# Exercises 3: Dynamic Logic

TECHNISCHE UNIVERSITÄT DARMSTADT

**The solutions to the exercises will be discussed on Monday, 18th May.**

## Problem 1 Interpreting Dynamic Logic Formulas

What is the meaning of the following DL formulas? Are the formulas valid? Give a brief justification for your answers. We consider the type **int** to be the mathematical whole numbers, i.e. without overflow. You may assume the following definitions:

```
\programVariables {
  int i, old_i, j;
  boolean b;
}
```

a) $(\texttt{i} > \texttt{j}) \rightarrow \langle \texttt{j = j - i;} \rangle (\texttt{j} < 0)$

b) $(i > 0) \rightarrow \langle \texttt{while ( i != 0 ) \{ i = i - 2; \}} \rangle (\texttt{i} \doteq 0)$

c) $[\texttt{while ( i != 0 ) \{ i = i - 2; \}}](\texttt{i} \doteq 0)$

d) $(\texttt{old\_i} \doteq \texttt{i}) \rightarrow \langle \texttt{j = 0; while (i > 0) \{ j++; i = i - 1; \}} \rangle (\texttt{i} \doteq 0 \rightarrow \texttt{j} \doteq \texttt{old\_i})$

e) $\exists$ **boolean** bool; $\big(\texttt{b} \doteq \texttt{bool} \rightarrow \langle \texttt{if (b) \{ i = 10; \} else \{ j = -10; \}} \rangle (\texttt{i} > \texttt{j})\big)$

f) $\exists$ **boolean** bool; $\langle \texttt{b = bool; if (b) \{ i = 10; \} else \{ j = -10; \}} \rangle (\texttt{i} > \texttt{j})$

## Problem 2 Semantics of Dynamic Logic

Justify formally (using the semantics definition) the following equivalence:

$$\langle \texttt{p} \rangle \phi \text{ iff. } \neg [\texttt{p}] \neg \phi$$

## Problem 3 Updates

Simplify the updates of the following formulas using the update simplification rules of the previous lecture:

- $\{x := x + y\}\{y := x + y\}\langle \texttt{p} \rangle \phi$
- $\{x := x + y\}\{x := 3\}\langle \texttt{p} \rangle \phi$

Assume that neither program p nor formula $\phi$ containing program variable x. Which other simplification rule would be possible? Prove that the suggested simplification rule is sound.

## Problem 4 Unwind-Loop rule

The unwindLoop rule as presented in the lecture is a simplified version of the actual one for Java as it does not consider continues, breaks, returns etc. Provide a version of the unwindLoop rule for loops with labeled break statements.