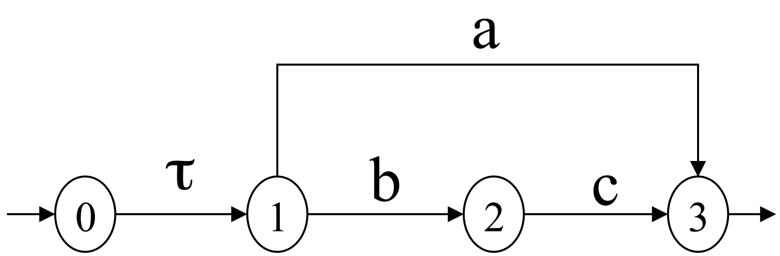


$G_1 \times G_2$

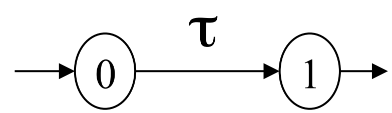


$(G_1 \times G_2)/\approx_{\Sigma'}$

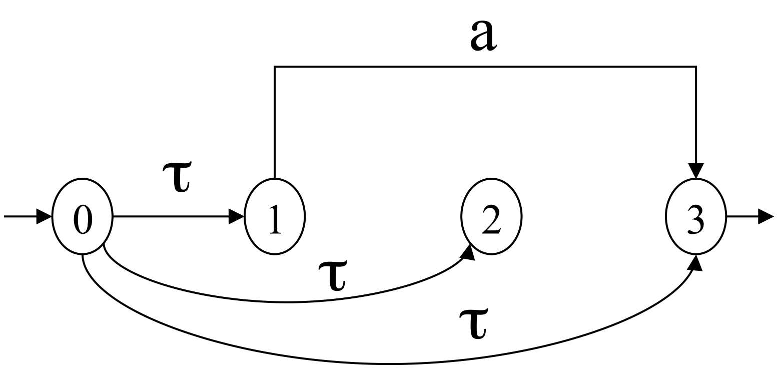
Example 1 (cont.)



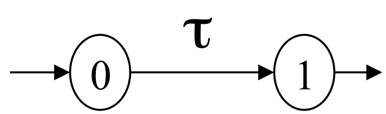
G_1



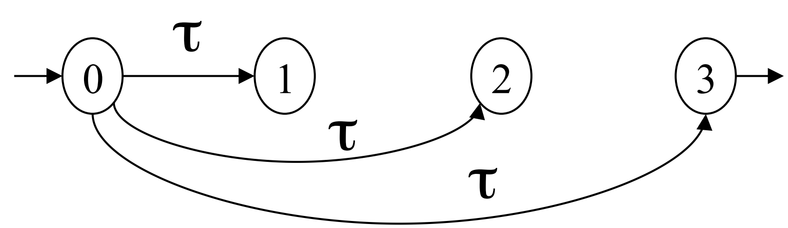
G_2



$G_1/\approx_{\Sigma 1\cap\Sigma'}$



$G_2/\approx_{\Sigma 2\cap\Sigma'}$

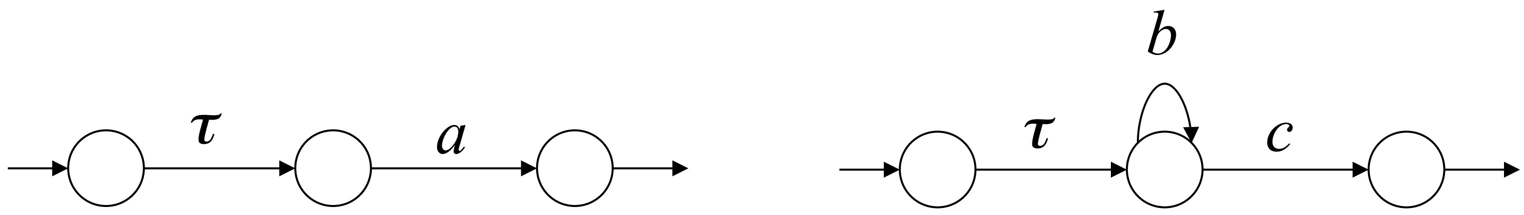


$(G_1/\approx_{\Sigma 1\cap\Sigma'})\times(G_2/\approx_{\Sigma 2\cap\Sigma'})$

Example 1 (cont.)

- Clearly, $(G_1 \times G_2)/\approx_{\Sigma'} \cong (G_1/\approx_{\Sigma 1 \cap \Sigma'}) \times (G_2/\approx_{\Sigma 2 \cap \Sigma'})$
- Thus, the condition of marking awareness is only sufficient.

Example 2



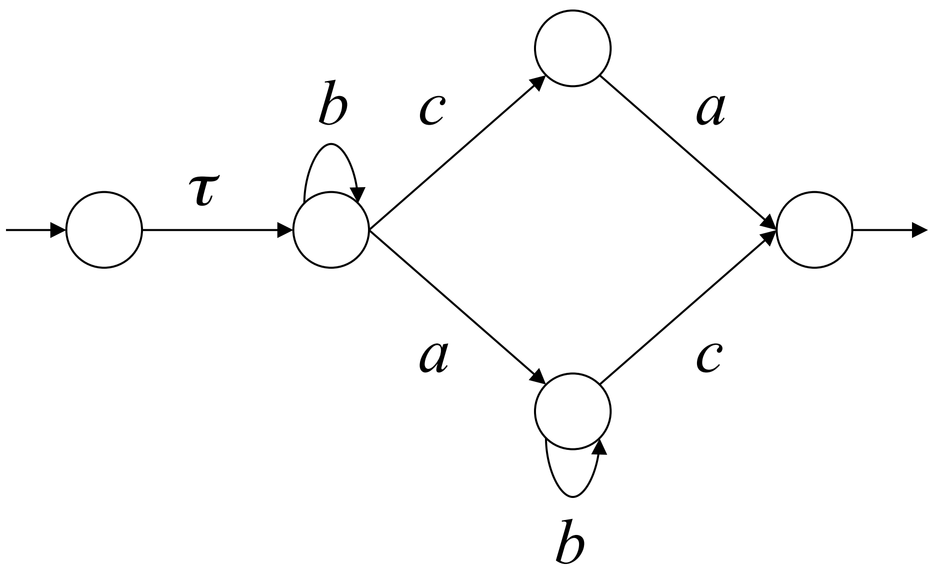
G_1 $\qquad$ G_2

$\Sigma_1=\{\tau,a\}$

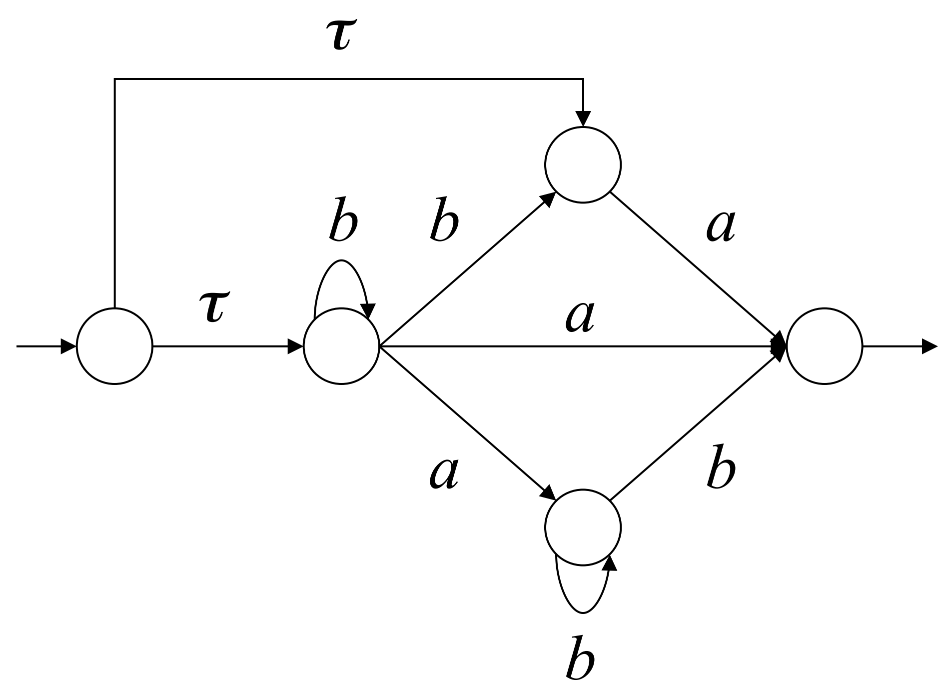
$\Sigma_2=\{\tau,b,c\}$

$\Sigma'=\{\tau,a,b\}$

Example 2 (cont.)



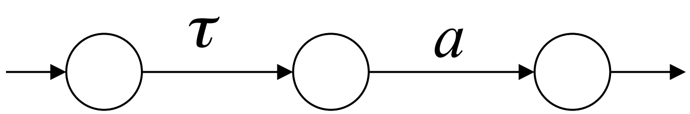
$G_1 \times G_2$

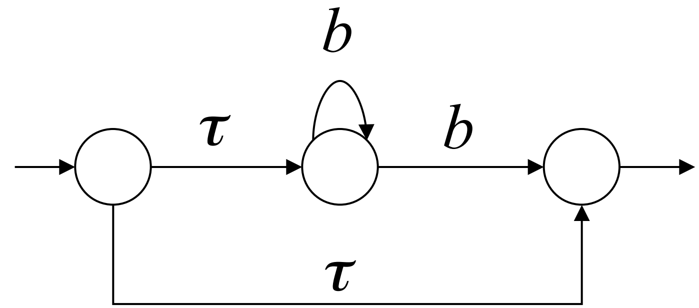


$(G_1 \times G_2)/\approx_{\Sigma'}$

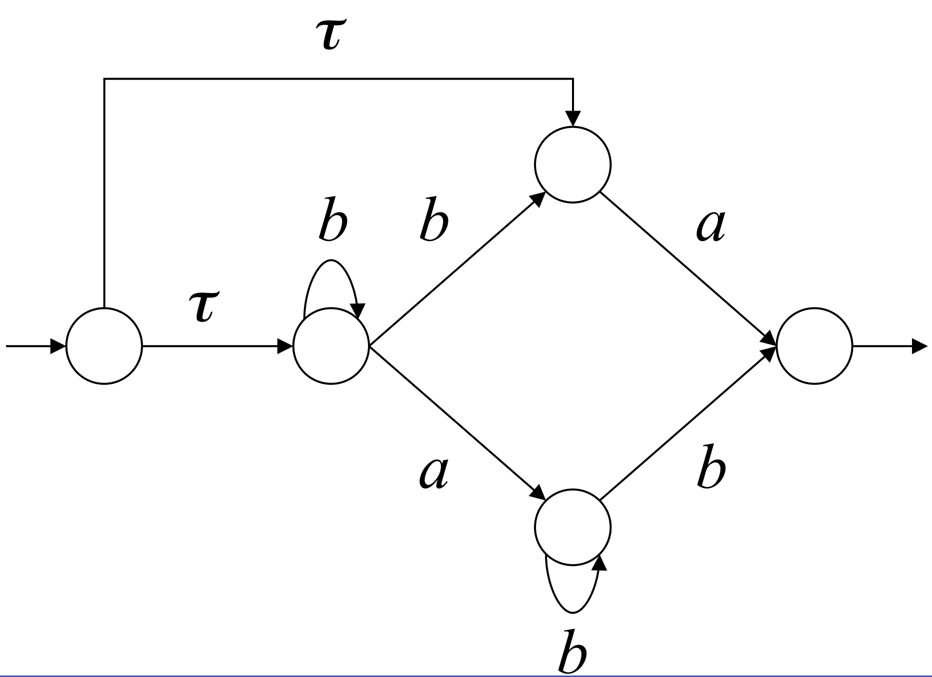
Example 2 (cont.)

$G_1/\approx_{\Sigma1\cap\Sigma'}$





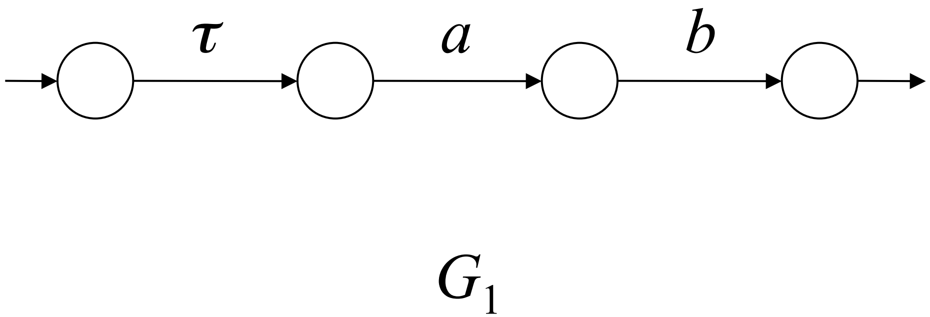
$G_2/\approx_{\Sigma2\cap\Sigma'}$

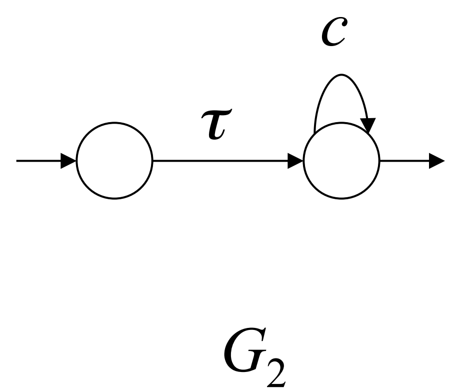


$(G_1/\approx_{\Sigma1\cap\Sigma'})\times(G_2/\approx_{\Sigma2\cap\Sigma'})$

- Clearly, $(G_1 \times G_2)/\approx_{\Sigma'} \cong (G_1/\approx_{\Sigma 1 \cap \Sigma'}) \times (G_2/\approx_{\Sigma 2 \cap \Sigma'})$

Example 3



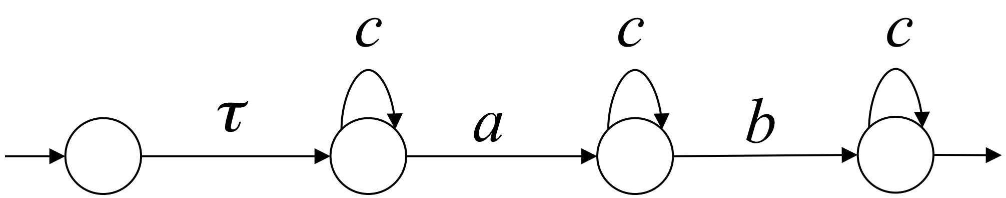


$\Sigma_1 = \{\tau, a, b\}$

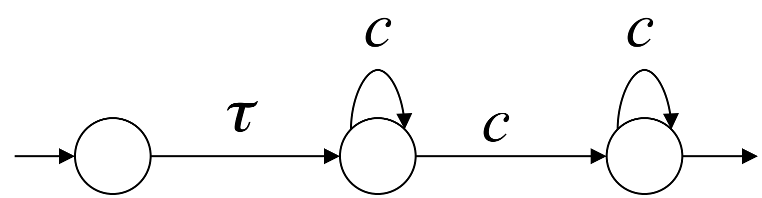
$\Sigma_2 = \{\tau, c\}$

$\Sigma' = \{\tau, c\}$

Example 3 (cont.)

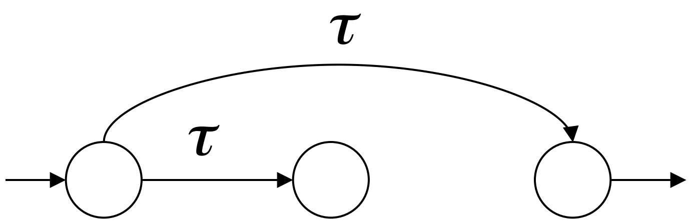


$G_1 \times G_2$

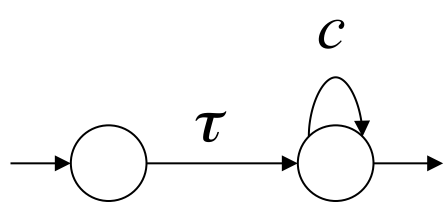


$(G_1 \times G_2)/\approx_{\Sigma'}$

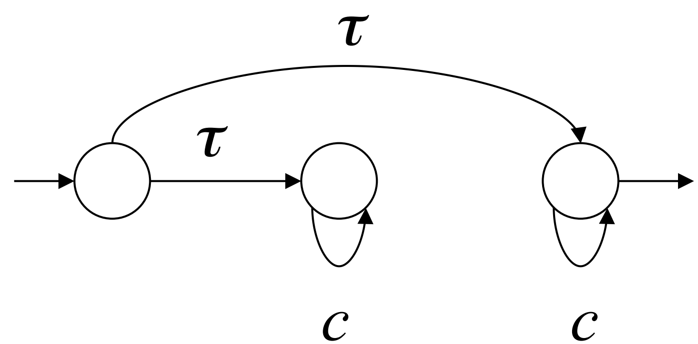
Example 3 (cont.)



$G_1/\approx_{\Sigma1\cap\Sigma'}$



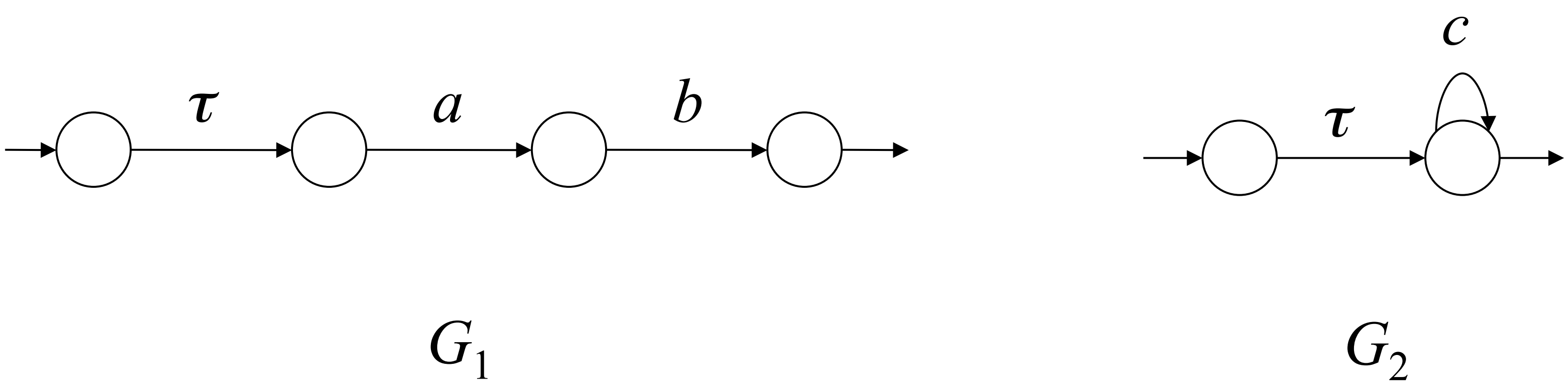
$G_2/\approx_{\Sigma2\cap\Sigma'}$



$(G_1/\approx_{\Sigma1\cap\Sigma'})\times(G_2/\approx_{\Sigma2\cap\Sigma'})$

- Clearly, $(G_1 \times G_2)/\approx_{\Sigma'} \sqsubseteq (G_1/\approx_{\Sigma 1 \cap \Sigma'}) \times (G_2/\approx_{\Sigma 2 \cap \Sigma'})$
- But, it is not true that $(G_1 \times G_2)/\approx_{\Sigma'} \cong (G_1/\approx_{\Sigma 1 \cap \Sigma'}) \times (G_2/\approx_{\Sigma 2 \cap \Sigma'})$

Example 3 (revisit)

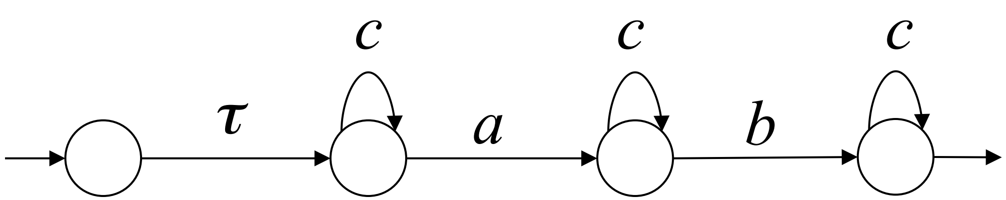


$\Sigma_1=\{\tau,a,b\}$

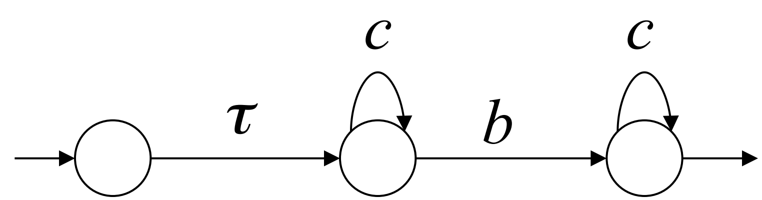
$\Sigma_2=\{\tau,c\}$

$\Sigma'=\{\tau,b,c\}$

Example 3 (cont.)

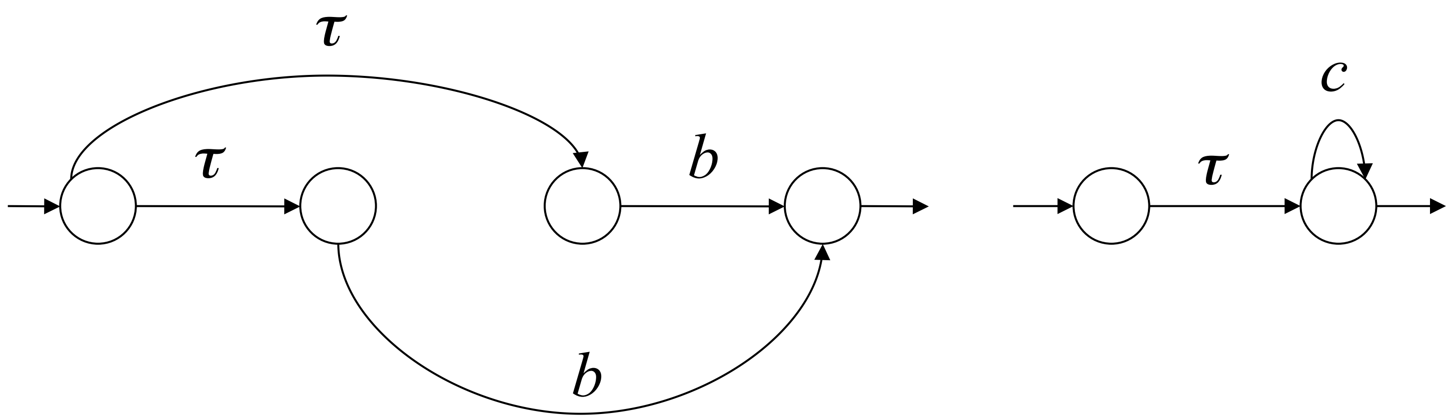


$G_1 \times G_2$

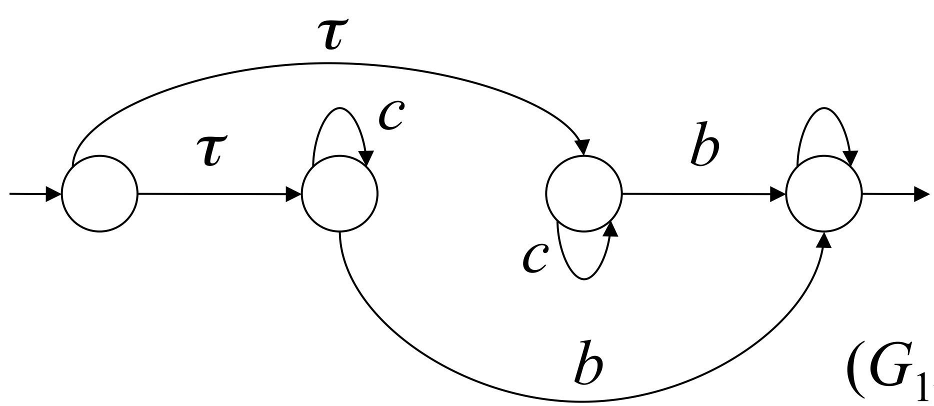


$(G_1 \times G_2)/\approx_{\Sigma'}$

Example 3 (cont.)



$G_1/{\approx_{\Sigma 1 \cap \Sigma'}}$ $G_2/{\approx_{\Sigma 2 \cap \Sigma'}}$



$(G_1/{\approx_{\Sigma 1 \cap \Sigma'}}) \times (G_2/{\approx_{\Sigma 2 \cap \Sigma'}})$

- We can check that, $(G_1 \times G_2)/\approx_{\Sigma'} \cong (G_1/\approx_{\Sigma 1 \cap \Sigma'}) \times (G_2/\approx_{\Sigma 2 \cap \Sigma'})$

Main Result

- **Theorem:** Given Σ and $\Sigma' \subseteq \Sigma$, let $G \in \phi(\Sigma)$ and $S \in \phi(\Sigma')$. Then
  - $B((G/\approx_{\Sigma'}) \times S) = \emptyset \Rightarrow B(G \times S) = \emptyset$
  - G is marking aware w.r.t. $\Sigma' \Rightarrow [B((G/\approx_{\Sigma'}) \times S) = \emptyset \Leftrightarrow B(G \times S) = \emptyset]$

A Computational Challenge

- Let $\{\Sigma_i | i \in I = \{1,2,\ldots,n\}\}$ be a collection of local alphabets.
- For any $J \subseteq I$, let $\Sigma_J := \cup_{j\in J}\Sigma_j$.
- Let $G_i \in \phi(\Sigma_i)$ for each $i \in I$, and $\Sigma' \subseteq \Sigma_I$.
- We want to compute $(\times_{i\in I} G_i)/\approx_{\Sigma'}$ efficiently.

Sequential Abstraction over Product (SAP)

- For k=1,2,…,n,
  - $J(k) := \{1,2,\ldots,k\}$ and $T(k) := \Sigma_{Jk} \cap (\Sigma_{I-Jk} \cup \Sigma')$
  - If k=1 then $W_1 := G_1/\approx_{T(1)}$
  - If k>1 then $W_k := (W_{k-1} \times G_k)/\approx_{T(k)}$

- **Proposition 6**

  Suppose W_n is computed by SAP. Then $(\times_{i\in I} G_i)/\approx_{\Sigma'} \sqsubseteq W_n$.

Conclusions

- Advantages of this approach
  - It possesses the good aspects of an observer
  - It does not have the bad aspects of an observer

- Potential disadvantages of this approach
  - Abstraction creates more transitions, which might complicate synthesis
  - The marking awareness condition is sufficient but not necessary