6CCGD005W Formal Methods: In-class test

Formatting of this document is not important as the actual test will take place in Blackboard. Each question is worth 10 points; in the actual test the order of questions will be randomised.

Question 1

Given the sets
Animal = {Bat, Crocodile, Dog, Parrot, Toad, Wasp}
Flying <: Animal & Flying = {Bat, Parrot, Wasp}
Furry <: Animal & Furry = {Bat, Dog}
Evaluate the following expressions. (2 marks each)

1. Flying \/ Furry
2. Animal - Furry
3. card(Furry /\ Flying)
4. card({Flying})
5. POW(Furry)

Question 2

Given the sets
A <: NAT & A = {1,2,3,4,5}
B <: NAT & B = {2,3,5,7}
For each of the following formulas, explain briefly what they mean and decide which ones are true. (2 marks each)

1. (A <: B) <=> (B < A)
2. !(n).(n : A => n+1 : B)
3. #(n).(n : B & n+1 :A)
4. !(n).(n : A => #(m).(m : B & m < n))
5. #(n).(n : A & !(m).(m : B => m < n))

Question 3

Given the sets
Animal = {Bat, Crocodile, Dog, Parrot, Toad, Wasp}
Skin = {Fur, Feathers, Neither}
Size = {Tiny, Small, Medium, Large}
and the relations
skinType : Animal <-> Skin & skinType = {Bat|->Fur, Crocodile|->Neither, Dog|->Fur, Parrot|->Feathers, Toad|->Neither, Wasp|->Neither}
sizes : Animal <-> Size & sizes = {Bat|->Small, Crocodile|->Medium, Crocodile|->Large, Dog|->Small, Dog|->Medium, Parrot|->Small, Toad|->Tiny, Toad|->Small, Wasp|->Tiny}
Determine the following. (2 marks each)

1. The kind of relation of **skinType** (Is it a (partial) function? If so, is it injective and/or surjective?)

2. dom(Animal*Skin - skinType)
3. skinType $\sim$
4. (sizes$\sim$; skinType)
5. sizes[{Toad}]

Question 4

Given the set
Names = {George, John, Mary, Paul, Peter, Ringo}
and the sequences
S : seq(NAT) & S = [John, Paul, George, Ringo]
T : seq(NAT) & T = [Peter, Paul, Mary]
Determine the following. (2 marks each)

1. S ^ T
2. ran(T)
3. John -> S
4. S /\ T
5. tail(S)

Question 5

Write a B machine called Gadgets satisfying the following requirements:

It allows us to create gadgets from machine parts. There are three types of gadget (clunky, dodgy and questionable) and three types of machine parts (cogs, hinges and levers).

A clunky gadget needs a cog and a hinge; a dodgy gadget needs a cog and a lever; a questionable gadget needs a hinge and a lever. Initially we have 5 of each part.

The machine should keep track of how many of each part you have.

The machine must provide an operation to try to create any of the types of gadgets for which we have enough parts, and update the supply afterwards.

Question 6

Given the following incomplete B machine:

```
MACHINE Palindromes
VARIABLES
 pals
INVARIANT
 pals <: seq(NAT) & TODO
INITIALISATION
 pals  := {}
OPERATIONS
 addEmpty = TODO
 append(nn) = TODO
END // TwoSets
```

Provide the following:

1. The missing part of the invariant which says that all the sequences in pals are palindromes (i.e. have the same numbers when read forwards and backwards, such as [1,10,5,10,1] or [0,23,23,0]) (4 marks)
2. The definition of the addEmpty operation which adds an empty sequence to pals (3 marks)
3. The definition of the append operation which adds the given number nn to the front and back of all the sequences in pals (3 marks)

Question 7

Consider this incomplete machine:

```
MACHINE Tokens

SETS
 SHAPES = {Circle, Ellipse, Square, Triangle};
 COLOURS = {Red, Yellow, Green, Blue};
 RESPONSES = {Yes, No}

VARIABLES
 tokens
```

1. The tokens variable should keep track of the tokens (combinations of a shape and a colour) created so far. Provide the invariant and initialisation for it. (2 marks)
2. Add an operation coloursSeen(shape) which returns the set of colours that the given shape brand has been created with. (3 marks)
3. Add an operation shapesSeen(colour) which returns the set of shapes that have been created with the given colour. (3 marks)
4. Add an operation allDone which returns Yes if we have created all possible combinations, and No otherwise. (2 marks)

Question 8

Explain the role that each of the following parts plays in the definition of a machine: (2 marks each)

1. PROPERTIES
2. INVARIANT
3. INITIALISATION
4. OPERATIONS
5. ASSERTIONS

Question 9

Define a B-Machine for the tiny airline company OneFlight that has the following requirements:

- It specifies a single plane's flight route.
- The airline serves the following cities: Berlin, Dublin, Geneva, London, Madrid, Paris and Rome.
- The plane's flight route is a sequence of cities, starting from the departure city to the destination city.
- The flight route has a maximum length, i.e. maximum number of cities.
- It is a one-way flight, so no city can occur on the route more than once.

Your B machine should include the following:

1. Any sets, constants, variables, state invariant and initialisation that the flight route requires. (7 marks)
2. The following enquiry operation on the flight route:

   RouteStatus - reports via a suitable message whether the flight route is empty, full, only has the departure city or can be extended, i.e. not full. (3 marks)

Question 10

Given the following B machine that partially specifies a ghost train ride at a fair ground.

```
MACHINE GhostTrainRide

 SETS

 PEOPLE = { Joe, Bob, Ian, Mary, Sue, Mia, Tom, Tim, Zoe, Bill }

 CONSTANTS

 RideCapacity, MaxQueuing

 PROPERTIES

 10 <= card( PEOPLE )

 &

 RideCapacity : NAT1 & RideCapacity < card(PEOPLE)

 &

 MaxQueuing : NAT1 & MaxQueuing <= ( RideCapacity * 2 )

 VARIABLES

 onRide, // people on the ride

 rideQueue // people queue to go on the ride

 INVARIANT

 onRide <: PEOPLE &

 rideQueue <: PEOPLE &

 onRide /\ rideQueue = {} &

 card(onRide) <= RideCapacity &

 card(rideQueue) <= MaxQueuing

 INITIALISATION

 onRide := {} || rideQueue := {}
```

Using "plain English" only, answer the following questions.

1. Explain the meaning of the three PROPERTIES predicates. (3 marks)
2. Explain the meaning of the five INVARIANT predicates. (7 marks)