Examination, Models of Computation (DIT310/DIT311/TDA184)

- Date and time: 2019-01-16, 8:30–12:30.
- Author/examiner: Nils Anders Danielsson. Telephone number: 1680. Visits to the examination rooms: ∼9:30 and ∼11:30.
- Authorised aids (except for aids that are always permitted): None.
- The GU grades Pass (G) and Pass with Distinction (VG) correspond to the Chalmers grades 3 and 5, respectively.
- To get grade n on the exam you have to be awarded grade n or higher on at least n exercises.
- A completely correct solution of one exercise is awarded the grade 5. Solutions with minor mistakes *might* get the grade 5, and solutions with larger mistakes might get lower grades.
- Exercises can contain parts and/or requirements that are only required for a certain grade. To get grade n on such an exercise you have to get grade n or higher on every part marked with grade n or lower (and every unmarked part), and you have to fulfil every requirement marked with grade n or lower (as well as every unmarked requirement).
- Do not hand in solutions for several exercises on the same sheet.
- Write your examination code on each sheet.
- Solutions can be rejected if they are hard to read, unstructured, or poorly motivated.
- After correction the graded exams are available in the student office in room 4482 of the EDIT building. If you want to discuss the grading you can, within three weeks after the result has been reported, contact the examiner and set up a time for a meeting (in which case you should not remove the exam from the student office).

1. (a) *For grade 3:* Give an example of a set A for which $(\mathbb{N} \to A) \to A$ is countable, and give an example of a set B for which $(\mathbb{N} \to B) \to B$ is not countable. You do not need to provide proofs.

   (b) *For grade 4:* Either prove that the set

   $$(\{ 0 \} \to \mathbb{N}) \to (\{ 0 \} \to \mathbb{N})$$

   is countable, or that it is not countable. You can use theorems from the lecture slides without providing proofs for them.

2. Give concrete syntax for the χ expression e for which the standard χ encoding (as presented in the lectures), given using concrete syntax, is

   $$\begin{aligned}\ulcorner e \urcorner = \mathsf{Case}(&\mathsf{Var}(\mathsf{Zero}()),\\ &\mathsf{Cons}(\mathsf{Branch}(\mathsf{Suc}(\mathsf{Zero}()),\\ &\qquad\qquad \mathsf{Cons}(\mathsf{Zero}(), \mathsf{Nil}()),\\ &\qquad\qquad \mathsf{Var}(\mathsf{Zero}())),\\ &\quad \mathsf{Nil}()))).\end{aligned}$$

   Assume that the number 0 corresponds to the variable x, and that the number 1 corresponds to the constructor True.

3. Is the following function χ-decidable?

   $$\begin{aligned}&\mathit{is\text{-}total}_3 \in \mathit{CExp} \to \mathit{Bool}\\ &\mathit{is\text{-}total}_3\ e =\\ &\quad \textbf{if}\ \forall\, b_1 \in \mathit{Bool}.\ \exists\, b_2 \in \mathit{Bool}.\ [\![e \ulcorner b_1 \urcorner]\!] = \ulcorner b_2 \urcorner\ \textbf{then}\ \mathsf{true}\ \textbf{else}\ \mathsf{false}\end{aligned}$$

   Here *CExp* is a set containing the abstract syntax of every closed χ expression, and $\ulcorner b \urcorner$ is the standard encoding of the boolean b.

   For grade 3: Motivate your answer.

   For grade 4: Provide a proof. You are allowed to make use of Rice's theorem, the fact that the halting problem is undecidable, and the fact that the *terminates-in* function from the lectures (which decides whether an expression terminates in at most a certain number of steps) is decidable, but not other results stating that some function is or is not computable (unless you provide proofs).

   For grade 5: You may not use Rice's theorem (unless you provide a proof).

4. Is the following function χ-decidable?

   $$\begin{aligned}&\mathit{is\text{-}total}_4 \in \mathit{Fun} \to \mathit{Bool}\\ &\mathit{is\text{-}total}_4\ e =\\ &\quad \textbf{if}\ \forall\, b_1 \in \mathit{Bool}.\ \exists\, b_2 \in \mathit{Bool}.\ [\![e \ulcorner b_1 \urcorner]\!] = \ulcorner b_2 \urcorner\ \textbf{then}\ \mathsf{true}\ \textbf{else}\ \mathsf{false}\end{aligned}$$

   Here *Fun* is the following set:

   $$\{ e \in \mathit{CExp} \mid \exists\, f \in \mathit{Bool} \rightharpoonup \mathit{Bool}.\ e \text{ witnesses the } \chi\text{-computability of } f \}$$

   The encoding function for *CExp* is used also for *Fun*.

   The grade criteria of the previous exercise apply to this one as well.

5. (a) *For grade 3:* What is the value of $[\![p]\!]\ (\mathsf{nil}, 2)$, where p is an element of PRF_1 that is defined by

$$p = \mathsf{rec}\ (\mathsf{comp}\ \mathsf{suc}\ (\mathsf{nil}, \mathsf{zero}))\ (\mathsf{comp}\ \mathsf{suc}\ (\mathsf{nil}, \mathsf{proj}\ 1))?$$

(b) *For grade 4:* Give a simple description of the function $f \in \mathbb{N} \to \mathbb{N}$ defined by $f\,n = [\![p]\!]\ (\mathsf{nil}, n)$. The description should not involve any reference to the language PRF or the program p.

(c) *For grade 5:* Prove that your description is correct.

For reference, here is the abstract syntax of PRF:

$$\frac{}{\mathsf{zero} \in PRF_0} \qquad \frac{}{\mathsf{suc} \in PRF_1} \qquad \frac{i, n \in \mathbb{N} \qquad 0 \le i < n}{\mathsf{proj}\ i \in PRF_n}$$

$$\frac{f \in PRF_m \qquad gs \in (PRF_n)^m}{\mathsf{comp}\ f\ gs \in PRF_n} \qquad \frac{f \in PRF_n \qquad g \in PRF_{2+n}}{\mathsf{rec}\ f\ g \in PRF_{1+n}}$$

The denotational semantics is defined in the following way (for any $m, n \in \mathbb{N}$):

$$\begin{aligned}
&[\![_]\!] \in PRF_n \to (\mathbb{N}^n \to \mathbb{N}) \\
&[\![\mathsf{zero}]\!]\ \mathsf{nil} && = 0 \\
&[\![\mathsf{suc}]\!]\ (\mathsf{nil}, n) && = 1 + n \\
&[\![\mathsf{proj}\ i]\!]\ \rho && = \mathit{index}\ \rho\ i \\
&[\![\mathsf{comp}\ f\ gs]\!]\ \rho && = [\![f]\!]\ ([\![gs]\!]\star\ \rho) \\
&[\![\mathsf{rec}\ f\ g]\!]\ (\rho, \mathsf{zero}) && = [\![f]\!]\ \rho \\
&[\![\mathsf{rec}\ f\ g]\!]\ (\rho, \mathsf{suc}\ n) && = [\![g]\!]\ (\rho, [\![\mathsf{rec}\ f\ g]\!]\ (\rho, n), n)
\end{aligned}$$

$$\begin{aligned}
&[\![_]\!]\star \in (PRF_m)^n \to (\mathbb{N}^m \to \mathbb{N}^n) \\
&[\![\mathsf{nil}]\!]\star\ \rho && = \mathsf{nil} \\
&[\![fs, f]\!]\star\ \rho && = [\![fs]\!]\star\ \rho, [\![f]\!]\ \rho
\end{aligned}$$

The *index* function is defined as follows (for any set A and $n \in \mathbb{N}$):

$$\begin{aligned}
&\mathit{index} \in A^n \to \{\, i \in \mathbb{N} \mid 0 \le i < n \,\} \to A \\
&\mathit{index}\ (xs, x)\ \mathsf{zero} && = x \\
&\mathit{index}\ (xs, x)\ (\mathsf{suc}\ i) && = \mathit{index}\ xs\ i
\end{aligned}$$

6. Consider the variant of Turing machines that we get if the semantics of the "move left" and "move right" instructions are changed so that they move two steps at a time. The *move* function from the lectures is replaced by the following one (where *move* is the original function):

$$
\begin{aligned}
&\mathit{move}' \in \{\mathsf{L}, \mathsf{R}\} \to \mathit{Tape} \to \mathit{Tape} \\
&\mathit{move}' \ \mathsf{L} \ t = \mathit{move} \ \mathsf{L} \ (\mathit{move} \ \mathsf{L} \ t) \\
&\mathit{move}' \ \mathsf{R} \ t = \mathit{move} \ \mathsf{R} \ (\mathit{move} \ \mathsf{R} \ t)
\end{aligned}
$$

Let us leave all other definitions, including the rest of the semantics, the definition of computability, and the encoding function for natural numbers, unchanged.

Is every Turing-computable partial function $f \in \mathbb{N} \rightharpoonup \mathbb{N}$ computable using this kind of machine?

For grade 3: Motivate your answer.

For grade 4: Provide a proof.