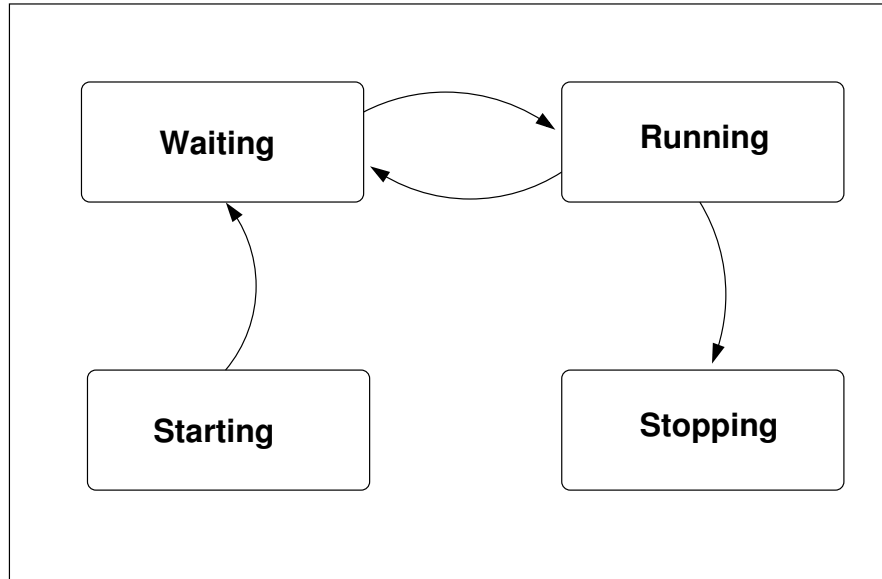


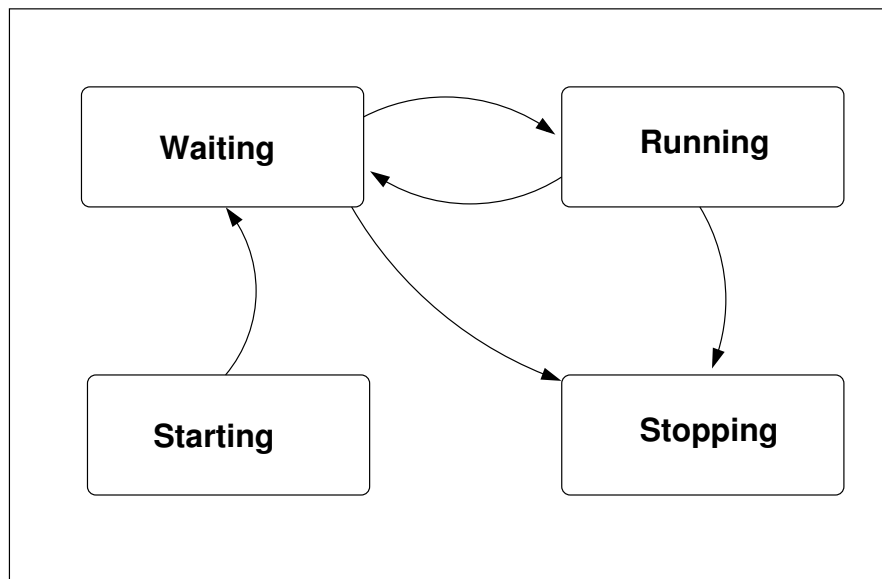
Exercise 1 *ex1-tut1.zip*

Express the following states machine using an Event B machines and check properties on the resulting models.



Exercise 2 *ex2-tut1.zip*

Express the following states machine using an Event B machines and check properties on the resulting models.



Exercise 3 *ex4-tut1.zip*

We consider a finite sequence of integers v_1, \dots, v_n where n is the length of the sequence and is supposed to be fixed. Write an Event B specification modelling the computation of the value

of the summation of the sequence v . You should define carefully v , n and the summation of a finite sequence of integers.

Exercise 4 *ex5-tut1.zip*

Express the following property in Event B :

We assume to have p resources which may be shared by n processes. If a process uses a given resource, the resource can not be used by another process. A process can use only at most one resource.

Exercise 5 *ex6-tut1.zip*

A Petri net is a tuple $R=(S,T,F,K,M,W)$

- S is a finite set of places.
- T is a finite set of transitions.
- $S \cap T = \emptyset$
- F is the flow relation : $F \subseteq S \times T \cup T \times S$
- K is expressing the capacity of each place :
 $K \in S \rightarrow \text{Nat} \cup \{\omega\}$
- M is representing the initial marking of each place :
 $M \in S \rightarrow \text{Nat} \cup \{\omega\}$ and satisfies the following condition $\forall s \in S : M(s) \leq K(s)$.
- W is the weight of each edge :
 $W \in F \rightarrow \text{Nat} \cup \{\omega\}$

The state of a Petri net R is defined by a set of markings :

- a marking M for R is a function from S to $\text{Nat} \cup \{\omega\}$:
 $M \in S \rightarrow \text{Nat} \cup \{\omega\}$ and it satisfies the condition $\forall s \in S : M(s) \leq K(s)$.
- a transition t of T is ready to fire for a marking M of R , if
 1. $\forall s \in \{s' \in S \mid (s',t) \in F\} :$
 $M(s) \geq W(s,t).$
 2. $\forall s \in \{s' \in S \mid (t,s') \in F\} :$
 $M(s) \leq K(s) - W(s,t).$

- $t \in T : \text{Pre}(t) = \{s' \in S : (s',t) \in F\}$ and $\text{Post}(t) = \{s' \in S : (t,s') \in F\}$

The simulation of a Petri net is defined by a relation linking three elements : a marking M , a marking M' and a transition t as follows :

- the new marking M' is defined as follows from M :

$$M'(s) = \begin{cases} M(s) - W(s,t), & \text{SI } s \in \text{PRE}(t) - \text{POST}(t) \\ M(s) + W(t,s), & \text{SI } s \in \text{POST}(t) - \text{PRE}(t) \\ M(s) - W(s,t) + W(t,s), & \text{SI } s \in \text{PRE}(t) \cap \text{POST}(t) \\ M(s), & \text{SINON} \end{cases}$$

We consider the following Petri net :



Question 5.1 *Translate this Petri net in Event B.*

Question 5.2 *Express safety properties that you can discover from the diagram.*

Exercise 6 (*ex7-tut1.zip*)

We consider the following abstract machine /

```

MACHINEM1
VARIABLES
     $x$ 
INVARIANTS
...
EVENTS
EVENT INITIALISATION
    BEGIN
         $act1 : x := -10$ 
    END
    EVENT evt1
        WHEN
             $grd1 : x \geq -1$ 
        THEN
             $act1 : x := x+1$ 
        END
    EVENT evt2
        WHEN
             $grd1 : x \leq -1$ 
             $grd2 : x \geq -44$ 
        THEN
             $act1 : x := x-1$ 
        END
END

```

We have possible candidates as invariant. For each question, explain why the assertion is or is not an inductive invariant. For each question, explain why the assertion is or is not a safety property.

Question 6.1 (M1)

```

 $inv1 : x \in \mathbb{Z}$ 
 $inv3 : x \leq -1$ 

```

Question 6.2 (M2)

```

 $inv1 : x \in \mathbb{Z}$ 
 $inv3 : x \leq -3$ 

```

Question 6.3 (M3)

```

 $inv1 : x \in \mathbb{Z}$ 
 $inv4 : -45 \leq x \wedge x \leq -10$ 

```

Question 6.4 (M4)

```

 $inv1 : x \in \mathbb{Z}$ 
 $inv3 : x \leq -3$ 
 $inv4 : -45 \leq x \wedge x \leq -10$ 
 $inv2 : x \leq -1$ 

```

Exercice 7 *ex8-tut1.zip*

A semaphore s is a shared variable accessible by two operations : $P(s)$ and $V(s)$. Informally, we can describe the effect of these two operations as follows :

- $P(s)$ is testing if the value of s is greater than 0 and is not equal to 0. If the value of s is 0, the process which is executing $P(s)$ is inserted in a queue.
- $V(s)$ is increasing the value of s by one, if the queue is non empty. If the queue is non empty, the first waiting process of the queue is awoken and becomes a lively process.

Using the Event B modelling features, describe a system using the primitives.