CMPSCI 250: Introduction to Computation

Lecture #21: Non-Regular Languages and the Myhill-Nerode Theorem

David Mix Barrington

18 April 2013

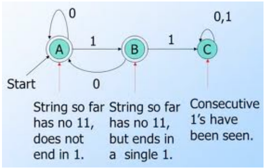
Non-Regular Languages and Myhill-Nerode

- The Strings For Each State of a DFA
- L-Distinguishable Strings
- Languages With No DFA’s
- The Relation of L-Equivalence
- More than k Classes Means More Than k States
- Constructing a DFA From the Relation
- The Minimal DFA For a Language

The Strings for Each State of a DFA

- A DFA with k states divides the strings in Σ* into k categories, based on what state it takes each string to.

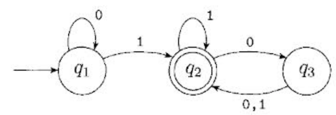
- In this example, a three-state DFA whose language is all strings that don't have a 11 substring, we can say that certain strings go to state A, others to B, and others to C. The language of the DFA is the union of the A class and the B class. If we changed the final state set, the language would be some other union of some of these three classes.



- There are limits to the kind of classes a DFA can divide Σ* into. We'll use these to prove limits on the power of DFA's to decide languages.

iClicker Question 1: Strings for a Given State

- In the pictured DFA, what is the set of strings that take the DFA to state q_1 when started in state q_1?
- (a) the language 0*
- (b) the language ∅
- (c) the language 1(1+0(0+1))*
- (d) the language (0+1)*



A Definition: L-Distinguishable Strings

- Let $L \subseteq \Sigma^*$ be any language. Two strings u and v are **L-distinguishable** (or **L-inequivalent**) if there exists a string w such that $uw \in L \oplus vw \in L$. They are **L-equivalent** if for every string w, $uw \in L \leftrightarrow vw \in L$ (we write this as $u \equiv_L v$).

- For example, let L be the language of the DFA on the previous slide, the set of strings with no "11", which is denoted by the regular expression $(0+10)^*(1+\lambda)$. The strings u = 1001 and v = 1000 are L-distinguishable, because we can take w to be the string 1. Then uw = 10011 is not in L, but vw = 10001 is.

- Any string u in L is L-distinguishable from any string v not in L, because we can always take $w = \lambda$. Then uw is in L and vw is not.

- Suppose that a DFA M takes L-distinguishable string u and v to the same state. Then it also takes uw and vw to the same state. The language L(M) *cannot* be L, because if it were that state would be both final and non-final.

iClicker Question #2: L-Distinguishable Strings

- Let L be the language $(a + ba^*ba^*b)^*$, of strings with a number of b's that is divisible by 3. Which pair of strings is not L-distinguishable?

- (a) bbab and aab

- (b) aaa and bbb

- (c) aaab and baabaab

- (d) λ and bbaaabb

Sets of Pairwise L-Distinguishable Strings

- We just saw that if a DFA takes two L-distinguishable strings to the same state, it cannot have L as its language. What if S is a set of **pairwise L-distinguishable** strings, meaning that any two distinct strings in S are L-distinguishable?

- In that case, any DFA that has L as its language must have *at least as many* states as S has strings. Why? If there were fewer states than strings, the DFA must take two or more strings to the same state by the **Pigeonhole Principle**. And it can't take two L-distinguishable strings to the same state.

- In our example, the strings 1001, 1000, and 11 are pairwise L-distinguishable for our language $(0+10)^*(1+\lambda)$. That means that no DFA with fewer than three states could possibly have L as its language. The three-state DFA we have is thus a **minimal DFA** for L.

Languages With No DFA's

- If S is an *infinite* set of pairwise L-distinguishable strings, no correct DFA for L can exist *at all*.

- The easiest language to prove unrecognizable by any DFA is EQ, defined as $\{a^nb^n: n \geq 0\}$ or $\{\lambda, ab, aabb, aaabbb,...\}$. Here our set S is $\{a^i: i \geq 0\}$ or $\{\lambda, a, aa, aaa,...\}$. If i and j are two distinct natural numbers, then the strings a^i and a^j are EQ-distinguishable because a^ib^i is in EQ and a^jb^i is not.

- For another example, consider the language Paren $\subseteq \{L, R\}^*$ which contains all strings of L's and R's that represent balanced sets of parentheses. Paren has such a set, $\{L^i: i \geq 0\}$, because if $i \neq j$ then L^iR^i is in Paren but L^jR^i is not. So any two distinct strings in the set are L-distinguishable. No DFA for Paren exists, and thus Paren is not a DFA-recognizable language.

The Language Prime Has No DFA

- Let Prime be the language $\{a^n$: n is a prime number$\}$. It doesn't seem likely that any DFA could decide Prime, but this is a little tricky to prove.

- Let i and j be two naturals with $i > j$. We'd like to show that a^i and a^j are Prime-distinguishable, by finding a string a^k such that $a^i a^k \in$ Prime and $a^j a^k \notin$ Prime. We need a natural k such that $i + k$ is prime and $j + k$ not, or vice versa.

- Pick a prime p bigger than both i and j (since there are infinitely many primes). Does $k = p - j$ work? It depends on whether $i + (p - j)$ is prime -- if it isn't we win because $j + (p - j)$ *is* prime. If it is prime, look at $k = p + i - 2j$. Now $j + k$ is the prime $p + (i - j)$, so if $i + k = p + 2(i - j)$ is not prime we win.

- We find a value of k that works unless *all* the numbers p, $p + (i - j)$, $p + 2(i - j)$,..., $p + r(i - j)$,... are prime. But $p + p(i - j)$ is not prime as it is divisible by p.

The Relation of L-Equivalence

- The relation of L-equivalence is aptly named because we can easily prove that it is an **equivalence relation** -- it is **reflexive**, **symmetric**, and **transitive**. Clearly $\forall w$: $uw \in L \leftrightarrow uw \in L$, so it is reflexive. If we have that $\forall w$: $uw \in L \leftrightarrow vw \in L$, we may conclude that $\forall w$: $vw \in L \leftrightarrow uw \in L$, and thus it is symmetric. Transitivity is equally simple to prove.

- We know that any equivalence relation **partitions** its base set into **equivalence classes**. The **Myhill-Nerode Theorem** says that for any language L, there exists a DFA for L with k or fewer states if and only if the L-equivalence relation's partition has k or fewer classes. That is, if the number of classes is a natural k then there is a **minimal DFA** with k states, and if the number of classes is infinite then there is no DFA at all.

- It's easiest to think of the theorem as "k or fewer states $\leftrightarrow$ k or fewer classes".

iClicker Question #3: Equivalence Classes

- If R is an **equivalence relation** on a set X, the **equivalence class** of an element w is the set of elements of X that are "equivalent" to w, that is, the set {z: R(z, w)}. Define a relation R on {a, b}* so that R(u, v) means "u and v both begin with the same letter and end with the same letter". What is the equivalence class of the string bbaa for this relation?

- (a) the set of all strings that begin with b and end with a, that is, $b\Sigma^*a$.

- (b) the set of all strings except bbaa itself

- (c) {bbaa}

- (d) the empty set $\emptyset$

More Than k Classes Means More Than k States

- We've essentially already proved half of this theorem. We can take "k or fewer states → k or fewer classes" and take its contrapositive, to get "more than k classes → more than k states".

- Let L be an arbitrary language and assume that the L-equivalence relation has more than k (non-empty) equivalence classes. Let $x_1,\ldots,x_{k+1}$ be one string from each of the first k + 1 classes. Since any two distinct strings in this set are in different classes, by definition they are not L-equivalent, and this means that they are L-distinguishable.

- By our result from earlier in this lecture, since there exists a set of k + 1 pairwise L-distinguishable strings, no DFA with k or fewer states can have L as its language.

- This proves the first half of the Myhill-Nerode Theorem.

Constructing a DFA From the Relation

- Now to prove the other half, “k or fewer classes → k or fewer states”. In fact we will prove that if there are exactly k classes, we can build a DFA with exactly k states. This DFA will necessarily be the smallest possible for the language, because a smaller one would contradict the half we have proved.

- Let L be an arbitrary language and assume that the classes of the relation are $C_1, \ldots, C_k$. We will build a DFA with states $q_1, \ldots, q_k$, each state corresponding to one of the classes.

- The initial state will be the state for the class containing λ. The final states will be any states that contain strings that are in L. The transition function is defined as follows. To compute $\delta(q_i, a)$, where $a \in \Sigma$, let w be any string in the class C_i and define $\delta(q_i, a)$ to be the state for the class containing the string wa.

- It’s not obvious that this δ function is **well-defined**, since its definition contains an arbitrary choice. We must show that any choice yields the same result.

Completing the Proof

- Let u and v be two strings in the class C_i. We need to show that ua and va are in the same class as each other. That is, for any u, v, and a, we must show $u \equiv_L v \rightarrow ua \equiv_L va$. Assume that $\forall w$: $uw \in L \leftrightarrow vw \in L$. Let z be an arbitrary string. Then $uaz \in L \leftrightarrow vaz \in L$, because we can specialize the statement we have to az. We have proved $\forall z$: $uaz \in L \leftrightarrow vaz \in L$ or $ua \equiv_L va$.

- Now we prove that for this new DFA and for any string w, $\delta^*(i, w) = q_j \leftrightarrow w \in C_j$. (Here "i" is the initial state of the DFA.) We prove this by induction on w. Clearly $\delta^*(i, \lambda) = i$, which matches the class of λ. Assume as IH that $\delta^*(i, w) = x$ matches the class of w. Then for any a, $\delta^*(i, wa)$ is defined as $\delta(x, a)$ which matches the class of wa by the definition, which is what we want.

- If two strings are in the same class, either both are in L or both are not in L. So L is the union of the classes corresponding to our final states. Since the DFA takes a string to the state for its class, $\delta^*(i, w) \in F \leftrightarrow w \in L$.

The Minimal DFA and Minimizing DFA's

- Let X be a regular language and M be any DFA such that L(M) = X. We will show that the minimal DFA, constructed from the classes of the L-equivalence relation, is **contained within** M.

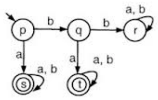
- We begin by eliminating any unreachable states of M, which does not change M's language.

- Remember that a correct DFA cannot take two L-distinguishable strings to the same state. So for any state p of M, the strings w such that $\delta(i, w) = p$ are all L-equivalent to each other. Each state of M is thus associated with one of the classes of the L-equivalence relation.

- The states of M are thus partitioned into classes themselves. If we combine each class into a single state, we get the minimal DFA. In Discussion #12 on Monday we will see, and then practice, a specific algorithm to find these classes and thus construct the minimal DFA equivalent to any given DFA.

iClicker Question #4: A Non-Minimal DFA

- Here is a five-state DFA that is *not* a minimal DFA for its language X (which is $(a + ba)\Sigma^*$). Which of these four statements about this DFA is **false**?



- (a) The two final states s and t could be merged into one without changing the language of the DFA.

- (b) The strings b and bb are X-distinguishable.

- (c) States q and r could be merged into one without changing the language of the DFA.

- (d) The set {λ, a, b, bb} is a pairwise X-distinguishable set of strings.