Lund University and Institute of Technology
Computer Science
Lennart Andersson

2005–05–28, 8.00–12.00
DAT130, EDA145
Examination

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. Provide a derivation tree for $\langle S, \sigma \rangle \to \sigma'$, where $S = $ `while` $1 \leq$ `x do (s:=s+y; x:=x-1)`, $\sigma = [s \mapsto 0, x \mapsto 2, y \mapsto 5]$, and some σ' using the natural semantics for **While**.
2. Define structural operational semantics for **While** extended with `if` b `then` S without relying on the semantics for the standard two-branch `if`-statement.
3. Show that $(\lambda x \,.\, \lambda y \,.\, y(x\ x\ y))(\lambda x \,.\, \lambda y \,.\, y(x\ x\ y))$ is a fixed point combinator.
4. Assume that $\mathcal{S}'_{cs}[\![S_i]\!]\ c\ \sigma = c(\mathcal{S}_{ds}[\![S_i]\!]\ \sigma)$ for $i = 1, 2$. Show that $\mathcal{S}'_{cs}[\![S_1; S_2]\!]\ c\ \sigma = c(\mathcal{S}_{ds}[\![S_1; S_2]\!]\ \sigma)$.
5. If you want to construct a program that can make proofs using axioms and rules like those in Table 2.1 in Nielson you need to represent axioms and rules by appropriate data types. Define such data types in Haskell or corresponding classes in Java or C++. Assume that there are data types (classes) for representing statements, **Stm**, and states, **State**. You may ignore that some rules have a side condition. You will get a bonus if your representation includes this condition. In this case you may assume that there is a data type representing boolean expressions, **Bexp**.
6. List all monotone functions in $\{\,a, b, c\,\} \to \{\,A, B\,\}$, where $a \sqsubseteq b \sqsubseteq c$ and $A \sqsubseteq B$ and show that one of them is continuous.