CS 4510-X

The Myhill-Nerode Theorem

Name: Abrahim Ladha, Frederic Faulkner

2.1 Pumping

The pumping lemma is not a perfect characterization of non-regular languages. There exist languages which can satisfy the conditions of the pumping lemma, but be non-regular.

2.2 Myhill-Nerode

Lucky for us, there does exist a perfect characterization. It helps us prove non-regular languages to be non-regular, or regular languages to be regular. It also implies a unique and minimal[1] DFA for each regular language.

To give a high level idea, for any DFA D, and two strings which end on the same state in D, say x, y. Since it is deterministic, there is one outgoing transition from that state on 0, so $x0, y0$ will also end in the same state given x, y ended in the same state. We can abuse this! Lets generalize.

For a language L, let $\sim_L$ be an equivalence relation on Σ^* defined by $x \sim_L y$ if for all $z \in \Sigma^*$, $xz \in L \iff yz \in L$. Note that taking $z = \varepsilon$ shows that, if $x \sim_L y$, then either both x and y are members of L, or neither is.

Theorem 2.1 *L is regular if and only if $\sim_L$ partitions Σ^* into a finite number of equivalence classes.*

($\Longrightarrow$) Suppose L is regular, and thus has a DFA to decide it, lets denote as D. For each $x \in \Sigma^*$, running x on D will stop in some state. Let $[q_i]$ be the set of all strings which stop on state q_i. Notice there are a finite number of such sets, one for each state, and $\Sigma^* = \cup_{i=1}^{n} [q_i]$, with each disjoint[2]. Let $x, y \in [q_i]$, and let $z \in \Sigma^*$, and let q_j be the state D finishes if it starts in q_i and reads z. Then xz and yz both cause the machine to end in q_j. If q_j is an accept state, $xz, yz \in L$; if q_j is a non-accept state, $xz, yz \notin L$. This implies $xz \in L \iff yz \in L$ whenever $x \sim y$. Each set $[q_i]$ is then an equivalence class of $\sim_L$, and there are a finite number of them.

($\Longleftarrow$) Let $\sim_L$ partition Σ^* into a finite number of equivalence classes. We can construct a DFA to decide L. Recall a DFA has the form $(Q, \Sigma, \delta, q_0, F)$. Let $q_i \in Q$ be the states and $[q_i]$ be the equivalence class for q_i.

- Q : Form n states, one for each equivalence class of $\sim_L$.
- Σ : is the alphabet of L.
- δ : For each $q_i \in Q$, $a \in \Sigma$, and $x \in [q_i]$, define $\delta(q_i, a) =$ the state for the class which contains xa. [3]

[1] with respect to the number of states
[2] And for q_f the final state, $L = [q_f]$
[3] For this function to be well defined, we need that $x \sim_L y \implies \text{xz} \sim_L yz$. Do you see why?

- q_0 : Let the start state be the state for the equivalence class which contains ε
- F : For each state $q_j \in Q$, determine if an element of L is contained in the equivalence class for q_j.

This is a well defined DFA, which decides L. This implies that L is regular.

2.3 Usage

To use the theorem to prove a language is regular, show that $\sim_L$ exhibits a finite number of equivalence classes.

As an example, we prove $\{1^{2n} \mid n \in \mathbb{N}\}$ is regular[4]. Consider the sets $[q_0], [q_1], [q_2]$ where $[q_2]$ is all string containing at least one zero, $[q_0]$ is all even strings of ones, and $[q_1]$ is odd strings of ones. Notice that each of these sets partition $\Sigma^* = [q_0] \cup [q_1] \cup [q_2]$, and they are pairwise disjoint. For any $x, y \in [q_2]$ then $xz, yz \in [q_2]$ since they still contain a zero. For x, y both even or both odd length, and z no zeroes, then the parity of xz, yz is the same if the parity of x, y is. If z has a zero, then xz, yz again both are in $[q_2]$.

To use the theorem to prove a language is non-regular, give an infinite set S of strings, such that for each pair of strings $x, y \in S$ there is a least one string z such that $xz \in L, yz \notin L$ or vice versa. Then each string of S must belong to a separate equivalence class of $\sim_L$, so if S is infinite, there are infinitely many equivalence classes. Note that you may have a different z for each pair x, y.

As an example, we prove $\{0^n 1^n \mid n \in \mathbb{N}\}$ is not regular. Let $S = \{0^i | i \in \mathbb{N}\}$. Then, for any two elements $x, y = 0^j, 0^k$ in S with $j \neq k$, $z = 1^j$ gives $xz \in L$, but $yz \notin L$.

Note that, if you are trying to prove a language regular, there is almost always a simpler method available than the Myhill-Nerode theorem. Similarly, to prove the non-regularity of a language on exams and homeworks, we encourage you to use the pumping lemma rather than the Myhill-Nerode theorem. It is a powerful tool, but can be difficult to use correctly.

2.4 Problems

Turn in number 1 and two of the remaining problems.

1. Prove that $\sim_L$ is an equivalence relation.
2. Give an example of a non-regular language that cannot be proved to be non-regular using the pumping lemma. (Prove your work, do not just state the language.)
3. Let $L_{n,c} = \{1^k \mid k \equiv c \pmod{n}\}$. Prove this language is regular for any c, n WITHOUT constructing a DFA.
4. Prove that the DFA from the proof in 1.2 is minimal (i.e. there is no DFA for L with fewer states.)
5. Let $L = \{1^{n^2} \mid n \in \mathbb{N}\}$. Prove that L is not regular using the theorem.

Of additional interest to you may be problems 1.47,1.48[5] from the Sipser book.

[4] You should see this is regular quickly as this is just $(11)^*$, and existence of a regular expression implies it is regular.

[5] 1.34, 1.35 in the first edition