CS 361 Handout: Myhill-Nerode

Shikha Singh

Let L be a language over the alphabet Σ. Two strings x and y are *indistinguishable with respect to* L if for any $z \in \Sigma^*$, $xz \in L$ if and only if $yz \in L$. In other words, x and y are either both in L or both not in L, and appending the same string to both x and y yields two strings that are either both in L or both not in L.

The notion of indistinguishability allows us to define the following equivalence relation $\equiv_L$ on Σ^*. We say $x \equiv_L y$ if x and y are indistinguishable. By definition of indistinguishability, $x \equiv_L y$ if and only if $y \equiv_L x$, and we always have $x \equiv_L x$. It is also easy to see that if $x \equiv_L y$ and $y \equiv_L w$ then we must have $x \equiv_L w$. Thus the relation $\equiv_L$ on Σ^* is an equivalence relation since it is reflexive, symmetric and transitive. This relation is called the Myhill-Nerode relation after the people who introduced it.

Consider $L = \{w \mid |w| \text{ is even}\}$ and let $[x]$ be an equivalence class for x under $\equiv_L$. Then we have two equivalence classes, first the class $[e]$ of all strings $e \in L$ which have even length and second the equivalence class $[o]$ consisting of all strings $o \notin L$ of odd length.

The intuition behind strings being indistinguishable or not follows from considering the finite automaton for the languages and its computation on the strings.

Indistinguishability and DFAs Let $M = (Q, \Sigma, \delta, s, F)$ be a DFA for a language L. Consider any two strings $x, y \in \Sigma^*$. We say $x \sim_M y$ if and only if M reaches the same state on both x and y, that is, then there is a state $q \in Q$ such that starting at s, M reaches state q after reading string x and starting at s, M reaches the same state q after reading string y.

Claim 1. If $x \sim_M y$ then $x \equiv_{L(M)} y$.

Since x and y drive the machine M to the same state, appending z to the input results in identical computations ending in the same accepting or nonaccepting state.

Lemma 1. Let L be a language over the alphabet Σ. If the relation $\equiv_L$ over Σ^* has k equivalence classes, then every DFA for L must have at least k states.

Proof. If L is not regular, then there is no DFA for L, much less a DFA with less than k states. Now suppose L is regular and let M be the DFA such that $L(M) = L$. Suppose M has less than k states. Then by the pigeonhole principle there exists strings $x, y \in \Sigma^*$ such that x and y are in different equivalence classes of $\equiv_L$ but they drive M to the same state. Since x and y are not indistinguishable, there exists some $z \in \Sigma^*$ that distinguishes them, that is, there exists $z \in \Sigma^*$ such that $xz \in L$ but $yz \notin L$ (or vice versa). Since M reaches exactly the same state on x and y it can either accept both xz and yz or reject both xz and yz which leads to a contradiction. $\square$

Theorem 1 (Myhill-Nerode)**.** Let L be a language over Σ. Then L is regular if and only if the relation $\equiv_L$ over Σ^* has a finite number of equivalence classes.

Proof. If there are infinitely many equivalence classes, then it follows from Lemma 1 that no DFA can decide L, and so L is not regular.

The proof of the other direction will be an exercise on the next assignment. It requires us to show that if the relation $\equiv_L$ over Σ^* has a finite number of equivalence classes, then we can define a DFA for L that has a state corresponding to each equivalence class. $\square$

Using Myhill-Nerode to prove that a language L is not regular.

Example 1. Consider the language $L = \{a^n b^n \mid n \in \mathbb{N}\}$. Prove that L is not regular.

Let us use Myhill-Nerode to prove L is not regular, instead of using the pumping lemma. If we show that the language has infinite number of equivalence classes, we can conclude from Myhill-Nerode theorem that it is not regular.

Consider the string a^i for $i \in \mathbb{N}$. For each such string, there is a single extension such that the resulting string is in L (this extension is b^i) and all other extensions result in strings not in L. Therefore $\equiv_L$ contains one equivalence class for each $i \in \mathbb{N}$ corresponding to the starting string a^i. Since the number of equivalence classes are infinite, L is not regular.

Example 2. Consider the language $L = \{a^n \mid n \text{ is a power of } 2\}$. Prove that L is not regular.

If we show that the language has infinite number of equivalence classes, we can conclude from Myhill-Nerode theorem that it is not regular.

Consider the infinite set $S = \{a^{2^n} \mid n \in \mathbb{N}\}$. (It is infinite because it has one element for each natural number.) Let a^{2^i} and a^{2^j} be any two distinct strings in S. Without loss of generality, let $i < j$. Consider the strings $a^{2^i} a^{2^i}$ and $a^{2^i} a^{2^j}$. Then we know that $a^{2^i} a^{2^i} \in L$ because it has length 2^{i+1} but the string $a^{2^i} a^{2^j} \notin L$ because it has length $2^i(1 + 2^{j-i})$ which cannot be a power of 2.

Since S has an infinite set of strings that are all distinguishable relative to the language L, the relation $\equiv_L$ has infinitely many equivalence classes and thus by the Myhill-Nerode theorem L is not regular.