Parallelism and communication

"real" parallel system
$$P = P_1 \parallel \ldots \parallel P_n$$

compositional design $\uparrow$ $\downarrow$ analysis semantics

transition system
$$\mathcal{T} = \mathcal{T}_1 \parallel \ldots \parallel \mathcal{T}_n$$

goal: define semantic parallel operators on transition systems or program graphs that model "real" parallel operators

Interleaving operator $|||$ for TS

- interleaving of concurrent, independent actions of parallel processes (modelled by TS)
- representation by nondeterministic choice:
  "which subprocess performs the next step?"

$$\mathit{effect}(\alpha|||\beta) \;=\; \mathit{effect}(\alpha;\beta+\beta;\alpha)$$

parallel execution of α and β on *two processors* $\hat{=}$ serial execution on a *single processor* in arbitrary order

Interleaving operator ||| for TS

$$\mathcal{T}_1 = (S_1, Act_1, \longrightarrow_1, S_{0,1}, AP_1, L_1)$$

$$\mathcal{T}_2 = (S_2, Act_2, \longrightarrow_2, S_{0,2}, AP_2, L_2)$$

The transition system $\mathcal{T}_1 \,|||\, \mathcal{T}_2$ is defined by:

$$\mathcal{T}_1 \,|||\, \mathcal{T}_2 = (S_1 \times S_2, Act_1 \cup Act_2, \longrightarrow, S_{0,1} \times S_{0,2}, AP, L)$$

where the transition relation $\longrightarrow$ is given by:

$$\frac{s_1 \xrightarrow{\alpha}_1 s_1'}{\langle s_1, s_2\rangle \xrightarrow{\alpha} \langle s_1', s_2\rangle} \qquad \frac{s_2 \xrightarrow{\alpha}_2 s_2'}{\langle s_1, s_2\rangle \xrightarrow{\alpha} \langle s_1, s_2'\rangle}$$

atomic propositions: $AP = AP_1 \uplus AP_2$

labeling function: $L(\langle s_1, s_2\rangle) = L_1(s_1) \cup L_2(s_2)$

SOS-notation (structured operational semantics)

just a simple notation for operational semantics

$$\frac{\textbf{premise}}{\textbf{conclusion}}$$

E.g., *"the relation $\longrightarrow$ is given by ..."*

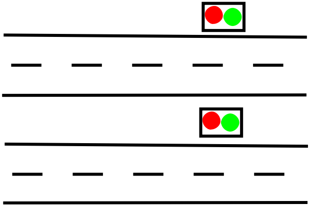
$$\frac{s_1 \xrightarrow{\alpha}_1 s_1'}{\langle s_1, s_2 \rangle \xrightarrow{\alpha} \langle s_1', s_2 \rangle} \qquad \frac{s_2 \xrightarrow{\alpha}_2 s_2'}{\langle s_1, s_2 \rangle \xrightarrow{\alpha} \langle s_1, s_2' \rangle}$$

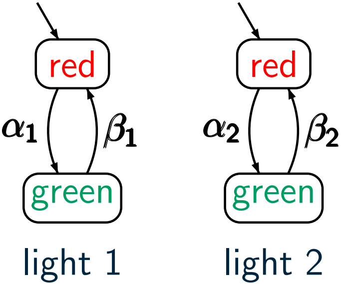
means that $\longrightarrow$ is the smallest relation such that:

(1) If $s_1 \xrightarrow{\alpha}_1 s_1'$, then $\langle s_1, s_2 \rangle \xrightarrow{\alpha} \langle s_1', s_2 \rangle$

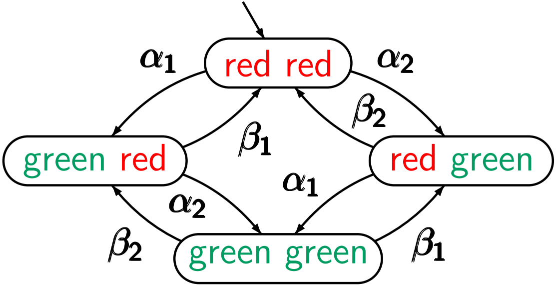
(2) If $s_2 \xrightarrow{\alpha}_2 s_2'$, then $\langle s_1, s_2 \rangle \xrightarrow{\alpha} \langle s_1, s_2' \rangle$

Useless lights for non-crossing streets





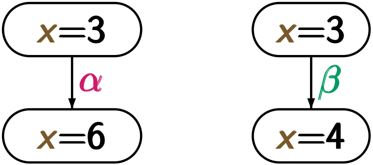
light 1 ||| light 2



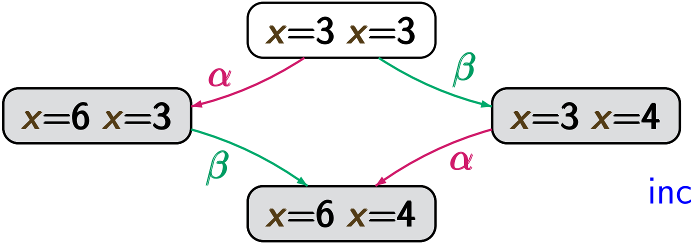
Interleaving fails for dependent actions

dependent actions $\alpha \mathrel{\widehat{=}} x{:=}2x$ and $\beta \mathrel{\widehat{=}} x{:=}x{+}1$

representations in transition systems



interleaving operator $|||$ for transition systems "fails"



inconsistent states

Interleaving for program graphs

. . . for modeling parallel systems with subprocesses communicating via shared variables

program graph $\mathcal{P}_1$
$(Loc_1, \ldots, \hookrightarrow_1, \ldots)$

program graph $\mathcal{P}_2$
$(Loc_2, \ldots, \hookrightarrow_2, \ldots)$

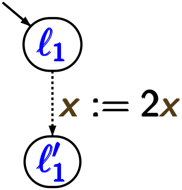
interleaving operator

$$\mathcal{P}_1 \,|||\, \mathcal{P}_2 \;=\; (Loc_1 \times Loc_2, \ldots, \hookrightarrow, \ldots)$$

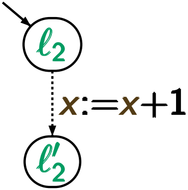
$$\frac{\ell_1 \overset{g:\alpha}{\hookrightarrow}_1 \ell_1'}{\langle \ell_1, \ell_2 \rangle \overset{g:\alpha}{\hookrightarrow} \langle \ell_1', \ell_2 \rangle} \qquad \frac{\ell_2 \overset{g:\alpha}{\hookrightarrow}_2 \ell_2'}{\langle \ell_1, \ell_2 \rangle \overset{g:\alpha}{\hookrightarrow} \langle \ell_1, \ell_2' \rangle}$$

Example: interleaving for PG

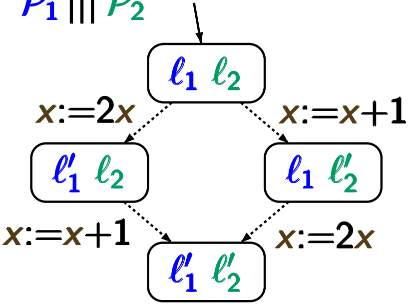
$\mathcal{P}_1$



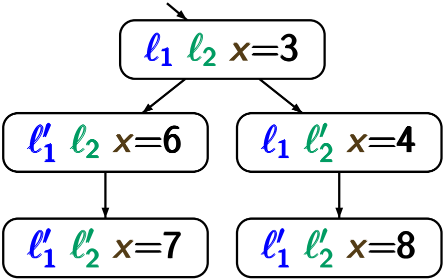
$\mathcal{P}_2$



$\mathcal{P}_1 \,|||\, \mathcal{P}_2$

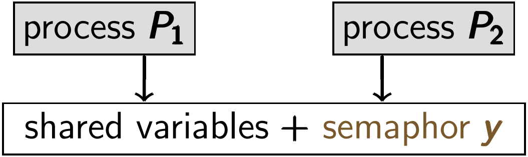


transition system $\mathcal{T}_{\mathcal{P}_1 \,|||\, \mathcal{P}_2}$



note: $\mathcal{T}_{\mathcal{P}_1 \,|||\, \mathcal{P}_2} \neq \mathcal{T}_{\mathcal{P}_1} \,|||\, \mathcal{T}_{\mathcal{P}_2}$

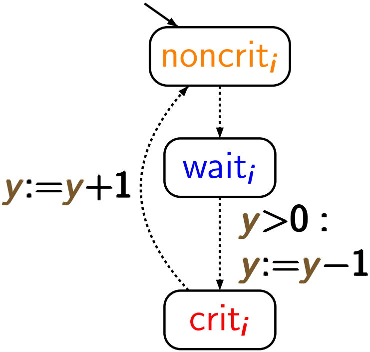
Mutual exclusion with semaphore



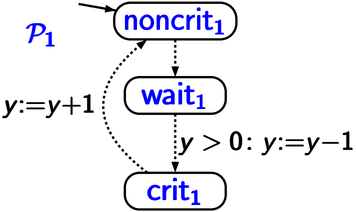
protocol for process P_i

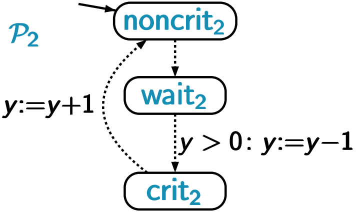
```
LOOP FOREVER
  noncritical actions;
  AWAIT  y > 0 DO
         y:=y−1
  OD
  critical actions;
  y:=y+1
END LOOP
```

program graph $\mathcal{P}_i$

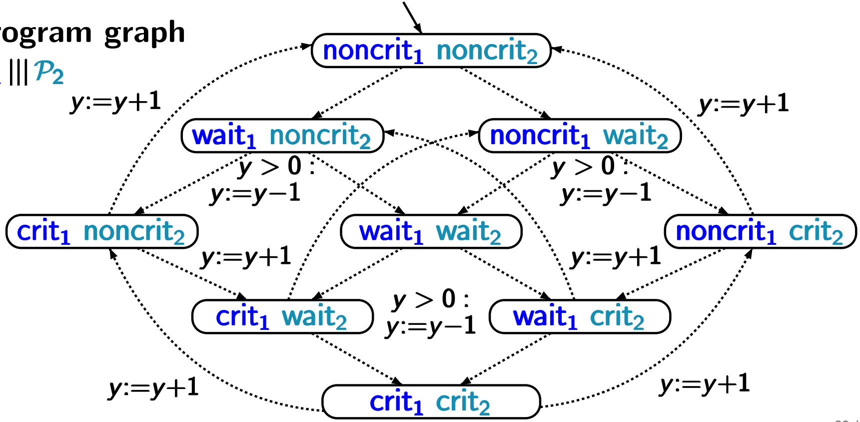


Mutual exclusion with semaphore

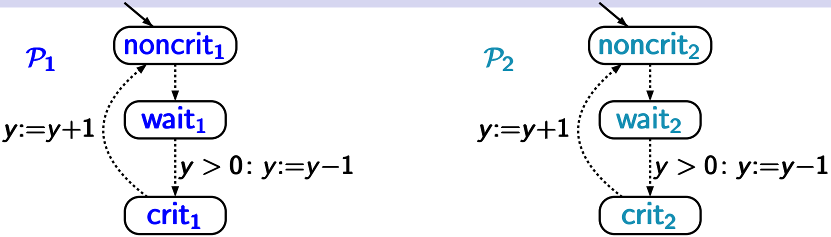




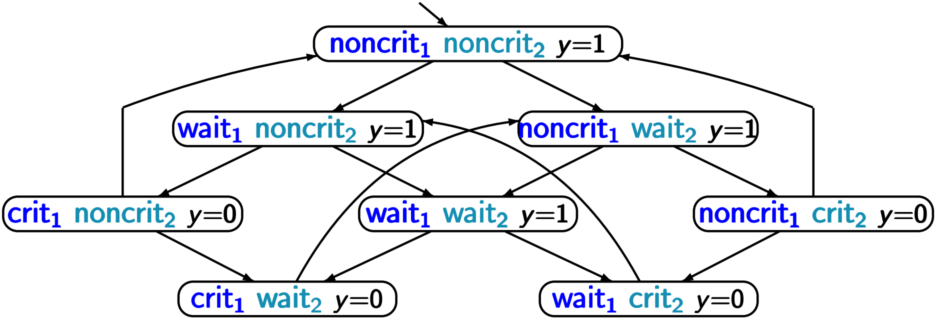
program graph
$\mathcal{P}_1 \,|||\, \mathcal{P}_2$



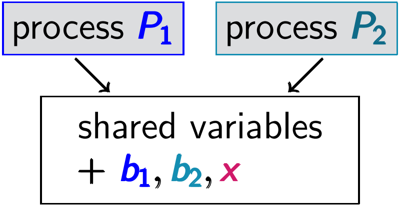
TS for mutual exclusion with semaphore

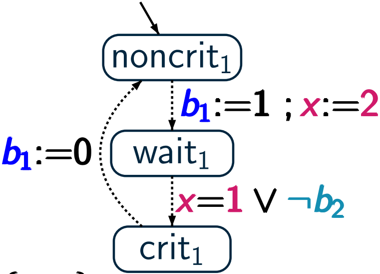


reachable fragment of the transition system $\mathcal{T}_{\mathcal{P}_1 \,|||\, \mathcal{P}_2}$



Peterson algorithm for mutual exclusion

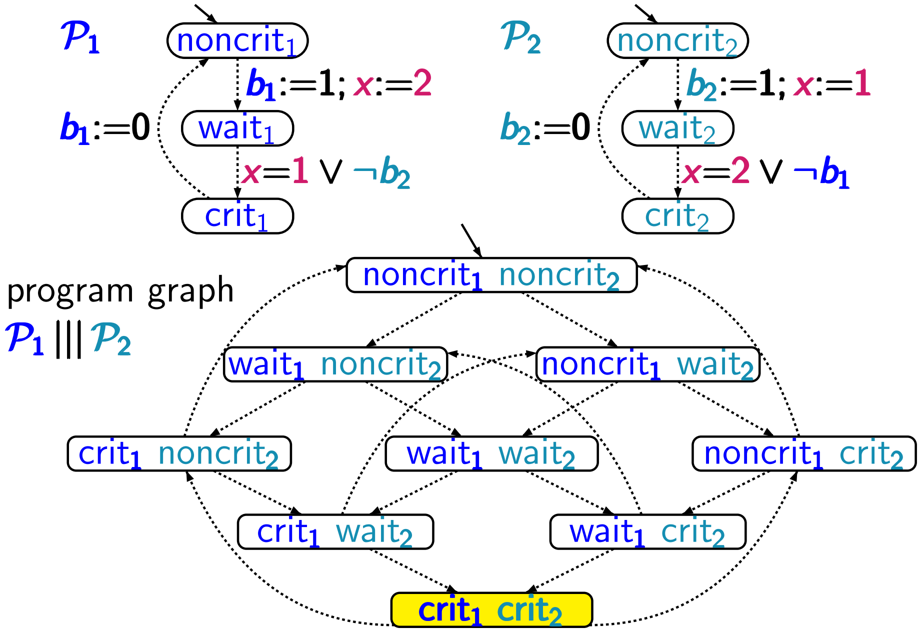




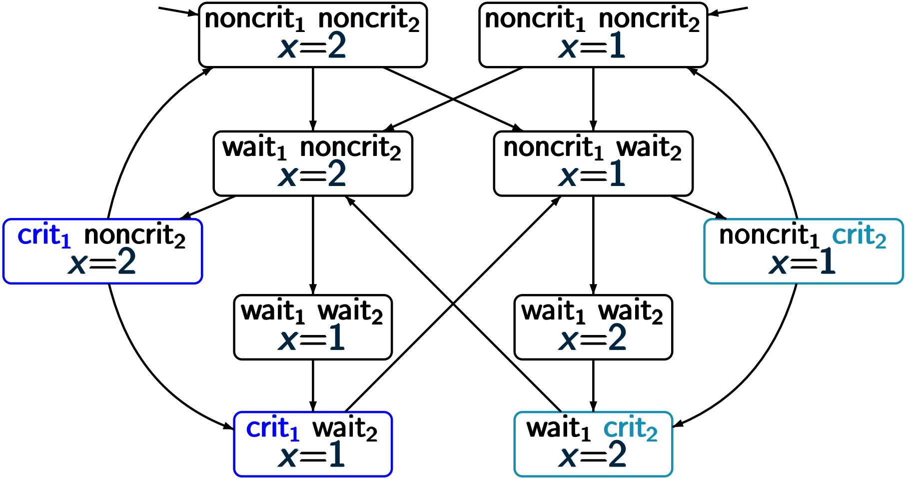
b_1, b_2 Boolean variables, $x \in \{1, 2\}$

```
LOOP FOREVER                          (* protocol for P1 *)
    noncritical actions;
    atomic{b1:=1 ; x:=2};
    AWAIT x=1 ∨ ¬b2 DO critical section OD
    b1:=0
END LOOP
```

Program graphs for Peterson algorithm



TS for the Peterson algorithm

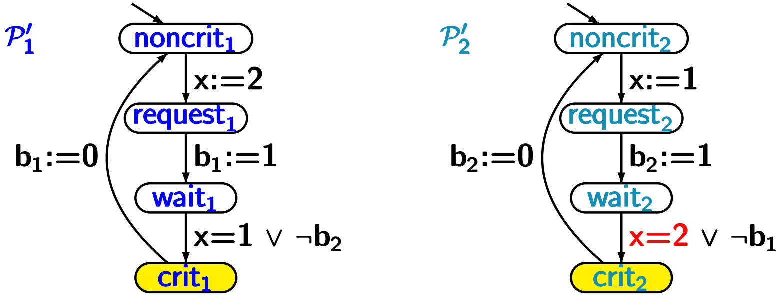


value of b_1 is given by $\textbf{wait}_1 \vee \textbf{crit}_1$

value of b_2 is given by $\textbf{wait}_2 \vee \textbf{crit}_2$

+ unreachable states

Variant of Peterson algorithm incorrect!



possible executions

noncrit$_1$	noncrit$_2$	x=1	$\neg b_1$	$\neg b_2$
noncrit$_1$	request$_2$	x=1	$\neg b_1$	$\neg b_2$
request$_1$	request$_2$	x=2	$\neg b_1$	$\neg b_2$
wait$_1$	request$_2$	x=2	b_1	$\neg b_2$
crit$_1$	request$_2$	x=2	b_1	$\neg b_2$
crit$_1$	wait$_2$	x=2	b_1	b_2
crit$_1$	crit$_2$	x=2	b_1	b_2

Operators for parallelism and communication

- true concurrency: interleaving operator $|||$ for TS
  (no communication, no dependencies)
- communication via shared variables
  - ∗ description of subsystems by program graphs
  - ∗ interleaving $|||$ for program graphs
  - ∗ TS is obtained by "unfolding"
- synchronous message passing $\longleftarrow$ data abstract
  - ∗ operator $\|_{\mathit{Syn}}$ for TS
  - ∗ interleaving for independent actions
  - ∗ synchronization over actions in Syn
- channel systems
  communication via shared variables + via channels
- synchronous product

Synchronous message passing

$\mathcal{T}_1 = (S_1, Act_1, \rightarrow_1, \ldots)$, $\mathcal{T}_2 = (S_2, Act_2, \rightarrow_2, \ldots)$ TS

$Syn \subseteq Act_1 \cap Act_2$ set of synchronization actions

composite transition system:

$$\mathcal{T}_1 \|_{Syn} \mathcal{T}_2 = (S_1 \times S_2, Act_1 \cup Act_2, \rightarrow, \ldots)$$

for modeling the concurrent execution of $\mathcal{T}_1$ and $\mathcal{T}_2$ with synchronization over all actions in Syn

$\mathcal{T}_1 = (S_1, Act_1, \rightarrow_1, \ldots)$, $\mathcal{T}_2 = (S_2, Act_2, \rightarrow_2, \ldots)$ TS

$Syn \subseteq Act_1 \cap Act_2$ set of synchronization actions

composite transition system:

$$\mathcal{T}_1 \|_{Syn} \mathcal{T}_2 = (S_1 \times S_2, Act_1 \cup Act_2, \rightarrow, \ldots)$$

interleaving for all actions $\alpha \in Act_i \setminus Syn$:

$$\frac{s_1 \xrightarrow{\alpha}_1 s_1'}{\langle s_1, s_2 \rangle \xrightarrow{\alpha} \langle s_1', s_2 \rangle} \qquad \frac{s_2 \xrightarrow{\alpha}_2 s_2'}{\langle s_1, s_2 \rangle \xrightarrow{\alpha} \langle s_1, s_2' \rangle}$$

handshaking (rendezvous) for all $\alpha \in Syn$:

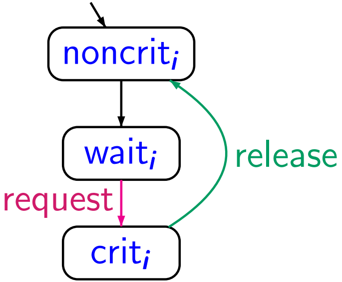
$$\frac{s_1 \xrightarrow{\alpha}_1 s_1' \quad \wedge \quad s_2 \xrightarrow{\alpha}_2 s_2'}{\langle s_1, s_2 \rangle \xrightarrow{\alpha} \langle s_1', s_2' \rangle}$$

Mutual exclusion with an arbiter

protocol for process P_i

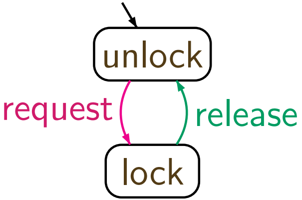
```
LOOP FOREVER DO
      noncritical actions
      request
      critical section
      release
      noncritical actions
OD
```

transition system $\mathcal{T}_i$



Arbiter:

selects nondeterministically a synchronization partner $\mathcal{T}_1$ or $\mathcal{T}_2$



Mutual exclusion with an arbiter

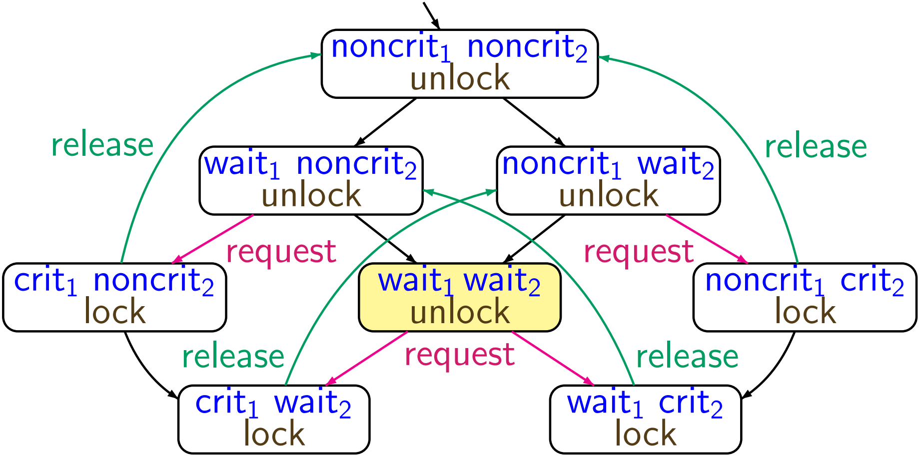
$$(\mathcal{T}_1 \ ||| \ \mathcal{T}_2) \ \|_{Syn} \ \textit{Arbiter} \text{ where } Syn = \{\text{request}, \text{release}\}$$

"pure" interleaving for TS

handshaking for actions request and release

Mutual exclusion with an arbiter

$(\mathcal{T}_1 \,|||\, \mathcal{T}_2) \parallel_{Syn} \textit{Arbiter}$ where $Syn = \{\text{request}, \text{release}\}$



nondeterministic choice: who enters the critical section**?**

synchronization operator $\|_{Syn}$ for
three or more processes

Synchronous message passing

$$\left.\begin{array}{lcl} \mathcal{T}_1 & = & (S_1, Act_1, \rightarrow_1, \ldots) \\ \mathcal{T}_2 & = & (S_2, Act_2, \rightarrow_2, \ldots) \\ \mathcal{T}_3 & = & (S_3, Act_3, \rightarrow_3, \ldots) \\ \mathcal{T}_4 & = & (S_4, Act_4, \rightarrow_4, \ldots) \\ \vdots & & \vdots \end{array}\right\} \text{transition systems}$$

for $Syn \subseteq Act_1 \cup Act_2 \cup Act_3 \cup Act_4 \cup \ldots$

$$\boxed{\begin{array}{l} \mathcal{T}_1 \,\|_{Syn}\, \mathcal{T}_2 \,\|_{Syn}\, \mathcal{T}_3 \,\|_{Syn}\, \mathcal{T}_4 \,\|_{Syn}\, \ldots \stackrel{\text{def}}{=} \\ \qquad \Big(\big((\mathcal{T}_1 \,\|_{Syn}\, \mathcal{T}_2) \,\|_{Syn} \mathcal{T}_3\big) \,\|_{Syn}\, \mathcal{T}_4\Big) \,\|_{Syn}\, \ldots \end{array}}$$

where, e.g., $\mathcal{T}_1 \,\|_{Syn}\, \mathcal{T}_2 \stackrel{\text{def}}{=} \mathcal{T}_1 \,\|_{H}\, \mathcal{T}_2$

with $H = Syn \cap Act_1 \cap Act_2$

Parallel operator $\|$

$$
\begin{array}{lcl}
\mathcal{T}_1 & = & (S_1, Act_1, \rightarrow_1, \ldots) \\
\mathcal{T}_2 & = & (S_2, Act_2, \rightarrow_2, \ldots) \\
\mathcal{T}_3 & = & (S_3, Act_3, \rightarrow_3, \ldots) \\
\mathcal{T}_4 & = & (S_4, Act_4, \rightarrow_4, \ldots) \\
\vdots & & \vdots
\end{array}
$$

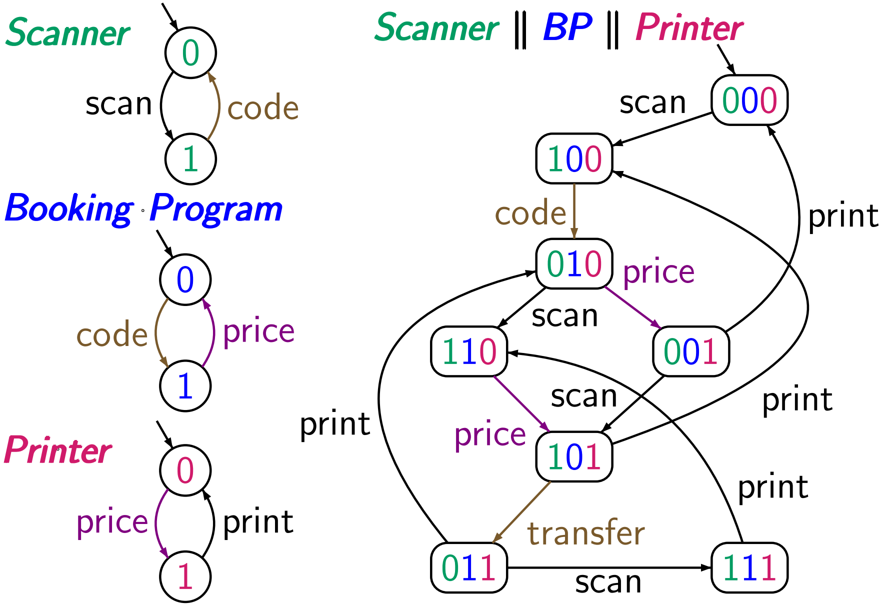
transition systems s.t.
$Act_i \cap Act_j \cap Act_k = \emptyset$
if i, j, k are pairwise distinct

$$
\boxed{\begin{array}{l}
\mathcal{T}_1 \,\|\, \mathcal{T}_2 \,\|\, \mathcal{T}_3 \,\|\, \mathcal{T}_4 \,\|\, \ldots \stackrel{\text{def}}{=} \\
\quad (((\mathcal{T}_1 \,\|_{Syn_{1,2}}\, \mathcal{T}_2) \,\|_{Syn_{1,2,3}}\, \mathcal{T}_3) \,\|_{Syn_{1,2,3,4}}\, \mathcal{T}_4) \ldots
\end{array}}
$$

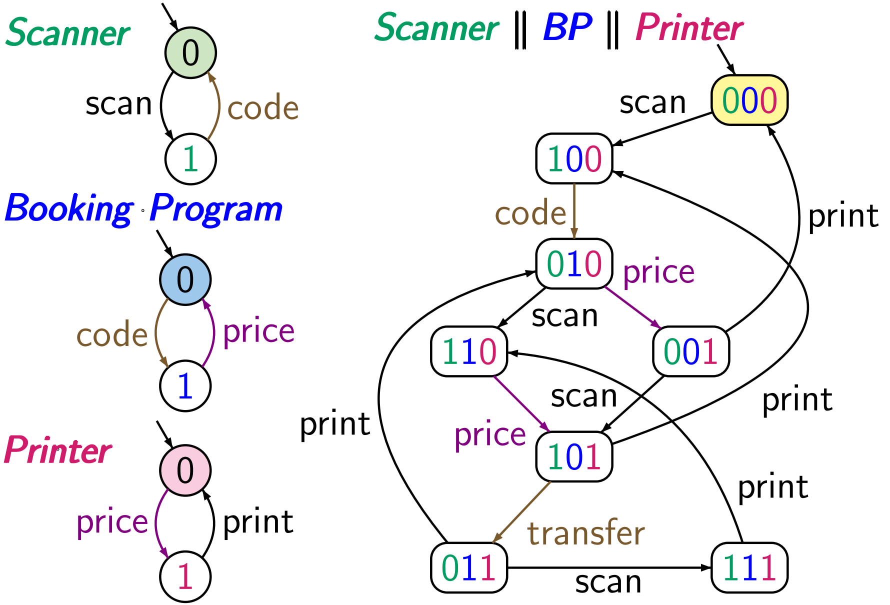
where

$$
\begin{array}{lcl}
Syn_{1,2} & = & Act_1 \cap Act_2 \\
Syn_{1,2,3} & = & (Act_1 \cup Act_2) \cap Act_3 \\
Syn_{1,2,3,4} & = & (Act_1 \cup Act_2 \cup Act_3) \cap Act_4 \\
\vdots & & \vdots
\end{array}
$$

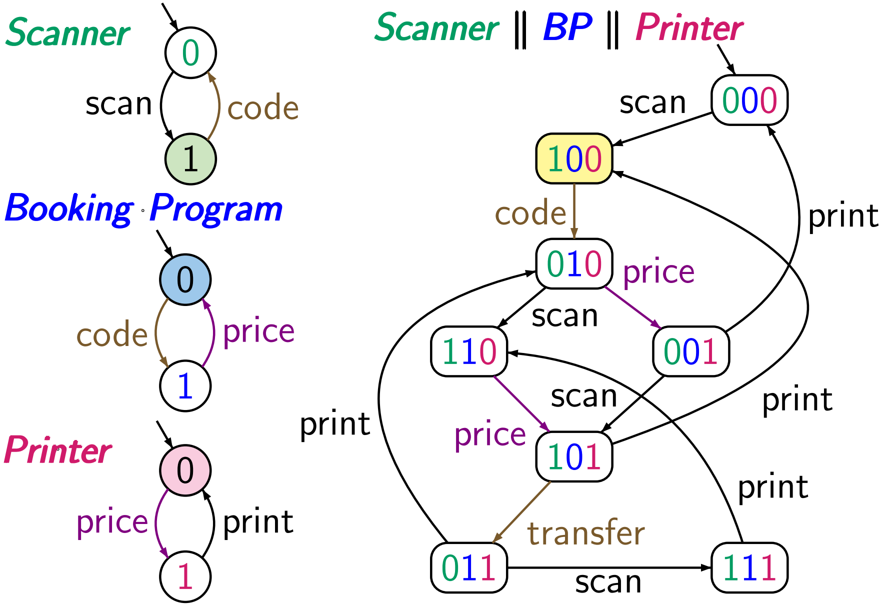
Booking system in supermarket



Booking system in supermarket



Booking system in supermarket



Booking system in supermarket