CS412 Solution sheet 4

A bit of wp and some sequences

1. (a) $(x > y)[3, 2/x, y] = 3 > 2 = true$

   (b) $(x > y)[x + y, 2x/x, y]$
   $= x + y > 2x$
   $= y > x$

   (c) $(\forall x, y \bullet (x \in \mathbb{N} \wedge y \in \mathbb{N} \Rightarrow P(x, y)))[3/x]$
   $= (\forall x, y \bullet (x \in \mathbb{N} \wedge y \in \mathbb{N} \Rightarrow P(x, y)))$
   There are no free occurrences of x.

   (d) $(\forall x \bullet (x \in \mathbb{N} \Rightarrow (P(x, y) \wedge \exists y \bullet (Q(x, y))))))[3/y]$
   $= (\forall x \bullet (x \in \mathbb{N} \Rightarrow (P(x, 3) \wedge \exists y \bullet (y \in \mathbb{N} \wedge Q(x, y)))))$
   Only the first occurrence of y is free.

2. (a) $[x, y := x + y, 2x]x > y$
   $= \quad x + y > 2x$
   $= \quad y > x$

   (b) $[x := 1\,..\,n \;\|\; y := 1\,..\,m]\,\exists x \bullet (x \subseteq \mathbb{N} \wedge x \subset y)$
   $= \quad \exists x \bullet (x \subseteq \mathbb{N} \wedge x \subset 1\,..\,m)$
   $= \quad m > 0$

   (c) $\left[\begin{array}{l} \textbf{IF} \quad x < y \\ \textbf{THEN} \quad x := y \\ \textbf{END} \end{array}\right] x = y$
   $= \quad (x < y \wedge y = y) \vee (x \geq y \wedge x = y)$
   $= \quad x < y \vee ((x > y \vee x = y) \wedge x = y)$
   $= \quad x < y \vee (x > y \wedge x = y) \vee (x = y \wedge x = y)$
   $= \quad x < y \vee false \vee (x = y \wedge x = y)$
   $= \quad x < y \vee x = y)$
   $= \quad x \leq y$

   (d) $\left[\begin{array}{l} \textbf{PRE} \quad x < y \\ \textbf{THEN} \quad x := y \\ \textbf{END} \end{array}\right] x = y$
   $= \quad x < y \wedge y = y$
   $= \quad x < y$

   (e) $\left[\begin{array}{l} \textbf{CASE} \quad \mathit{clearance} \quad \textbf{OF} \\ \textbf{EITHER} \quad \mathit{unk} \quad \textbf{THEN} \quad \mathit{action} := \mathit{alert} \\ \textbf{OR} \quad \mathit{conf} \quad \textbf{THEN} \quad \mathit{permission} := \mathit{denied} \\ \textbf{OR} \quad \mathit{sec} \quad \textbf{THEN} \quad \mathit{permission} := \mathit{denied} \\ \textbf{ELSE} \quad \mathit{permission} := \mathit{granted} \\ \textbf{END} \end{array}\right] \mathit{permission} = \mathit{granted}$
   $= \quad (\mathit{clearance} = \mathit{unk} \Rightarrow \mathit{permission} = \mathit{granted} \;\wedge$

$$
\begin{aligned}
& \quad clearance = conf \Rightarrow denied = granted \;\wedge \\
& \quad clearance = sec \Rightarrow denied = granted \;\wedge \\
& \quad clearance = tops \Rightarrow granted = granted) \\
= & \; (clearance = unk \Rightarrow permission = granted \;\wedge \\
& \quad clearance \notin \{conf, sec\} \;\wedge \\
& \quad clearance \notin \{conf, sec, unk\} \Rightarrow true) \\
= & \; (clearance = unk \Rightarrow permission = granted \;\wedge \\
& \quad clearance \notin \{conf, sec\})
\end{aligned}
$$

3. $s_1 = [1, 1, 0]$ and $s_2 = [1, 0, 1]$

   (a) Write s_1 and s_2 using set notation.
   $s_1 = \{1 \mapsto 1, 2 \mapsto 1, 3 \mapsto 0\}$ and $s_2 = \{1 \mapsto 1, 2 \mapsto 0, 3 \mapsto 1\}$

   (b) What are:

      i. $s_1 \frown s_2 = [1, 1, 0, 1, 0, 1]$
      ii. $s_1 \cup s_2 = \{1 \mapsto 1, 2 \mapsto 0, 2 \mapsto 1, 3 \mapsto 0, 3 \mapsto 1\}$
      iii. $s_1 \cap s_2 = \{1 \mapsto 1\} = [1]$
      In this case, the answer is a sequence, but in general it won't be.
      iv. $s_1(2) = 1$
      v. $s_1 \oplus s_2 = s_2$
      vi. $rev\ (s_1 \frown s_2) = [1, 0, 1, 0, 1, 1]$
      vii. $s_1 \rhd \{1\} = \{1 \mapsto 1, 2 \mapsto 1\} = [1, 1]$
      Another case where the result would not generally be a sequence.
      viii. $s_1 \ntriangleright \{1\} = \{3 \mapsto 0\}$
      ix. $head\ (tail\ (s_2)) = 0$
      x. $\lambda\, i \bullet (i \in \mathbb{N}_1 \mid i + 1) = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, \ldots]$

4. Show how each of the following can be defined using the old set, relation, function notation.

   (a) $\mathit{last}(ss) = ss(\mathit{card}(ss))$

   (b) $\mathit{tail}(ss) = \{nn, ii \mid nn \in 1 \mathinner{..} \mathit{card}(ss) - 1 \wedge ii = ss(nn+1)\}$

   (c) $\mathit{perm}(X) = \{ss \mid ss : \mathrm{iseq}(X) \wedge \mathit{ran}(ss) = X\}$
   Or what about: $\mathit{perm}(X) = \mathrm{iseq}(X) \cap (\mathbb{N} \twoheadrightarrow X)$

5. Machine to represent a nondeterministic communication system

```
MACHINE           Comms

SETS              USR;MSG

VARIABLES         intransit, mailboxes

INVARIANT         intransit : POW(MSG * USR) &
                  mailboxes : USR --> seq(MSG)

INITIALISATION    intransit := {} || mailboxes := USR * {<>}

OPERATIONS

  send(mm:MSG;uu:USR) = intransit := intransit \/ {mm |-> uu};

/* I'm assuming here that messages are all distinct, perhaps by
   means of a timestamp.
*/

  deliver =
    ANY     xx, yy
    WHERE   xx:MSG & yy:USR & xx |-> yy : intransit
    THEN    intransit := intransit - {xx |-> yy} ||
            CHOICE  mailboxes(yy) := mailboxes(yy) <- xx
            OR      skip
            END
    END

/* I've chosen to represent the nondeterminism at this stage. A message
   can either be delivered - or it can be lost.
*/

END
```