LOGIC & AUTOMATA — ASSIGNMENT 1

Due: 29 February, 5pm

Note: marks do **not** reflect difficulty.

1. (*7 marks*) Consider MSO formulae over a vocabulary that has only one binary relation symbol $<$. We are interested in structures where $<$ is interpreted as a linear order, i.e. we are looking at structures $\langle A, < \rangle$, where $A$ is a set, and $<$ is a linear order on it.

   Prove that there is no MSO sentence that checks whether the cardinality of $A$ is a perfect square, i.e. of the form $n^2$ for some $n \in \mathbb{N}$.

   *Hint*: Use Büchi's theorem for strings.

2. (*8 marks*) Consider the following nondeterministic automaton model on strings: $(Q, q_0, F, \delta_\rightarrow, \delta_\leftarrow)$, where $Q$ is a set of states, $q_0$ is a single initial state, states in $F \subseteq Q$ are final, and $\delta_\rightarrow, \delta_\leftarrow : Q \times \Sigma \rightarrow 2^Q$ are two transition functions.

   The automaton starts in state $q_0$, reading the first symbol of a string. Every time the automaton is in a state $q$, reading symbol $a_i$ of a string $a_0 \dots a_{n-1}$, it nondeterministically chooses one of the two transition functions and does the following:

   - For $\delta_\rightarrow$: it selects a state $q' \in \delta_\rightarrow(q, a)$, and moves one position to the right, i.e. it now reads $a_{i+1}$ in state $q'$. In the special case when $i = n - 1$ (i.e. there are no more symbols on the right), the automaton accepts if $q' \in F$ and rejects otherwise.
   - For $\delta_\leftarrow$: it selects a state $q' \in \delta_\leftarrow(q, a)$, and moves one position to the left, i.e. it now reads $a_{i-1}$ in state $q'$. In the special case when $i = 0$ (i.e. there are no more symbols on the left), the automaton rejects.

   Thus, these automata can go back-and-forth, like Turing machines. But we further impose a condition that such an automaton can visit a position *at most twice* (if it visits any position for a third time, it rejects).

   These automata clearly generalise the usual NFAs (when one transition function is empty), so they accept all regular languages. Your goal is to prove the converse *using Büchi's theorem*: every language they accept is regular.

3. (*15 marks*) The goal of this problem is to prove that MSO captures regular languages of unranked trees, using a translation of unranked trees into binary trees.

   - (*4 points*) Show that every regular language of unranked trees is definable in MSO, by coding unranked tree automata. It suffices to give a high-level description of the coding (i.e., "a sentence $\exists X \exists Y \ \varphi$ where $\varphi$ says that $X$ and $Y$ are sets such that ...").

     Next we define a translation from $\Sigma$-labeled unranked trees into $\Sigma \cup \{\perp\}$-labeled binary trees, where $\perp$ is a new alphabet symbol. We first define a mapping $r : \mathbb{N}^* \rightarrow \{0, 1\}^*$ as follows:

     (a) $r(\varepsilon) = \varepsilon$;

     (b) if $r(s) = w$, then $r(s \cdot 0) = w \cdot 0$ (first child is mapped to the left successor), and if $s = s' \cdot i$, then $r(s' \cdot (i + 1)) = w \cdot 1$ (next sibling is mapped to the right successor).

     If $T = (D, \lambda)$, let $D'$ be the completion of $r(D)$: that is, if we have a string in $r(D)$ that has only left or only right successor, we add the missing successor to it. We define $\lambda' : D' \rightarrow \Sigma \cup \{\perp\}$ as follows: if $w = r(s)$, then $\lambda'(w) = \lambda(s)$; otherwise $\lambda'(w) = \perp$. Finally, $r(T)$ is $(D', \lambda')$.
   - (*4 marks*) Prove that every sentence over unranked trees can be expressed over translations: that is, for each sentence $\psi$ over $\Sigma$-labeled unranked trees, there is a sentence $\psi^r$ over binary $\Sigma \cup \{\perp\}$-labeled trees such that $T \models \psi$ iff $r(T) \models \psi^r$.
   - (*3 marks*) Show that there is a tree automaton that recognizes trees of the form $r(T)$.

- (*2 marks*) What remains to be proved to conclude that MSO over unranked trees can be translated into unranked tree automata? How does one conclude the proof from that statement?
- (*2 marks*) Prove that statement.