

## Formal Methods: (Past Paper)

Q1:

a) types

mode = <working> | <idle> | <broken>

b) state IndustrialRobot of

initialmode :: Mode

Robot :: Mode

inv mk-IndustrialRobot (i, r)  $\triangleq$  initialmode = <idle>  $\wedge$  robot = initialmode

init mk-Industrial (i, r)  $\triangleq$  Robot = initialmode  $\wedge$  r = i

(i)(i) setmode (m:mode)

ext wr Robot :: mode

pre TRUE

post Robot = <m>

(ii) getmode () m:mode

ext rd Robot :: mode

pre TRUE

post m = Robot

## Question 28

~~[PERMISSION], [LANDED]~~ → [AIRCRAFT]

① Axiomatic definition & | MAX: N

② State Schema

Airport

$\text{permission} : \text{PPERMISSION}$ $\text{Landed} : \text{PLANDED}$ $\text{Landed} \leq \text{permission}$ $\# \text{Landed} \leq \text{MAX}$
--------------------------------------------------------------------------------------------------------------------------------------------------------------

③ Init- state

InitAirport

$\text{Airport}$ $\text{Landed, permission} = \emptyset$
-------------------------------------------------------------

④

Give Permission

$\Delta \text{Airport}$ $\text{aircraft?} : \text{AIRCRAFT}$ <del><math>\text{permission} \neq \text{permission}</math></del> $\text{aircraft?} \neq \text{permission}$ $\text{permission}' = \text{permission} \cup \{\text{aircraft?}\}$
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

recalLanding

$\Delta \text{Airport}$ $\text{aircraft?} : \text{AIRCRAFT}$ $\text{aircraft} \in \text{Permission}$ $\# \text{Landed} < \text{MAX}$ $\text{Landed}' = \text{Landed} \cup \{\text{aircraft?}\}$
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

numberWaiting

$\Delta \text{Airport}$ $\text{count} ! : \text{N}$ $\text{count} = \#(\text{Permission} \setminus \text{Landed})$
--------------------------------------------------------------------------------------------------------------------------

Q3:

hyper values

limit: N

state CarParking of

carsCount: [Z]

inv mk - CarParking(c)  $\Delta$  c  $\leq$  LIMIT

init mk - CarParking(count)  $\Delta$  count = NIL

Enter()

ext wr carsCount: [Z]

pre carsCount  $<$  LIMIT

post carsCount =  $\overline{\text{carsCount} + 1}$

Depart()

ext wr carsCount: [Z]

pre carsCount  $> 0$

post carsCount =  $\overline{\text{carsCount} - 1}$

QuerySpace() count: ~~0~~ Z

ext wr carsCount: [Z]

pre TRUE

post count = LIMIT - carsCount



# Question 4:

[PATIENT]

i) ~~Limit~~ LIMIT : N

ii) ~~Patient Register~~  
 $reg : \mathbb{P} \text{ PATIENT}$   
 $\# reg \leq \text{LIMIT}$

iii) Init  
~~Patient Register~~  
 $reg = \emptyset$

iv) add Patient  
~~Patient Register~~  $\Delta \text{ Patient Register}$   
 $\text{Patient} ? : \text{PATIENT}$   
 $\text{patient} \notin reg$   
 $\# reg < \text{LIMIT}$   
 $reg' = reg \cup \{ \text{Patient} ? \}$

remove Patient  
 $\Delta \text{ Patient Register}$   
 $\text{Patient} ? : \text{PATIENT}$   
 $\text{Patient} \in reg$   
 $reg' = reg \setminus \{ \text{Patient} ? \}$

getPatients  
 $\sqsubseteq \text{ Patient Register}$   
 $\text{output} ! : \mathbb{P} \text{ PATIENT}$   
 $\text{output} ! = reg$

number Register  
 $\sqsubseteq \text{ Patient Register}$   
 $\text{count} ! : N$   
 $\text{count} ! = \# reg$

Question 58-

[Room]

STATUS := OCCUPIED | VACANT

limit : N

Hotel

room : Room  $\rightarrow$  STATUS

# room  $\leq$  limit

Init

Hotel

room =  $\emptyset$

Add Room

$\Delta$  Hotel

~~room~~? : Room

$x? \notin$  room

# room  $<$  limit

$room' = room \cup \{x? \rightarrow VACANT\}$

occupy Room

$\Delta$  Hotel

$x? : Room$

~~room~~ (x?) = VACANT

$room' = room \oplus \{x? \rightarrow OCCUPIED\}$

Vacant Room

$\Delta$  Hotel

~~room~~ (x?) = VACANT

$x? : Room$

room (x?) = OCCUPIED

$room' = room \oplus \{x? \rightarrow VACANT\}$

(d) occupancy Count = # (room  $\Delta$  {occupied})

extra