Fundamentals of Computer Science, Spring 2014

Assignment 4

Due date: February 27, 2014

Mandatory exercises

1) Construct a context-free grammar for the language

$$L = \{(ab)^n c^m \mid n, m \in \mathbb{N} \land n \neq m\}.$$

2) Construct a Turing machine $M = (Q, \Sigma, \Gamma, \delta, \Box, q_0, F)$ with input alphabet $\Sigma = \{0, 1\}$ that computes the reversal function. In other words, for every binary word w we have,

$$q_0 w \Rightarrow^* q_f w^R,$$

where q_f is some accepting state and w^R is the reversal of w. For example, the reversal of 00101 is 10100.

Define all transitions of M and briefly explain how M works.

Voluntary exercises (for higher grades than 3)

3) A *2-stack automaton* is a pushdown automaton with 2 stacks, i.e., an automaton $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$ where

$$\delta : Q \times \Sigma \times \Gamma \times \Gamma \to \mathit{Fin}(\mathcal{P}(Q \times \Gamma^* \times \Gamma^*)).$$

The transition function should be interpreted as follows. If

$$(p, w_1, w_2) \in \delta(p, a, s_1, s_2),$$

then if the automaton is in state p, reads an a in the input word, the symbol s_1 is at the top of stack 1, and the symbol s_2 is at the top of stack 2, then it can go to state q while replacing s_1 with the word w_1 on the top of stack 1 and replacing s_2 with the word w_2 on the top of the second stack.

Argue that 2-stack automata are equivalent to Turing machines, i.e., that for every Turing machine M that accepts a language $L(M)$, there is a 2-stack automaton that also accepts $L(M)$.