Automata and Logic (CSC2428)
Lecture 6

Prof. Leonid Libkin
notes by Arie Gurfinkel

February 24, 2005

1 Safra's Construction

Theorem 1 (McNaughton,Safra) *For every Büchi automaton A with n states, there exists a deterministic Rabin automaton A' with $2^{O(n \log n)}$ states and $O(n)$ accepting pairs such that $L_\omega(A') = L_\omega(A)$.*

We start by showing that the usual powerset construction does not work. For example, consider an automaton A and an automation 2^A constructed by the powerset construction, shown in Figure 1. The two automata are not equivalent, $L_\omega(A) = \emptyset \neq a^\omega = L_\omega(2^A)$. The problem is that the fact that the final state $\{q_0, q_1\}$ of the powerset automaton 2^A occurs infinitely often on a run does not guarantee that the automaton A has a run on which its final state q_1 occurs infinitely often. Thus, we need to augment the powerset construction to track final states separately.

Consider the following example. Let A be a Büchi automaton, Y_0 be a state in the corresponding powerset automaton, and X_1 be a state reachable from Y_0 on a finite word u_0. Furthermore, let $F_1 = X_1 \cap F$ be the set of final states of A contained in X_1, and Y_1 be a state that is reachable from both X_1 and F_1 on some finite word v_0. This situation is depicted in Figure 1(c). Then, for every state q in Y_1, there exists a run of A on u_0v_0 that passes through at least one accepting state in F_1. Continuing with this construction, we can show that for every state in Y_2 there exists a run of A that passes through two accepting states, etc. Note that this does not guarantee that A has a run that passes through the *same* accepting state twice. However, since the number of accepting states of A is finite, then existence of an infinitely many sets X_i, Y_i, and F_i for an ω-word w guarantees that A has an accepting run on ω. To capture this intuition, we start by introducing the notion of a Safra Tree.

Definition 1 (Safra Tree) *Given a set Q of n states, a* Safra Tree *is a tree T such that*

- *nodes of T are labeled by $\{1, \ldots, 2n\}$,*
- *there exists a function $\lambda : T \to 2^Q \setminus \{\emptyset\}$ such that*
  - *for any two sibling nodes v_1 and v_2, $\lambda(v_1) \cap \lambda(v_2) = \emptyset$,*
  - *for every node v, $\bigcup_{v_i \in child(v)} \lambda(v_i) \subset \lambda(v)$,*

(a) (b) (c)

Figure 1: (a) A one-letter Büchi automaton A, and (b) the result of a powerset consutrction 2^A.

- *and some leaves of T are marked with a special symbol "!".*

It is easy to see that each node v has at most n children, and the height of the tree is at most $n - 1$.

Lemma 1 *If T is a Safra Tree over a set of states Q, then T has at most $n = |Q|$ nodes. Furthermore, the number of Safra Trees over Q in $2^{O(n \log n)}$.*

Proof:
For each state $q \in Q$ there exists a unique tree node $v \in T$ such that $q \in \lambda(v)$ and $\forall v' \in child(v) \cdot q \notin \lambda(v')$. Thus, we can define a partial function $f : Q \to T$ that for each state q associates its unique node v. The function f is onto, hence, $|T| \leq |Q|$.

To prove the second part of the lemma, we note that each Safra Tree T is uniquely determined by:

- a function $f_1 : Q \to \{0, \ldots, 2n\}$ mapping each state to a node index,
- a partial function parent : $\{1, \ldots, 2n\} \to \{1, \ldots, 2n\}$ mapping each node to its parent,
- a partial function next_sibling : $\{1, \ldots, 2n\} \to \{1, \ldots, 2n\}$ mapping each node to its immediate sibling, and
- a set $M \subseteq \{1, \ldots, 2n\}$ of marked nodes.

By a simple counting argument it follows that the number of Safra Trees is at most:

$$(2n + 1)^n + 2(2n)^{2n} + 2^{2n} = 2^{O(n \log n)}$$

□

We now proceed with constructing a deterministic Rabin automaton. Given a Büchi automaton $A = (Q, Q_0, \delta, F)$, a Rabin automaton $A' = (\mathcal{T}, T_0, \delta', \{(L_i, U_i) \mid i \in [1, \ldots, 2n])$ is constructed as follows.

- $\mathcal{T}$ is the set of all Safra Trees over Q,
- for a tree $T \in \mathcal{T}$, and $a \in \Sigma$, $\delta'(T, a)$ is defined in the following steps:

1. for each node $v \in T$, add a new rightmost child v', labeled with $\lambda(v) \cap F$, and assign it the smallest available index
2. for each node $v \in T$, replace its label with $\delta(\lambda(v), a)$, where $\delta(X, a) = \bigcup_{x \in X} \delta(x, a)$
3. to convert the resulting tree to a Safra Tree: (a) for any two siblings v_1 and v_2 such that $X = \lambda(v_1) \cap \lambda(v_2) \neq \emptyset$, remove states in X from the rightmost sibling, and (b) remove all nodes with an empty label
4. for every node $v \in T$ such that $\lambda(v) = \bigcup_{v_i \in child(v)} \lambda(v_i)$, delete the subtree rooted at v, and mark v with "!"

- T_0 is a tree with a single node labeled by Q_0
- $L_i = \{T \in \mathcal{T} \mid T$ has *no* nodes with index $i\}$
- $U_i = \{T \in \mathcal{T} \mid$ a node with index i is *marked* in T$\}$

Intuitively, the acceptance condition of A' ensures that there exists a node v that, after some finite prefix, is never removed from the tree, and this node becomes marked infinitely often.

The following theorem shows that the construction is optimal.

Theorem 2 (Safra) *There is no translation from a Büchi automaton to a deterministic Rabin automaton such that the Rabin automaton has $2^{O(n)}$ states and $O(n)$ accepting pairs.*

2 Weak S1S is not that weak

Theorem 3 *wS1S = S1S*

Proof:
For any S1S formula φ, we can construct a corresponding Büchi automaton A^B_φ, and then apply Safra's construction to obtain a deterministic Rabin automaton A^R_φ. We now show that for any deterministic Rabin automaton $A' = (Q, q_0, \delta, \{L_i, U_i\})$ there exists a wMSO formula ψ.

Let A_{Z_i}, where $Z \in \{L, U\}$, denote a DFA obtained from A' by letting Z_i be the set of final states. For example, $A_{L_1} = (Q, q_0, \delta, L_1)$. From the proof of Büchi's theorem, we know that there exists a wMSO formula $\alpha^{Z_i}(\vec{X}, x)$ that is true if and only if the automaton A_{Z_i} is in one of its final states when reading position x of the string representation of $\vec{X}$. Consider a wMSO formula

$$\psi(\vec{X}) = \bigvee_i \big(\exists x \forall y (y > x \rightarrow \neg \alpha_{L_i}(\vec{X}, y)\big) \wedge \big(\forall x \exists y (y > x \rightarrow \alpha_{U_i}(\vec{X}, y)\big)$$

The formula $\psi(\vec{X})$ is true if and only if there exists an i such that for the unique run of A' on the string encoding of $\vec{X}$, the states in L_i are visited finitely many times, and states in U_i are visited infinitely many times. That is, $\psi(\vec{X})$ is true if and only if A' accepts the string encoding of $\vec{X}$.

$\square$

3 Linear Temporal Logic

Let Σ be a finite set of propositional letters, the syntax of Linear Temporal Logic (LTL) is defined by the following grammar:

$$\varphi \ := \ a \mid \varphi \vee \varphi \mid X\varphi \mid Y\varphi \mid \varphi\, U\, \varphi \mid \varphi\, S\, \varphi$$

where $a \in \Sigma$. The intuitive interpretation of the temporal modalities is: X – next, Y – yesterday, U – until, and S – since.

For a word $w \in \Sigma^\omega$, let w_i denote the ith letter of w, i.e., $w = w_0 w_1 \ldots$. The semantics of an LTL formula φ is defined with respect to a Σ^ω word w and a natural number $n \in \mathbb{N}$ as follows:

$$\begin{array}{lcl}
(w,n) \models a & \leftrightarrow & a = w_n \\
(w,n) \models \neg\varphi & \leftrightarrow & (w,n) \not\models \varphi \\
(w,n) \models \varphi \vee \psi & \leftrightarrow & (w,n) \models \varphi \text{ or } (w,n) \models \psi \\
(w,n) \models X\varphi & \leftrightarrow & (w,n+1) \models \varphi \\
(w,n) \models Y\varphi & \leftrightarrow & n > 0 \text{ and } (w,n-1) \models \varphi \\
(w,n) \models \varphi\, U\, \psi & \leftrightarrow & \exists k > n \cdot (w,k) \models \psi \text{ and } \forall i \in [n,k) \cdot (w,i) \models \varphi \\
(w,n) \models \varphi\, S\, \psi & \leftrightarrow & n > 0 \text{ and } \exists k < n \cdot (w,k) \models \psi \text{ and } \forall i \in (k,n](w,i) \models \varphi
\end{array}$$

Additionally, we define the following useful syntactic abbreviations:

$$\top \ = \ \bigvee_{a\in\Sigma} a \qquad F\varphi \ = \ \top U \varphi \qquad G\varphi \ = \ \neg F \neg\varphi$$

where $\top$ stands for true, $F\varphi$ for φ is true in the *future*, and $G\varphi$ for φ is true *globally* (or always).

Over Σ^ω strings, LTL is equivalent to FO.

Theorem 4 (Kamp, 1968) *Let w be an ω-word and n a natural number, then (a) for each LTL formula φ there exists an $FO(<,(P_a)_{a\in\Sigma})$ formula $\alpha(x)$ such that*

$$(w,n) \models \varphi \ \leftrightarrow \ \mathcal{M}_w \models \alpha(n)$$

and (b) for each $FO(<,(P_a)_{a\in\Sigma})$ formula $\alpha(x)$ there exists an LTL formula φ such that

$$\mathcal{M}_w \models \alpha(n) \ \leftrightarrow \ (w,n) \models \varphi$$

Corollary 1 *For each FO sentence β there exists an LTL formula φ such that $\mathcal{M}_w \models \beta \leftrightarrow (w,0) \models \varphi$*