Answer key

MODULE: (2022) 6SENG005C.1 Formal Methods (IIT Sri Lanka)

2022 In-Class Test

Question 1

10 points

Given the following B definitions:

CAR = {Ford, Toyota, Mini, Nissan, Renault, Fiat, Ferrari, RollsRoyce, AstonMartin }
COUNTRY = { UK, France, Italy, Japan, USA)
BudgetCars <: CAR
BudgetCars = { Ford, Fiat, Nissan, Renault }
LuxuryCars <: CAR
LuxuryCars = { Ferrari, RollsRoyce, AstonMartin }
built_in : CAR -> COUNTRY
built in = { Ford |-> USA, Toyota |-> Japan, Mini |-> UK, Nissan |-> Japan, Renault |-> France, Ferrari |-> Italy, Fiat |-> Italy, RollsRoyce |-> UK, AstonMartin |-> UK }

Evaluate the following expressions:

(a) LuxuryCars /\ { Mini, AstonMartin } **[1 mark]**
(b) card(built_in) **[1 mark]**
(c) built_in(AstonMartin) **[1 mark]**
(d) BudgetCars - { Ford, Toyota, Nissan, Ferrari } **[1 mark]**
(e) dom(built_in) **[2 marks]**
(f) built_in |> { UK } **[2 marks]**
(g) P(LuxuryCars) **[2 marks]**

Question 2

10 points

The following is an example of an abstract machine's operation:

yy <-- operation(xx) =
PRE PC
THEN Subst
END

Explain the overall purpose of an operation and the role that each part plays in specifying the operation.

Question 3

10 points

Given the following definitions of the relations R1 and R2:

R1 : NAT <->NAT
R1 = { 0 |-> 0, 1 |-> 2, 2 |-> 3, 3 |-> 3, 3 |-> 4, 3 |-> 5, 4 |-> 5 }

R2 : NAT <->NAT
R2 = { 0 |-> 1, 3 |-> 3, 4 |-> 5, 4 |-> 6, 5 |-> 5, 6 |-> 7 }

(a) Evaluate the following expressions:

(i) R2 [{ 3, 4 }] **[2 marks]**
(ii) R2 <+ { 3 |-> 7 } **[2 marks]**
(iii) R1 <+ { 3 |-> 7 } **[3 marks]**

(b) Explain why R1 and R2 are relations but not functions. **[3 marks]**

Question 4

10 points

Given the following B definitions:

Person = { Paul, Sue, Ian, John, Tom, Jim, Mary }

Day = { Mon, Tue, Wed, Thu, Fri, Sat, Sun }

favouriteday = { Paul |-> Sat, Paul |-> Sun, Sue |-> Sun, Ian |-> Wed, John |-> Fri, Tom |-> Tue }

working = { Mon |-> Paul, Tue |-> Ian, Wed |-> Tom, Thu |-> Paul, Fri |-> Sue }

birthday = { Paul |-> Mon, Sue |-> Tue, Ian |-> Wed, John |-> Thu, Tom |-> Fri, Jim |-> Sat, Mary |-> Sun }

For each of the above relations favouriteday, working and birthday give its type definition and give a justification for your choice.
That is, for each one give an explanation of why you think it is just a relation, or a function, and what type of function, i.e. total, partial, injective, surjective or bijective.

Question 5

10 points

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:

(a) Any sets, constants, variables, state invariant and initialization that the flight route requires. **[7 marks]**

(b) 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 6

10 points

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.

(a) Explain the meaning of the three PROPERTIES predicates. **[3 marks]**

(b) Explain the meaning of the five INVARIANT predicates. **[7 marks]**

Question 7

10 points

With reference to the GhostTrainRide B machine of the previous question, add the following two operations to the machine, ensuring that the operations are compatible with its existing declarations.

(a) joinRideQueue(person) - the person joins the queue for the ghost train ride. **[5 marks]**

(b) getOnRide(person) - the person leaves the queue for the ride and gets on the ghost train ride. **[5 marks]**

Question 8

10 points

Determine the missing assertions for the following program using pre-condition propagation.

```
[Assertion 1]
x:=x+5;
[Assertion 2]
x:=y;
[Assertion 3]
y:=5
[x>y]
```

Question 9

10 points

Determine the missing assertions for the following program using pre-condition propagation.

```
[Assertion 1]
IF x>0 THEN
  [Assertion 2]
  y:=0
  [Assertion 3]
ELSE
  [Assertion 4]
  x:=y
  [Assertion 5]
```

END
[x>=y]

Question 10

10 points

Which of the following Hoare triples are valid? Provide counterexamples for the invalid ones.

(a)
[y>0]
y:=x
[x>0]

(b)
[x>0]
x:=x+1
[x=x+1]

(c)
[x>y]
y:=x+1
[y>x]

MODULE: (2022) 6SENG005C.1 Formal Methods (IIT Sri Lanka)

2022 In-Class Test

Question 1

10 points

Given the following B definitions:

CAR = {Ford, Toyota, Mini, Nissan, Renault, Fiat, Ferrari, RollsRoyce, AstonMartin }
COUNTRY = { UK, France, Italy, Japan, USA)
BudgetCars <: CAR
BudgetCars = { Ford, Fiat, Nissan, Renault }
LuxuryCars <: CAR
LuxuryCars = { Ferrari, RollsRoyce, AstonMartin }
built_in : CAR -> COUNTRY
built in = { Ford |-> USA, Toyota |-> Japan, Mini |-> UK, Nissan |-> Japan, Renault |-> France, Ferrari |-> Italy, Fiat |-> Italy, RollsRoyce |-> UK, AstonMartin |-> UK }

Evaluate the following expressions:

(a) LuxuryCars /\ { Mini, AstonMartin } **[1 mark]**
(b) card(built_in) **[1 mark]**
(c) built_in(AstonMartin) **[1 mark]**
(d) BudgetCars - { Ford, Toyota, Nissan, Ferrari } **[1 mark]**
(e) dom(built_in) **[2 marks]**
(f) built_in |> { UK } **[2 marks]**
(g) P(LuxuryCars) **[2 marks]**

Question 2

10 points

The following is an example of an abstract machine's operation:

yy <-- operation(xx) =
PRE PC
THEN Subst
END

Explain the overall purpose of an operation and the role that each part plays in specifying the operation.

Question 3

10 points

Given the following definitions of the relations R1 and R2:

R1 : NAT <->NAT
R1 = { 0 |-> 0, 1 |-> 2, 2 |-> 3, 3 |-> 3, 3 |-> 4, 3 |-> 5, 4 |-> 5 }

R2 : NAT <->NAT
R2 = { 0 |-> 1, 3 |-> 3, 4 |-> 5, 4 |-> 6, 5 |-> 5, 6 |-> 7 }

(a) Evaluate the following expressions:

(i) R2 [{ 3, 4 }] **[2 marks]**
(ii) R2 <+ { 3 |-> 7 } **[2 marks]**
(iii) R1 <+ { 3 |-> 7 } **[3 marks]**

(b) Explain why R1 and R2 are relations but not functions. **[3 marks]**

Question 4

10 points

Given the following B definitions:

Person = { Paul, Sue, Ian, John, Tom, Jim, Mary }

Day = { Mon, Tue, Wed, Thu, Fri, Sat, Sun } favouriteday = { Paul |-> Sat, Paul |-> Sun, Sue |-> Sun, Ian |-> Wed, John |-> Fri, Tom |-> Tue }

working = { Mon |-> Paul, Tue |-> Ian, Wed |-> Tom, Thu |-> Paul, Fri |-> Sue }

birthday = { Paul |-> Mon, Sue |-> Tue, Ian |-> Wed, John |-> Thu, Tom |-> Fri, Jim |-> Sat, Mary |-> Sun }

For each of the above relations favouriteday, working and birthday give its type definition and give a justification for your choice.
That is, for each one give an explanation of why you think it is just a relation, or a function, and what type of function, i.e. total, partial, injective, surjective or bijective.

Question 5

10 points

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:

(a) Any sets, constants, variables, state invariant and initialization that the flight route requires. **[7 marks]**

(b) 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 6

10 points

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.

(a) Explain the meaning of the three PROPERTIES predicates. **[3 marks]**

(b) Explain the meaning of the five INVARIANT predicates. **[7 marks]**

Question 7

10 points

With reference to the GhostTrainRide B machine of the previous question, add the following two operations to the machine, ensuring that the operations are compatible with its existing declarations.

(a) joinRideQueue(person) - the person joins the queue for the ghost train ride. **[5 marks]**

(b) getOnRide(person) - the person leaves the queue for the ride and gets on the ghost train ride. **[5 marks]**

Question 8

10 points

Determine the missing assertions for the following program using pre-condition propagation.

```
[Assertion 1]
x:=x+5;
[Assertion 2]
x:=y;
[Assertion 3]
y:=5
[x>y]
```

Question 9

10 points

Determine the missing assertions for the following program using pre-condition propagation.

```
[Assertion 1]
IF x>0 THEN
  [Assertion 2]
```

```
  y:=0
  [Assertion 3]
ELSE
  [Assertion 4]
  x:=y
  [Assertion 5]
END
[x>=y]
```

Question 10

10 points

Which of the following Hoare triples are valid? Provide counterexamples for the invalid ones.

(a)
[y>0]
y:=x
[x>0]

(b)
[x>0]
x:=x+1
[x=x+1]

(c)
[x>y]
y:=x+1
[y>x]