Lund University
Computer Science
Lennart Andersson

Examination
DAT130, EDA145
2007–05–29, 8.00–12.00

Examination in Programming language theory

This exam has 6 problems, each worth 5 marks. For passing the exam at most 15 marks will be required.

The following texts may be used during the exam:
Nielson, Nielson, Semantics with Applications.
Andersson, Programming language theory, Lecture notes.

---

1 Consider arithmetic expressions with abstract grammar

$$a ::= n \mid x \mid a_1 + a_2$$

Let substitution be defined as in Nielsen, page 16.

Show by structural induction that $(a[x \mapsto a_1])[x \mapsto a_2] = a[x \mapsto (a_1[x \mapsto a_2])]$ for all a.

2 In a *guarded command*

```
if  b₁ → S₁
 ▌  b₂ → S₂
fi
```

$$\begin{array}{llll} \texttt{if} & b_1 & \rightarrow & S_1 \\ \mathbf{|} & b_2 & \rightarrow & S_2 \\ \texttt{fi} & & & \end{array}$$

the conditions b_1 and b_2 may be true or false independently. If just one of them is true the corresponding statement is executed, if both are true one of the statements is chosen nondeterministicly for execution, and if both are false none of them is executed.

Extend **While** with this statement giving it structural operational semantics.

3 Assume that S_1 and S_2 are equivalent according to the natural operational semantics for **While**. Show that `while` b `do` S_1 is equivalent to `while` b `do` S_2.

4 Lists can be represented in λ-calculus in the following way.

First, a triple can be represented by $\langle P, Q, R \rangle \triangleq \lambda t.t\,P\,Q\,R$, e.g. $\langle \lambda x.x, \lambda x.x, \lambda x.x \rangle \triangleq \lambda t.t\,(\lambda x.x)\,(\lambda x.x)\,(\lambda x.x)$.

The components of a triple can be extracted applying it to $\lambda xyz.x$, $\lambda xyz.y$ and $\lambda xyz.z$.

The empty list is represented by $[\,] \triangleq \langle T, I, I \rangle$.

A nonempty list with head (first element) M and tail N is represented by

$$[M|N] \triangleq \langle F, M, N \rangle$$

F, T and I are defined in the section on λ-calculus of the Lecture notes. $[\,]$ and $[M|N]$ are just convenient names for the combinators inspired by Prolog syntax.

Define a combinator that will append two lists.

5 Extend **While** with a loop statement `loop` S_1; `exitif` $b; S_2$ `pool`. Informally $S_1; S_2$ will be executed repeatedly and the execution of the statement will terminate before executing S_2 the first time the expression b has the value `tt`.

Define the semantics of this statement by extending the rules for partial correctness (Table 6.1). The new rule(s) should not refer to any `while` statement.

6 Let

$$g_i\ \sigma \triangleq \begin{cases} \sigma[\mathtt{y} \mapsto ?] & \text{if } \sigma\,\mathtt{x} < i \text{ and } \sigma\,\mathtt{x} \neq 0 \\ \underline{\text{undef}} & \text{otherwise} \end{cases}$$

where $i \in \mathbb{Z}$ is an integer and ? is a meaningful expression.

Select the expression such that $Y \triangleq \{\, g_i \mid i \in \mathbb{Z} \,\}$

a) is a chain

b) is not a chain

in the ccpo ($\mathbf{State} \hookrightarrow \mathbf{State}, \subseteq$). Both answers require explanations.