# DD2457 Program Semantics and Analysis

Examination Problems
29 May 2017, 14:00 - 19:00

*Dilian Gurov*
tel: 08-790 8198

Give solutions in English or Swedish, each problem beginning on a new sheet. Write your name on all sheets. The maximal number of points is given for each problem. The total number of points is 30. Up to two bonus points per section will be taken into account. The course book, the handouts, own notes taken in class, as well as reference material are admissible at the exam.

## 1 Level E

For passing level E you need 8 (out of 10) points from this section.

Consider the following program:

$$x := 0;\ \textbf{while}\ \neg(y \leq 0)\ \textbf{do}\ x := x + 1;\ y := y - 1$$

which has the purpose to copy (incrementally) the initial value of variable $y$ to the final value of variable $x$, implicitly assuming that the former is not negative.

1. Specify the program by means of a *Hoare triple*. The specification should make sense on its own, i.e., it should tell the user how to use the program without knowing the implementation. $\boxed{2\text{p}}$

2. Suggest a *loop invariant* that is suitable for verifying the program. $\boxed{1\text{p}}$

   Your next task will be to verify the program against your specification and loop invariant by means of *symbolic execution* (see handouts).

3. That is, first, convert the Hoare triple and loop invariant into a *loop-free program* in the **While** language extended with **assume**, **assert** and **havoc** statements. $\boxed{1\text{p}}$

4. Next, explore all paths. $\boxed{3\text{p}}$

5. And finally, collect all resulting *verification conditions* and argue semi-formally for their validity. $\boxed{3\text{p}}$

   Note that if you chose too weak a loop invariant, this would show itself only here, resulting in verification conditions that are not valid (the right-hand side of the implication will not be a logical consequence of the left-hand side).

## 2 Level C

For grade D you need to have passed level E and obtained 5 (out of 12) points from this section. For passing level C you need 8 points from this section.

1. Consider the *denotational semantics* of **While**. In class, we suggested the following definition for the **repeat** $S$ **until** $b$ statement: 6p

$$\mathcal{S}_{ds}[\![\textbf{repeat } S \textbf{ until } b]\!] \stackrel{\mathsf{def}}{=} \mathsf{FIX}\ H_{S,b}$$

where:

$$H_{S,b}\ (g) \stackrel{\mathsf{def}}{=} \textbf{cond}(\mathcal{B}[\![b]\!]\,, id, g) \circ \mathcal{S}_{ds}[\![S]\!]$$

and where:

$$\mathsf{FIX}\ H_{S,b} = \bigcup_{i \geq 0} H^{i}_{S,b}(\emptyset)$$

Use denotational semantics to prove that the statement:

$$\textbf{repeat } S \textbf{ until } b$$

is *equivalent* to the unfolding:

$$S;\ \textbf{if } b \textbf{ then skip else } (\textbf{repeat } S \textbf{ until } b)$$

2. For a Hoare triple $\{P\}\,S\,\{Q\}$, the pair $(P, Q)$ is often called the *contract* of the program $S$. In fact, for many applications it is meaningful to separate a contract $C = (P, Q)$ from its possible implementations. We can then say that a particular implementation $S$ *meets* its contract $C$, denoted $S \models_{par} C$, if and only if the corresponding Hoare triple is semantically valid, that is $\models_{par} \{P\}\,S\,\{Q\}$. 6p

   Now, your task is to provide a *denotational semantics for contracts*. That is, define the denotation $\mathcal{S}_{ds}[\![C]\!]$ of a contract $C = (P, Q)$ as a single mathematical object (of what type?), in the spirit of how we defined the denotation of statements. Your definition should allow the relation $S \models_{par} C$ to be equivalently expressed as a simple relationship between the denotations of $S$ and $C$. Show this relationship and motivate your reasoning.

## 3 Level A

For grade B you need to have passed level C and obtained 3 (out of 8) points from this section. For grade A you need 6 points from this section.

Consider again the *denotational semantics* of **While** and its extension with **repeat** $S$ **until** $b$ as proposed in Problem C1 above.

1. Explain what the $i$-th approximant of $H_{S,b}$ captures. 2p

2. Compute iteratively the denotation of the statement **repeat** $x := x - 2$ **until** $x \leq 0$. That is, start by simplifying $H_{x:=x-2,\ x\leq 0}\ (g)$ as much as possible, by evaluating all occurrences of semantic functions. Then, compute the first few fixed-point approximants of $H_{x:=x-2,\ x\leq 0}$, and guess the $i$-th approximant. Finally, present the fixed point, which is also the denotation of the statement. Simplify all denotations as much as possible. 6p

---

*Good luck!*