DD2457 Program Semantics and Analysis

Dilian Gurov
KTH CSC
tel: 08-790 8198

EXAMINATION PROBLEMS
26 May 2011, 8:00 - 13:00

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 (out of 8) points from this section.

1. In a first step from a structural operational semantics (SOS) to an abstract machine semantics (AM), we can start by refining the rules just enough to obtain axiom rules only. Observe that in our SOS for **While** it is just the rules for sequential composition that are not axioms! To accomplish this first step, we can change the configurations to be of shape $\langle \gamma, s \rangle$, where $\gamma \in \mathbf{Stm}^*$ is a (possibly empty) sequence of statements (instead of a single statement). Then, the rule for **skip** could look like this: $\boxed{6p}$

$$[\mathsf{skip}_{\mathsf{SAM}}] \quad \langle \mathbf{skip} : \gamma, s \rangle \Rightarrow \langle \gamma, s \rangle$$

   The idea is that the rules always inspect the head of the statement sequence only. Initial configurations are again of shape $\langle S, s \rangle$, i.e. they have a single–element list as a first component, while final configurations are of shape $\langle \varepsilon, s \rangle$, where ε denotes the empty sequence, or (by abuse of notation) are simply written as s.

   (a) Complete the suggested operational semantics (SAM) in the style discussed above. Note that it has to cosist of axiom rules only!

   (b) Use your semantics to execute the program:

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

   from a state s such that $s(x) = 17$ and $s(y) = 5$. Since there are axiom rules only, program executions can be presented in the style we used for abstract machines, i.e. simply as sequences of configurations from an initial to a final configuration.

2. Consider the following program for computing factorials: $\boxed{2p}$

$$y := 1; \mathbf{while}\ \neg(x = 0)\ \mathbf{do}\ (y := y * x; x := x - 1)$$

   Specify the program by means of a Hoare triple, i.e. with a precondition and a postcondition. Make sure that from the specification it is clear how to use the program without knowing its implementation.

   Please note that there is a later problem in Section C that is based on your solution!

2 Level C

For grade D you need to have passed level E and obtained 4 (out of 10) points from this section. For passing level C you need 7 points from this section.

1. Consider the extension of the **While** programming language with the non–determinstic construct S_1 **or** S_2 discussed in class and in the course book. Adapt the denotational semantics of this extension of **While** to non–deterministic programs. Note that the domain of statement denotations now becomes the set of *total* maps $\boxed{5\text{p}}$

$$\mathbf{State} \to \mathcal{P}(\mathbf{State})$$

with the meaning: $s' \in \mathcal{S}_{ds}[\![S]\!]\,(s)$ whenever there is an execution of S from s that terminates in s'.

Please note that there is a later problem in Section A that is based on your solution!

2. Consider the program for computing factorials and the specification you gave to it in Problem 2 of Section E. $\boxed{5\text{p}}$

   (a) Suggest a suitable loop invariant.

   (b) Use the verification condition generator presented in class (see handouts) to generate a verification condition from your annotated program.

   (c) Justify the obtained verification condition, i.e. argue for its validity.

---

3 Level A

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

1. Use your denotational semantics from Problem 1 in Section C to compute the denotation of the following program: $\boxed{7\text{p}}$

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

That is:

   (a) Determine the functional F for this loop, simplifying as much as possible to eliminate any use of semantic functions.

   (b) Apply the iterative fixed–point construction to compute approximants until reaching a fixed–point. The construction should converge for this program, right?

   (c) Explain intuitively the meaning of the program as suggested by its denotation.

2. Use structural induction to show that for all statements $S \in \mathbf{Stm}$ and assertions $P \in \mathbf{Assn}$ we have: $\boxed{5\text{p}}$

$$\vdash_{par} \{P\}\, S\, \{true\}$$

where $\vdash_{par}$ denotes derivability in the deductive system for partial correctness of the axiomatic semantics of **While** (see handouts).

Good luck, and please fill out the course evaluation form (see course web page)!