Exercise 1 Heap-manipulating OO programs

Exercise Sheet 12: Specification and Verification with Higher-Order Logic (Summer Term 2014)

In this exercise we will use the calculus from the lecture to do some pen- and paper exercises about the following JavaKE program:

```
1 interface Cell {
2    int set(int val);
3    int get();
4 }

6 class StandardCell implements Cell {
7    int x;
8    int set(int par) {
9        x = par;
10    };
11    int get() {
12        res = x;
13    }
14 }
15 }

7 class PrevCell implements Cell {
8    boolean f;
9    int x1;
10   int x2;
11   int set(int par) {
12       boolean t;
13       t = f;
14       f = !t;
15       if (!t) x1 = par;
16       else x2 = par;
17   }
18   int get() {
19       if (f) res = x1;
20       else res = x2;
21   }
22   int getPrev() {
23       if (f) res = x2;
24       else res = x1;
25   }
26 }
```

a) Give a weak precondition \( P \) so that the following triple is valid in the context of the above program:

\[
\{ P \}
\begin{align*}
& c1.set(4); \\
& c2.set(2); \\
& \text{int } x = c1.get(); \\
& \text{int } y = c2.get(); \\
& \text{res } = x \cdot 10 + y; \\
& \{ \text{res } = 42 \}
\end{align*}
\]

b) Define a heap abstraction \( \text{cell}(e,x) \) which states that reference “\( e \)” points to a cell object with a current value of “\( x \)”.

c) Prove the following statement on paper, using the rules from the lecture:

\[
\{ c \neq \text{null} \} r = c.\text{set}(x) \{ \text{cell}(c,x) \}
\]

You can assume that the static type of \( c \) in this program part is \( \text{Cell} \).

For every step in the proof, write down the name of the used rule.

Try to get to the following intermediate goal first:

\[
\{ c \neq \text{null} \land c = \text{C} \land x = X \} r = c.\text{set}(x) \{ \text{cell}(C,X) \}
\]