Automata and Logic (CSC2428)
Lecture 5

Lecturer: Prof. Leonid Libkin
Scribe: Jorge Baier

February 11th, 2005

1 MSO-definability of Infinite Strings: Part one

Definition 1 (Infinite string) *An infinite string (ω-string) s over alphabet Σ is a mapping $\mathbb{N} \to \Sigma$.*

Furthermore, the set of all ω-strings over Σ is denoted by Σ^ω.

Definition 2 (Büchi automaton) *A Büchi automaton over ω-strings is a tuple $(\Sigma, Q, Q_0, F, \delta)$, where Q is a finite set of states, $Q_0 \subseteq Q$ is a set of initial states, and $\delta : Q \times \Sigma \to 2^Q$ is a transition function, and $F \subseteq Q$ is the set of final states.*

As before, we need to define what is a run of a Büchi automaton.

Definition 3 *A run of a Büchi automaton $A = (\Sigma, Q, Q_0, F, \delta)$ over the ω-string $s = a_0a_1a_2\cdots$ is a function $\rho : \mathbb{N} \to Q$ such that*

1. $\rho(0) \in Q_0$.
2. $\rho(i+1) \in \delta(\rho(i), a_i)$.

In order to define the language accepted by the Büchi automaton A of the above definition, it is useful to define the following set.

$$Inf(\rho) = \{q \in Q \mid |\{u \mid \rho(u) = q\}| = \omega\},$$

for any run ρ of A. $Inf(\rho)$ corresponds to the set of states seen infinitely often by A on string s.

An ω-string s is accepted by a Büchi automaton A iff there exists a run ρ of A on s such that $Inf(\rho) \cap F \neq \varnothing$.

The set of all ω-strings accepted by A is denoted by $L_\omega(A)$. The language $L_\omega(A)$ is an ω-regular language.

As with finite strings, an ω-string s over Σ can be viewed as a structure $\mathcal{M}_s = (\mathbb{N}, <, (P_a)_{a\in\Sigma})$, where, as before, P_a is the set of positions where symbol a occurs in s.

We say that a set X of ω-strings is MSO-definable iff there is an MSO sentence φ such that:

$$s \in X \Leftrightarrow \mathcal{M}_s \models \varphi.$$

The following theorem is the analog of the theorem we had before but for ω-strings.

Theorem 1 (Büchi 1960±1) *A set of ω-strings is ω-regular iff is MSO-definable.*

Proof: It turns out that one of the $\Rightarrow$ is very easy since it is almost the same proof as that of the finite case.

We need to prove that if $X \subseteq \Sigma^\omega$ is ω-regular then X is MSO definable. Given a Büchi automaton $A = (\Sigma, Q, Q_0, F, \delta)$ we will write a formula φ_A that intuitively guesses a run of A and that is such that $\mathcal{M}_s \models \varphi_A$ iff $s \in L_\omega(A)$.

Assume $Q = \{q_0, q_1, \ldots, q_{n-1}\}$. The run is represented by the predicates X_i ($i \in \{0, \ldots, n-1\}$) that contain the set of positions of string s such that A, while processing s, is in state q_i.

The formula φ_A is very similar to the one presented in the first lecture, therefore we only define in logic the part that is different. The different part corresponds to the acceptance condition.

$$\varphi_A = \exists X_0 X_1 \cdots X_{n-1} \begin{pmatrix} \text{“All } X_i\text{'s are a partition of the universe”} \wedge \\ \text{“}X_i\text{'s respect the definition of } \delta\text{”} \wedge \\ \bigvee_{q_i\in F} \forall x \exists y\, x < y \wedge X_i(y) \end{pmatrix}$$

Intuitively $\bigwedge_{q_i\in F} \forall x \exists y\, x < y \wedge X_i(y)$ says that there is at least one final state such that whichever position of s where it occurs it not the last. This proves the $\Rightarrow$ direction.

Unfortunately, to prove the $\Leftarrow$ direction, we cannot use the same proof of the finite case. The reason is that complementation and intersection of Büchi automata do not work as in the finite case.

Before completing the proof we examine some closure properties ω-regular languages.

2 Closure properties of ω-regular languages

Given two NFA's A and A' such that $L(A), L(A') \subseteq \Sigma^*$ it is easy to construct NFA's for $L(A) \cup L(B)$, $L(A) \cap L(B)$ and $\Sigma^* - L(A)$. To do this we use, respectively, the union construction, the product construction, and flipping accepting for nonaccepting states and vice versa on a deterministic version of A.

We now review whether or not those constructions work with Büchi automata. In the following, let $A_1 = (\Sigma, Q_1, Q_1^0, F_1, \delta_1)$ and $A_2 = (\Sigma, Q_1, Q_2^0, F_2, \delta_2)$ be Büchi automata such that $Q_1 \cap Q_2 \neq \varnothing$.

2.1 ω-regular languages are closed under union

The union construction in NFA's is defined as

$$A_1 \cup A_2 = (\Sigma, Q_1 \cup Q_2, Q_1^0 \cup Q_2^0, F_1 \cup F_2, \delta_1 \cup \delta_2).$$

It is easy to see that $L_\omega(A_1 \cup A_2) = L_\omega(A_1) \cup L_\omega(A_2)$ since if $s \in L_\omega(A_1)$ or $s \in L_\omega(A_2)$ then clearly $s \in L_\omega(A_1 \cup A_2)$ ($A_1 \cup A_2$ just has to guess the correct initial state). On the other hand if $s \in L_\omega(A_1) \cup L_\omega(A_2)$ then there exists an accepting run on s. Since the states of A_1 and A_2 are disjoint, it is easy to see that this run is also to a run of A_1 or A_2, and therefore $s \in L_\omega(A_1)$ or $s \in L_\omega(A_2)$.

2.2 ω-regular languages are closed under intersection

The product construction in NFA's is defined as follows.

$$A_1 \times A_2 = (\Sigma, Q_1 \times Q_2, Q_1^0 \times Q_2^0, F_1^0 \times F_2^0, \delta),$$

where $\delta((q_1, q_2), a) = (\delta(q_1, a), \delta(q_2, a))$, for all $(q_1, q_2) \in Q_1 \times Q_2, a \in \Sigma$. Over regular languages, $L(A_1 \times A_2) = L(A_1) \cap L(A_2)$. Nevertheless, that is not the case over ω-regular languages.

To see that, consider the following example. Let $A_1 = (\{a\}, \{q_1, q_2\}, \{q_1\}, \{q_1\}, \delta)$ and $A_2 = (\{a\}, \{q_1', q_2'\}, \{q_1'\}, \{q_2'\}, \delta)$. And let δ be such that $\delta(q_1, a) = \{q_2\}$, $\delta(q_2, a) = \{q_1\}$, $\delta(q_1', a) = \{q_2'\}$, and $\delta(q_2', a) = \{q_1'\}$. Notice that A_1 and A_2 only differ on their accepting state.

Observe that the (unique) run of A_1 and A_2 on string $s = aaa\cdots$ are respectively $q_1q_2q_1q_2\cdots$ and $q_1'q_2'q_1'q_2'\cdots$, therefore $s \in L_\omega(A_1) \cap L_\omega(A_2)$. On the other hand, the run of $A_1 \times A_2$ on s is $(q_1, q_1')(q_2, q_2')(q_1, q_1')\cdots$. However, $A_1 \times A_2$ does not accept s since its set of final states is $\{(q_1, q_2)\}$.

Nevertheless, it is possible to define another version of the product, $A_1 \otimes A_2$, to overcome this problem. Intuitively, $A_1 \otimes A_2$ runs in two *tracks*. Initially, $A_1 \otimes A_2$ is in track 1, where it simulates a run of A_1 until it sees a final state of A_1. Then, it "jumps" to track 2 and simulates now A_2 until it sees a final state of A_2, jumping back to track 1, where the cycle repeats. The accepting states of $A_1 \otimes A_2$ are the accepting states of A_1 on track 1. Thus, $A_1 \otimes A_2$ accepts an ω-string s iff there exist runs in A_1 and A_2 that accept s. Therefore, $L_\omega(A_1 \otimes A_2) = L_\omega(A_1) \cap L_\omega(A_2)$.

Formally,

$$A_1 \otimes A_2 = (\Sigma, (Q_1 \times \{1\}) \times (Q_2 \times \{2\}), (Q_1^0 \times \{1\}) \times (Q_2^0 \times \{2\}), (F_1 \times \{1\}) \times (Q_2 \times \{2\}), \delta).$$

The definition of δ is left as an excercise.

2.3 Non-empty ω-regular languages contain periodic strings

An ω-string $a_0a_1a_2\cdots$ is periodic if there exists naturals k and n such that

$$\forall m \geq k \; a_m = a_{m+n}.$$

Let A be a Büchi automaton and $s \in L_\omega(A)$. There is an accepting run ρ of A on s such that $q \in \mathit{Inf}(\rho)$, for some q. Then, there exist finite strings s_0 and s_1 such that $s_0 s_1$ is a prefix of s, and such that A, after processing s_0 is in q, and after processing $s_0 s_1$ is also in q. Then, the periodic string $s' = s_0 s_1 s_1 s_1 \cdots$ is accepted by A since there is a run on s' that contains q infinitely many times.

2.4 Does the complement construction for NFA's work with Büchi automata?

In regular languages, we take an NFA, do the powerset construction and finally flip accepting for nonaccepting states and vice versa.

Definition 4 (Deterministic Büchi automaton) *A deterministic Büchi automaton* $A = (\Sigma, Q, Q_0, F, \delta)$ *is a Büchi automaton such that* $|Q_0| = 1$ *and* $|\delta(q, a)| \leq 1$, *for all* $q \in Q$, $a \in \Sigma$.

Proposition 1 *There are ω-regular languages that are not accepted by deterministic Büchi automata.*

Proof: Consider the language generated by $(a+b)^* b^\omega$, i.e. the ω-strings that have finitely many a's. Suppose $A = (\{a, b\}, Q, Q_0, F, \delta)$ is a deterministic Büchi automaton that accepts $L((a+b)^* b^\omega)$. Since $ab^\omega \in L_\omega(A)$, then there exists a finite number n_1 such that $ab^{n_1} \in L(A)$ (i.e. $\delta(q_0, ab^{n_1}) = q_1$, for some $q_1 \in F$). Furthermore since $ab^{n_1}ab^\omega \in L_\omega(A)$, then there must exist an n_2 such that $\delta(q_2, ab^{n_1}ab^{n_2})$, for some $q_2 \in F$. Following the same reasoning $\delta(q_i, ab^{n_1}ab^{n_2} \cdots ab^{n_i})$, for some $q_i \in F$ and $\delta(q_j, ab^{n_1}ab^{n_2} \cdots ab^{n_i} \cdots ab^{n_j})$ such that $q_j = q_i$ and $j \geq i$, because there are a finite number of final states. We conclude that, since A is deterministic

$$s = ab^{n_1}ab^{n_2} \cdots ab^{n_{i-1}}(ab^{n_i} \cdots ab^{n_j})^\omega \in L_\omega(A),$$

but that is a contradiction because s has an infinite number of a's. $\square$

We conclude that the powerset construction does not work for Büchi automata since we cannot always determinize a Büchi automaton. However, we will see later in this lecture that it is possible to construct a Büchi automaton for the complement of any ω-regular language.

3 ω-regular expressions

Definition 5 (ω-regular expression) *ω-regular expression is inductively defined by the following.*

- *$\varnothing$ is a regular expression.*
- *If e_r is a regular expression and e_1 and e_2 are ω-regular expressions, then the following are ω-regular expressions:*
  - $e_1 \cup e_2$, $(L(e_1 \cup e_2) = L(e_1) \cup L(e_2))$.
  - $e_r \cdot e_2$, $(L(e_r \cdot e_2) = L(e_r) \cdot L(e_2))$.
  - e_r^ω, $(L(e_r^\omega) = \{w_0 w_1 \cdots \mid \mathit{for}\ i \geq 0\ w_i \in L(e_r)\})$.

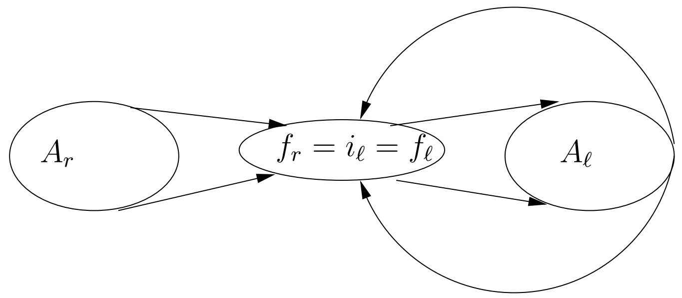

Figure 1: Büchi automaton for $r\ell^\omega$. Final state of A_r, f_r, is merged with the initial and final state of A_ℓ.

Theorem 2 (Kleene for ω-strings) *Sets of the form $L(e)$, for an ω-regular expression e, are precisely the ω-regular languages.*

Proof: First, we prove that if e is a regular expression, then $L(e)$ is an ω-regular language.

Any ω regular expression can be written as a finite union of expressions of the form $r_i\ell_i^\omega$, where r_i and ℓ_i are regular expressions. Therefore, it suffices to prove that we can construct Büchi automata for any expression of the form $r\ell^\omega$ (for r and ℓ regular expressions), since the class of ω-regular languages is closed under union. The proof of the latter is the same as in the case of finite automata and therefore we do not give the proof.

Suppose $e = r\ell^\omega$ for regular expressions r and ℓ. Suppose A_r and A_ℓ are normalized finite automata[1] for r and ℓ. We can then construct the automaton shown in figure 1 for $L(e)$.

In the other direction, we prove that if A is a Büchi automaton, then we can construct a regular expression for $L_\omega(A)$. Consider the following two languages.

$$L_1(q_0, q) = \{x \in \Sigma^* \,|\, \text{that there is a run of } A \text{ on } x \text{ that starts in } q_0 \text{ and ends in } q\}$$
$$L_2(q) = \{x \in \Sigma^* \,|\, \text{that there is a run of } A \text{ on } x \text{ that starts in } q \text{ and ends in } q\}$$

It is clear that it is possible to construct regular expressions $e_1(q_0, q)$ and $e_2(q)$ for $L_1(q_0, q)$ and $L_2(q)$ since are regular languages. Moreover,

$$L_\omega(A) = \bigcup_{q_0 \in Q_0} \bigcup_{q \in F} e_1(q_0, q) e_2^w(q),$$

which concludes the proof.

Corollary 1 *Every ω-regular language is a finite union of languages of the form $L(e_1 e_2^\omega)$, where e_1 and e_2 are regular expressions.*

[1] A normalized finite automaton contains exactly one finite state, from which no transitions emerge.

4 ω-regular are closed under complement

Theorem 3 (Büchi) *For every Büchi automaton A, there exists a possibly non-deterministic Büchi automaton A' such that $L_\omega(A') = \Sigma^\omega - L_\omega(A)$.*

Proof: The original proof of Büchi uses Ramsey Theory. For our proof, we first need to introduce Müller and Rabin automata.

Definition 6 (Müller automaton) *A Müller Automaton is a tuple $A = (\Sigma, Q, Q_0, \delta, \mathcal{F} \subseteq 2^Q)$, where Σ, Q and $Q_0 \subseteq Q$ have the same definition as in Büchi automata.*

Definition 7 (Accepting run of a Müller automaton) *A run ρ is accepting for a Müller Automaton A iff $Inf(\rho) \in \mathcal{F}$.*

Definition 8 (Rabin automaton) *A Rabin automaton is a tuple $A = (\Sigma, Q, Q_0, \delta, \{(v_i, u_i) \,|\, v_i, u_i \subseteq 2^Q\})$, where Σ, Q and $Q_0 \subseteq Q$ and δ have the same definition as in Büchi automata.*

Definition 9 (Accepting run of a Rabin automaton) *A run ρ is accepting for a Rabin Automaton A iff $v_i \cap Inf(\rho) = \varnothing$ and $u_i \cap Inf(\rho) \neq \varnothing$.*

Theorem 4 *Non-deterministic Büchi, Rabin and Müller automata accept exactly the class of ω-regular languages.*

Proof:

1. (Büchi automaton $\Rightarrow$ Müller) It is easy to construct a Müller automaton A' equivalent to a Büchi automaton $A = (\Sigma, Q, Q_0, \delta, F)$. Indeed, define $A' = (\Sigma, Q, Q_0, \delta, \mathcal{F})$ with
$$\mathcal{F} = \{x \in 2^Q \,|\, x \cap F \neq \varnothing\}.$$
2. (Rabin $\Rightarrow$ Müller) Given a Rabin automaton $A = (\Sigma, Q, Q_0, \delta, F)$, the Müller automaton $A' = (\Sigma, Q, Q_0, \delta, \mathcal{F})$, where
$$\mathcal{F} = \{x \,|\, \text{there exist } (u_i, v_i) \in F, \text{and } u_i \cap x = \varnothing, v_i \cap x \neq \varnothing\},$$
is such that $L(A) = L(A')$.
3. (Müller $\Rightarrow$ Büchi). Suppose $A_M = (\Sigma, Q, Q_0, \delta, \mathcal{F})$ is a Müller automaton. We can construct a Büchi automaton $A_B = (\Sigma, Q', Q'_0, \delta', \{p\})$ in the following way. First A_B gueses one particular set $F \in \mathcal{F}$. Then it begins simulating A_M. Each time A_B sees a state $q \in F$ of A_M it "remembers it" in its finite control. Once A_B has seen all the states in F, it enters into state p, and the remembering cycle starts all over again.
4. (Büchi $\Rightarrow$ Rabin) This is McNaughton's theorem which is shown below.

Theorem 5 (McNaughton 1968, Safra 1988) *For every nondeterministic Büchi automaton A with n states, there exists an equivalent Rabin automaton with $O(2^{n \log n})$ states and $O(n)$ accepting pairs.*

Corollary 2 *For every nondeterministic Büchi automaton A with n states, there exists an equivalent deterministic Müller automaton with $O(2^{n \log n})$ states.*

Corollary 3 (Büchi's Theorem) *Over Σ^ω, regular = MSO definable.*

Proof: The same as in the final case. To handle negation though we do not use the complement construction but rather we translate the Büchi automaton to a Rabin Automaton, we flip accepting states for non-accepting and vice versa, and finally we translate the resulting automaton to a Büchi automaton.

5 S1S is decidable

The S1S logic is MSO over the structure $(\mathbb{N}, succ)$. As before, for every formula $\varphi(X_1, \ldots, X_n)$ of S1S, it is possible to construct a Büchi automaton A_φ such that $\varphi(X_1, \ldots, X_n)$ is true iff A_φ accepts a nonempty language. The proof of this is the same as in the finite case.

One important question to ask here is what is the complexity of determining whether a Büchi automaton $A = (\Sigma, Q, Q_0, \delta, F)$ accepts an nonempty language. The problem reduces to graph reachability (which by the way is NLOGSPACE-complete), since we have to determine whether there exist states $q_0 \in Q_0$ and $q_f \in F$ such that there is a path from q_0 to q_f and that there is a path from q_f to q_f.

Next time we'll prove a more interesting result, namely that wSIS=SIS.