Lecture 09: Myhill-Nerode Theorem

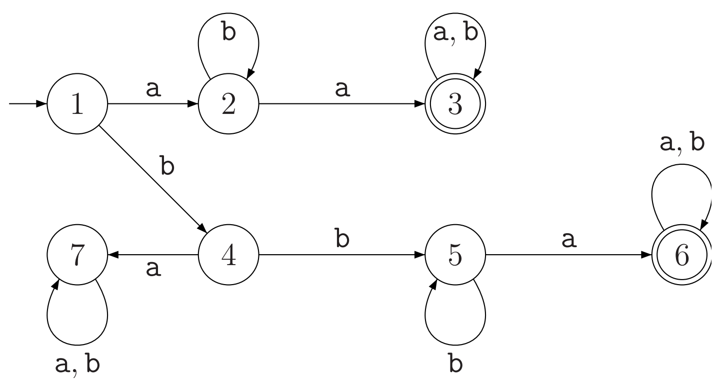
16 February 2010

In this lecture, we will see that every language has a unique minimal DFA. We will see this fact from two perspectives. First, we will see a practical algorithm for minimizing a DFA, and provide a theoretical analysis of the situation.

1 On the number of states of DFA

1.1 Starting a DFA from different states

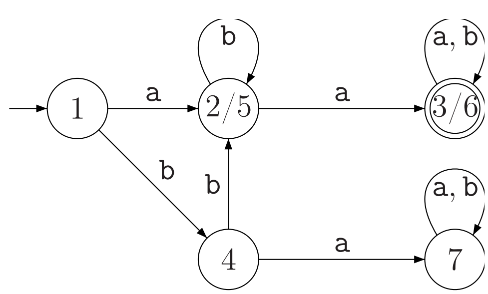
Consider the DFA on the right. It has a particular defined start state. However, we could start it from any of its states. If the original DFA was named M, define M_q to be the DFA with its start state changed to state q. Then the language L_q, is the one accepted if you start at q.



For example, in this picture, L_3 is $(\mathtt{a}+\mathtt{b})^*$, and L_6 is the same. Also, L_2 and L_5 are both $\mathtt{b}^*\mathtt{a}(\mathtt{a}+\mathtt{b})^*$. Finally, L_7 is $\emptyset$.

Suppose that $L_q = L_r$, for two states q and r. Then once we get to q or r, the DFA is going to do the same thing from then on (i.e., its going to accept or reject *exactly* the same strings).

So these two states can be merged. In particular, in the above automata, we can merge 2 and 5 and the states 3 and 6. We can the new automata, depicted on the right.



1.2 Suffix Languages

Let Σ be some alphabet.

Definition 1.1 Let $L \subseteq \Sigma^*$ be any language.

The ***suffix language*** of L with respect to a word $x \in \Sigma^*$ is defined as

$$[\![L/x]\!] = \left\{ y \;\middle|\; x\,y \in L \right\}.$$

In words, $[\![L/x]\!]$ is the language made out of all the words, such that if we append x to them as a prefix, we get a word in L.

The ***class of suffix languages*** of L is

$$\mathcal{C}(L) = \left\{ [\![L/x]\!] \;\middle|\; x \in \Sigma^* \right\}.$$

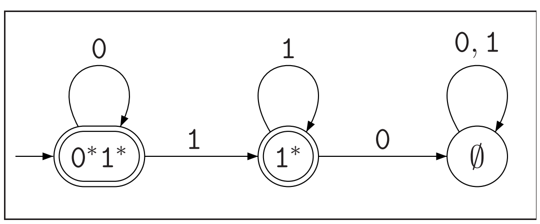
Example 1.2 For example, if $L = \mathtt{0}^*\mathtt{1}^*$, then:

- $[\![L/\epsilon]\!] = \mathtt{0}^*\mathtt{1}^* = L$
- $[\![L/\mathtt{0}]\!] = \mathtt{0}^*\mathtt{1}^* = L$
- $[\![L/\mathtt{0}^i]\!] = \mathtt{0}^*\mathtt{1}^* = L$, for any $i \in \mathbb{N}$
- $[\![L/\mathtt{1}]\!] = \mathtt{1}^*$
- $[\![L/\mathtt{1}^i]\!] = \mathtt{1}^*$, for any $i \geq 1$
- $[\![L/\mathtt{10}]\!] = \left\{ y \;\middle|\; \mathtt{10}y \in L \right\} = \emptyset$.

Hence there are only three suffix languages for L: $\mathtt{0}^*\mathtt{1}^*$, $\mathtt{1}^*$, $\emptyset$. So $\mathcal{C}(L) = \left\{ \mathtt{0}^*\mathtt{1}^*,\ \mathtt{1}^*,\ \emptyset \right\}$.

As the above example demonstrates, if there is a word x, such that any word w that have x as a prefix is not in L, then $[\![L/x]\!] = \emptyset$, which implies that $\emptyset$ is one of the suffix languages of L.

Example 1.3 The above suggests the following automata for the language of Example 1.2: $L = \mathtt{0}^*\mathtt{1}^*$.



And clearly, this is the automata with the smallest number of states that accepts this language.

1.2.1 Regular languages have few suffix languages

Now, consider a DFA $M = (Q, \Sigma, \delta, q_0, F)$ accepting some language L. Let $x \in \Sigma^*$, and let M reach the state q on reading x. The suffix language $[\![L/x]\!]$ is precisely the set of strings w, such that xw is in L. But this is exactly the same as L_q. That is, $[\![L/x]\!] = L_q$, where q is the state reached by M on reading x. Hence the suffix languages of a regular language accepted by a DFA are precisely those languages L_q, where $q \in Q$.

Notice that the definition of suffix languages is more general, because it can also be applied to non-regular languages.

Lemma 1.4 *For a regular language L, the number of different suffix languages it has is bounded; that is $\mathcal{C}(L)$ is bounded by a constant (that depends on L).*

Proof: Consider the DFA $M = (Q, \Sigma, \delta, q_0, F)$ that accepts L. For any string x, the suffix language $[\![L/x]\!]$ is just the languages associated with L_q, where q is the state M is in after reading x.

Indeed, the suffix language $[\![L/x]\!]$ is the set of strings w such that $xw \in L$. Since the DFA reaches q on x, it is clear that the suffix language of x is precisely the language accepted by M starting from the state q, which is L_q. Hence, for every $x \in \Sigma^*$, $[\![L/x]\!] = L_{\delta(q_0,x)}$, where q is the state the automaton reaches on x.

As such, any suffix language of L is realizable as the language of a state of M. Since the number of states of M is some constant k, it follows that the number of suffix languages of L is bounded by k. ■

An immediate implication of the above lemma is the following.

Lemma 1.5 *If a language L has infinite number of suffix languages, then L is not regular.*

1.2.2 The suffix languages of a non-regular language

Consider the language $L = \left\{ \mathtt{a}^n\mathtt{b}^n \;\middle|\; n \in \mathbb{N} \right\}$. The suffix language of L for $\mathtt{a}^i$ is

$$[\![L/\mathtt{a}^i]\!] = \left\{ \mathtt{a}^{n-i}\mathtt{b}^n \;\middle|\; n \in \mathbb{N} \right\}.$$

Note, that $\mathtt{b}^i \in [\![L/\mathtt{a}^i]\!]$, but this is the only string made out of only $\mathtt{b}$s that is in this language. As such, for any i, j, where i and j are different, the suffix language of L with respect to $\mathtt{a}^i$ is different from that of L with respect to $\mathtt{a}^j$ (i.e. $[\![L/\mathtt{a}^i]\!] \neq [\![L/\mathtt{a}^j]\!]$). Hence L has infinitely many suffix languages, and hence is not regular, by Lemma 1.5.

Let us summarize what we had seen so far:

- Any state of a DFA of a language L is associated with a suffix language of L.
- If two states are associated with the same suffix language, that we can merge them into a single state.
- At least one non-regular language $\left\{ \mathtt{a}^n\mathtt{b}^n \;\middle|\; n \in \mathbb{N} \right\}$ has an infinite number of suffix languages.

It is thus natural to conjecture that the number of suffix languages of a language, is a good indicator of how many states an automata for this language would require. And this is indeed true, as the following section testifies.

2 Regular Languages and Suffix Languages

2.1 A few easy observations

Lemma 2.1 *If $\epsilon \in [\![L/x]\!]$ if and only if $x \in L$.*

Proof: By definition, if $\epsilon \in [\![L/x]\!]$ then $x = x\epsilon \in L$. Similarly, if $x \in L$, then $x\epsilon \in L$, which implies that $\epsilon \in [\![L/x]\!]$. ■

Lemma 2.2 *Let L be a language over alphabet Σ. For all $x, y \in \Sigma^*$ we have that if $[\![L/x]\!] = [\![L/y]\!]$ then for all $\mathtt{a} \in \Sigma$ we have $[\![L/x\mathtt{a}]\!] = [\![L/y\mathtt{a}]\!]$.*

Proof: If $w \in [\![L/x\mathtt{a}]\!]$, then (by definition) $x\mathtt{a}w \in L$. But then, $\mathtt{a}w \in [\![L/x]\!]$. Since $[\![L/x]\!] = [\![L/y]\!]$, this implies that $\mathtt{a}w \in [\![L/y]\!]$, which implies that $y\mathtt{a}w \in L$, which implies that $w \in [\![L/y\mathtt{a}]\!]$. This implies that $[\![L/x\mathtt{a}]\!] \subseteq [\![L/y\mathtt{a}]\!]$, a symmetric argument implies that $[\![L/y\mathtt{a}]\!] \subseteq [\![L/x\mathtt{a}]\!]$. We conclude that $[\![L/x\mathtt{a}]\!] = [\![L/y\mathtt{a}]\!]$. ■

2.2 Regular languages and suffix languages

We can now state a characterization of regular languages in term of suffix languages.

Theorem 2.3 (Myhill-Nerode theorem.) *A language $L \subseteq \Sigma^*$ is regular if and only if the number of suffix languages of L is finite (i.e. $\mathcal{C}(L)$ is finite).*

Moreover, if $\mathcal{C}(L)$ contains exactly k languages, we can build a DFA for L that has k states; also, any *DFA accepting L must have k states.*

Proof: If L is regular, then $\mathcal{C}(L)$ is a finite set by Lemma 1.4.

Second, let us show that if $\mathcal{C}(L)$ is finite, then L is regular. Let the suffix languages of L be

$$\mathcal{C}(L) = \left\{ [\![L/x_1]\!], [\![L/x_2]\!], \ldots, [\![L/x_k]\!] \right\}. \tag{1}$$

Note that for any $y \in \Sigma^*$, $[\![L/y]\!] = [\![L/x_j]\!]$, for some $j \in \{1, \ldots, k\}$.

We will construct a DFA whose states are the various suffix languages of L; hence we will have k states in the DFA. Moreover, the DFA will be designed such that after reading y, the DFA will end up in the state $[\![L/y]\!]$.

The DFA is $M = (Q, \Sigma, q_0, \delta, F)$ where

- $Q = \left\{ [\![L/x_1]\!], [\![L/x_2]\!], \ldots, [\![L/x_k]\!] \right\}$
- $q_0 = [\![L/\epsilon]\!]$,
- $F = \left\{ [\![L/x]\!] \;\middle|\; \epsilon \in [\![L/x]\!] \right\}$. Note, that by Lemma 2.1, if $\epsilon \in [\![L/x]\!]$ then $x \in L$.
- $\delta\big([\![L/x]\!], \mathtt{a}\big) = [\![L/x\mathtt{a}]\!]$ for every $\mathtt{a} \in \Sigma$.

The transition function δ is well-defined because of Lemma 2.2.

We can now prove, by induction on the length of x, that after reading x, the DFA reaches the state $[\![L/x]\!]$. If $x \in L$, then $\epsilon \in [\![L/x]\!]$, which implies that $\delta(q_0, x) = [\![L/x]\!] \in F$. Thus,

$x \in L(M)$. Similarly, if $x \in L(M)$, then $[\![L/x]\!] \in F$, which implies that $\epsilon \in [\![L/x]\!]$, and by Lemma 2.1 this implies that $x \in L$. As such, $L(M) = L$.

We had shown that the DFA M accepts L, which implies that L is regular, furthermore M has k states.

We next prove that *any* DFA for L must have at least k states. So, let $N = (Q', \Sigma, \delta_N q_{\text{init}}, F)$ any DFA accepting L. The language L has k suffix languages, generated by the strings $x_1, x_2, \ldots, x_k$, see Eq. (1).

For any $i \neq j$, we have that $[\![L/x_i]\!] \neq [\![L/x_j]\!]$. As such, there must exist a word w such that $w \in [\![L/x_j]\!]$ and $w \notin [\![L/x_j]\!]$ (the symmetric case where $w \in [\![L/x_j]\!] \setminus [\![L/x_i]\!]$ is handled in a similar fashion. But then, $x_i w \in L$ and $x_j w \notin L$. Namely, $N(q_{\text{init}}, x) \neq N(q_{\text{init}}, y)$, and the two states that N reaches for x_i and x_j respectively, are distinguishable. Formally, let $q_i = \delta(q_{\text{init}}, x_i)$, for $i = 1, \ldots, k$. All these states are pairwise distinguishable, which implies that N must have at least k states. ■

Remark 2.4 The full Myhill-Nerode theorem also shows that all minimal DFAs for L are isomorphic, i.e. have identical transitions as well as the same number of states, but we will not show that part.

This is done by arguing that any DFA for L that has k states must be *identical* to the DFA we created above. This is a bit more involved notationally, and is proved by showing a $1-1$ correspondence between the two DFAs and arguing they must be connected the same way. We omit this part of the theorem and proof.

2.3 Examples

Let us explain the theorem we just proved using an example.

Consider the language $L \subseteq \{\mathtt{a}, \mathtt{b}\}^*$:

$$L = \left\{ w \;\middle|\; w \text{ has an odd number of } \mathtt{a}\text{'s} \right\}.$$

The suffix language of $x \in \Sigma^*$, where x has an even number of a's is:

$$[\![L/x]\!] = \left\{ w \;\middle|\; w \text{ has an odd number of } \mathtt{a}\text{'s} \right\} = L.$$

The suffix language of $x \in \Sigma^*$, where x has an odd number of a's is:

$$[\![L/x]\!] = \left\{ w \;\middle|\; w \text{ has an even number of } \mathtt{a}\text{'s} \right\}.$$

Hence there are only two distinct suffix languages for L. By the theorem, we know L must be regular and the minimal DFA for L has two states. Going with the construction of the DFA mentioned in the proof of the theorem, we see that we have two states, $q_0 = [\![L/\epsilon]\!]$ and $q_1 = [\![L/\mathtt{a}]\!]$. The transitions are as follows:

- From $q_0 = [\![L/\epsilon]\!]$, on a we go to $[\![L/\mathtt{a}]\!]$, which is the state q_1.
- From $q_0 = [\![L/\epsilon]\!]$, on b we go to $[\![L/b]\!]$, which is same as $[\![L/\epsilon]\!]$, i.e. the state q_0.
- From $q_1 = [\![L/\mathtt{a}]\!]$, on a we go to $[\![L/aa]\!]$, which is same as $[\![L/\epsilon]\!]$, i.e. the state q_0.

- From $q_1 = \llbracket L/a \rrbracket$, on b we go to $\llbracket L/ab \rrbracket$, which is same as $\llbracket L/a \rrbracket$, i.e. the state q_1.

The initial state is $\llbracket L/\epsilon \rrbracket$ which is the state q_0, and the final states are those states $\llbracket L/x \rrbracket$ that have ϵ in them, which is the set $\{q_1\}$.

We hence have a DFA for L, and in fact this is the minimal automaton accepting L.