CS412 solutions to exercise sheet 2

Relations

1. $CHILD = \{Anna, Ben, Carol, Dave\}$
   $SUBJECT = \{maths, french, physics, history\}$
   $reg = \{Anna \mapsto maths, Anna \mapsto history, Ben \mapsto maths,$
   $Ben \mapsto physics, Ben \mapsto history, Dave \mapsto maths\}$

   (a) $\{Anna, Ben, Dave\}$

   (b) $\{maths, physics, history\}$

   (c) $\{Anna \mapsto maths, Anna \mapsto history, Ben \mapsto maths, Ben \mapsto french$
   $Ben \mapsto physics, Ben \mapsto history, Dave \mapsto maths\}$

   (d) $\{Ben \mapsto maths, Ben \mapsto physics, Ben \mapsto history\}$

   (e) $\{Anna \mapsto maths, Anna \mapsto history, Dave \mapsto maths\}$

   (f) $\{Anna \mapsto maths, Anna \mapsto history, Dave \mapsto maths\}$

   (g) $\{Anna \mapsto maths, Ben \mapsto maths, Ben \mapsto physics, Dave \mapsto maths\}$

   (h) reg

   (i) $\{maths\}$

   (j) $maths \mapsto Anna, history \mapsto Anna, maths \mapsto Ben,$
   $physics \mapsto Ben, history \mapsto Ben, maths \mapsto Dave\}$

   (k) $\{Anna \mapsto Anna, Anna \mapsto Ben, Anna \mapsto Dave,$
   $Ben \mapsto Ben, Ben \mapsto Anna, Ben \mapsto Dave,$
   $Dave \mapsto Anna, Dave \mapsto Ben, Dave \mapsto Dave\}$

   (l) $\{Anna \mapsto maths, Anna \mapsto history, Ben \mapsto maths,$
   $Ben \mapsto french, Ben \mapsto physics, Ben \mapsto history, Dave \mapsto maths\}$

   (m) $\{Anna \mapsto maths, Anna \mapsto history, Ben \mapsto french,$
   $Dave \mapsto maths\}$

2. (a) There is always a number of ways to express things - you may have alternatives.

   i. $MODULES - \text{ran}(register)$ or perhaps
   $\{m \mid m \in MODULE \land$
   $(\forall s \bullet (s \in STUDENT \Rightarrow s \mapsto m \notin register))\}$

   ii. $register^{-1}[\{cs400\}]$ or
   $\{s \mid s \in STUDENT \land s \mapsto cs400 \in register\}$

   iii. $CBS \lhd register$

   iv. $card(CBS - \text{dom}(register))$

(b) Write B operations:

i. $s\_enrol(ss, mm) \mathrel{\widehat=}$
PRE $ss \in STUDENT \land mm \in MODULE \land ss \mapsto mm \notin register$
THEN $register := register \cup \{ss \mapsto mm\}$
END
The type info could be in the PRE part instead. Could omit the check to see if it's already there if we're not bothered.

ii. $mms \leftarrow s\_mods(ss) \mathrel{\widehat=}$
PRE $ss \in STUDENT$
THEN $mms := register[\{ss\}]$
END

iii. $oo \leftarrow remove(mm) \mathrel{\widehat=}$
PRE $mm \in MODULE$
THEN $register := register \rhd \{mm\} \;||\; oo := register^{-1}[\{mm\}]$
END

iv. You'd probably first need to decide what's required here. The following op lists all new registrations, and all those now gone.

$$added, removed \leftarrow changes \mathrel{\widehat=}$$
$$added := register - prereg \;||\; removed := prereg - register$$

3. MACHINE
    LabSys

```
SETS
    LABS; TOPICS; STUDENTS

ABSTRACT_CONSTANTS
    lab_topics

PROPERTIES
    lab_topics : LABS <-> TOPICS
/* Could have lab_topics as a variable - this way assumes it's fixed. */

VARIABLES
    attendance

INVARIANT
    attendance : STUDENTS <-> LABS

INITIALISATION
    attendance := {}

OPERATIONS
    ll <-- attended(ss) =
       PRE ss:STUDENTS
       THEN ll := attendance[{ss}]
       END;

    nl <-- notattended = nl := LABS - ran(attendance);

    ss <-- notat(ll) =
       PRE ll:LABS
       THEN ss := STUDENTS - (attendance~[{ll}])
       END;

    tt <-- covered(ss) =
       PRE  ss:STUDENTS
       THEN tt := (attendance ; lab_topics)[{ss}]
       END;

   attend_lab(ss,ll) =
       PRE  ss:STUDENTS & ll:LABS & ss |-> ll /: attendance
       THEN attendance := attendance \/ {ss |-> ll}
       END

END
/* I'm being a bit lax here allowing sets as outputs - these in general
* are not implementable so we'd need to be more careful with something
* we were planning to develop. Here, we're really interested in using the
* notation. */
```

4. (a) Here, the constant *lab*1 denotes the special lab, requiring changes to the CONSTANTS and PROPERTIES sections. The INVARIANT is updated to reflect the requirement that no other labs can be attended unless *lab*1 has been attended. The operation to record an attendance must now also check that *lab*1 is taken first.

```
MACHINE
    LabSys2

SETS
    LABS; TOPICS; STUDENTS

ABSTRACT_CONSTANTS
    lab_topics, lab1

PROPERTIES
    lab_topics : LABS <-> TOPICS & lab1: LABS

VARIABLES
    attendance

INVARIANT
    attendance : STUDENTS <-> LABS &
    !xx.(xx:STUDENTS => (attendance[{xx}] = {} or lab1:attendance[{xx}]))

INITIALISATION
    attendance := {}

OPERATIONS
    ll <-- attended(ss) =
       PRE ss:STUDENTS
       THEN ll := attendance[{ss}]
       END;

    nl <-- notattended = nl := LABS - ran(attendance);

    ss <-- notat(ll) =
       PRE ll:LABS
       THEN ss := STUDENTS - (attendance~[{ll}])
       END;

    tt <-- covered(ss) =
       PRE  ss:STUDENTS
       THEN tt := (attendance ; lab_topics)[{ss}]
       END;

   attend_lab(ss,ll) =
       PRE  ss:STUDENTS & ll:LABS & ss |-> ll /: attendance &
        ((lab1 /: attendance[{ss}]) or ll = lab1)
       THEN attendance := attendance \/ {ss |-> ll}
       END
END
```

(b) This time, the variable *registered* has been added to record the set of registered students. The invariant is strengthened so that only registered students may have attended labs. It is initialised to empty. We might decide it makes sense to offer the various query operations only for those registered on the course, and in these cases could add preconditions. Similarly, the *notat* operation may well only record missing students who are not registered. Registration is straightforwartd. When deregistering we must ensure that the invariant is preserved - so either we can deregister only those students who haven't attended labs, or we could alter the attendance record - or we might want to think again about the invariant.

```
MACHINE
    LabSys3

SETS
    LABS; TOPICS; STUDENTS

ABSTRACT_CONSTANTS
    lab_topics, lab1

PROPERTIES
    lab_topics : LABS <-> TOPICS & lab1: LABS

VARIABLES
    attendance, registered

INVARIANT
    attendance : STUDENTS <-> LABS & registered <: STUDENTS &
    !xx.(xx:STUDENTS => (attendance[{xx}] = {} or lab1:attendance[{xx}])) &
    dom(attendance) <: registered

INITIALISATION
    attendance := {} || registered := {}

OPERATIONS
    ll <-- attended(ss) =
       PRE ss:STUDENTS & ss:registered
       THEN ll := attendance[{ss}]
       END;

    nl <-- notattended = nl := LABS - ran(attendance);

    ss <-- notat(ll) =
       PRE ll:LABS
       THEN ss := registered - (attendance~[{ll}])
       END;

    tt <-- covered(ss) =
       PRE  ss:STUDENTS & ss:registered
       THEN tt := (attendance ; lab_topics)[{ss}]
       END;

   attend_lab(ss,ll) =
       PRE  ss:STUDENTS & ll:LABS & ss |-> ll /: attendance &
        ((lab1 /: attendance[{ss}]) or ll = lab1)
```

```
       THEN attendance := attendance \/ {ss |-> ll}
       END;

    reg(ss) =
        PRE ss:STUDENTS & ss /: registered
        THEN registered := registered \/ {ss}
        END;

    dereg(ss) =
        PRE ss:STUDENTS & ss : registered & ss /: dom(attendance)
        THEN registered := registered - {ss}
        END
END
```