Technische Universität München
Fakultät für Informatik
Lehrstuhl für Effiziente Algorithmen (LEA)
Prof. Dr. Ernst W. Mayr
Moritz Fuchs

Winter Semester 2014/15
Problem set 10
December 16, 2014

Automata and Formal Languages

Due January 13, 2015 before class!

Exercise 1 (Presburger Arithmetic - 10 points)

(a) Using the algorithm from the lecture, construct an automaton recognizing the formula $x + y \leq 4$.

(b) Construct an automaton recognizing the formula $\exists y : x + y \leq 4$.

(c) Construct an automaton recognizing the formula $x \geq 2 \wedge \exists y : x + y \leq 4$.

Exercise 2 (MSO IV - 10 points)

We interpret the monadic second order logic over finite words with the standard interpretation of $<$ as 'less than' relation.
Let MSO' be a modification of the standard monadic second-order logic given by the following syntax. Let X, Y, Z be second-order logical variables and Σ an alphabet. An MSO' formula over Σ is defined by the following grammar, where $a \in \Sigma$:

$$\varphi ::= X \subseteq Q_a \mid X < Y \mid Sing(X) \mid X \subseteq Y \mid \neg\varphi \mid (\varphi \vee \varphi) \mid \exists X \varphi$$

Although we quantify over set variables only, we want this logic to be equally "powerful" as the original MSO. As there are no first-order variables, the first-order predicates $<$ will be replaced by the second-order predicates, so new atomic formulas are introduced: $Sing(X)$ (meaning singleton), $X \subseteq Y$ (meaning subset inclusion), $X \subseteq Q_a$ for every $a \in \Sigma$ (meaning all elements of X are labelled by a), and $X < Y$ (true for singletons $X = \{x\}, Y = \{y\}$ satisfying $x < y$).

Show that MSO and MSO' are equally expressive, i.e., a language is definable in MSO iff it is definable in MSO'.

Hint: Express the newly defined predicates in the original MSO and vice versa.

Exercise 3 (Plus - 10 points)

Express the addition using MSO. More precisely, find a formula $plus(X, Y, Z) \in MSO$ that is true iff $x + y = z$, where x, y, z are numbers encoded by the sets X, Y, Z, respectively, in the binary lsbf. You are allowed to use the successor macro $Succ$.

Exercise 4 (Infinite words - 10 points)

You are given finite words $u, v, x, y \in \Sigma^*$ which represent the ω-words $w = uv^\omega$ and $z = xy^\omega$. Give an algorithm for deciding $w \stackrel{?}{=} z$.