

Tutorial N° 02 : Petri Nets (Modeling Part)

Exercise 01 : Algorithmic structures

Model using PNs (This is a condition):

- 1) A test if C then [treatment(s) if] else [treatment(s) else]
- 2) A loop “for i = 1, N do [treatment(s)]”
- 3) A loop “while C do [treatment(s)]”
- 4) A loop “repeat [treatment(s)] until C”

Exercise 02:

❖ Model using PNs:

- 1) Addition of two integers X and Y.
- 2) Subtraction of two integers X and Y.
- 3) Multiplication of two integers X and Y.
- 4) Division of two integers X and Y.
- 5) The parallelism of three processes
- 6) Synchronization of five processes.
- 7) Semaphore (call by signal) for two processes.

❖ Construct a Petri net system that has the following equation: $2H_2 + O_2 \rightarrow 2H_2O$

Exercise 03 : Producer / Consumer

A producer/consumer system involves a producer process, a consumer process and a shared resource in the form of a buffer; the producer process deposits products there which are then withdrawn by the consumer process with the synchronization constraint requiring the consumer to wait for at least one production before being able to consume.

- Build the petri net corresponding to a producer/consumer system with a buffer of unlimited size.

Exercise 04 : Access to a resource in mutual exclusion



Reminder of the principle of mutual exclusion between processes: At any given time only one process accesses/uses a resource that it requested and obtained. After using the resource, the process releases it.

We want to manage access to a resource by two processes. These processes can be either at rest (in relation to the resource), or in demand for the resource, or in use of the resource.

- Build the Petri net modeling this management.

Exercise 05 : Mutual Exclusion

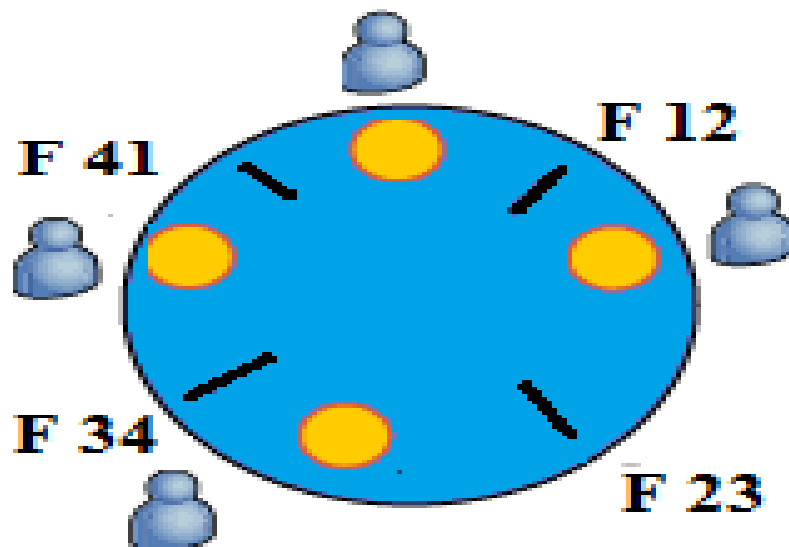
Two machines must interact with a database. Machines can read, write or remain idle.

- Model the situation using Petri nets, ensuring that the machines cannot write at the same time?

Exercise 06 : The philosophers' dinner problem

Four philosophers $\phi_1, \phi_2, \phi_3, \phi_4$ divide their time between rest (thought) and meals. Before eating they must successively take the cutlery located on their right then the cutlery located on their left. We have seen that we obtain a deadlock very quickly with this strategy (which is one that can arise in the general case).

- Model the philosophers' problem using a Petri net, carefully avoiding the deadlock situation.



Exercise 07

From a bridge, frogs jump into a stream, nondeterministically choose one of the two banks to swim to, and then hop to the bridge to start over again. A lovely girl picks up every third frog from the stream, kisses the frog, and puts the frog back on the bridge.

- Model this fairy tale as a Petri net. Suppose that three frogs are initially on the bridge.