Notes on Nondeterministic Finite State Automata

Scribe: Matthew Pecha Lecturer/Editor: Chris Eppolito

1 May 2020

Last time we discussed deterministic automata; writing such an automaton can be difficult in general.

Example 1. Write an automaton accepting only strings of the form $11x11$ where $x \in \{0,1\}^*$.

Rather than approach this problem directly (which is a bit difficult), we will introduce a new model for precisely this type of problem. The new model will relax the constraints on transitions

Definition. A *nondeterministic finite state automaton* is a quintuple M = (S, s_0, F, I, t) with

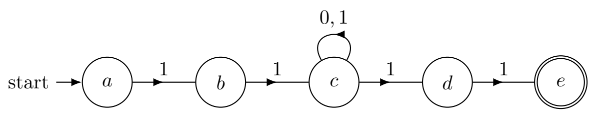
1. a finite set S of *states* with a distinguished *initial state* $s_0 \in S$ and a set $F \subseteq S$ of final states,
2. a finite set I of *input characters*,
3. a transition function $t\colon S \times I \to \mathbb{P}(S)$.

A word $w \in A^*$ is *recognized* by M when there is a w-labeled path from the initial state to a final state.

Note that multiple paths labeled by the same word w starting from the initial state are possible. Thus this model trades deterministic behavior for easy description.

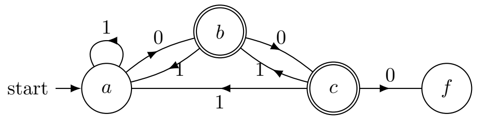
Our conventions for drawing nondeterministic automata are the same as before.

Example 2. The nondeterministic automaton below recognizes bit strings of the form $11x11$.



Showing that $L_M = \{11x11 : x \in \{0,1\}^*\}$ amounts to an easy case analysis.

Example 3. The nondeterministic automaton below recognizes bit strings without three consecutive 0's.



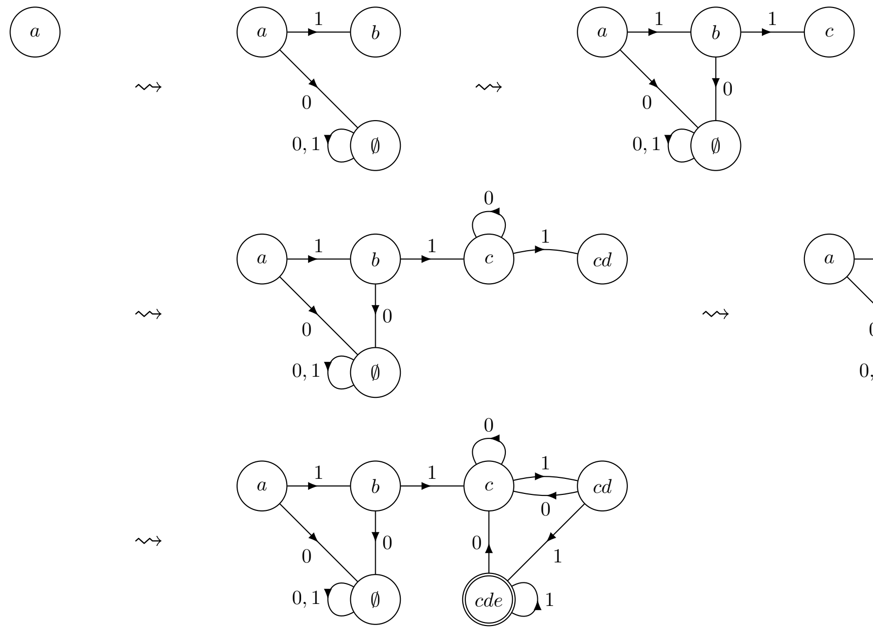
We claim that the languages accepted the deterministic finite state automata are precisely the languages accepted by nondeterministic finite state automata.

If L is accepted by an automaton M, then letting $t'(s,i) := \{t(s,i)\}$ we have L is trivially accepted by

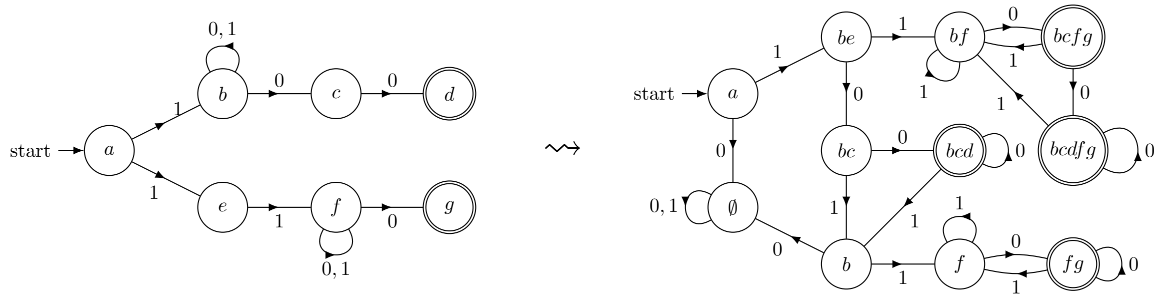
$$M' := (S(M), s_0(M), F(M), I(M), t').$$

Given a nondeterministic automaton, we must construct a deterministic automaton accepting its language. Before giving technical details, we will illustrate the idea of the construction on the automaton we gave above. The rough idea of this construction is to replace states by sets of states. The new final states will be those subsets which contain an old final state. Transitions and states are built up recursively together. For ease of notation we drop the set notation below.

Example 4. The construction outlined above is implemented below for the automaton of Example 2.



Example 5. The automaton below recognizes bit strings of the form either $1x00$ or $11x0$ for some $x \in \{0,1\}^*$. Applying the above construction, we also obtain a deterministic automaton to do the same below.



Now we return to outline a proof of the claim from earlier.

Proposition. *Let M be a nondeterministic automaton. Consider the sequence $N_k = (S_k, t_k)$ defined by*

1. *For $k = 0$ we have $S_0 = \{\{s_0(M)\}\}$ and $t_0 \colon \emptyset \times I \to \emptyset$.*
2. *Given $N_k = (S_k, t_k)$, define*

$$S_{k+1} = S_k \cup \left\{ \bigcup_{\alpha \in \omega} t_M(\alpha, i) : i \in I, \omega \in S_k \right\} \qquad \text{and} \qquad t_{k+1}(\omega, i) := \bigcup_{\alpha \in \omega} t_M(\alpha, i).$$

There is an integer $k \in \mathbb{Z}^+$ such that $N_m = N_k$ for all $m \geq k$. Letting $F' := \{\omega \in S_m : F_M \cap \omega \neq \emptyset\}$, we have $M' := (S_m, \{s_0\}, F', I, t_m)$ is a deterministic automaton.

Proposition. *Let M be a nondeterministic automaton, let M' be the corresponding deterministic automaton, let $w \in I^*$, let $\alpha \in S$, and let $\omega \in S'$ be the terminus of the (unique!) w-labeled walk in M' starting at $\{s_0\}$. There is a w-labeled walk from s_0 to α in M if and only if $\alpha \in \omega$.*

Having these two technical (but simple) propositions, the following corollary is easy, and proves the claim.

Corollary. *For all nondeterministic automata M we have $L_M = L_{M'}$.*