Automata and Logic (CSC2428)
Lecture 8

Prof. Leonid Libkin
notes by Yong Yuan

March 10, 2005

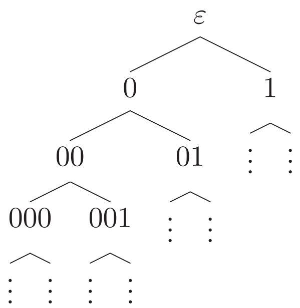
1 Proof of Rabin's Complementation Theorm

The goal of this lecture is to prove the Rabin's Complementation Theorem. In order to achieve this goal, we will first introduce *infinite tree domain*:

Definition A (binary)*infinite tree domain* is the set $D = \{0, 1\}^*$, such that if a tree $s \in D$, then

- both $s0, s1 \in D$
- neither $s0, s1 \in D$

Here is an example:



Definition A Σ-labeled infinite tree is a tuple $T = (D, \lambda_T)$ where

- the set $D = \{0, 1\}^*$
- the mapping $\lambda_T : \{0, 1\}^* \to \Sigma$

That is, each node of a $\Sigma - \textit{labeled}$ tree is labled with an expression $\in \Sigma$. A Σ-labeled tree can be viewed as a structure $S = (<_0, <_1, (P_a)_{a\in\Sigma})$, where $<_0$ is corresponding to the right successor of a node, and $<_1$ is corresponding to the left successor of a node. We use $\mathcal{T}_\Sigma^\omega$ to denote a Σ-labeled infinite tree.

Theorem 1.1 (Rabin 1970) *Given a structure in MSO theory:* $S2S = (\{\{0, 1\}^*\}, <_0, <_1)$. $S2S$ *is decidable.*

Recall that $S1S = (0^*, <_0)$, and $S1S$ is decidable, which can be proved by reduction to Büchi Automata. The general idea of proving that $S2S$ is decidable is the same as that of proving the decidability of $S1S$. That is, we can reduce S2S formula to infinite tree automata, and then show the emptiness problem for the infinite tree automata is decidable. The proof of the case for complementation is the hard part.

Definition An automaton (non-deterministic) over an infinite $\Sigma - \textit{lebeled}$ tree s is a tuple $\mathcal{A} = (Q, q_0, \Delta, Acc)$, where

- Q : the set of states in the automaton

- Δ: $Q \times \Sigma \rightarrow 2^{Q \times Q}$
- Acc: the acceptance condition over an ω-string.

Definition A run of an automaton $\mathcal{A}$ over an infinite tree is a mapping $\rho_{\mathcal{A}} : \{0,1\}^* \rightarrow Q \in \mathcal{T}_Q^{\omega}$, such that

- $\rho(\epsilon) = q_0$
- if $\rho(s) = q$, $\lambda_T(s) = a$, then $(\rho(s0), \rho(s1)) \in \delta(q,a)$

Here is an example:

$$\begin{array}{c} q \quad a \\ \swarrow \quad \searrow \\ q_1 \quad q_2 \end{array} \qquad \Rightarrow (q_1, q_2) \in \delta(q,a)$$

Definition An infinite path is a sequence of nodes $s_0 s_1 \ldots$ s.t. $s_0 = s_\epsilon$, and s_{i+1} is either $s_i 0$ or $s_i 1$. $\rho_{\mathcal{A}} \mid \pi$ is a sequence $\rho_{\mathcal{A}}(s_0)\rho_{\mathcal{A}}(s_1) \ldots \in Q^{\omega}$

A run of an automaton $\mathcal{A}$, $\rho_{\mathcal{A}}$, is accepting if for each path π, $\rho_{\mathcal{A}} \mid \pi$ satisfies the accepting condition Acc. For example, given Büchi acceptance condition $F \subseteq Q, \forall \pi \cdot Inf(\rho_{\mathcal{A}} \mid \pi) \cap F \neq \emptyset$. If Muller acceptance condition $F \subseteq 2^Q$ is given, then $\forall \pi \cdot Inf(\rho_{\mathcal{A}} \mid \pi) \in F - \mathcal{C}$

Exercise: For automata over infinite trees, Büchi is strictly weaker than Muller: Büchi $\subset$ Muller.

we can consider this example: Given a run, if only finitely many a's accur along each path,this run will only be accepted by Muller automata. Such run can be produced with the following transition function δ: $\delta(q,a) = (q_a, q_a)$, and $\delta(q,b) = (q_b, q_b)$.

Definition A *Rabin Chain Automaton* is a tuple $\mathcal{A} = (Q, q_0, \delta, Acc)$. The definitions of Q, q_0, and δ are the same as those of Rabin Automata. The acceptance condition Acc is a chain of subsets: $E_0 \subset F_0 \subset E_1 \subset F_1 \ldots \subset E_m \subset F_m \subseteq Q$, where for any path π, there exists an integer $k \leq m$, such that $Inf(\rho_{\mathcal{A}} \mid \pi) \cap E_k \neq \emptyset \wedge Inf(\rho_{\mathcal{A}} \mid \pi) \cap F_k \neq \emptyset$

Proposition 1.2 *Rabin Chain and Muller automata are equivalent.*

Proof Rabin Chain $\Longrightarrow$ Muller: easy. It is left as an exercise.

Muller $\Longrightarrow$ Rabin Chain:

Give an automaton $A = (Q, q_0, \delta, \mathcal{F})$, where $|Q| = n, Q = \{0,1,2,\ldots,n-1\}$, $q_0 = 0$, Δ is the graph of δ (i.e., $\Delta = \{(q,a,q',q'')|(q',q'') \in \delta(q,a)\}$, we will prove the direction Muller $\Longrightarrow$ Rabin Chain using the concept of latest appearance record (LAR). An LAR is a pair: (permutation of Q, a pointer to an element of Q). We will use $\natural$ as a pointer to the element left to it. For example, $(i_0, i_1, \ldots, \imath_j\natural, \ldots, i_{n-1})$ means the pointer is pointing to i_j.

The *Intuition* behind the proof is that we can use LAR to track the history of states when converting a Muller automaton to a Rabin Chain Automaton.

Let $S \in Q^\omega$. $S = s_0 s_1 s_2 \ldots s_m$ will be associated with LARs $L(s) = L(s_0)L(s_1)\ldots L(s_n)$. $(L(s_i), s_{i+1}) \longmapsto L(s_{i+1})$. For example, $(0, 2, 4\natural, 3, 1, 2) \longmapsto (0, 4\natural, 3, 1, 2,)$

Let $F = Inf(S), |F| = p$, then the first $n - p$ positions in an LAR will be pointed finitely times, and the positions $n - p + 1$ will be pointed at infinitely often.

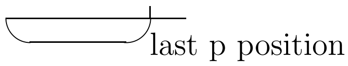


Figure 1: Illustration of LAR

We can construct a Rabin Chain automaton from a Muller automaton as follows. The initial states are $(1\natural, \ldots, n-1, 0)$. If we have $(i, a, i', i'') \in \Delta$, then let $f = (L, i) \longmapsto ((\ldots \natural, i), a, f(L, i'), f(L, i''))$.

Therefore, we can have $E_i = \{$L $\mid$ pointer is at a position p such that $p < i$, or $p = i \wedge L|i \notin F\}$, where $L|i$ indicates the position from i to the end of the permutation. If E_i is empty, we get $E_i \subseteq F_i$. In this case, we can cut E_1 and F_1 in order to make sure $E_i \subset F_i$.

This proposition implies that we can replace Muller automata with Rabin Chain automata in the later proofs. For the game used in the later proof, A Rabin Chain automaton's winning strategy is memoryless, while a Muller automaton's needs finite history.

Let A be a Muller or Rabin Chain automaton, and $L_\Sigma^{\mathcal{T}} = \{T \in \mathcal{T}_\Sigma^\omega$ accepted by A$\}$

Definition A set of tree in $\mathcal{T}_\Sigma^\omega$ is MSO-definable iff they are accepted by Muller (or Rabin Chain) tree automata.

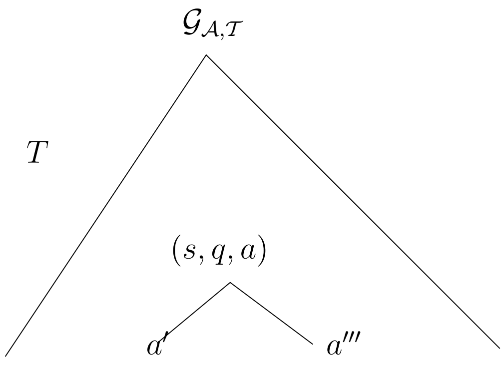
Tree automata $\Longrightarrow$ MSO-definable is easy. The proof is left as an exercise.

We will prove Tree $\Longrightarrow$ Automata. We will focus on complementation problem for automata on infinite trees because proving closure under union, projection, and projection for infinite trees is relatively easy.

Complementation of Rabin Chain Automata If $\mathcal{A}$ is a Rabin Chain Automaton, then there can be constructed a Rabin Chain Automaton $\mathcal{A}'$ s.t. $\mathcal{L}_\omega^T(\mathcal{A}') = \mathcal{T}_\Sigma^\omega - \mathcal{L}_\omega^T(\mathcal{A})$

We introduce a two-player Pathfinder game $\mathcal{G}_{\mathcal{A},\mathcal{T}}$, where $\mathcal{A}$ is an automaton, and $\mathcal{T}$ is a tree. The two players move alteratively. One player is an automaton, which picks a transition that has the form (s, q, a, q', q''). The other player is a pathfinder, which chooses direction in its move. For example, if the left direction is chosen, we get $(s0, a', q')$. If the right direction is chose, we get $(s1, a'', q'')$

The intuition is to show that the automata will accept a tree, while the pathfinder tries to pick the easiest way to violate the acceptance condition.



a node is of the form $(s \in \{0,1\}^*, a, q)$

Figure 2: Illustration of the moves

Each play of the game produces a sequence $(q_0 q_1 \ldots \in Q^\omega$. The automaton has a winning strategy if no matter how the pathfinder plays, the sequence produced by the play satisfies the acceptance condition.

Observation: The automata has a winning strategy in the game $\mathcal{G}_{\mathcal{A},\mathcal{T}}$ if and only $\mathcal{T} \in \mathcal{L}_\omega^{\mathcal{T}}(\mathcal{A}')$

We define the following sets:

- V_0 = set of positions for the automaton
- V_1 = set of positions for the pathfinder
- E = the set of edges between V_0 and V_1: $E \subseteq (V_0 \times V_1) \cup (V_1 \times V_0)$. That is, an edge exists only if a transition exists between a position in V_0 and a position in V_1.

Let position $P_0 \in V_0$, $P_1 \in V_1$, ..., $P_{2k} \in V_0$, $P_{2k+1} \in V_1$, ..., as game continues.

Strategy for the automata: $P_0 P_1 \ldots P_{2k} \to P_{2k+1} \in V_1$

Strategy for the pathfinder: $P_0 P_1 \ldots P_{2k+1} \in V_1 \to P_{2k+2} \in V_0$

A memoryless strategy for the Rabin Chain automaton would be $P_{2k} \to P_{2k+1} \in V_1, P_{2k+1} \to P_{2k+2} \in V_0$

Theorem 1.3 (Determinacy) *For every position, either the automaton or the pathfinder has a memoryless winning strategy.*

If $\mathcal{A}$ is a Rabin Chain automaton, then $\mathcal{T} \notin \mathcal{L}_\omega^{\mathcal{T}}$ iff the automaton does not have a winning strategy in $\mathcal{G}_{\mathcal{A},\mathcal{T}}$, which is true iff the pathfinder has a winning strategy in $\mathcal{G}_{\mathcal{A},\mathcal{T}}$. Please note, the winning strategy will be memoryless.

To prove the complementation problem, we need to find an automaton $\mathcal{A}'$ s.t. $\mathcal{A}'$ accepts $\mathcal{T}$ iff the pathfinder has a winning strategy from the initial position in $\mathcal{G}_{\mathcal{A},\mathcal{T}}$.

The pathfinder looks at the position $(s \in \{0,1\}^*, a \in \Sigma, (q, a, q', q'') \in \Delta)$, and then chooses a direction from the set $\{0, 1\}$.

To find the pathfinder's memoryless strategy, we first define the mapping $\sigma : \{0,1\}^* \times \Sigma \times \Delta \to \{0,1\}$. Let $\theta = \{f : \Sigma \times \Delta \to \{0,1\}\}$, then $\sigma = \{\sigma_s \in \theta, s \in \{0,1\}^*\}$

We know that $\sigma \in \mathcal{T}_\theta^\omega$, an infinite tree over the alphabet θ. Therefore, $\sigma \in \mathcal{T}_\theta^\omega \wedge T \in \mathcal{T}_\Sigma^\omega \implies (\sigma, T) \in \mathcal{T}_{\theta\times\Sigma}^\omega$.

A pathfinder has a memoryless winning strategy in the game $\mathcal{G}_{\mathcal{A},\mathcal{T}}$ iff $\exists \sigma \in \mathcal{T}_\theta^\omega$, s.t. for every path π through an input tree and for every sequence of the moves $\tau_0\tau_1\tau_2 \ldots \tau_k$ produced by the automaton, if $\tau_0\tau_1 \ldots$ is consistent with π, then the resulting sequence of states violates the Rabin Chain acceptance condition. For convenience, we introduce the following notations:

- $C_1(\sigma)$: for every path π through an input tree
- $C_2(\sigma, \pi)$: for every sequence of the moves $\tau_0\tau_1\tau_2 \ldots \tau_k$ produced by the automaton
- $C_3(\sigma, \pi, \tau)$: if $\tau_0\tau_1 \ldots$ is consistent with π, then the resulting sequence of states violates the Rabin Chain acceptance condition.

$C_3(\sigma, \pi, \tau)$ is an FO condition over π, τ, and σ. There is a deterministic string Muller automaton that verifies C_3 along the path π.

$C_2(\sigma, \pi)$: $\forall \tau, C_3(\sigma, \pi, \tau) \cdot \neg\exists\tau \cdot \neg C_3(\sigma, \pi, \tau) \implies \exists$ deterministic Muller automaton for C_2.

Since C_2 is deterministic, there is a deterministic Muller Tree automaton that verifies if every path π satisfies C_2.

$C_1 = \exists\sigma \cdot C_1(\sigma) \implies$ guess a σ and then check C_1.