On Minimization of Deterministic Finite Automata
The Myhill-Nerode Theorem and Uniqueness of Minimal DFAs

Ben Gunning

February 25, 2016

Abstract

The Myhill-Nerode Theorem is an important and interesting result in the study of the class of all regular languages. In particular, this result establishes an additional necessary and sufficient condition for the classification of regular languages beyond the usual usage of finite automata and regular expressions. The result further enables proof of the existence and uniqueness of a minimal deterministic finite automaton recognizing a given regular language. This result gives rise to the task of deterministic finite automaton (DFA) minimization.

Foreword

The following report is my essay submission for Course Project 1 in Professor Chiang's Theory of Computing class at the University of Notre Dame. The topic of this paper, the Myhill-Nerode Theorem, is relevant and related to the course material in that it sets forth an additional necessary and sufficient criterion for a regular language, hence relating closely to our previous discussion of finite automata and regular expressions. As of now, this paper is the product of roughly 3 hours of research and work.

1 Indistinguishability of Strings

In order to proceed with a discussion of the Myhill-Nerode Theorem, we must first establish some foundational definitions and results.

Definition 1.1 (Indistinguishability)**.** Let L be a language such that $x, y \in L$. We say that x and y are *indistinguishable by L* if the following criterion holds:

$$\forall z \in L \quad (xz \in L \wedge yz \in L) \vee (xz \notin L \wedge yz \notin L)$$

Two strings x and y are called *distinguishable* if they are not indistinguishable. This is equivalent to the existence of a string $z \in L$ such that exactly one of xz, yz is an element of L.

Definition 1.2 (Equivalence Relation)**.** Let $\sim$ be a relation on a language L. $\sim$ is called an *equivalence relation* if each of the following criteria is satisfied:

- $\forall x \in L \quad x \sim x$
- $\forall x, y \in L \quad x \sim y \Leftrightarrow y \sim x$
- $\forall x, y, z \in L \quad x \sim y \wedge y \sim z \Rightarrow x \sim z$

We now prove a short but useful result that will prove handy in our later discussion of the Myhill-Nerode Theorem.

Lemma 1.3. *Given a language L and $\sim$ such that $x \sim y$ if and only if x and y are indistinguishable by L, the relation $\sim$ is an equivalence relation.*

Proof. The reflexivity of $\sim$ is trivial, as is symmetry. The only property that warrants further discussion is transitivity.
Let $x, y, z \in L$ such that $x \sim y$ and $y \sim z$. Fix $w \in L$. By the indistinguishability of x and y, $xw \in L \Leftrightarrow yw \in L$. Similarly, $yw \in L \Leftrightarrow zw \in L$. Therefore, $xw \in L \Leftrightarrow zw \in L$, so $x \sim z$.
The satisfaction of these properties proves that $\sim$ is an equivalence relation over L. ■

Note that because indistinguishability defines an equivalence relation on a language, then it necessarily follows (from the properties of an equivalence relation) that indistinguishability yields a set of one or more equivalence classes over L.

2 Pairwise Distinguishability and the Index

Having defined and established the notion of indistinguishability of strings, we turn our attention to subsets of languages.

Definition 2.1 (Pairwise Distinguishability)**.** Let L be a language and $X \subseteq L$ be a set of strings in L. We say that X is *pairwise distinguishable by L* if $\forall x, y \in X$ x and y are distinguishable by L.

Definition 2.2 (Index)**.** We define the *index of L* to be the maximum over the sizes (number of elements) of all sets that are pairwise distinguishable in L.

The notion of the index is closely associated with the notion of equivalence classes explained in the last section. In fact, it is intuitive to recognize that the index of a given language is the number of equivalence classes under indistinguishability by said language. A basic understanding of the pigeonhole principle tells us that we may pick one element from each equivalence class while maintaining pairwise distinguishability of the set. However, when we have exhausted all of our equivalence classes, we will be unable to pick from any of our equivalence classes without having two indistinguishable strings and violating the condition of pairwise distinguishability.
Equipped with an intuitive comprehension of pairwise distinguishability, we now introduce two lemmas: the first seeks to establish an upper bound on the index of a given set, while the second provides a sufficient condition for regularity of a language.

Lemma 2.3. *Given a regular language L recognized by a deterministic finite automaton $M = (Q, \Sigma, \delta_0, q_0, F)$, L has index $\leq |Q|$.*

Proof. We proceed with a proof by contradiction. Suppose $X = \{x_1, ..., x_{k+1}\}$ is pairwise distinguishable by L. Then it must be true that at least two of the elements of X end in some state $q \in Q$ when passed as inputs into M (by the pigeonhole principle). Without loss of generality, call these elements x_1, x_2.
x_1 and x_2 are, by definition, distinguishable. Then $\exists y \in L | x_1 y \in L \wedge x_2 y \notin L$. We assume here, of course, that our choices of x_1 and x_2 may be swapped with one another to make the previous statement hold. However, M recognizes $x_1 y$ if and only if M recognizes $x_2 y$, so M does not recognize L.
This is a contradiction. Therefore the index of L is at most $|Q|$. ■

Lemma 2.4. *Let L be a language with finite index k. Then $\exists M$ a deterministic finite automaton such that M recognizes L and M has k states.*

Proof. We shall prove this result by construction. Consider $M = (Q, \Sigma, \delta_0, q_0, F)$ constructed as follows:

1. Q is the set of equivalence classes of L under indistinguishability. Clearly, $|Q| = k$ as desired. Additionally, Q is well-defined because the equivalence classes of L are disjoint and partition L.

2. Σ is simply the alphabet corresponding to the set of all symbols appearing in any string of L.

3. $\delta_0 : Q \text{ x } \Sigma \to Q$ is defined so that $\delta_0(q, a) = [qa]$.

4. $q_0 = [\epsilon]$.

5. F is the set of all equivalence classes q of L under indistinguishability satisfying $\forall x \in q \quad x\epsilon \in L$.

M recognizes L and satisfies the property that it must possess exactly k states. This completes our proof. ■

With these two results proven, we are now ready for the Myhill-Nerode Theorem.

3 Myhill-Nerode

Theorem 3.1 (Myhill-Nerode Theorem)**.** *A language L is regular if and only if it has finite index. Furthermore, the index of L is the number of states contained in the smallest deterministic finite automaton recognizing L.*

Proof. Somewhat anticlimactically, the theorem follows immediately from the two lemmas in the previous section. Lemma 2.3 gives us the forward implication, while Lemma 2.4 gives us the reverse. Together, the two lemmas give the second result as well. ■

Corollary 3.2. *The deterministic finite automaton defined in Lemma 2.4 is the unique minimal deterministic finite automaton recognizing a language L.*

Proof. We know that M is minimal, but is it unique? Suppose that $M' = (Q', \Sigma, \delta_0', q_0', F')$ is a deterministic finite automaton that recognizes L and satisfies the condition $|Q'| = k$.
Suppose that there exists some state $q' \in Q'$ such that there does not exist $q \in Q$ such that the strings that end up in q are the same strings that end up in q'. This means that there exist distinguishable strings $x, y \in L$ that both end up in some state $q'' \in Q'$ by the pigeonhole principle. The logic applied in Lemma 2.3 gives us a contradiction, so $Q' = Q$. It immediately follows from the fact that M' recognizes L that $q_0' = q_0$, $F' = F$, and $\delta_0' = \delta_0$. ■

4 Applications

The Myhill-Nerode Theorem and the uniqueness of the minimal deterministic finite automaton recognizing a given regular language has interesting applications. Most notably,

References

[1] Nerode, Anil. "Linear Automaton Transformations." *Proceedings of the American Mathematical Society* August 1958: 541-544. Print.

[2] Sipser, Michael. *Introduction to the Theory of Computation.* Boston: Course Technology, 2006. Print.