Computation Theory Example Sheet 2

Lecture 5

Further reading:

- Chapter 4 and 5 in M. Sipser's "Introduction to Computation Theory".
- The handout Computation Theory: Supplementary notes on decidability

Exercise 1 [Halting problem]

(a) Define what it means for an RM to decide the *halting problem.* (See [**2008P5Q10 (a)**] or [**2005P3Q7 (a)**] or [**2000P3Q9 (a)**])

(b) Prove that no such machine can exist. (See [**2017P6Q3**] or [**2005P3Q7 (c)**])

(c) Define what it means for a function to be *uncomputable*.

Exercise 2 [Unary functions] Does it make a significant difference that we are concentrating on *unary functions* (instead of n-ary)?

Exercise 3 [Characteristic function]

(a) Define the *characteristic function* χ_S of a set S. Give an example.

(b) Define what it means for a set to be *decidable/undecidable*? (See [**2014P6Q3 (d)**] or [**2005P4Q9 (a)**] or [**1995P3Q9 (c)**])

Exercise 4 Describe two strategies for proving that a function is uncomputable. Describe a strategy for proving that a set is undecidable.

Exercise 5 [Uncomputable/Undecidable] Do as many of the following until you are confident with proving undecidability/uncomputability:

(a) Show that the set $\{x \mid \phi_x(0)\downarrow\}$ is undecidable.

(b) Show that the function $\{(x, 0) \mid \phi_x(x)\downarrow\}$ is uncomputable.

(c) Show that $\{e \mid \phi_e \text{a total function}\}$ is undecidable.

(d) Attempt [**2015P6Q3 (c)**] (you can take recursive to mean RM computable here).

(e) Attempt [**2014P6Q3 (e)**].

(f) Attempt [**2011P6Q3 (b)**].

(g) Attempt [**2009P6Q3 (b)**].

(h) Attempt [**2005P4Q9 (b)**].

(i) (optional) [**2008P5Q10 (c)**].

Exercise 6 Show that there is a register machine computable partial function $f : \mathbb{N} \to \mathbb{N}$ such that both $\{x \in \mathbb{N} \mid f(x)\downarrow\}$ and $\{y \in \mathbb{N} \mid \exists x \in \mathbb{N}. f(x) = y\}$ are register machine undecidable.

[Exercise 4 in Lecturer's handout]

Exercise 7 Suppose S_1 and S_2 are subsets of $\mathbb{N}$. Suppose $f : \mathbb{N} \to \mathbb{N}$ is register machine computable function satisfying: for all $x \in \mathbb{N}$, x is an element of S_1 if and only if $f(x)$ is an element of S_2. Show that if S_2 is RM decidable, then so is S_1.

[Exercise 5 in Lecturer's handout]

Exercise 8

(a) Show that the set of codes $\langle e, e' \rangle$ of pairs of numbers e and e' satisfying $\phi_e = \phi_{e'}$ is undecidable.

[**Exercise 8 in Lecturer's handout**]

(b) Show that for any fixed e, the set of codes of equivalent programs is undecidable.

(c) Show that checking if a program e changes the contents of register n is undecidable.

(d) Attempt [**1996P3Q9 (c)**].

Exercise 9 [Recursive enumerability] *Recursive enumerability* is a concept that does not appear in the lecture notes, but it appears in several past papers.

(a) Attempt [**2012P6Q4**] (or [**2020P6Q5**]).

(b) Attempt [**2009P6Q4**].

(c) Attempt [**2008P6Q10**].

(d) (optional) Attempt [**1996P4Q8**].

(e) (optional) Attempt [**2000P4Q8**].

Exercise 10 Are there any practical implications of undecidability?

Exercise 11 Define your own undecidable problem.

Lecture 6 / Lecture 7 (first part)

Further reading:

- Chapter 3.1 and 3.3 in M. Sipser's "Introduction to Computation Theory".

Exercise 12

(a) Define *Turing Machines (TMs)*. (See [**2012P6Q3 (a)**] or [**2006P3Q7 (b)(i)**] or [**2004P3Q7 (b),(c)**])

(b) Define a *TM computation*.

(c) Define what it means for a partial function to be *TM computable*. (See [**2012P6Q3 (b)**])

(d) Explain briefly how to enumerate all possible TM computations, so that a given computation can be characterised by a single natural number code c. (See [**2001P3Q9 (c)**])

Exercise 13 For the example Turing machine given on **L6S12**, give the register machine program implementing $(S, T, D) := \delta(S, T)$, as described on **L6S30**. [**Note (by lecturer):** Tedious!—maybe just do a bit.] [**Note (by supervisor):** Implement this in Java if you prefer]

[**Exercise 7 in Lecturer's handout**]

Exercise 14 [RMs can simulate TMs]

(a) Describe the steps in simulating a TM using an RM. (See [**2004P3Q7**] or [**1998P3Q9**])

(b) Attempt [**2012P6Q3 (d)**].

Exercise 15 [TM computable]

(a) Explain how lists of naturals are represented on TM tapes.

(b) Define what it means for a partial function to be *TM computable*.

Exercise 16 [TMs can simulate RMs] Describe the high-level steps for simulating an RM using a TM.

Exercise 17 [TM problems]

(a) Given a Turing machine, is it decidable whether or not for all possible initial configurations the machine will not halt after 100 steps of transition? Justify your answer. **[2006P3Q7 (b)(ii)]**.

(b) Show that it is not possible to compute the maximum distance travelled by the Turing machine head from its initial position during halting computations as a function of the code c. Any results that you use should be stated clearly. **[2001P3Q9 (c)]**

(c) Show that it is not possible to compute a bound on the distance of the head from its starting position during HALTing Turing machine computations. **[1997P3Q9 (c)]**

(d) Show that there is no way of deciding by algorithm whether the blank character will be printed during the course of a general Turing machine computation. **[1993P5Q10 (b)]**

(e) Create your own TM undecidable program.

Exercise 18 Design a TM that checks if the string $s \in \{0, 1\}^*$ on the input is a palindrome.

Exercise 19 [Multi-tape TM] (optional)

(a) Show that a TM with multiple tapes and heads can be simulated by a classical TM.

(b) Does this machine have any advantage over classical TMs?

Exercise 20 [Non-deterministic TM] (optional - connection to Complexity Theory) Show that a *non-deterministic TM*, i.e. one that can perform more than one operations at a single step (similar analogy between DFAs and NFAs) is equivalent to a TM. Does this machine have any advantage over classical TMs?