Homework 3

Finite Automata on Infinite Words and Trees  Winter semester, 2009-2010

Note: Graphical representations of automata are accepted.

Exercise 1 (30 points). Consider the alphabet $\Sigma = \{0, 1\}$. Construct Müller, Rabin, and Streett automata which accept each of the following languages:

1. $\{w \mid$ the symbol 0 appears in $w$ exactly twice $\}$
2. $\{w \mid$ every 0 which appears in $w$ is followed immediately by 11 $\}$
3. $\{w \mid w$ contains finitely many 1’s $\}$
4. $\{w \mid w$ contains finitely many substrings 11 $\}$
5. $\{w \mid$ either $w$ contains only 1’s, or $w$ contains infinitely many 0’s and infinitely many 1’s$\}$

Exercise 2 (20 points). Let $\Sigma = \{a, b, c\}$. Let $A_1, A_2, A_3, A_4$ be the Müller automata whose states and transitions are as pictured in Figure 1, and whose acceptance conditions are defined as follows: $F_1 = \{\{2, 3\}\}, F_2 = \{\{1\}, \{1, 4\}\}, F_3 = \{\{3\}, \{1, 2\}\}$, and $F_4 = \{\{1, 2, 3\}\}$. Determine

![Figure 1: Graph representation of the automata in Exercise 2.](image-url)
the languages $L(A_1)$, $L(A_2)$, $L(A_3)$, and $L(A_4)$. You may represent the languages by $\omega$-regular expressions (e.g. $(aa + bb)^*(ab)^\omega$), but try to keep the expressions as simple as possible, and explain your reasoning.

**Exercise 3** (15 points). Prove that every Müller-recognizable language is a Boolean combination (union, intersection, complementation) of languages $\overline{W}_i$, where each $W_i$ is a FA-recognizable language (of finite words).

**Exercise 4** (25 points). A parity automaton is a tuple $(Q, \Sigma, T, \{q_f\}, c : Q \rightarrow \mathbb{N})$, where $T$ is a transition function. A run $r$ of a parity automaton is successful if and only if $\max\{c(q) | q \in \text{Inf}(r)\}$ is even.

1. Show how to construct Müller, Rabin, and Streett automata which recognize the same language as a given parity automaton. Explain briefly how your constructions work.

2. Prove that parity automata are closed under negation.

**Exercise 5** (10 points). Review of complementation of finite word automata. Use the power set construction to convert the following NFA into an equivalent DFA.