Exercises for Computability and Complexity, Spring 2017, Sheet 1 – Solutions

Please return your solutions in class, in the Thursday lecture on Feb 9.

Note: You may work in teams up to size 2.

Exercise 1 Show that a TM whose read/write head are restricted to the left and right moves $\{\leftarrow, \rightarrow\}$, can compute the same functions as the TMs from the definition in the lecture notes whose heads can pick motions from the set $\{\leftarrow, \rightarrow, -\}$.

Solution. Let *M* be a TM according to the lecture notes definition, and let

$p \in K$	$\sigma \in \Sigma$	$\delta(p, \sigma)$
q	a	$(q', b, -)$

be a transition that uses the standstill motion "-". Then you get the same ultimate effect of using this rule if you replace it in the TM's transition table by the following rules:

$p \in K$	$\sigma \in \Sigma$	$\delta(p, \sigma)$
q	a	$(q'_{return}, b, \rightarrow)$
q'_{return}	#	$(q', \#, \leftarrow)$
(one such rule for every tape symbol #)		

where q'_{return} is a new state which tells the TM to return to the left in the next step and switch to state q'.

Exercise 2 Give a formal definition of a version of TMs that use a 2-dimensional grid of memory cells instead of a 1-dimensional tape. Start with a plain English description of your basic ideas and intuitions of how to make a 2-dim grid useful for computations in the TM spirit, that is, what special grid cell symbols you want to use, how to initialize the grid, how to administer input. Then repeat-adapt definition 3.1 in formal rigor, and also give a formal definition of a configuration. Note: there are many ways how a 2-dim TM can be set up in a reasonable way, so there is not a singular "correct" definition.

Solution (one possibility)

Definition. A 2D-Turing machine is a structure $M = (K, \Sigma, \delta, s)$, where K is a finite set of *states*, $s \in K$ is the *initial state*, the *alphabet* Σ is a set of (tape) *symbols*, and where K and Σ are disjoint. We assume that Σ always contains the special symbols $\sqcup, \rhd, \triangledown, \diamond$,called the *blank, leftborder, upperborder and corner* symbol. Finally, δ is a *transition function*, where

$$\delta: K \times \Sigma \rightarrow (K \cup \{h, \text{"yes"}, \text{"no"}\}) \times \Sigma \times \{\leftarrow, \rightarrow, \uparrow, \downarrow, -\}.$$

We assume that h (the *halting state*), "yes" (the *accepting state*), "no" (the *rejecting state*), and the *cursor directions* $\leftarrow$ for "left", $\rightarrow$for "right", $\uparrow$ for "up", $\downarrow$ for "down" and $-$ for "stay", are extra symbols not in $K \cup \Sigma$. Furthermore, we require that for any $q \in K$,

$$\delta(q, \rhd) \in (K \cup \{h, \text{"yes"}, \text{"no"}\}) \times \{\rhd\} \times \{\rightarrow, \uparrow, \downarrow, -\},$$
$$\delta(q, \triangledown) \in (K \cup \{h, \text{"yes"}, \text{"no"}\}) \times \{\triangledown\} \times \{\leftarrow, \rightarrow, \downarrow, -\},$$
$$\delta(q, \diamond) \in (K \cup \{h, \text{"yes"}, \text{"no"}\}) \times \{\diamond\} \times \{\rightarrow, \downarrow, -\}.$$

Conventions. The memory of a 2DTM is a right- and down-infinite grid of square cells c_{ij}, where $i, j > 0$. At startup, the 2DTM is in state s and the cursor is on cell c_{11}. The initial inscription of the memory grid is

- first row: a leftmost ◇ followed by all ▿ 's
- first column: a top ◇ followed by all ▹'s
- all other cells carry the symbol ⊔ exept finitely many which might carry other symbols from Σ. If some cell has a non-⊔ symbol, all cells above and to the left also have non-⊔ symbols. The 2-dim pattern of all these non-⊔ symbols is the *input pattern*.

Definition. A *configuration* of a 2D-TM is a quadruple (q, P, i, j) where $q \in (K \cup \{h, \text{"yes"}, \text{"no"}\})$, P is a *pattern*, that is a map from $\mathbb{Z}^2 \to \Sigma \backslash \{\triangleright, \triangledown, \diamond\}$, which has only finitely many non-⊔ values, and where i, j are integers > 0.

Exercise 3 Give a transition table for a TM that computes the function $f(n) = 2n$. The TM should have the tape alphabet $\{0, 1, \triangleright, \sqcup\}$ and numbers are coded as binary strings by writing them to base 2.

Solution. That's an easy one. Multiplying n by 2 means to append a 0 at the binary representation of n. A table for such a TM:

$p \in K$	$\sigma \in \Sigma$	$\delta(q, \sigma)$	comment
s	▷	$(s, \triangleright, \rightarrow)$	get started
s	0	$(s, 0, \rightarrow)$	reading a 0, just move on to the right
s	1	$(s, 1, \rightarrow)$	reading a 1, just move on to the right
s	⊔	$(h, 1, -)$	hitting the first blank, replace it by 1, halt