Advanced Topics in Automata – Final Exam

Submission: August 8th, 2003

1. Give the best algorithm you can find for the membership problem for NBWs. That is, given an NBW A and a word $w = \alpha\beta^\omega$ (where $\alpha, \beta \in \Sigma^*$), devise an algorithm that decides whether $w \in L(A)$.

   Analyze the time and space complexity of your algorithm.

2. A deterministic automaton has exactly one run over each word. We determine whether the word is accepted or not by checking whether this run is fair or not. A *full* automaton has at least one fair run over each word. We determine acceptance by checking whether these runs start in an initial state or not.

   Consider, a (presumably) NBW $N = \langle \Sigma, Q, Q_0, \delta, F \rangle$. We say that N is *full* if it satisfies two conditions.

   (a) For every word $w = w_0, w_1, \ldots \in \Sigma^\omega$, there exists at least one sequence $r = t_0, t_1, \ldots \in Q^\omega$, such that:

   - For all i we have $t_{i+1} \in \delta(t_i, w_i)$.
   - $inf(r) \cap F \neq \emptyset$.

   We call sequences that satisfy these two conditions *uninitialized runs* over w.

   (b) There exists a partition $Q = Q_0 \cup Q_1$ such that for every word w, all the uninitialized runs of N over w start either in states from Q_0 or in states from Q_1 and not both.

   We say that a word w is accepted by N if the uninitialized runs over w start in Q_0.

   Full automata are interesting because when we follow a run of such an automaton at every stage we can determine whether the suffix is in the language of the automaton or not. Furthermore, when we consider the product of a full automaton and another automaton, the result has the language of the second. The full automaton is an *observer* that gives information on the rest of the run of the second automaton.

   Use the following facts to show that for every NBW we can construct a full automaton accepting the same language.

   - For every NBW we can construct a weak alternating automaton accepting the same language and another that accepts the complementary language.

     In Exercise 7 we established that the two weak alternating automata are the duals of each another.

- For an ABW A, let $\overline{A}$ denote ACW complement of A. Let Q denote the set of states of A and $\overline{Q} = \{\overline{q} \mid q \in Q\}$ denote the set of states of $\overline{A}$. Every word is accepted either by A_q or by $\overline{A}_{\overline{q}}$, where A_q is the automaton A with q as initial state. It is never the case that some word is accepted by both A_q and $\overline{A}_{\overline{q}}$.
- We can represent a subset of the states of A as $\{0,1\}^Q$ (namely, a function $f : Q \to \{0,1\}$). A 0 stands for 'the state is not in the subset' and a 1 stands for 'the state is in the subset'. Having a nondeterministic automaton in state $Q' \subseteq Q$ means that the suffix should be accepted from all the states in Q'. Equivalently, consider the set $\{-1,1\}^Q$, where -1 stands for '$\overline{q}$ should accept the suffix' and 1 stands for 'q should accept the suffix'.

3. We are familiar with the Büchi, co-Büchi, and Müller acceptance conditions. There are also 'strong' acceptance conditions that restrict both the set of states occurring finitely often and the set of states occurring infinitely often, while avoiding the exponential blow up incurred by using the Müller condition.

   A Streett condition consists of a set of pairs of sets of states. We require that for every pair, if the first set is visited infinitely often then so is the second set. Formally, we have the following.

   A nondeterministic Streett automaton is $A = \langle \Sigma, Q, Q_0, \delta, \alpha \rangle$, where Σ is the alphabet, Q is the set of states, $Q_0 \subseteq Q$ is the set of initial states, $\rho : Q \times \Sigma \to 2^Q$ is the transition relation and $\alpha = \{\langle L_1, U_1 \rangle, \ldots, \langle L_k, U_k \rangle\}$ is a Streett acceptance condition, where $L_i \subseteq Q$ and $U_i \subseteq Q$ are sets of states.

   A run of A on a word $w = w_0 w_1 \cdots$ is a sequence of states $q_0 q_1 \cdots$, such that $q_0 \in Q_0$ and for all i we have $q_{i+1} \in \delta(q_i, w_i)$. A run is accepting if for every $1 \leq j \leq k$ we have $inf(r) \cap L_j \neq \emptyset$ implies $inf(r) \cap U_j \neq \emptyset$ (or, in LTL format, $\bigwedge_{i=1}^{k} (\Box \Diamond L_i \to \Box \Diamond U_i)$).

   Use the following family of languages to show that Streett automata are exponentially more succinct than Büchi automata. For every language L_n describe a (small) Streett automaton that accepts the language. Show that every Büchi automaton that accepts L_n must have at least 2^n states.

   Let $\Sigma = \{0,1,2\}$. Every word in Σ^ω can be viewed as a word in $(\Sigma^n)^\omega$, that is, an infinite sequence of n-vectors over Σ. Let $u = a_0, \ldots, a_{n-1} \in \Sigma^n$. We say that i is *0-active* (resp. *1-active*) in u, for $0 \leq i < n$ if $a_i = 0$ (resp. $a_i = 1$). Let $w \in \Sigma^\omega$, then $w = u_0 u_1 \cdots$, where $u_j \in \Sigma^n$ forall $j \geq 0$. We say that i is *0-active* (resp. *1-active*) in u_j for $0 \leq i < n$, if i is 0-active (resp. 1-active) in u_j for infinitely many j's. Let L_n be the set of words in Σ^ω with a symmetric activity record. Formally,

   $$L_n = \{w \in \Sigma^\omega \mid i \text{ is 0-active in } w \text{ iff } i \text{ is 1 active in } w\}$$

   You can try thinking first about the problem where the alphabet is $\{0,1\} \times 2^{\{1,\ldots,n\}}$, and for which a letter $\langle 0, N \rangle$ with $N \subseteq \{1, \ldots, n\}$ represents the word $a_0, \ldots, a_n$ such that $a_i = 0$ if $i \in N$ and $a_i = 2$ otherwise. Similarly for $\langle 1, N \rangle$. Now show that over this large alphabet a Büchi automaton cannot have less than 2^n states. Extend your result to the constant alphabet above.

Remember that in every run over an infinite word, from some point on all the states that appear in the run are states that appear infinitely often.

4. A Dyck set over the alphabet $\Sigma = \{(_i\ ,\ )_i \mid i \in I\}$, for some set I, is the set of words where the parentheses are balanced in the usual sense. That is, the opening parenthesis $(_i$ is an obligation that $)_i$ will be found later in the word in a legal position with respect to balancing. (In particular, a word in a Dyck set cannot start with a closing parenthesis.) We can define a sym-Dyck set over the same alphabet by allowing balancing in a symmetric fashion: not only does a closing parenthesis balance an opening one, but also an opening parenthesis balances a closing one.

   - Define formally sym-Dyck sets.
   - Show that sym-Dyck sets are context free.
   - Prove the parentheses theorem we proved in class using sym-Dyck sets instead of Dyck sets. Give the full proof of the theorem, and mark accurately (in some typographical way: e.g., shading, underline, boldface, coloring, etc.) all the locations where your proof differs from the proof given in class.

5. Good Luck.