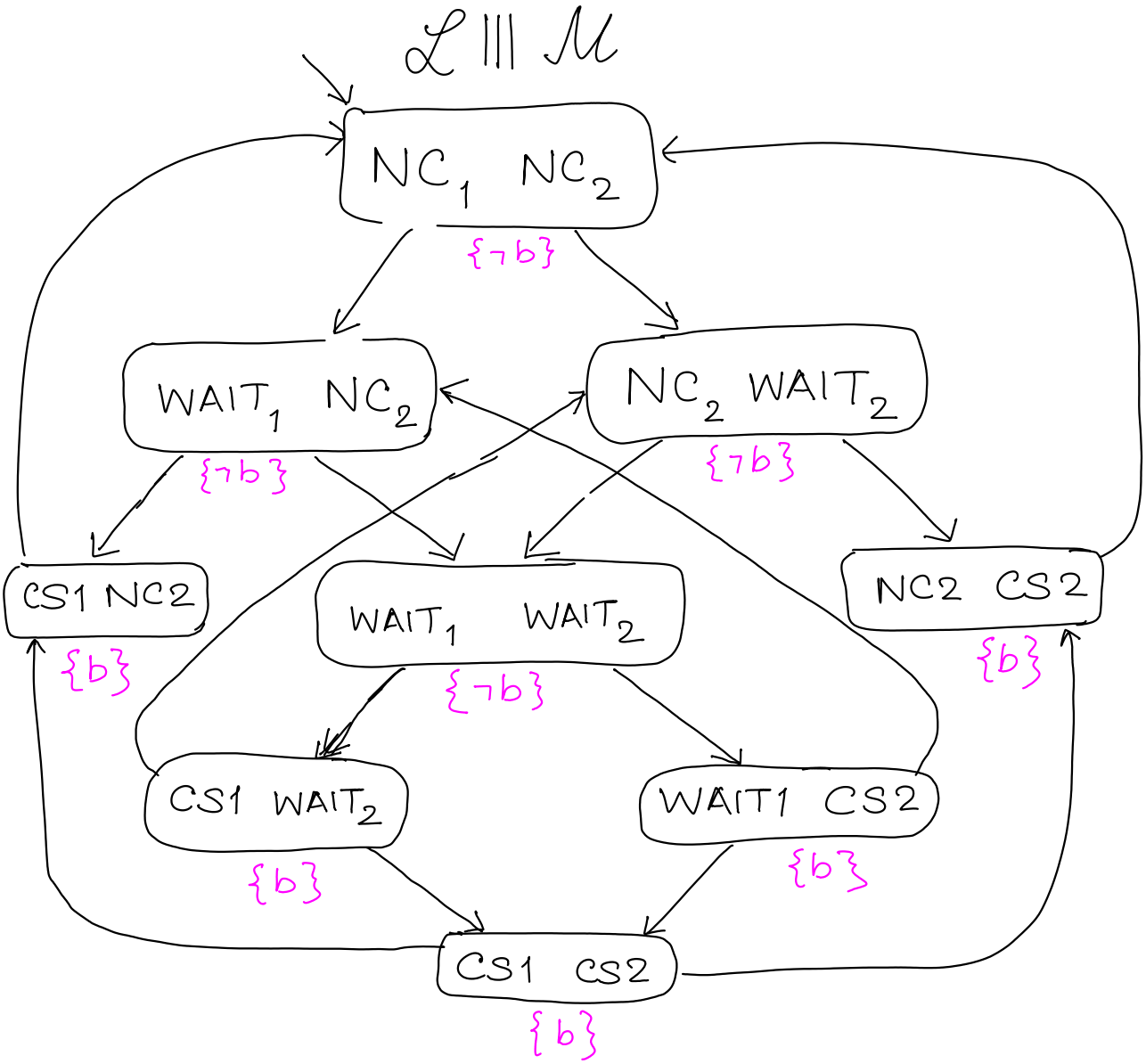
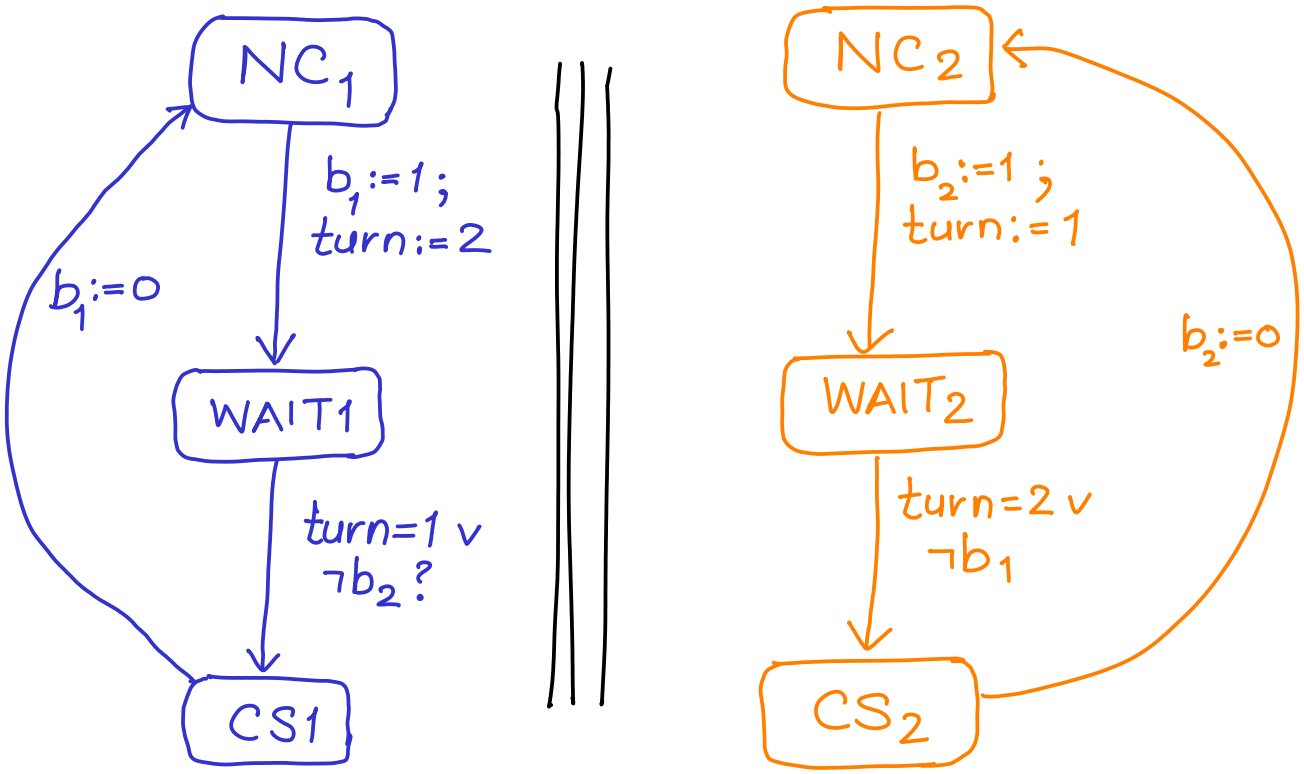


The above diagram shows the previously defined mutual exclusion algorithm as a decorated TS rather than as a LTS. In both cases, it is clear only from the state which process may have set or reset the boolean variable

Peterson's mutual Exclusion algorithm



In this case there are 3 shared variables

b_1	written by P_1	read by P_2	}	correspond to requests by each process for access to resource
b_2	written by P_2	read by P_1	}	
turn	written and read by both			

Questions. Q1 is given after Q2

2. Can Peterson's algorithm be extended to 3 (or more) processes P_0, P_1 and P_2 by simply changing the assignment to turn in P_i by $turn := (i+1) \bmod 3$ and the check to $turn = i \vee \neg b_{turn}$

a) Does this extension guarantee mutual exclusion?

b) Does it guarantee freedom from deadlock?

c) Does it guarantee that every process that requests access will eventually be granted access

1. The two process version of the algorithm has two atomic actions that may be regarded as compound viz

$b_i := 1 \; ; \; turn := \bar{i}$

$turn = i \; \vee \; \neg b_{\bar{i}}$

Suppose that these actions are not atomic and the only atomic actions are

(i) simple assignments to variables e.g.

$b_i := 1$ $\quad$ $b_i := 0$ } writing without any evaluation

(ii) simple condition evaluation e.g.

$turn = i$ $\quad$ $\neg b_{\bar{i}}$ } simply reading variables and comparing.

How do these conditions affect the algorithm?

Synchronization & Synchronized Products

The interleaving operator $|||$ may be easily seen to be

(i) commutative (upto isomorphism)
(ii) associative (upto isomorphism)

Def Let $\mathcal{L}_i = \langle S_i, A_i, \longrightarrow_i, I_i, AP_i, D_i \rangle$ for $i = 1, \ldots, n$ be a collection of DLTSs. Let

$$\varphi \subseteq Sync \subseteq \bigtimes_{i=1}^{n} (A_i \cup \{\_\}) - \{(\_, \ldots, \_)\}$$

where '$\_$' denotes an empty action. Then

$$\|_{Sync}^{n}_{i=1} \mathcal{L}_i = \langle S, A, \longrightarrow, I, AP, D \rangle$$

where
$$S = \bigtimes_{i=1}^{n} S_i \qquad D((s_1, \ldots, s_n)) = \bigcup_{i=1}^{n} D_i(s_i)$$
$$A = \bigtimes_{i=1}^{n} A_i \qquad AP = \bigcup_{i=1}^{n} AP_i$$
$$I = \bigtimes_{i=1}^{n} I_i$$

and $s \xrightarrow{\vec{a}} t$ for any $s, t \in S$ provided

$s = (s_1, \ldots, s_n)$, $\vec{a} = (a_1, \ldots, a_n)$

$t = (t_1, \ldots, t_n)$, $\forall i : 1 \le i \le n : a_i \in A_i \cup \{\_\}$,

$s_i \xrightarrow{a_i} t_i \vee (a_i = \_ \wedge s_i = t_i)$