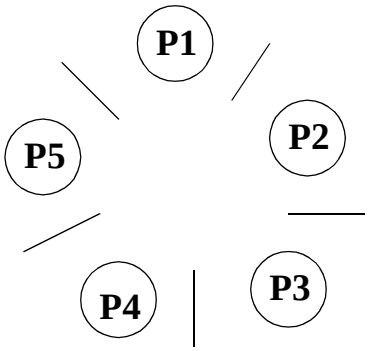
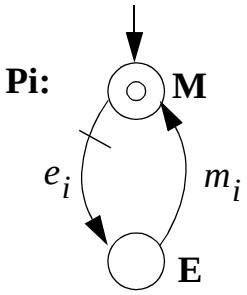
Control of Discrete Event Systems - Exercises

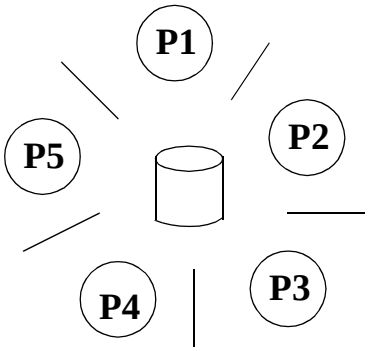
Systems:

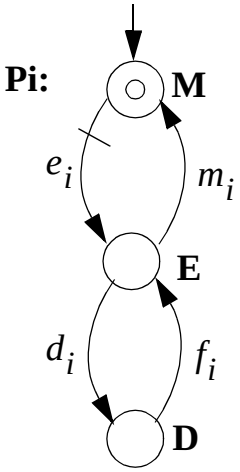
1. Dinner of Philosophers



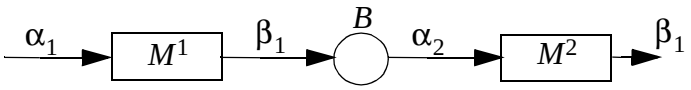


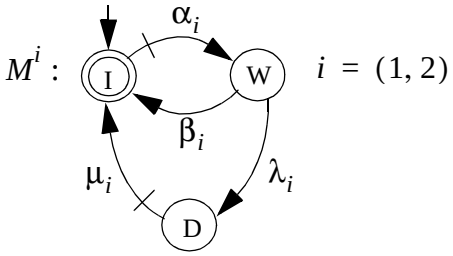
2. Dinner of Philosophers with "sake"





3. Small Factory

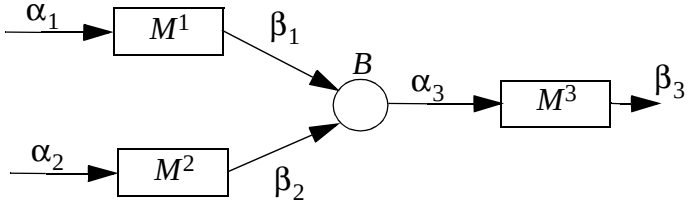


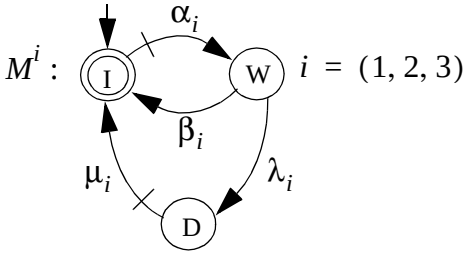


$i = (1, 2)$

Buffer Capacity: 1

4. Big Factory

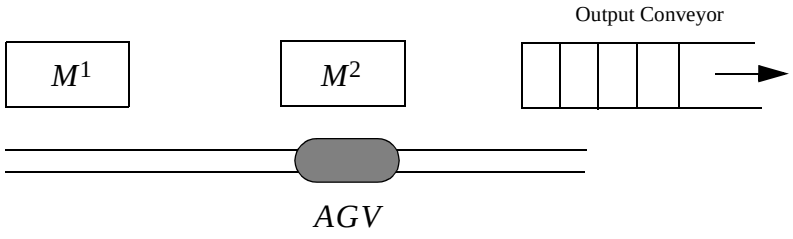




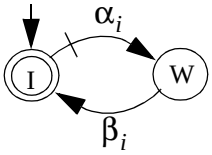
$i = (1, 2, 3)$

Buffer Capacity: 2

5. Workcell with AGV

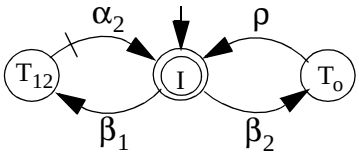


M^i :

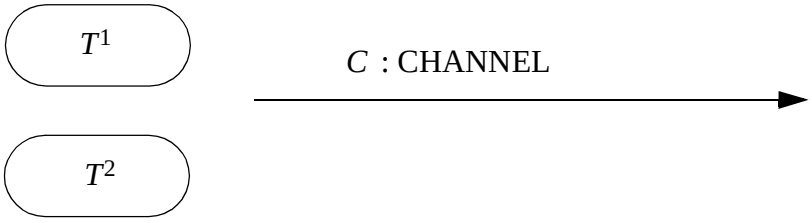


$i = (1, 2)$

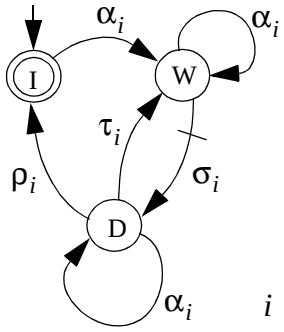
AGV :



6. Transmitters



T^i :



$i = (1, 2)$

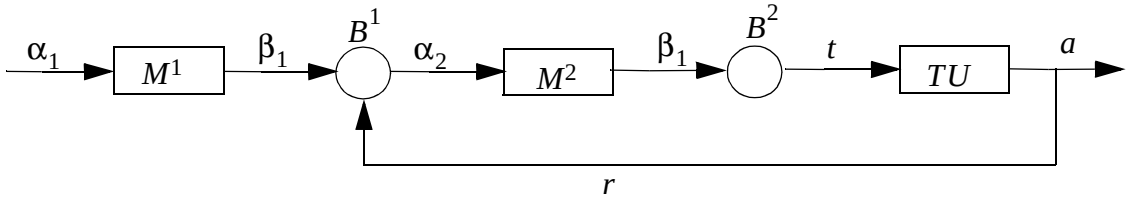
α_i : arrival of message to be transmitted

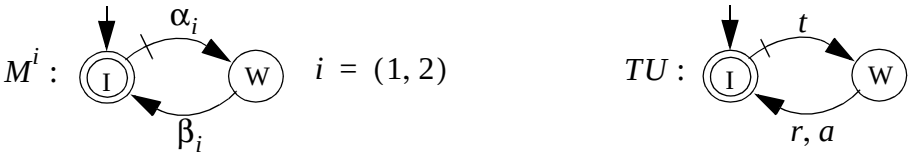
σ_i : start of message transmission

τ_i : timeout for no acknowledged message

ρ_i : reset for new transmission

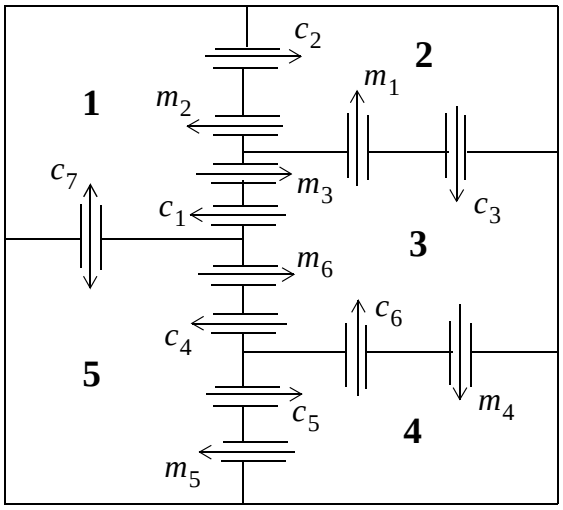
7. Transfer Line





Buffers Capacities: 1

8. Cat and Mouse



Initial condition: Cat in room 2

Mouse in room 4

Exercices:

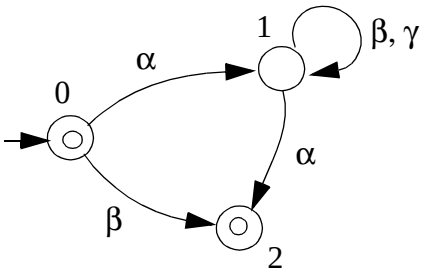
A. Basic (languages)

A1) With the alphabet $\Sigma = \{a, b, c\}$ find regular expressions for
a) the language formed by words with at least one a and one b;
b) the language formed by words with exactly one c.

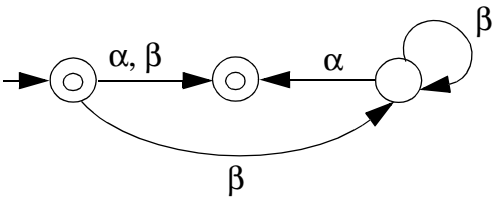
A2) With the alphabet $\Sigma = \{0, 1\}$ find regular expressions for
a) the language formed by words with no 1 preceeding 0;
b) the language formed by words with 2 or more symbols followed by 3 or more 0's.

A3) Build automata which accepts the languages
a) $a(ba)^*a^*$;
b) $\beta^*(\alpha\beta + \alpha\alpha)^*$;
c) $\alpha^*\beta\alpha(\alpha + \beta)$;
d) $(\alpha\beta)^*\alpha\beta\alpha\alpha$.

A4) Find regular expressions for the languages accepted by



(a)



(b)

B. Basic (DES models)

Build the composed generator (by hand and by TCT or CONDES) for

B1) Philosophers 1 and 2 in system 1;
B2) Philosophers 1, 2 and 3 in system 1;
B3) Philosophers 1 and 3 in system 2;
B4) to B8) The plants in systems 3 to 7.

B9) Build generators for the cat and for the mouse, and then obtain the generator for the composed system.

C. Control Synthesis (basic)

Solve the Supervisory Control Problem for the following systems and specifications:

C1) Philosophers 1, 2 and 3 in System 1;
Specification: avoid states where two adjacent philosophers eat simultaneously.

C2) Philosophers 1 and 2 in System 1;
Specifications: i) As in C1; and
ii) Fairness: at any time, the difference between the number of times each philosopher has eatten shouldn't be greater than 1.

C3) Philosophers 1 and 3 in System 2;
Specification: avoid states where both philosophers drink simultaneously.

C4) Philosophers 1 and 3 in System 2;
Specifications: i) As in C3; and
ii) Fairness: as in C2.

C5) System 3;
Specification: avoid overflow and underflow of buffer B.

C6) System 3;
Specifications: i) As in C5; and
ii) If both machines are broken down, then M^2 must be repaired before M^1.

C7) System 4;
Specifications: i) As in C5; and
ii) M^1, M^2 are repaired in order of breakdown; and
iii) M^3 has priority of repair over M^1, M^2.

C8) System 5;
Specification: avoid blocking. Why the TRIM component of the plant model cannot be used as a supervisor for the system?

C9) System 6;
Specification: avoid both transmitters to use the channel simultaneously.

C10) System 7;
Specifications: i) avoid overflow and underflow of buffer B^1;
ii) idem B^2.

Remark: This example shows the relevance of marking in the formulation of some control problems. Observe that if you don't use marking conveniently (marking all states in the

specs for example) in the generators for the specifications, the resulting supervisor may lead the closed-loop system to a deadlock (a state from where the system can not evolve). Discuss why this situation occurs.

C11) System 8;
Specifications: i) avoid the cat and the mouse to occupy the same room;
ii) allow the cat and the mouse to always return to their initial room.

D. Control Synthesis (extensions)

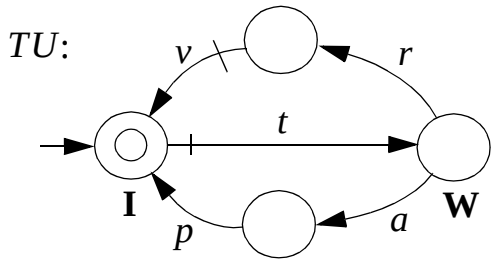
D1) Verify the possibility of solving C2, C4, C5, C7 and C10 using the modular approach. Whenever possible, solve it. Discuss the special case of C10.

D2) Consider the following hierarchical control problem for system 7:

Suppose an optimal low-level supervisor is used to control the system to ensure that neither of the buffers is subject to underflow or overflow (as in C10);

As the above solution as the starting point to the development of hierarchical structure consider that

i) the model for the Test Unit is now



ii) the initially significant events for the manager are

taking a fresh workpiece - low level event α_1 signaled as high level event τ_1;

pass test - event *a* signaled as τ_2;

fail test - event *r* signaled as τ_3

iii) High level Specification: If two consecutive test failures occur, allow TU to operate just once more, then shut down the system.

Solve the problem making the necessary modifications (if any) to get *control consistency* and *strict hierarchical consistency.*

E. Proofs

E1) Prove that the set of controllable languages contained in a language K is closed under

the operation of union of languages;

E2) Idem, for the L_m-closed languages;

E3) Prove that if K is prefix-closed, then so it is $supC(K)$;

E4) Prove that if K is Lm-closed, then so it is $supC(K)$;

E5) Prove Proposition 1 of the modular synthesis.