Bisimulation

→ moyen de faire de l'abstraction

→ lien avec CTL

Implementation relations

- A binary relation on transition systems
  - when does a transition systems correctly implement another?
- Important for system synthesis
  - stepwise refinement of a system specification TS into an "implementation" TS'
- Important for system analysis
  - use the implementation relation as a means for abstraction
  - replace $TS \models \varphi$ by $TS' \models \varphi$ where $|\, TS' \,| \ll |\, TS \,|$ such that:

$$TS \models \varphi \text{ iff } TS' \models \varphi \quad \text{or} \quad TS' \models \varphi \;\Rightarrow\; TS \models \varphi$$

$\Rightarrow$ Focus on state-based bisimulation and simulation

- logical characterization: which logical formulas are preserved by bisimulation?

Comment rendre des ST plus petits ?

ex LTL φ

$ST \stackrel{?}{\models} \varphi$

$s_1 \to s_2 \to \cdots$ chemin de ST

$\to$ mot

$L(s_1)\, L(s_2) \cdots$ 2^{AP}

$ST \to$ minimiser ST'

$L(ST) = L(ST')$

LTL : ST ~ automate non-déterministe !

$\to$ pas de minimisation

Bisimulation equivalence

Let $TS_i = (S_i, Act_i, \rightarrow_i, I_i, AP, L_i)$, i=1, 2, be transition systems.
A **bisimulation** for (TS_1, TS_2) is a **binary relation** $\mathcal{R} \subseteq S_1 \times S_2$ such that:

1. $\forall s_1 \in I_1 \; \exists s_2 \in I_2 .\; (s_1, s_2) \in \mathcal{R}$ **and** $\forall s_2 \in I_2 \; \exists s_1 \in I_1 .\; (s_1, s_2) \in \mathcal{R}$
2. for all states $s_1 \in S_1$, $s_2 \in S_2$ with $(s_1, s_2) \in \mathcal{R}$ it holds:

   2.1 $L_1(s_1) = L_2(s_2)$

   2.2 if $s_1' \in Post(s_1)$ then there exists $s_2' \in Post(s_2)$ with $(s_1', s_2') \in \mathcal{R}$

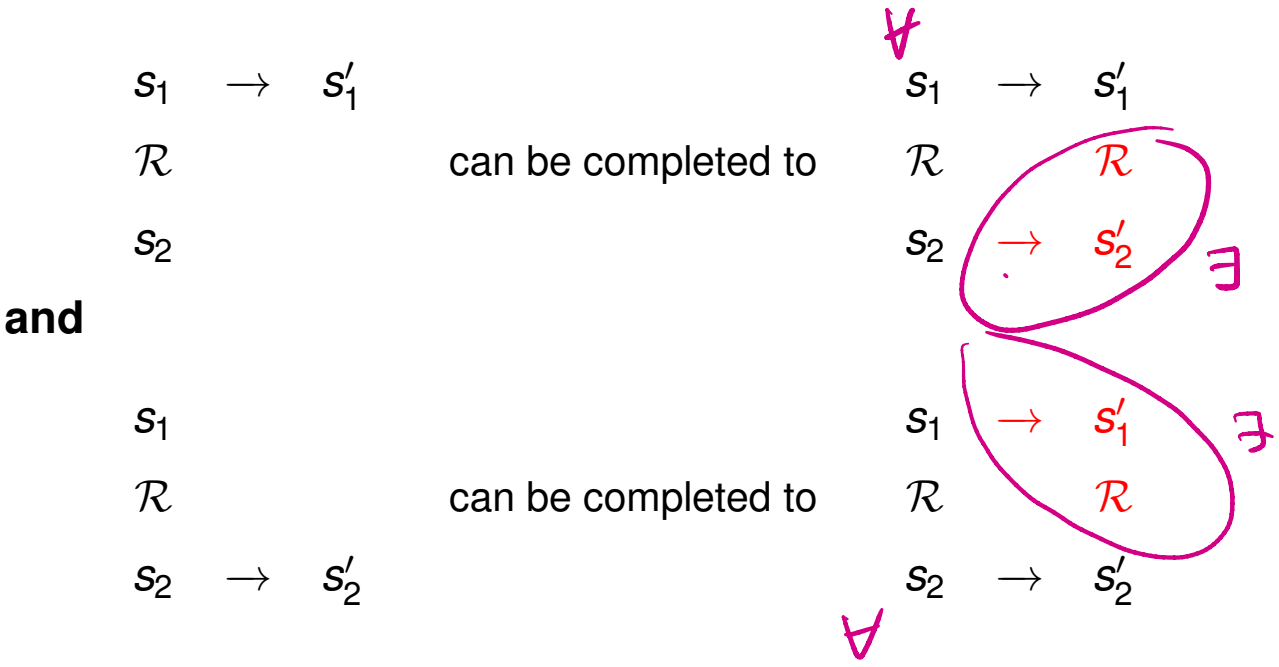
   2.3 if $s_2' \in Post(s_2)$ then there exists $s_1' \in Post(s_1)$ with $(s_1', s_2') \in \mathcal{R}$

bi

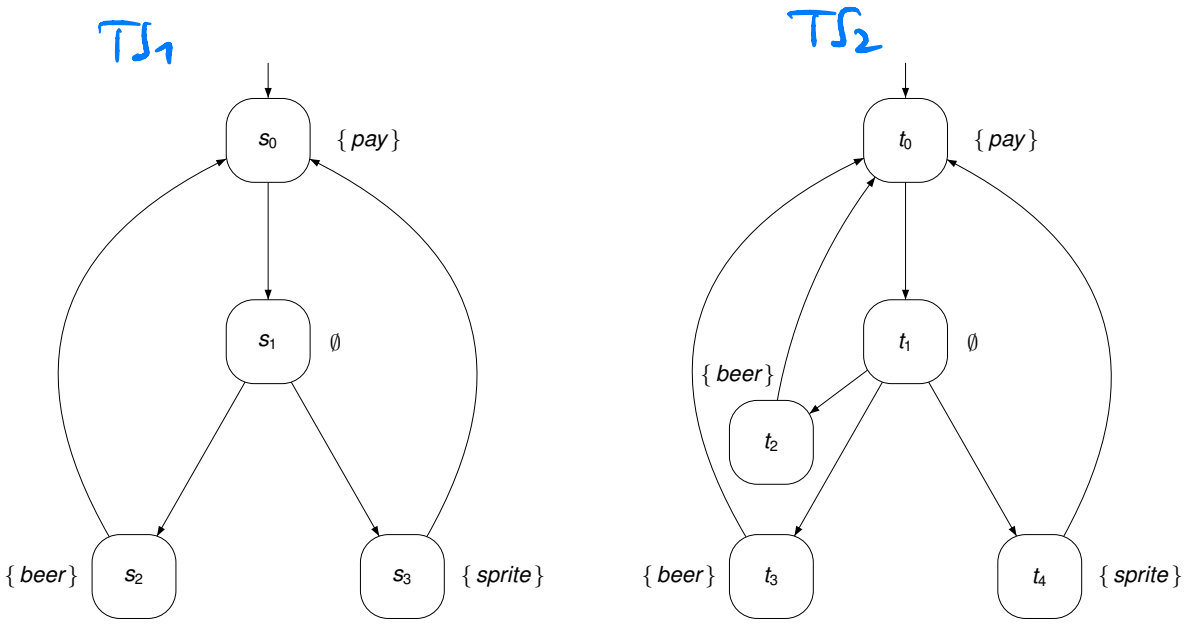
TS_1 and TS_2 are bisimilar, denoted $TS_1 \sim TS_2$, if there exists a bisimulation for (TS_1, TS_2)

Bisimulation equivalence

$\mathcal{R} \subseteq S_1 \times S_2$



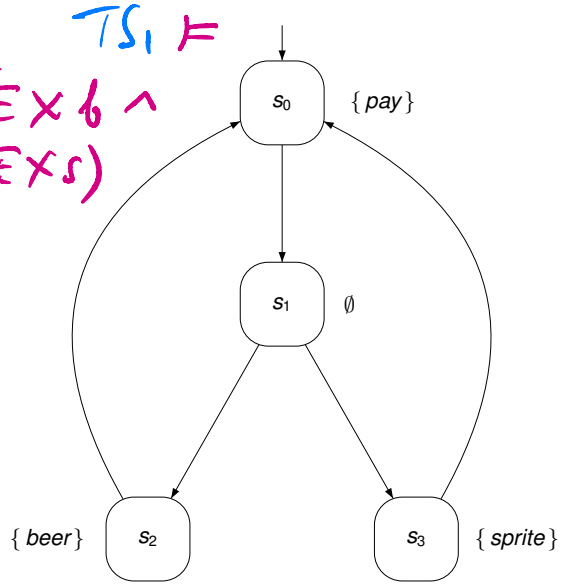
Example (1)

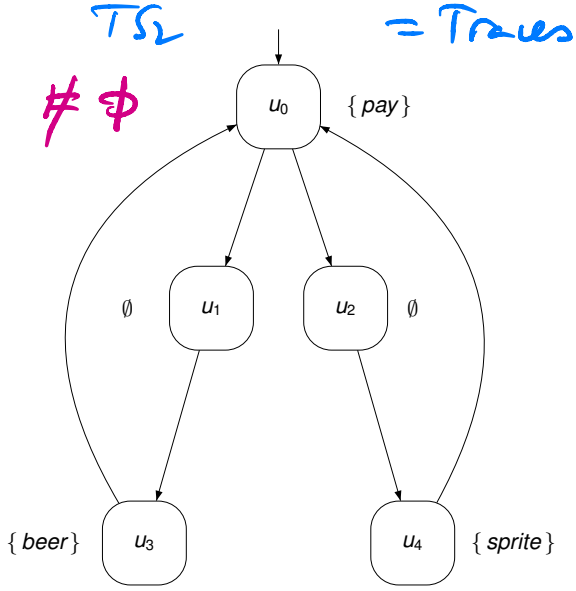


$$\mathcal{R} = \{(s_0, t_0), (s_1, t_1), (s_2, t_2), (s_2, t_3), (s_3, t_4)\}$$

is a bisimulation for (TS_1, TS_2) where $AP = \{ pay, beer, sprite \}$

Example (2)





$TS_1 \not\sim TS_3$ for $AP = \{ pay, beer, sprite \}$

But: $\{ (s_0, u_0), (s_1, u_1), (s_1, u_2), (s_2, u_3), (s_2, u_4), (s_3, u_3), (s_3, u_4) \}$

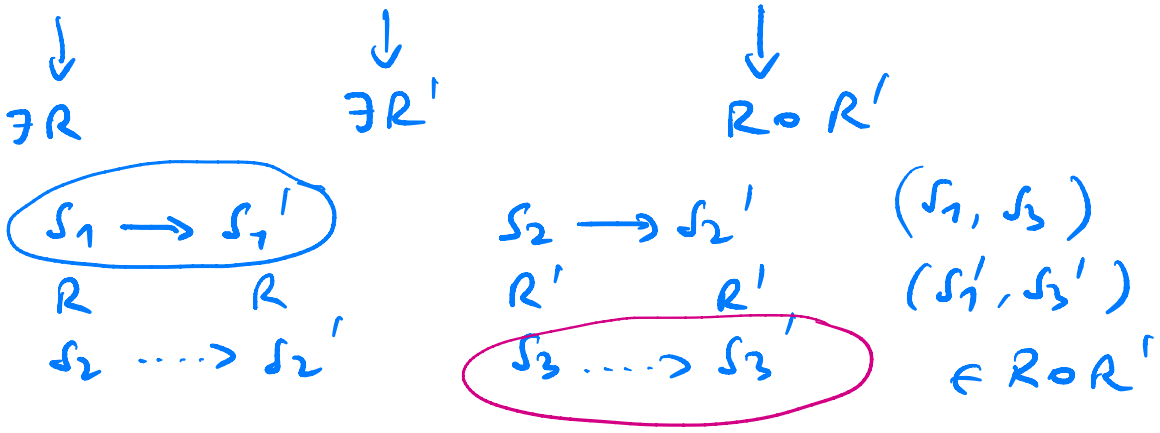
is a bisimulation for (TS_1, TS_3) for $AP = \{ pay, drink \}$

$\sim$ is an equivalence

$TS_1 \sim TS_2$ iff
exists R bisimulation
$R \subseteq S_1 \times S_2$

For any transition systems TS, TS_1, TS_2 and TS_3 over AP:

- $TS \sim TS$ (reflexivity) $\rightarrow$ $id_{|S}$
- $TS_1 \sim TS_2$ implies $TS_2 \sim TS_1$ (symmetry) $\rightarrow$ diff.
- $TS_1 \sim TS_2$ and $TS_2 \sim TS_3$ implies $TS_1 \sim TS_3$ (transitivity)



Bisimulation on paths

Whenever we have:

$$\exists \quad s_0 \to s_1 \to s_2 \to s_3 \to s_4 \ldots\ldots$$
$$\mathcal{R}$$
$$\exists \quad t_0$$

$L(s_i) = L(t_i)$, for $i \geq 0$

this can be completed to

$$\begin{array}{ccccccccc} s_0 & \to & s_1 & \to & s_2 & \to & s_3 & \to & s_4 \ldots\ldots \\ \mathcal{R} & & \mathcal{R} & & \mathcal{R} & & \mathcal{R} & & \mathcal{R} \\ t_0 & \to & t_1 & \to & t_2 & \to & t_3 & \to & t_4 \ldots\ldots \end{array}$$

proof: by induction on index i of state s_i

Bisimulation vs. trace equivalence

$Traces(sT) = \{ L(s_0)L(s_1)\ldots : s_0 \to s_1 \to \ldots , s_0 \in I \}$

$TS_1 \sim TS_2$ implies $Traces(TS_1) = Traces(TS_2)$

bisimilar transition systems thus satisfy the same LT properties!

$\exists\ sT_1, sT_2$ tq. $Traces(sT_1) = Traces(sT_2)$,
mais $sT_1 \not\sim sT_2$

(voir beer/sprite example)

Bisimulation on states

$\mathcal{R} \subseteq S \times S$ is a **bisimulation** on TS if for any $(s_1, s_2) \in \mathcal{R}$:

- $L(s_1) = L(s_2)$
- if $s_1' \in Post(s_1)$ then there exists an $s_2' \in Post(s_2)$ with $(s_1', s_2') \in \mathcal{R}$
- if $s_2' \in Post(s_2)$ then there exists an $s_1' \in Post(s_1)$ with $(s_1', s_2') \in \mathcal{R}$

s_1 and s_2 are **bisimilar**, $s_1 \sim_{TS} s_2$,
if $(s_1, s_2) \in \mathcal{R}$ for some bisimulation $\mathcal{R}$ for TS

$s_1 \sim_{TS} s_2$ if and only if $TS_{s_1} \sim TS_{s_2}$

TS_s is the transition system obtained from TS by declaring s as the initial state.

Coarsest bisimulation

$R_1, R_2 \subseteq S \times S$ bisimulations

$\Rightarrow$ $R_1 \cup R_2$ est bisimulation

$s_1 \longrightarrow s_1'$

$R_1 \cup R_2$ $\quad$ $R_1 \cup R_2$

$s_2 \dashrightarrow s_2'$

$\sim_{TS}$ is an equivalence and the coarsest bisimulation for *TS*

$\sim_{TS}$ rel. d'équivalence :

1) $s \sim_{TS} s$ (id est bisim.)

2) $s \sim_{TS} s' \Leftrightarrow \exists R$ bis. $(s, s') \in R \Rightarrow (s', s) \in R \Rightarrow s' \sim_{TS} s$

3) $s \sim_{TS} s' \sim_{TS} s''$

$(s, s') \in R \quad (s', s'') \in R' \Rightarrow (s, s'') \in R \cup R'$

- $\sim_{TS}$ est bisimulation

$$s \longrightarrow s_1$$

$\sim_{TS}$ $\quad$ $\sim_{TS}$

$$s' \dashrightarrow s_1'$$

$s \sim_{TS} s' \overset{def}{\iff} \exists R$ bis. $(s, s') \in R$

$\Rightarrow \exists\, s_1'$, $(s_1, s_1') \in R$

- $\sim_{TS}$ la plus grossière* bisimulation

R bisim.

$(s_1, s_2) \in R \overset{def}{\Rightarrow} s_1 \sim_{TS} s_2$

[* coarsest bisimulation, cad :

$\forall R$ bisimulation :

$(s_1, s_2) \in R \Rightarrow s_1 \sim_{TS} s_2$]

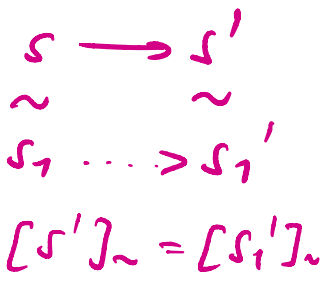
Quotient transition system

For $TS = (S, Act, \to, I, AP, L)$ and bisimulation $\sim_{TS} \subseteq S \times S$ on TS let

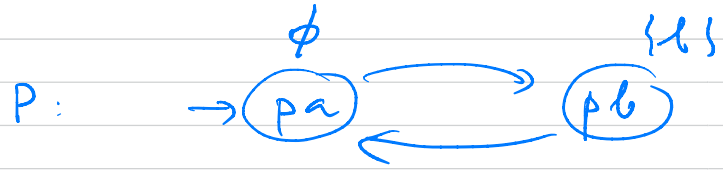
$TS/\sim_{TS} = (S', \{\tau\}, \to', I', AP, L')$ be the **quotient** of TS under $\sim_{TS}$

where

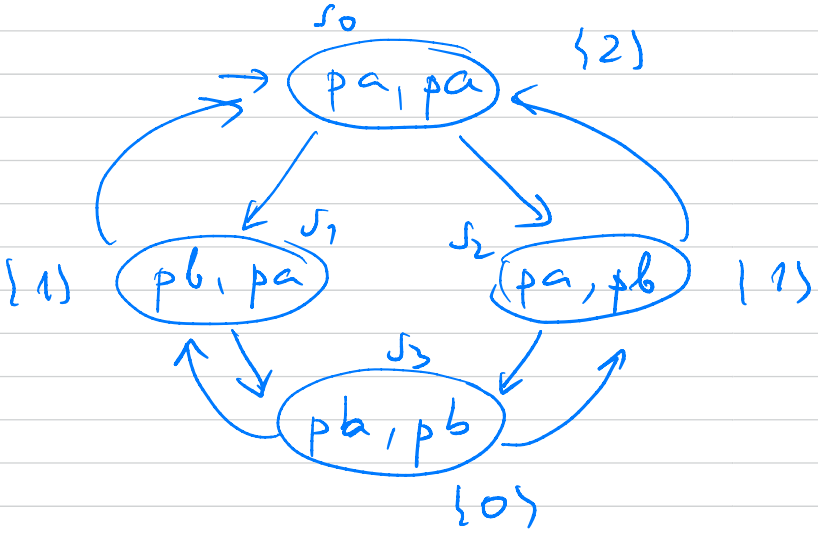
- $S' = S/\sim_{TS} = \{ [s]_\sim \mid s \in S \}$ with $[s]_\sim = \{ s' \in S \mid s \sim_{TS} s' \}$
- $\to'$ is defined by: $$\frac{s \xrightarrow{\alpha} s'}{[s]_\sim \xrightarrow{\tau}{}' [s']_\sim}$$
- $I' = \{ [s]_\sim \mid s \in I \}$
- $L'([s]_\sim) = L(s)$

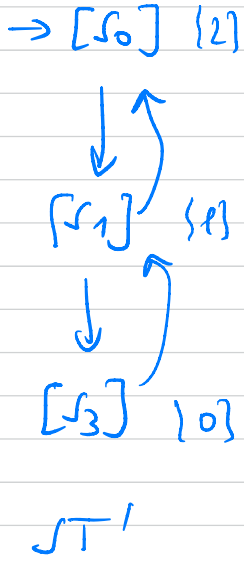


Ex $\underbrace{P \| P \| \ldots \| P}_{n}$



$n = 2$





ST'

$s_1 \sim_{ST} s_2$

$\text{taille}(ST) = 2^n$

$\text{taille}(ST') = n$

The Bakery algorithm

$$
P_1 :: \left[\begin{array}{l}
\textbf{loop forever do} \\
\quad \left[\begin{array}{ll}
 & \textbf{noncritical} \\
n_1 : & y_1 := y_2 + 1 \\
w_1 : & \textbf{await } (y_2 = 0 \vee y_1 < y_2) \\
c_1 : & \textbf{critical} \\
 & y_1 := 0
\end{array}\right]
\end{array}\right]
\;\|\;
P_2 :: \left[\begin{array}{l}
\textbf{loop forever do} \\
\quad \left[\begin{array}{ll}
 & \textbf{noncritical} \\
n_1 : & y_2 := y_1 + 1 \\
w_1 : & \textbf{await } (y_1 = 0 \vee y_2 < y_1) \\
c_1 : & \textbf{critical} \\
 & y_2 := 0
\end{array}\right]
\end{array}\right]
$$

Protocole sans deadlock, équitable

↓

$0 < y_1 = y_2$ pas possible

équité : si P_1 veut rentrer, P_2 ne peut pas rentrer 2 fois de suite

Example path fragment

process P_1	process P_2	y_1	y_2	effect
n_1	n_2	0	0	P_1 requests access to critical section
w_1	n_2	1	0	P_2 requests access to critical section
w_1	w_2	1	2	P_1 enters the critical section
c_1	w_2	1	2	P_1 leaves the critical section
n_1	w_2	0	2	P_1 requests access to critical section
w_1	w_2	3	2	P_2 enters the critical section
w_1	c_2	3	2	P_2 leaves the critical section
w_1	n_2	3	0	P_2 requests access to critical section
w_1	w_2	3	4	P_1 enters the critical section
...	...	..	..	...

Data abstraction

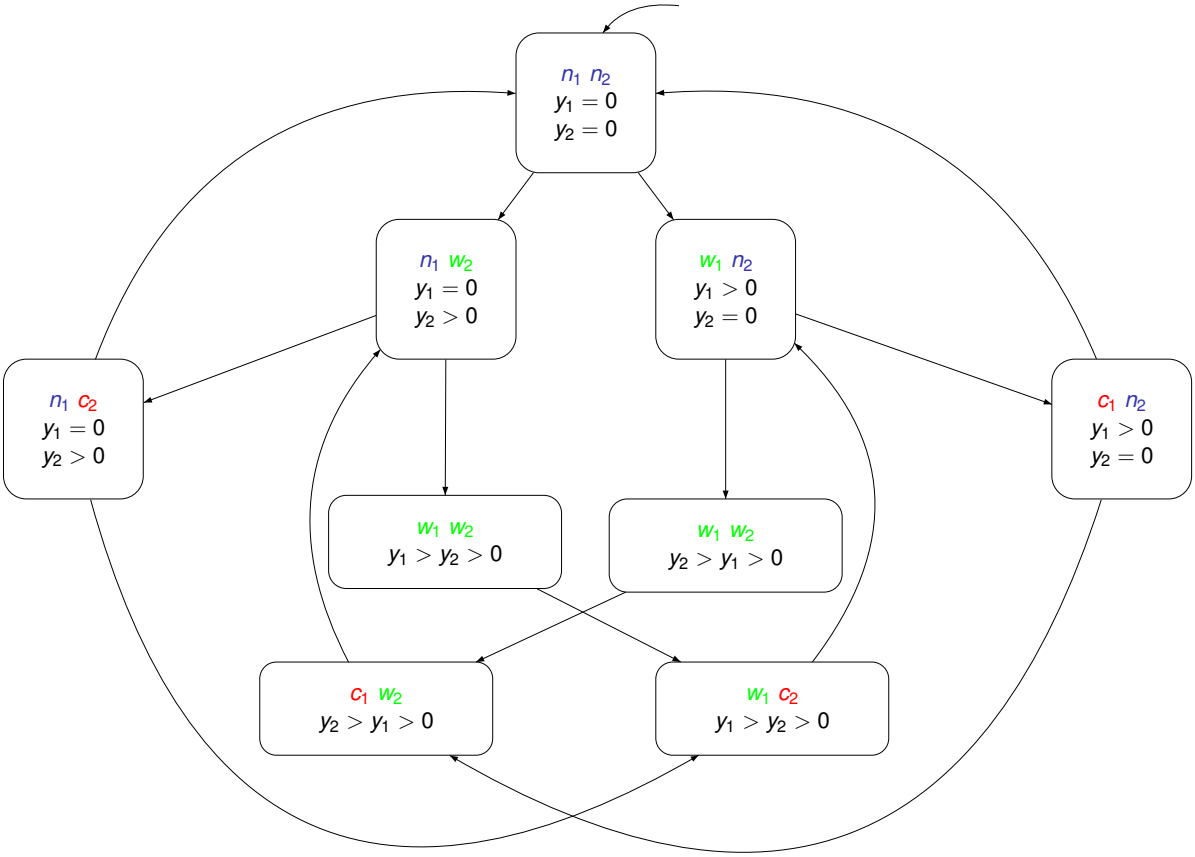
Function f maps a reachable state of TS_{Bak} onto an abstract one in TS^{abs}_{Bak}
Let $s = \langle \ell_1, \ell_2, y_1 = b_1, y_2 = b_2 \rangle$ be a state of TS_{Bak} with
$\ell_i \in \{ n_i, w_i, c_i \}$ and $b_i \in \mathbb{N}$
Then:

$$f(s) = \begin{cases} \langle \ell_1, \ell_2, y_1 = 0, y_2 = 0 \rangle & \text{if } b_1 = b_2 = 0 \\ \langle \ell_1, \ell_2, y_1 = 0, y_2 > 0 \rangle & \text{if } b_1 = 0 \text{ and } b_2 > 0 \\ \langle \ell_1, \ell_2, y_1 > 0, y_2 = 0 \rangle & \text{if } b_1 > 0 \text{ and } b_2 = 0 \\ \langle \ell_1, \ell_2, y_1 > y_2 > 0 \rangle & \text{if } b_1 > b_2 > 0 \\ \langle \ell_1, \ell_2, y_2 > y_1 > 0 \rangle & \text{if } b_2 > b_1 > 0 \end{cases}$$

$\mathcal{R} = \{ (s, f(s)) \mid s \in S \}$ is a bisimulation for $(TS_{Bak}, TS^{abs}_{Bak})$ ✓

for any subset of $AP = \{ noncrit_i, wait_i, crit_i \mid i = 1, 2 \}$

Bisimulation quotient



$$TS_{Bak}^{abs} = TS_{Bak} / \sim \quad \text{for} \quad AP = \{ crit_1, crit_2 \}$$

Remarks

- In this example, data abstraction yields a bisimulation relation
  - (typically, only a simulation relation is obtained, more later)
- $TS^{abs}_{Bak} \models \varphi$ with, e.g.,:
  - $\Box(\neg crit_1 \ \vee \ \neg crit_2)$ and
    $(\Box\Diamond wait_1 \ \Rightarrow \ \Box\Diamond crit_1) \ \wedge \ (\Box\Diamond wait_2 \ \Rightarrow \ \Box\Diamond crit_2)$
- Since $TS^{abs}_{Bak} \sim TS_{Bak}$, it follows $TS_{Bak} \models \varphi$
- Note: $Traces(TS^{abs}_{Bak}) = Traces(TS_{Bak})$