Introduction to Theoretical Computer Science

Tutorial Sheet – Week 4 tutorials

Here are a couple of routine exercises.

(1) Write down the code for an RM macro 'if $R_1 > R_2$ then goto I_j'. The macro must leave all registers unchanged after its execution. Assume a predefined GOTO macro.

(2) Give a simple recursive definition of a sequence coding function $\mathbb{N}^* \to \mathbb{N}$, based on the pairing function in the slides.

The following questions/comments are intended as prompts for discussion. Of course, you can ask/discuss about anything. Some of these topics we've touched on in discussion in lectures – this is an opportunity to think about them a bit more.

(3) Our register machines have a finite number of registers, each holding an unbounded number. Turing machines have an unbounded number of cells, each holding one of a finite set of symbols.

Suppose we allow register machine to have an unbounded number of registers, but each register is finite (e.g. 32 bits) – like current computer memory. With no changes to the instruction set, are these machines still Turing powerful? Why not?

Suppose now that we add a form of indirect addressing. For example, we might say that the register operand of an instruction can now be either i, as before, meaning R_i, or (i), meaning R_{R_i}. Does that help?

Why aren't Turing machines bitten by this issue? Can you adapt ideas from TMs to solve it?

Any other ideas?

(4) The proof of the Halting Problem relies on the lethal combination of *self-reference* (when the machine is run on itself) and *negation* (when we flip the result of the halting analyser). Here are some other famous contradictions/paradoxes. Discuss what they show or how they might be resolved.

(a) 'The barber shaves all and only the men who do not shave themselves.'

(b) 'The set of sets that are not members of themselves.'

(c) 'The smallest natural number not definable in under eleven words.'

(5) It's 'usually obvious' that any reasonable domain can be encoded into $\mathbb{N}$. Demonstrate this by giving encodings for: the rationals, lists of numbers, graphs, binary trees.