DD2457 Program Semantics and Analysis

Dilian Gurov
KTH CSC
tel: 08-790 8198

Examination Problems
14 December 2008, 14:00 - 19:00

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. 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 6 points from this section.

Consider the extension of **While** with non–deterministic choice S_1 **or** S_2 discussed in class and in the book.

1. In the structural operational semantics of this extended language, compute the *configuration graph* of the program $\boxed{\text{5p}}$

   $$\textbf{while } (0 \leq x) \wedge (x \leq 1) \textbf{ do } (x := x - 1 \textbf{ or } x := x + 1)$$

   from a state s such that $s(x) = 0$. Draw the graph as informatively as possible. Every transition of the graph should be justified by a derivation (but you can point out and omit derivations that are almost identical to an existing one).

2. Consider the following two types of termination properties for a (possibly non–deterministic) statement S from a state s: $\boxed{\text{3p}}$

   - *possible termination*, meaning that there is a terminating execution, and
   - *necessary termination*, meaning that all executions terminate.

   Formalise the two termination properties in both natural semantics and structural operational semantics. If you consider that it is not possible to formalise some case, give a justification why.

2 Level C

For grade D you need to have passed level E and obtained 4 points from this section. For passing level C you need 7 points from this section.

1. Recall the extension of **While** with division a_1/a_2 and exception–handling **try** S_1 **catch** S_2 considered in the first laboratory assignment. To adapt the semantics of **While**, we added the special error value $\perp$ to the set of integer values, letting $Z_\perp \overset{\text{def}}{=} Z \cup \{\perp\}$, and re-defined the evaluation function $\mathcal{A} : \textbf{AExp} \to (\textbf{State} \to Z_\perp)$ to capture division by zero as the source of producing an exception, and propagation of the error value by all arithmetic operations. Similarly, we added $\perp$ to the set of truth values, letting $\mathbf{T}_\perp \overset{\text{def}}{=} \mathbf{T} \cup \{\perp\}$, and re-defined the evaluation function $\mathcal{B} : \textbf{BExp} \to (\textbf{State} \to \mathbf{T}_\perp)$ so that the error value propagates. Finally, to distinguish between normal and exceptional termination, we introduced the set of extended states $\textbf{EState} \overset{\text{def}}{=} \textbf{State} \times \{\top, \perp\}$, where an extended state $(s, \top)$ is normal and $(s, \perp)$ is exceptional. By abuse of notation, we decided to let s denote the normal state $(s, \top)$, and $\hat{s}$ denote the exceptional state $(s, \perp)$.

(a) Adapt the direct style *denotational semantics* of statements to the extended language (assuming that mappings $\mathcal{A}$ and $\mathcal{B}$ are already adapted suitably, for instance as you have done in the laboratory assignment). Show only the changed or added defining clauses. $\boxed{\text{3p}}$
Hint: statement denotation is now of type $\mathcal{S}_{ds} : \textbf{Stm} \to (\textbf{EState} \hookrightarrow \textbf{EState})$. You can have separate defining clauses for normal and exceptional states.

(b) Use your denotational semantics to compute the denotation of the program: $\boxed{\text{2p}}$

$$x := 7; \textbf{try } x := x - 7; x := 7/x; x := x + 7 \textbf{ catch } x := x - 7$$

applied to an arbitrary normal initial state s.

2. Consider again the extension of **While** with non–deterministic choice S_1 **or** S_2.

(a) For a post–condition Q, express the weakest liberal pre–condition $wlp(S_1 \textbf{ or } S_2, Q)$ compositionally, that is in terms of the weakest liberal pre–conditions for S_1 and S_2. Justify your answer! $\boxed{\text{1p}}$
Note: we are taking the intesional view to Hoare logic, so pre– and post–conditions are assertions.

(b) Guided by your answer, extend the verification condition generator discussed in class by adding a defining clause for $vcg\,[\![S_1 \textbf{ or } S_2]\!]\,(P, Q)$. $\boxed{\text{1p}}$

(c) Verify the Hoare triple $\boxed{\text{3p}}$

$$\{\mathit{true}\}\, \textbf{while } (0 \le x) \wedge (x \le 1) \textbf{ do } (x := x - 1 \textbf{ or } x := x + 1)\, \{x < 0 \vee x > 1\}$$

by extracting a verification condition with your verification condition generator and justifying the verification condition.
Note: you need first to annotate the while loop with a suitable loop invariant.

3 Level A

For grade B you need to have passed level C and obtained 5 points from this section. For grade A you need 8 points from this section.

1. Show that statement **while** b **do** (**if** b **then** S_1 **else** S_2) is semantically equivalent to statement **while** b **do** S_1. Base your proof on a semantic style of your choice. $\boxed{\text{3p}}$

2. In the denotational semantics you developed above for **While** extended with division and exception handling, compute the denotational semantics of the statement

$$\textbf{while } 0 \le x \textbf{ do } x := x/y; y := y - 1$$

That is:

(a) determine the functional F for this loop, simplifying as much as possible; $\boxed{\text{3p}}$

(b) compute the first two approximants in the iterative fixed–point construction and explain intuitively their meaning; $\boxed{\text{3p}}$

(c) guess the i–th approximant and explain intuitively its meaning; $\boxed{\text{2p}}$

(d) present the denotation of the loop as the limit of the construction and explain intuitively its meaning. $\boxed{\text{1p}}$

Good luck!