CIA 2326 Tutorial Sheets 4 and 5 – Solutions

Tutorial 4

1) Given: \( \text{Process}_\text{name} = \{ \text{process}1, \text{process}2, \text{process}3, \text{process}4, \text{process}5 \} \)
\( \text{Process}_\text{ID} = \{1, 2, 3, 4\} \)
classify the following functions from \( \text{Process}_\text{ID} \) to \( \text{Process}_\text{name} \) as
i) total or partial;
ii) injective, bijective, surjective or partial only
a) \( \{1 \mapsto \text{process}1, 2 \mapsto \text{process}1\} \) partial only
b) \( \{1 \mapsto \text{process}1, 2 \mapsto \text{process}2\} \) partial injection
c) \( \{1 \mapsto \text{process}1, 2 \mapsto \text{process}2, 3 \mapsto \text{process}3, 4 \mapsto \text{process}2\} \) total only
d) \( \{1 \mapsto \text{process}1, 2 \mapsto \text{process}2, 3 \mapsto \text{process}3, 4 \mapsto \text{process}5\} \) total injection

2) Given: \( \text{set}_\text{A} = \{1, 2, 3, 4, 5\}, \text{set}_\text{B} = \{2, 4, 6, 8, 10\} \), classify the following functions from \( \text{set}_\text{A} \) to \( \text{set}_\text{B} \) as
i) total or partial;
ii) injective, bijective, surjective or partial only
a) \( \{1 \mapsto 6, 2 \mapsto 10\} \) partial injection
b) \( \{1 \mapsto 2, 2 \mapsto 4, 3 \mapsto 6, 4 \mapsto 8\} \) partial injection
c) \( \{1 \mapsto 2, 2 \mapsto 4, 3 \mapsto 6, 4 \mapsto 8, 5 \mapsto 10\} \) bijection

3) Given \( \text{STUDENT} = \{\text{ann, fred, pete, imran, jim}\}, \text{RESULT} = \{AA, BB, CC, DD, EE, FF\} \)
where before the update:
\[
\text{tested} = \{\text{ann} \mapsto BB, \text{fred} \mapsto EE, \text{pete} \mapsto AA, \text{imran} \mapsto BB\}
\]
and after the update:
\[
\text{tested} = \{\text{ann} \mapsto BB, \text{fred} \mapsto DD, \text{pete} \mapsto AA, \text{imran} \mapsto BB, \text{jim} \mapsto CC\}
\]
\( \text{tested} \iff \text{tested} \leq \text{update}_\text{function} \)
where \( \text{update}_\text{function} = \{\text{fred} \mapsto DD, \text{jim} \mapsto CC\} \)

4) Given:
\( A = \{a, b, c, d, e, f, g, h\}; \text{B} = \{1, 2, 3, 4, 5\} \)
\( \text{F1} = \{a \mapsto 2, b \mapsto 4, d \mapsto 1, e \mapsto 4, f \mapsto 2, h \mapsto 5\} \)
a) \( \text{dom F1} = \{a, b, d, e, f, h\} \)
b) \( \text{ran F1} = \{1, 2, 4, 5\} \)
c) \( F1^{-1} = \{2 \mapsto a, 4 \mapsto b, 1 \mapsto d, 4 \mapsto e, 2 \mapsto f, 5 \mapsto h\} \)
d) Is \( F1^{-1} \) a function? No – because we have \( (eg) 2 \mapsto a, 2 \mapsto f \) and \( a \neq f \).
e) (i) \( F1(b) = 4 \) (ii) \( F1(h) = 5 \).
f) \( F2 := F1 \cup \{g \mapsto 2\} \Rightarrow \text{F2} = \{a \mapsto 2, b \mapsto 4, d \mapsto 1, e \mapsto 4, f \mapsto 2, h \mapsto 5, g \mapsto 2\} \).
g) \( F3 := F1 - \{b \mapsto F1(b)\} \Rightarrow \text{F3} = \{a \mapsto 2, c \mapsto 4, d \mapsto 1, e \mapsto 4, f \mapsto 2, h \mapsto 5\} \)
\[
F_3 = \{ a \mapsto 2, d \mapsto 1, e \mapsto 4, f \mapsto 2, h \mapsto 5 \}.
\]

h) \( F_4 := F_1 \iff \{ a \mapsto 4, c \mapsto 3, g \mapsto F_1(d) \} \Rightarrow \)
\( F_4 = \{ a \mapsto 4, c \mapsto 3, g \mapsto 1, b \mapsto 4, d \mapsto 1, e \mapsto 4, f \mapsto 2, h \mapsto 5 \} \).

i) Calculate LHS and RHS, where \( R_1 = F_1, \)
\( R_2 = \{ a \mapsto 4, c \mapsto 3, g \mapsto F_1(d) \} \).
They should both equal \( F_4: \)
\( \{ a \mapsto 4, c \mapsto 3, g \mapsto 1, b \mapsto 4, d \mapsto 1, e \mapsto 4, f \mapsto 2, h \mapsto 5 \} \).

5) The following is a possible solution – which for completeness also includes the other clauses:

\begin{verbatim}
MACHINE Class_Man
SETS
   STUDENT; GRADE = \{ A1, A2, A3, B1, B2, B3, C1, C2, C3, D1, D2, D3, E1, E2, E3, FF, GG \}; RESULT = \{ pass, fail \}
CONSTANTS
   max_class_size
PROPERTIES
   max_class_size \in \mathbb{N}
VARIABLES
   enrolled, tested
INVARIANT
   enrolled \subseteq STUDENT \land
   tested \in STUDENT \rightarrow GRADE \land \text{dom} \text{tested} \subseteq \text{enrolled} \land
   \text{card} (\text{enrolled}) \leq \text{max\_class\_size}
OPERATIONS
   res \leftarrow Add\_Result (stud, grd) =
PRE
   stud \in \text{enrolled} \land grd \in GRADE \land stud \notin \text{dom} \text{tested}
THEN
   tested := tested \cup \{ stud \mapsto grd \} \mid
   IF \quad grd \in \{ E1, E2, E3, FF, GG \}
   THEN \quad res := fail
   ELSE \quad res := pass
END
\end{verbatim}

(b) Here is some English text to make the specification more readable, where the narrative is illustrated by means of some suitable sets.

The set of students are represented by \( STUDENT \), for example
\( STUDENT = \{ Ayaz, Betty, Chris, Dave, Rose, Mary, Bill, Tom \} \).
The set of students enrolled for a particular class are a subset of this set, for example
\( enrolled = \{ Ayaz, Betty, Chris, Dave, Bill, Tom \} \),
where there must be no more than max\_class\_size enrolled students. (For example max\_class\_size = 6.)

Some of the students have been tested and the grade of each tested student is recorded by means of a function: tested ∈ STUDENT → GRADE. tested is a set of pairs, for example: tested = {Ayaz → A1, Dave → D1}. Each tested student must have been enrolled and this is captured by the restriction that the domain of tested (i.e. the set of first members of the pairs) must be a subset of the enrolled students. In this case the domain of tested is {Ayaz, Dave} so this can be seen to be true.

For the operation Add\_Result to take place, the student input to it must have been enrolled but must not have been tested, so must not belong to the domain of tested. For example stud = Betty. The input grade (grd) must belong to the set of grades (for example grd = B1) The variable tested is updated with the result, and this is achieved by adding the pair Betty → B1 to tested, so that tested becomes {Betty → B1, Ayaz → A1, Dave → D1}. Since the input grade did not belong to the set {E1, E2, E3, FF, GG} then the output for the example is res := pass.

(c) Correct\_Grade(st, grd) =
PRE st ∈ dom tested ∧ grd ∈ GRADE
THEN
tested := tested + {st → grd}
END

(d) This question was incomplete - it did not state the input (an enrolled student). It should also have removed a student from enrolled. The answer provided just removes student from tested. For alternative solution see file of Class\_Man2.mch in /local/public/cas436.

Leaves\_Course(st) =
PRE st ∈ enrolled
THEN IF st ∈ dom tested THEN tested := tested + {st → tested(st)}
END

**Tutorial 5**

1 a) The machine, Air\_Booking models an Aircraft booking system. Sorry - there was no part (a)!

1 b) If SEATS = \{A1..A10, B1..B10, C1..C10, D1..D10\}
NAME = \{jones, rashid, smith, chandler, hawkins, brown, white, black\}
and the current state of Air\_Booking is

\[
\text{seats} = \{A1, A2, A3, B1, B2, B3, C1, C2, C3, D1, D2, D3\}
\]
\[
\text{passengers} = \{B3 → jones\}
\]

the final state of Air\_Booking after the sequential application of

\[
\text{MakeBooking}(B1, \text{rashid}) \text{ i.e. with input seat\_no = B1, name = rashid}
\]
\[
\text{MakeBooking}(B2, \text{smith})
\]
\[
\text{MakeBooking}(C1, \text{hawkins})
\]
is calculated as follows: seats is unaltered – there is no assignment for it after THEN.
Maplets $B1 \mapsto rashid, B2 \mapsto smith, C1 \mapsto hawkins$ are added sequentially so that $\text{passengers} = \{B3 \mapsto jones, B1 \mapsto rashid, B2 \mapsto smith, C1 \mapsto hawkins\}$

1) $\text{DeleteBooking}(\text{seat}_\text{no}) =$
PRE
$\text{seat}_\text{no} \in \text{dom passengers}$
THEN
$\text{passengers} := \text{passengers} - \{\text{seat}_\text{no} \mapsto \text{passengers(\text{seat}_\text{no})}\}$
END

The operation inputs a seat number $\text{seat}_\text{no}$, the passenger booked for this seat number is $\text{passengers(\text{seat}_\text{no})}$ and the pair $\text{seat}_\text{no} \mapsto \text{passengers(\text{seat}_\text{no})}$ is removed from the set of pairs $\text{passengers}$.

1 d) $\text{no\_seats} \leftarrow \text{SeatsSold} =$
BEGIN $\text{no\_seats} := \text{card(\text{dom passengers})}$ END

There is no precondition for this operation since there is no input and the database can be interrogated in any state. BEGIN .. END is for readability and not strictly necessary. The number of seats sold is the number of members in the domain of $\text{passengers}$.

2 a) NB I have only included enough narrative to indicate how it should be done. See also solution to other questions.

$MACHINE$
$\text{CarsForSale}$

$SETS$
$\text{CAR}$

$CONSTANTS$
$\text{NewCars}, \text{SecondHandCars}, \text{Prices}, \text{max\_no\_cars}$

$PROPERTIES$
$\text{NewCars} \subseteq \text{CAR} \land$
$\text{SecondHandCars} \subseteq \text{CAR} \land$
$\text{max\_no\_cars} \in \mathbb{N} \land$
$\text{Prices} \subseteq \mathbb{N}$
$\text{NewCars} \cap \text{SecondHandCars} = \{\}$

$\text{NewCars}, \text{SecondHandCars}$ are subsets of $\text{CAR}$ and they have an empty intersection – no members in common.
2 b) VARIABLES and INVARIANT clauses.

VARIABLES
   
forsale
INVARIANT
   
\( \text{forsale} \in \text{CAR} \rightarrow \mathbb{N} \land \)
   \( \text{dom forsale} \subseteq \text{NewCars} \cup \text{SecondHandCars} \land \)
   \( \text{card}(\text{forsale}) \leq \text{max\_no\_car} \)

The constraint that the union of sets \( \text{NewCars}, \text{SecondHandCars} \) is equal to the domain of the \text{forsale} variable ensures that all cars in the database are either new or secondhand. We have already ensured that \( \text{NewCars}, \text{SecondHandCars} \) have no members in common, so cars for sale cannot be both new and secondhand. Also there is a maximum number of cars, \( \text{max\_no\_car} \) for sale: the number of elements in the domain of \text{forsale} is less than or equal to the maximum number of cars.

2 c) INITIALISATION

\text{forsale} = \{\}

2 d) OPERATIONS

\text{AddNewCar}\,(\text{car}, \text{price}) =

PRE
   \( \text{car} \in \text{NewCars} \land \)
   \( \text{price} \in \text{Prices} \land \)
   \( \text{car} \notin \text{dom forsale} \)

THEN
   \( \text{forsale} := \text{forsale} \cup \{\text{car} \mapsto \text{price}\} \)

END

and the narrative ...

2 e) 

\text{price} \leftarrow \text{NewCarSold}\,(\text{car}) =

PRE
   \( \text{car} \in \text{NewCars} \land \)
   \( \text{car} \in \text{dom forsale} \)

THEN
   \( \text{forsale} := \text{forsale} - \{\text{car} \mapsto \text{forsale}(\text{car})\} \land \)
   \( \text{price} := \text{forsale}(\text{car}) \)

END
and the narrative ...

2 f)

\[
\text{price} \leftarrow \text{SecondHandPrice}(\text{car}) \\
\text{PRE} \\
\quad \text{car} \in \text{SecondHandCars} \land \\
\quad \text{car} \in \text{dom}\ \text{forsale} \\
\text{THEN} \\
\quad \text{price} := \text{forsale}(\text{car}) \\
\text{END}
\]

and the narrative ...