15-354: Computational Discrete Math — K. Sutner

Solutions Assignment 2

Solution: The Busy Beaver Function

Part A: Domination

Suppose f is a computable, strictly increasing function. Then, for any number n, we can construct a program P_n as follows:

```
// P_n:
       let  m = f(2*n);
       for( i = 0; i < m; i++ )
                print 1;
```

So P_n prints $f(2n)$-many ones. Written out as Turing machine P_n requires at most $\log n + c$ states for some constant c. But then for sufficiently large n (more precisely, we need $\log n + c \leq n$) we have

$$\beta(n) \geq f(2n) > f(n),$$

as required.

Part B: Non-Computability

Follows from part 1: Assume β is computable and consider $\beta'(n) = \max\big(\beta(i) \mid i \leq n\big)$. Then β' is also computable and is strictly increasing. But then by Domination we have $\beta'(n) < \beta(n)$ for sufficiently large n, a contradiction.

Part C: Wurzelbrunft

Given Wurzelbrunft's device we can compute $\beta(n)$ as follows. First generate a list of all programs of size n While the list has wildly exponential size it still finite and easy to generate in principle. Next use Wurzelbrunft's box to remove all those programs from the list that fail to terminate. Now run all the remaining programs to completion: since they all halt we can simply keep computing till the last machine has finished. Lastly, determine which machine has generated the most 1's.

Offering money for Wurzelbrunft's device is a bad idea since it contradicts the unsolvability of the Halting problem. You might as well purchase a perpetual motion machine (actually, that would make slightly more sense: Hilbert's old challenge of axiomatizing physics is still unsolved).

Solution: Write-First Turing Machines

Part A: Computation

As for ordinary Turing machines we define an instantaneous description to the be a word in $\Gamma^\star \times Q \times \Gamma^\star$ (assume Q and Γ to be disjoint). Then $x\,p\,y$ means that the non-blank tape content is xy, that the head is positioned at the first letter in y and that the machine is in state p. To define the one-step relation let $\gamma(p) = (b, d, f)$. Then

$$x\,p\,a\,y \vdash x\,b\,f(y_1)\,y_2 \dots y_m \qquad \text{if } d = +1$$
$$x\,p\,a\,y \vdash x_1 \dots x_{n-1}\,f(x_n)\,b\,y \qquad \text{if } d = -1$$

The multi-step relation is defined as the transitive closure of $\vdash$.

We can safely assume the initial IDs to be of the form $q_0\,0\,x$ (the head is positioned at the last blank before the actual input) and the final IDs to be of the form $q\,y$. The definition of computability is then the same as for read-first machines.

Part B: Simulation by Read-First

Left as an exercise.

Part C: Simulation by Write-First

Consider a read-first TM M with transition function $\delta : Q \times \Gamma \to \Gamma \times \Delta \times Q$. Recall that $\Gamma = \{0, 1\}$ and define a new state set Q' consisting of two disjoint copies Q_0 and Q_1 of Q. Consider an ID $xpay$ of the read-first machine and assume that the write-first TM M' is in ID xp_aay. Consider a transition $\delta(p, a) = (b, d, q)$ in M. We set $\gamma(p_a) = (b, d, f)$ where $f(i) = q_i$.

Note that the initial configuration is not a problem since we assume the head to be positioned at a blank (a 0 in out case). Then M' simulates M' in a direct step-by-step fashion.

Part D: Size

The given simulation requires a state set twice the size of the original in either direction. We do not know if there are improvements.