ch 6 continuation: verification

Alg 1. Reachability

Alg 2 coreachability$(\Sigma, \delta, Q_m, Q_x)$

let $k := 0$, $Q_0 = Q_m \setminus Q_x$

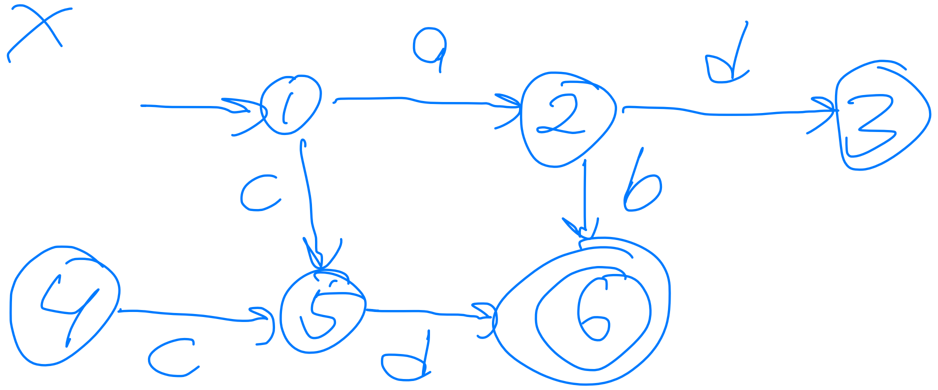
repeat

$k := k+1$

$Q_k := Q_{k-1} \cup \{q \mid \delta(q, \sigma) \in Q_{k-1}, \sigma \in \Sigma\} \setminus Q_x$

until $Q_{k-1} = Q_k$, return Q_k

Ex



Alg 1 Reachability

$Q_0 = \{1\}$

$Q_1 = \{1\} \cup \{2, 5\} = \{1, 2, 5\}$

$Q_2 = \{1, 2, 3, 5, 6\}$ $\quad Q_3 = Q_2 \stackrel{\text{def}}{=} Q_r$

Alg 2 coreachability

$Q_0 = \{6\}$

$Q_1 = \{2, 5, 6\}$

$Q_2 = \{1, 2, 4, 5, 6\}$

$Q_3 = Q_2 \stackrel{\text{def}}{=} Q_c$

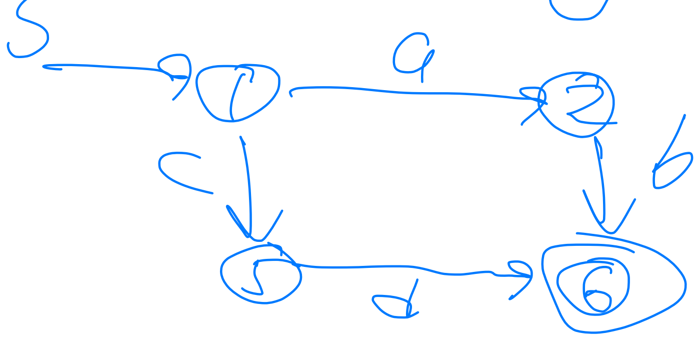
Trim automation includes only the states that are both reachable and coreachable.

Trim states are

$Q_r \cap Q_c = \{1, 2, 5, 6\}$

Nonblocking supervisor

S



verification based on reachability analysis

1) Is a specific state q reachable? $q \in Q_r$?

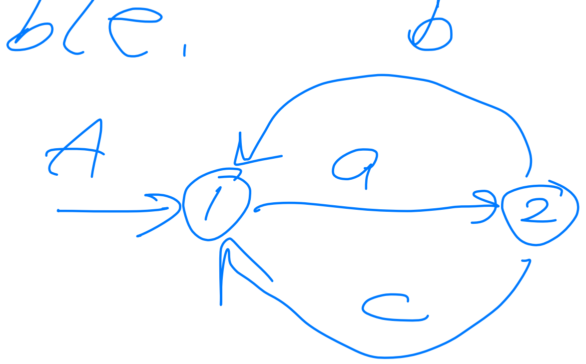
2) Forbidden states Q_f are given. $Q_f \cap Q_r = \emptyset$?

3) Are event sequences given by a specification Sp possible in an automaton A?

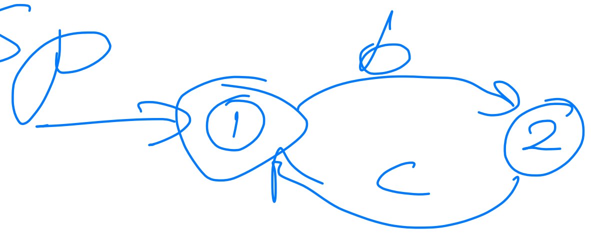
Generate AllSp. 1) If AllSp is nonblocking $\Rightarrow$ Sp is a part of the behaviour of A

2) If AllSp is blocking, the Sp event sequences are not possible.

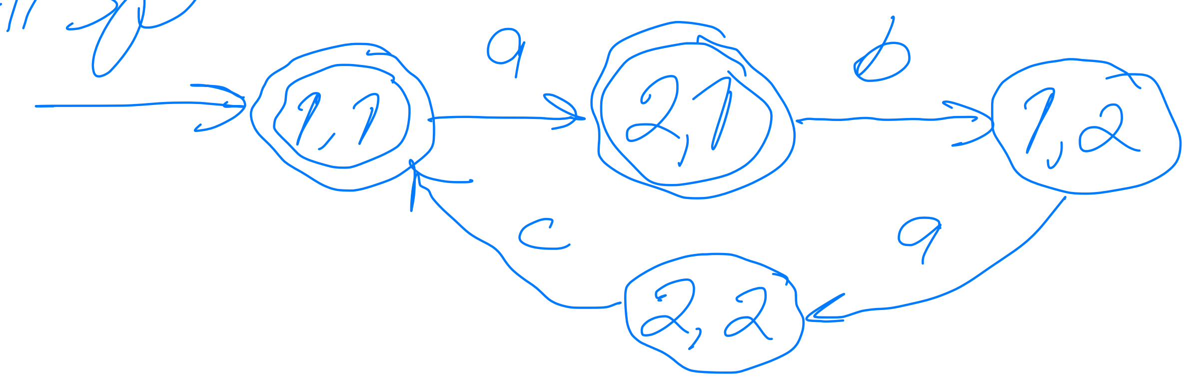
Ex A



a) Sp

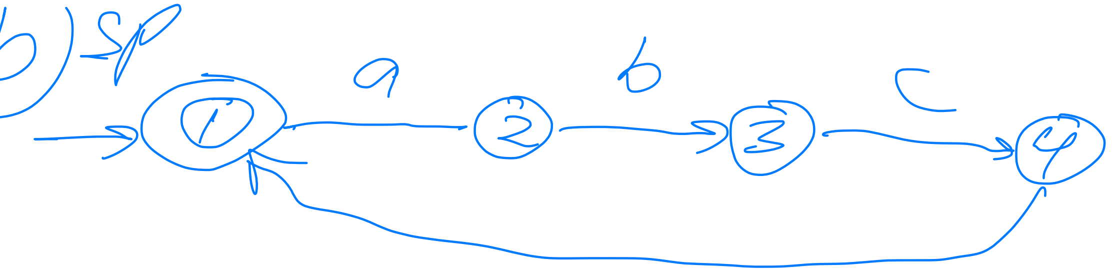


AllSp

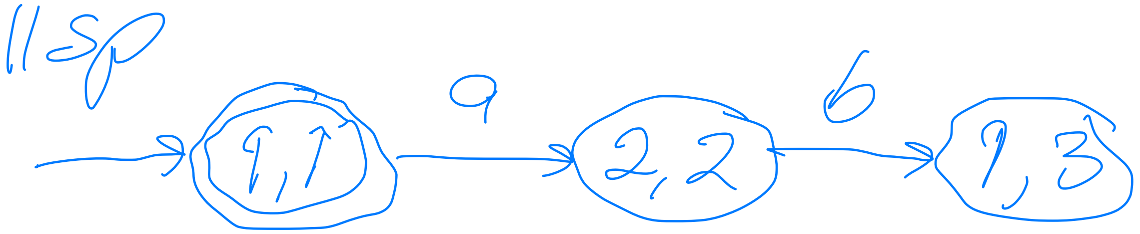


Sp is possible in A

b) Sp



AllSp



blocking state

$\Rightarrow$ Sp is not possible in A

Controllability

Plant $P = \langle Q^P, \Sigma^P, \delta^P, q_i^P \rangle$

supervisor $S =$
$\langle Q^S, \Sigma^S, \delta^S, q_i^S, Q_m^S, Q_x^S \rangle$

$$\Sigma^S \subseteq \Sigma^P$$

closed loop system: $P \| S$

$$\Sigma^P = \Sigma_c \dot{\cup} \Sigma_u, \quad \Sigma_c \cap \Sigma_u = \emptyset$$

Σ_c = set of controllable events which the supervisor can accept or prevent.

Σ_u = set of uncontrollable events which S cannot prevent.

Controllable supervisor S

$P \| S \quad Q_r^{P\|S} \subseteq Q^P \times Q^S$

$Q_r^{P\|S} = \text{Reachability}(\Sigma^{P\|S}, \delta^{P\|S}, q_i^{P\|S})$

Controllability is related to the uncontrollable events in P that are also a part of the supervisor alphabet Σ^S

$$\sigma_u \in \Sigma_u \cap \Sigma^S$$

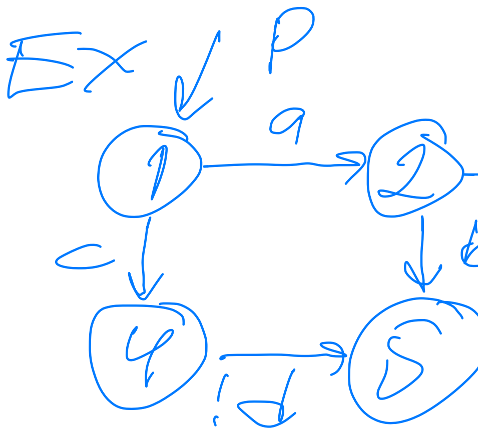
Definition of controllability

A supervisor S is controllable with respect to (wrt) a plant P and a set of uncontrollable events $\Sigma_u \subseteq \Sigma^P$ if, for all reachable states $\langle p, q \rangle \in Q_r^{P\|S}$ in the closed loop system $P\|S$, and for all uncontrollable events $\sigma_u \in \Sigma_u \cap \Sigma^S$:

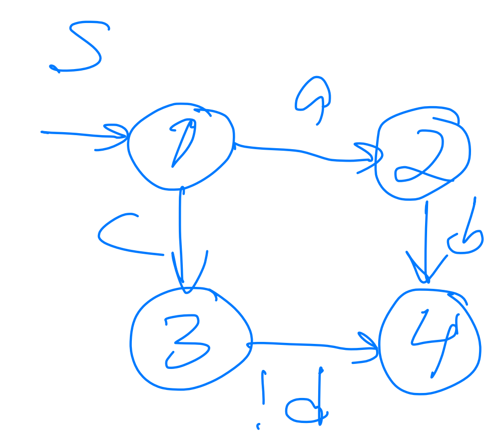
$$\delta^P(p, \sigma_u) \in Q^P \Rightarrow \delta^S(q, \sigma_u) \in Q^S$$

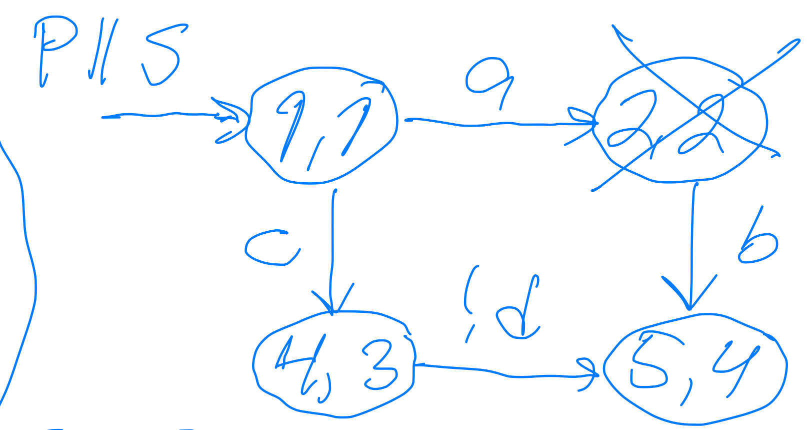
This means that S must be able to follow P when P executes an uncontrollable event $\sigma_u \in \Sigma_u \cap \Sigma^S$ in the closed loop system.

Ex



$\Sigma_u = \{d\}$





Note that d is uncontrollable

$(2,2)$ is an uncontrollable state since S cannot execute d, which P is able to do.

Uncontrollable states

$Q_{uc} = \{ \langle p,q \rangle \in Q_r^{P\|S} \mid$

$\exists \sigma_u \in \Sigma_u \cap \Sigma^S \wedge \delta^P(p,\sigma_u)$ exists $\wedge$ $\delta^S(q,\sigma_u)$ does not exist $\}$

7. Supervisor synthesis

Plant $P = P_1 \| P_2 \| \ldots \| P_n$

specification $Sp =$

$= Sp_1 \| Sp_2 \| \ldots \| Sp_m$

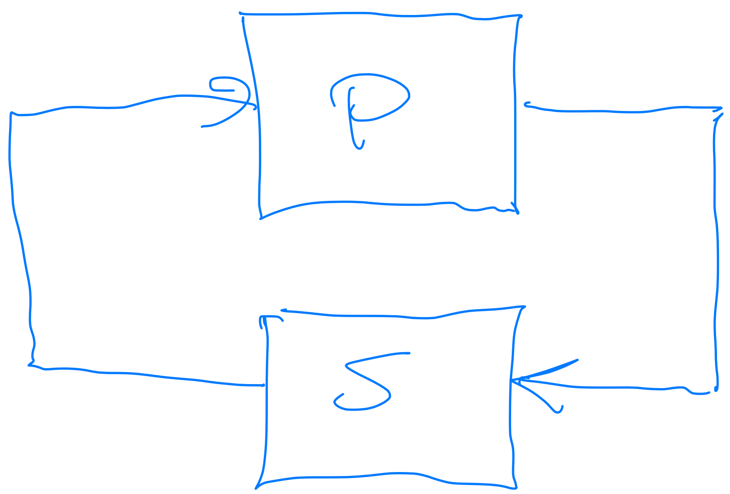
Total specification $S_0 = P \| Sp$ is also a first candidate for a possible supervisor S.

If S_0 has any blocking or uncontrollable states, such states are removed from S_0 to get a controllable and nonblocking supervisor $S \leq S_0$ (S is a subautomaton of S_0).

This supervisor S is also a model of the closed loop system, i.e. $S = P \| S$ when $\Sigma^S \subseteq \Sigma^P$

$\therefore$ we need to define what we mean by subautomaton ($\leq$) and equality ($=$)

closed loop system



closed loop model

P || S

Sub-automaton

$A = \langle Q^A, \Sigma^A, \delta^A, q_i^A, Q_m^A \rangle$

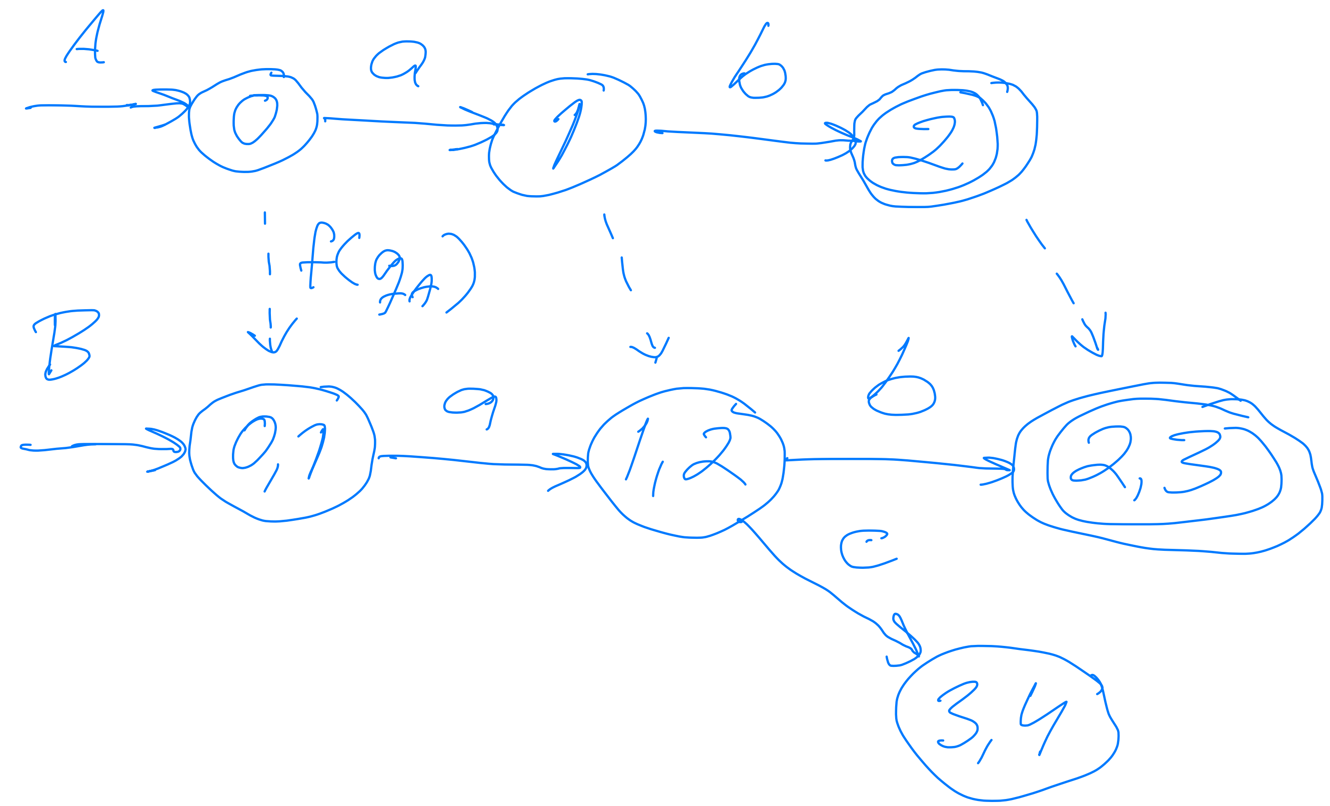
$B = \langle Q^B, \Sigma^B, \delta^B, q_i^B, Q_m^B \rangle$

A is a subautomaton of B, written $A \leq B$ if

$Q^A \subseteq Q^B$, $\Sigma^A = \Sigma^B$, $\delta^A \subseteq \delta^B$

$q_i^A = q_i^B$, $Q_m^A \subseteq Q_m^B$

The state names in Q^A can be different than corresponding state names in Q^B, if for all states $q_A \in Q^A$ there is a one-to-one mapping (bijective function) such that $q_B = f(q_A)$



$A \leq B$ since $f(0) = (0,1)$

$f(1) = (1,2)$

$f(2) = (2,3)$

Equivalence between automata

1) Language equivalence

$\Sigma^A = \Sigma^B$

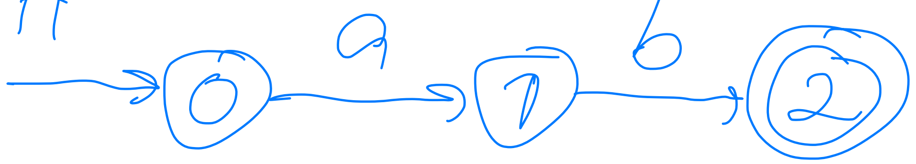
$L(A) = L(B)$ $\quad L_m(A) = L_m(B)$

2) Structural equivalence

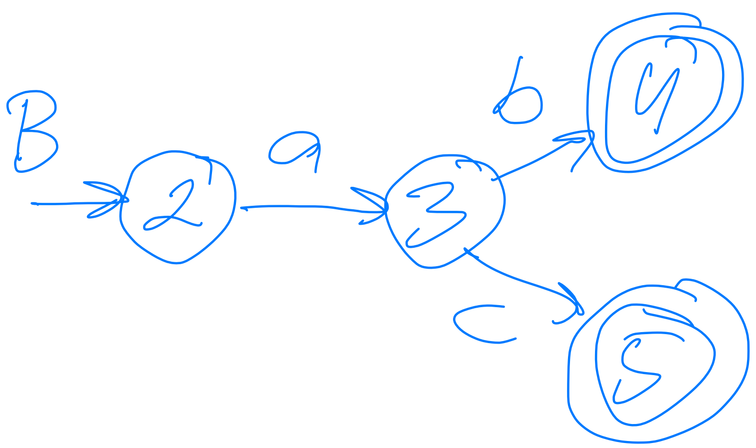
if $A \preceq B$ and $B \preceq A$

$\Rightarrow A = B$

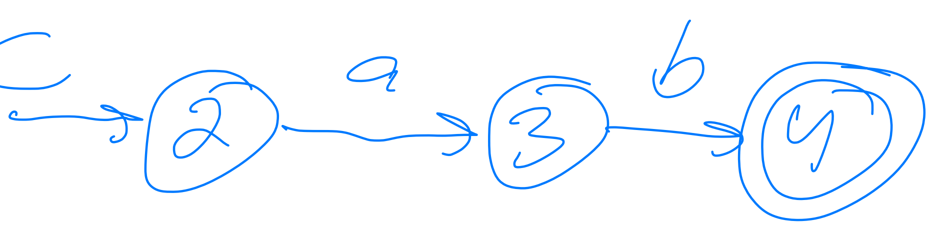
A



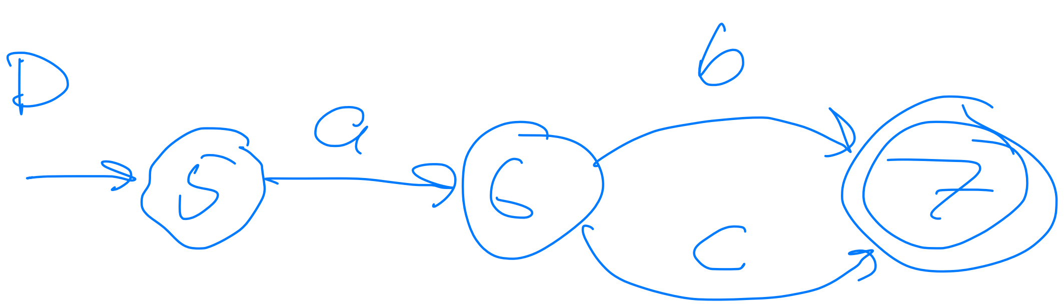
B



C



D



$A = C$ $\quad A \neq B$

$A \preceq B$ when $\Sigma^A = \{a, b, c\}$

$L(B) = \overline{a(b+c)} = L(D)$

$L_m(B) = a(b+c) = L_m(D)$

$\therefore$ B and D are language equivalent but not structurally equivalent $B \neq D$.

Refinement

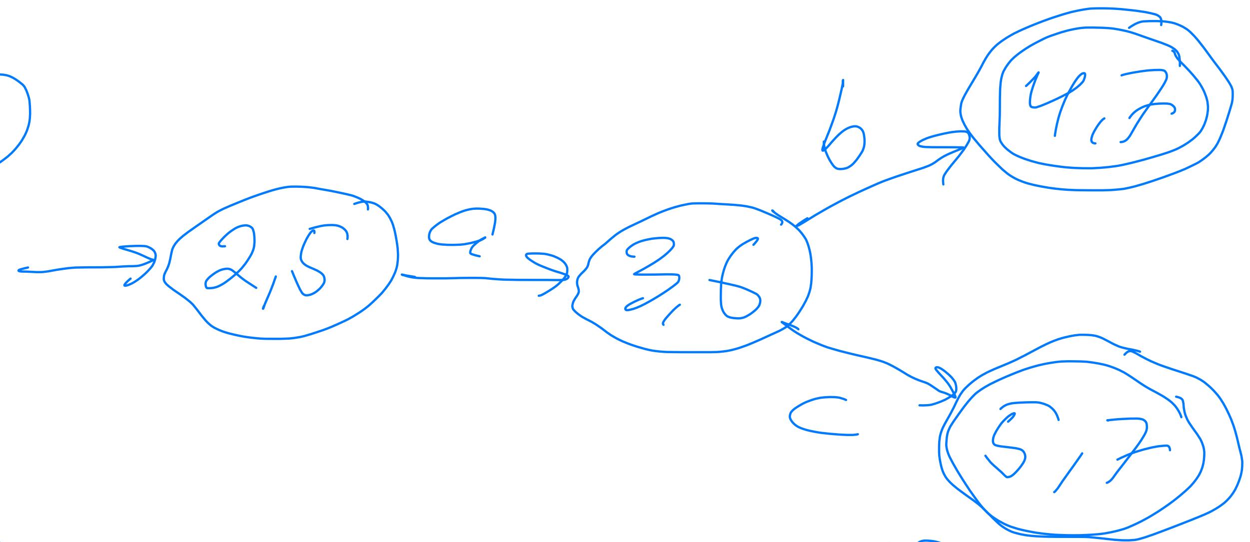
A refines B if $A \| B = A$

$A \preceq B \Rightarrow A \| B = A$

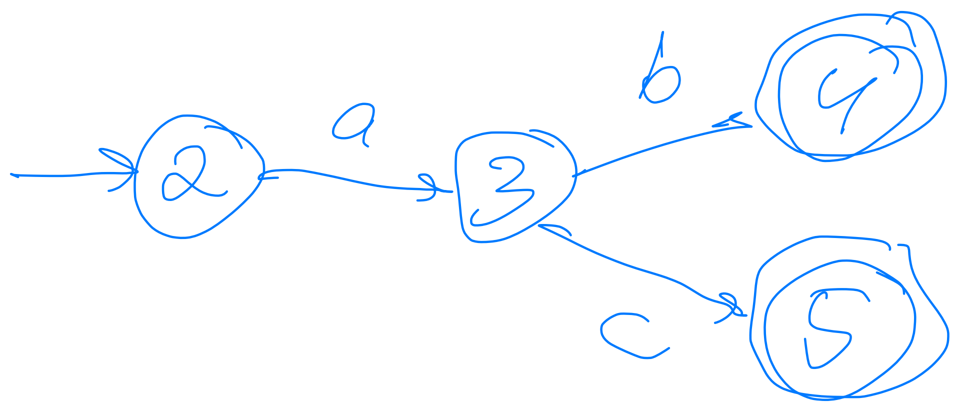
But refinement does not guarantee subautomaton.

B refines D since

$B \| D$



B



$\therefore \quad B \| D = B$

Lemma 7.1 If a supervisor S is constructed such that S is a subautomaton of $S_0 = P \| S_P$, i.e. $S \le S_0$, then S refines P, i.e. $P \| S = S$

Proof: $S \le S_0 \Rightarrow S = S_0 \| S$ i.e. S refines S_0.

Now $S_0 = P \| S_P$

$S_0 \| P = (P \| S_P) \| P = P \| (P \| S_P)$
$= (P \| P) \| S_P = P \| S_P = S_0 \Rightarrow$
S_0 refines P,
since also S refines S_0
S refines S_0 refines $P \Rightarrow$
S refines P $\quad S = P \| S$