Informatics 2 – Introduction to Algorithms and Data Structures

Tutorial 10: Register machines and computability

A slightly shorter sheet this week ...

1. We here consider *register machines* and the associated notion of an *RM-computable* partial function $\mathbb{N} \rightharpoonup \mathbb{N}$ or $\mathbb{N} \times \mathbb{N} \rightharpoonup \mathbb{N}$ (see Lecture 29).

   (a) Design a flowchart for a register machine that tests whether 'A < B'. The machine should have two exit points, one for each outcome.

   (b) Design a single machine that computes 'A div B' and 'A mod B' (assuming B is non-zero), storing the results in C and D respectively.

   (c) Show that if both $f : \mathbb{N} \to \mathbb{N}$ and $g : \mathbb{N} \to \mathbb{N}$ are RM-computable, then so is their *composition* h defined by $h(n) = g(f(n))$.

   (d) Show that if $e, f, g : \mathbb{N} \to \mathbb{N}$ are all RM-computable, then so is the function k defined by

   $$k(n) \ = \ \text{if } e(n) = 0 \text{ then } f(n) \text{ else } g(n)$$

   [Hint: recall that any specific register machine must have a fixed, finite number of registers, but this number may be as large as we please.]

2. We may informally define the *number of steps* taken by a register machine computation to be the number of times the flow of control passes through one of the basic components. For instance, running the adder machine from the lecture with initial state A=3, B=5 takes 22 steps.

   (a) One of the main ideas in Lecture 30 is that the predicate

   > the machine coded by m, when applied to the inputs coded by n, eventually halts

   can't itself be decided by a register machine. (The problem of deciding if this predicate is true or false for a given m and n is the so-called *halting problem*). What about the predicate 'the machine coded by m, when applied to the inputs coded by n, halts within k steps'? Would you expect this to be RM-decidable? Informally justify your answer.

   (b) (Harder.) Let T be the set of all codes for register machines that compute some *total* function $\mathbb{N} \to \mathbb{N}$ (that is, machines that halt on all inputs). It would be very useful if there were some register machine that could tell us, given any $m, m' \in T$, whether the machines represented by m and m' gave rise to the *same* total function. Show however that no such machine is possible.

   [Hint: show that if such a machine did exist, we could use it to solve the halting problem. You may draw on your answer to part (a) above.]