Tutorial 12

1) Let [CUSTOMER] be the set of all customers at a supermarket and [CHECKOUT] be the set of all checkout counters.

   (a) Specify a supermarket where the customer has to wait in a check-out queue before he can pay at the checkout. There are several checkouts. Name the state space as *Supermarket*.

   (b) Specify an initial state schema called *InitSupermarket* where a queue is empty for each checkout.

   (c) Specify an operation schema called *EnterCheckout* where a customer joins a queue for a checkout. (The customer and the checkout are inputs).

   (d) Specify an operation schema called *LeaveCheckout* where a customer finishes with the purchasing and leave the counter. The queue will be updated with the rest of the customers queuing at the counter.

   (e) Specify an operation schema called *CloseCheckout* where a queue closes, and all the waiting customers in the queue are appended to another queue (two checkouts are inputs).

2) Consider a scenario concerning a waiting list in a hospital.

> A hospital has a number of beds for the use of patients who require operations. When a patient is referred to the hospital for treatment, the system attempts to find a free bed. If a bed is available immediately, the patient is informed. If no free bed is available, the patient is placed on a waiting list. Waiting lists can grow indefinitely.
>
> When a patient occupying a bed is ready to leave the hospital, the bed becomes free. The system allocates the free bed to a patient in the waiting list based on a first come first serve basis. The system also allows a hospital administrator to enquire information about which patients are waiting, which beds are occupied and by whom, and which beds are free.

Given the types:

[BED]	- the set of all possible beds
[PATIENT]	- the set of all possible patients

The state of the waiting list is shown in the state space below:

$\_WaitingList\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_$

$beds : \mathbb{P}\ BED$

$occupiedBy : BED ⤔ PATIENT$

$waiting : seq\ PATIENT$

$dom\ occupiedBy \subseteq beds$

$ran\ waiting \cap ran\ \ occupiedBy = \varnothing$

$dom\ occupiedBy \subset beds \Rightarrow waiting = \langle\rangle$

Referring to the given state space schema for the *WaitingList* above, write the Z schemas for the following operations:

(a) An initial schema called *InitWaiting* which defines the initial condition of each of the components.

$\_\_InitWaiting\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_$

$WaitingList$

$beds = \varnothing$

(b) An operation schema called *HowManyBedsFree* which outputs the number of free beds (*numberFree!*).

$\_\_HowManyBedsFree\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_$

$\Xi\ WaitingList$

$numberFree! : \mathbb{N}$

$numberFree! = \#beds - \#occupiedBy$

(c) An operation schema called *HowLongIsList* which returns the size of the waiting list in the output component *lengthOfList!*.

$\_\_HowLongIsList\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_$

$\Xi\ WaitingList$

$lengthOfList! : \mathbb{N}$

$lengthOfList! = \#waiting$

(d) An operation schema called *LeaveWithNoWaiting* such that a patient leaves the hospital and there is no patient on the waiting list to occupy the free bed.

$LeaveWithNoWaiting$

$\Delta\ WaitingList$
$p? : PATIENT$

$p? \in \mathrm{ran}\ occupiedBy$
$waiting = \langle\rangle$

$beds' = beds$
$occupiedBy' = occupiedBy \rhd \{p?\}$
$waiting' = waiting$

OR

$LeaveWithNoWaiting$

$\Delta\ WaitingList$
$p? : PATIENT$
$b? : BED$

$b? \in beds$
$b? \mapsto p? \in occupiedBy\ \ OR\ \ occupiedBy\ b? = p?$
$waiting = \langle\rangle$

$beds' = beds$
$occupiedBy' = occupiedBy \setminus \{b? \mapsto p?\}$
$waiting' = waiting$

OR

$LeaveWithNoWaiting$

$\Delta\ WaitingList$
$p? : PATIENT$
$b? : BED$

$b? \in beds$
$b? \mapsto p? \in occupiedBy\ \ OR\ \ occupiedBy\ b? = p?$
$waiting = \langle\rangle$

$beds' = beds$
$occupiedBy' = occupiedBy \oplus \{b? \mapsto \varnothing\}$
$waiting' = waiting$

(e) An operation schema called *LeaveAndAssignToWaiting* such that a patient leaves the hospital and there is at least one patient on the waiting list to occupy the free bed.

__*LeaveAndAssignToWaiting*___

$\Delta\ WaitingList$
$p1?, p2? : PATIENT$
$b? : BED$

$b? \in beds$
$b? \mapsto p1? \in occupiedBy$ *OR* $occupiedBy\ b? = p1?$
$waiting \neq \langle\rangle$
$p2? \in ran\ waiting$
$p? = head\ waiting$

$beds' = beds$
$occupiedBy' = \{(occupiedBy \setminus \{b? \mapsto p1?\}), (occupiedBy \cup \{b? \mapsto p2?\})\}$
OR
$occupiedBy' = occupiedBy \oplus \{b? \mapsto p2?\}$
$waiting' = tail\ waiting$

OR

__*LeaveAndAssignToWaiting*___

$\Delta\ WaitingList$
$p? : PATIENT$
$b? : BED$

$b? \in beds$
$b? \mapsto p? \in occupiedBy$ *OR* $occupiedBy\ b? = p?$
$waiting \neq \langle\rangle$

$beds' = beds$
$occupiedBy' = \{(occupiedBy \setminus \{b? \mapsto p?\}), (occupiedBy \cup \{b? \mapsto head\ waiting?\})\}$
OR
$occupiedBy' = occupiedBy \oplus \{b? \mapsto head\ waiting?\}$
$waiting' = tail\ waiting$