CS412 Exercise sheet 5

Weakest precondition and machine consistency

1. Calculate and simplify (assuming a type-correct context):

   (a) $[xx :\in \{ii \mid ii \in \mathbb{N} \wedge ii < 5\}]xx < 2$

   (b) $\left[\begin{array}{l} \mathit{ANY}\ xx \\ \mathit{WHERE}\ xx : \mathit{MID} \wedge xx : S \\ \mathit{THEN}\ members := members \cup \{xx \mapsto 62\} \\ \mathit{END} \end{array}\right] members \in \mathit{MID} \twoheadrightarrow \mathbb{N}$

   (c) $\left[\begin{array}{l} \mathit{ANY}\ xx \\ \mathit{WHERE}\ xx : \mathbb{N} \wedge xx * xx < 10 \\ \mathit{THEN}\ yy := yy + xx \\ \mathit{END} \end{array}\right] yy < 8$

   (d) $[\mathit{CHOICE}\ mm := xx\ \mathit{OR}\ mm := yy\ \mathit{END}]mm = xx$

2. Show that $wp(P, Q) \vee wp(P, R)$ is *NOT* always equal to $wp(P, Q \vee R)$.
   Hint: think of a simple nondeterministic operation such as a coin toss.

3. What are:

   (a) $[\textbf{IF}\ xx \neq 0\ \textbf{THEN}\ oo := val\ /\ xx\ \textbf{END}]R$

   (b) $[\textbf{PRE}\ xx \neq 0\ \textbf{THEN}\ oo := val\ /\ xx\ \textbf{END}]R$

   How are these different and why?
   If $xx = 0$ initially, under what circumstances can the **IF** be guaranteed to establish R?
   If $xx = 0$ initially, under what circumstances can the **PRE** be guaranteed to establish R?
   What implications does this have for any implementation of these operations?

4. The *entrysys* machine has definitions:

   MACHINE *entrysys*
   SETS *PID*
   VARIABLES $inside, maxin$
   INVARIANT $inside \subseteq PID\ \wedge\ maxin \in \mathbb{N}_1\ \wedge\ card(inside) \leq maxin$
   INITIALISATION $inside := \{\}\ \ ||\ \ maxin := 500$
   OPERATIONS
   $\quad ww \leftarrow whosin\ \ \widehat{=}\ \ ww := inside;$
   $\quad enter(pp)\ \ \widehat{=}$ **PRE** $pp \in PID\ \wedge\ pp \notin inside\ \wedge\ card(inside) < maxin$
   THEN $inside := inside \cup \{pp\}$
   END
   END For this question work on paper using the proof conditions as given in lectures.

(a) An operation is added to allow the maximum limit on people in the building to be changed:

$$change\_lim(nn) \;\;\widehat{=}\;\; \textbf{PRE}\ nn : \mathbb{N}_1\ \textbf{THEN}\ maxin := nn\ \textbf{END}$$

Write down the correctness requirement for this operation and demonstrate that the operation is *incorrect* in its current form.

(b) If a proof fails for an operation, it may be appropriate to alter the invariant, or the operation, or both. What would you do in this case?

(c) It is proposed that a register be kept of those eligible to enter the building. For each identifier of a person allowed to enter the building the register will record the associated name and staff category. Adjust the machine to add this and define an operation, *catin*, to output the names of everyone in a given category currently in the building.

(d) Why is verification of consistency for operations such as *catin* trivial?

5. Check out the entrysys machine from above in the BToolkit. Generate the POs and see if the tool can prove them. If not -inspect the obligations that don't prove. Are there errors in your machine...?