Wednesday, 6 April 2022 11.17

C12.1 Verify the claim in Section 12.2 of Lecture 12 that for all $n \geq 0$, $m_0 \in \mathbb{Z}$ the following state-transformation relationship holds:

$$\langle i = n, m = m_0 \rangle$$
$$[\![\textbf{while } i > 0 \textbf{ do } m \leftarrow 2 * m; i \leftarrow i - 1 \textbf{ od}]\!]$$
$$\langle i = 0, m = m_0 * 2^n \rangle$$

(*Hint:* Induction on n.)

C12.2 Verify the following loop invariant claims in Section 12.4 of Lecture 12 hold for any $N \in \mathbb{Z}$:

(i)

$$\{m * 2^i = N\} m \leftarrow 2 * m;\ i \leftarrow i - 1 \{m * 2^i = N\}$$

(ii)

$$\{m * 2^i = N\}$$
$$\textbf{while } i > 0 \textbf{ do } m \leftarrow 2 * m;\ i \leftarrow i - 1 \textbf{ od}$$
$$\{m * 2^i = N\}$$

You don't need to go back to the detailed semantics of the programming language here; commonsense mathematical reasoning suffices.

12.4 Proving weak correctness via loop invariants

- An essential challenge in establishing the (weak) correctness of a program P is proving the correctness of the **while-do** loops.
- This can be done by means of *loop invariants*, which are also a useful way of thinking about loop design in everyday practical programming.
- A predicate I is an *invariant* for a program S if $\{I\}S\{I\}$.
- For example, one can easily verify that the predicate

$$I(m,i): \{m * 2^i = N\},$$

for any constant $N \in \mathbb{Z}$, is an invariant for the body of the loop in our previous exponentiation program:

$$\{m * 2^i = N\} m \leftarrow 2 * m;\ i \leftarrow i - 1 \{m * 2^i = N\}$$

Aalto University School of Science

CS-C2160 Theory of Computation / Lecture 12
Aalto University / Dept. Computer Science

- A program S is *weakly correct* with respect to a specification $\langle P, Q \rangle$, denoted $\{P\}S\{Q\}$, if given an initial state ω where $P(\omega)$ holds, program S will transform it into a state ω' where $Q(\omega')$ holds, assuming S halts.

Here the loop invariant is the predicate
$\{m * 2^i = N\}$

C12.2.

(i) Let $N \in \mathbb{Z}$ be arbitrary.
Then for precondition of state ω $\{N = m * 2^i\}$
we get that the assignment
$S = m \leftarrow 2 * m;\ i \leftarrow i - 1$
transforms it into state ω'. We have
$N = m * 2^{i-1} * 2 = m * 2^{i-1+1} = m * 2^i$.

(ii) Induction on the "number of iterations" of our while-loop.
Base case: holds by the given spesification $\langle P, Q \rangle$.
Induction: assume it holds for the kth iteration of the while loop.
Then for $k+1$st while loop:
By induction assumption and part (i) we get
$N = m * 2^{i-1} * 2 = m * 2^i$