Exercises for Computability and Complexity, Spring 2019, Sheet 3

Please return your solutions in class, in the Tuesday lecture on February 26. You may (like always in this course) work in teams of 2.

Exercise 1. Consider the ultra-simple TM M with tape alphabet $\Sigma = \{0, 1, \sqcup, \rhd\}$ and states $\{s, yes, no\}$ that has the following transition table:

$p \in K$	$\sigma \in \Sigma$	$\delta(q, \sigma)$
s	0	$(yes, 0, -)$
s	1	$(s, 1, \rightarrow)$
s	$\sqcup$	$(no, \sqcup, -)$
s	$\rhd$	$(s, \rhd, \rightarrow)$

What is the language $L(M)$ decided by M? Describe that language in plain English. Write a RAM program that decides the same language, in the following sense. Your RAM should compute a string function $f: \Sigma^* \rightarrow \{0,1\}$, such that $f(w) = 1$ iff w is in $L(M)$.

Problem 2 (easy) Show that the function $plus2: \mathbb{N} \rightarrow \mathbb{N}$, $plus2(n) = n + 2$, is primitive recursive.

Problem 3 (a little easier than medium) Show that the function $minus1: \mathbb{N} \rightarrow \mathbb{N}$, $minus1(n) = \max(0, n - 1)$ is primitive recursive. Hint: there is a very compact way of doing this, exploiting the fact that the primitive recursion scheme condition $h(n + 1, x) = g(n, h(n, x), x)$ has the "minus 1" operation already built in in the first arguments $n + 1$ and n passed to h and g.

Challenge problem (a bit demanding; optional) Show that the function $evensquare: \mathbb{N} \rightarrow \mathbb{N}$, defined by $evensquare(n) = n$ if n is uneven, else $evensquare(n) = n^2$, is primitive recursive. You may assume that $square: \mathbb{N} \rightarrow \mathbb{N}$, $square(n) = n^2$, is primitive recursive. Hint: you may find it helpful to construct *evensquare* from a number of helper functions which you construct before assembling *evensquare*.