CSE396 Notes on the Myhill-Nerode Theorem Fall 2007

Think of strings x and y as being racehorses, and strings z as being possible training programs for the horses. Let M be a training farm that does two things:

- M pins colored ribbons on the horses before they begin training, to predict which training programs the horses will best respond to—different programs will work well for different horses.
- After horse x goes through training program z, M *accepts* xz if x has become a good racehorse after the training z.

Thinking of courses not horses, M is a DFA, and the different colors are the different states M can be in upon reading a horse x. If M has k states, then there are k possible color labels.

The purpose of assigning colors to horses x and y is that if there is *some* training program z that makes xz a good racehorse but makes yz a poor one—or vice-versa—then x and y must not get the same color label. We can formalize this by writing

$$color(x) \neq color(y) \iff (\exists z \in \Sigma^*)[(xz \in L \ \&\ yz \notin L) \ \ or \ \ (xz \notin L \ \&\ yz \in L)]$$

where "L" means $L(M)$. We can write this in a more-compact form if we regard "L" as a *function* such that $L(x) = 1$ means $x \in L$ and $L(x) = 0$ means $x \notin L$. This is called the *characteristic function* of the language L. Also write $x \not\equiv_L y$ as shorthand for $color(x) \neq color(y)$. Then the above is equivalent to:

$$x \not\equiv_L y \iff (\exists z \in \Sigma^*)\ L(xz) \neq L(yz).$$

(Technically, this is the negative of the relation $x \equiv_L y \iff (\forall z \in \Sigma^*)\ L(xz) = L(yz)$. This defines an *equivalence relation*, which is important for the theory, but for now it's good enough to think in terms of horses and colors.)

Now suppose we have a set S of k horses in which every pair of horses $x, y \in S$ is such that $x \not\equiv_L y$. (The word "pair" makes $x \neq y$ understood.) Not only does this mean that the particular pair of horses x, y must have $color(x) \neq color(y)$, but—and this is important—it means that all the horses in S must be of different colors. That is, the k horses in S must have k different colors. Call such a set S a "distinctive set."

It follows that any DFA ("developer of fast animals"?) that wants to train this set S of k horses must use k different color labels. Put another way, if there is a DFA M that accepts the language L, then M must have (at least) k states. Believe it or not, we have almost completed the half of the proof of the *Myhill-Nerode Theorem* that you'll need to use in problems. John Myhill was a mathematics professor at UB(!) from 1958 when this theorem was proved to his passing in 1987, while Anil Nerode is very much alive and was my boss at Cornell. Here it is:

Myhill-Nerode Theorem, part I: Let $L \subseteq \Sigma^*$ be any language. Suppose there is an *infinite* set S of strings such that for all $x, y \in S$ $(x \neq y)$, $x \not\equiv_L y$. That is, suppose

$$(\forall x \neq y \in S)(\exists z \in \Sigma^*)\ L(xz) \neq L(yz).$$

Then L is not a regular language.

Proof: Suppose L is regular—then there is a DFA M such that $L(M) = L$. Let k be the number of states in M. Since S is infinite, we can take any subset $S' \subseteq S$ of $k + 1$ horses (i.e., strings), and it still holds that for all pairs $x, y \in S'$, $x \not\equiv_L y$. By the above reasoning, M needs at

least $k+1$ states to process all the horses in S', but this contradicts the fact that M had k states. Hence a DFA M such that $L(M) = L$ cannot exist, so L is not a regular language. [ ]

If you are comfortable with reasoning directly about infinite quantities, you could apply "the above reasoning" to say that any DFA M that accepts L must use a different color—i.e., be in a different state—for any string in S. Thus M must have inifnitely many states—but this contradicts the notion of a *finite* automaton.

Now "part II" of the Myhill-Nerode Theorem states the converse: for *every* non-regular language L, there is always an infinite set S that proves it is nonregular, i.e. such that for all distinct $x, y \in S$, $x \not\equiv_L y$. Technically this is Problem 1.35 on page 89 (the text's "X" is my "S"), and I may-or-may-not prove it in class. The point is that if you put the two parts together, the Myhill-Nerode Theorem in its entirety has a simple, crisp statement:

A language is non-regular if *and only if* if there is an infinite distinctive set for it.

The italicized words are what separate the Myhill-Nerode Theorem from the "Pumping Lemma" as used in the text. The "Pumping Lemma" only has a "part I"—it does not give an "if and only if" *characterization* of what it *means* for a language to be (non-)regular. What it means for a language to be regular is that there are no infinite distinctive sets—that the distinctions one needs to make between string prefixes x and y are finite. The "if and only if" makes the Myhill-Nerode theorem mathematically superior, IMHO. Moreover, IMHO this superiority shows up in a cleaner and simpler "script" for proving languages to be non-regular. Here are all the examples in the text, re-done via the (first part of the) Myhill-Nerode Theorem:

1.38/1.73, p80: Let $L = \{\, 0^n 1^n : n \geq 0 \,\}$. To prove that L is non-regular, take $S = \{\, 0^n : n \geq 0 \,\}$, i.e. $S = 0^*$. Clearly S is infinite (note by the way that S need not be a subset of L). We need to show that S is a distinctive set for L. To do so, we need to show

$$(\forall x \neq y \in S)(\exists z \in \Sigma^*)\ L(xz) \neq L(yz).$$

The "script" for this runs: **Let** any pair $x, y \in S$ with $x \neq y$ **be given**. By definition of S, there are numbers $m, n \geq 0$ with $m \neq n$ such that $x = 0^m$ and $y = 0^n$. **Take** $z = 1^m$. Then $L(xz) \neq L(yz)$, since $xz = 0^m 1^m \in L$ and $yz = 0^n 1^m \notin L$. **Since** $x, y \in S$ **were arbitrary**, S is an "infinite distinctive set" for L, so L is non-regular. (Here the boldface **take** goes with the $\exists$ quantifer, while all the other boldface words go with the $\forall$ quantifier.)

1.39/1.74, p81/80: Given $L = \{\, w : \#0(w) = \#1(w) \,\}$, take $S = 0^*$ again. Clearly S is infinite. Let any distinct $x, y \in S$ be given. Then there are natural numbers $m, n \geq 0$ with $m \neq n$ such that $x = 0^m$ and $y = 0^n$. Take $z = 1^m$. Then $xz = 0^m 1^m \in L$ but $yz = 1^n 0^m \notin L$, so $L(xz) \neq L(yz)$. Since $x, y \in S$ were arbitrary, S is an infinite distinctive set for L, and so L is non-regular. (Note that the argument was identical to that of 1.38—this often happens!)

1.40/1.75: $L = \{\, ww : w \in \{\, 0, 1 \,\}^* \,\}$. Take $S = \{\, 0^n 1 : n \geq 0 \,\}$, i.e. $S = 0^*1$. Clearly S is infinite. Let any distinct $x, y \in S$ be given. Then there are $m, n \geq 0$ with $m \neq n$ such that $x = 0^m 1$ and $y = 0^n 1$. Take $z = 0^m 1$. Then $xz = 0^m 10^m 1$ has the form "ww" with $w = 0^m 1$, so $xz \in L$. However, $yz = 0^m 10^n 1$ cannot similarly be broken down into two identical substrings because $m \neq n$, so $yz \notin L$. Thus $L(xz) \neq L(yz)$, and since x and y were arbitrary members of S, S is an infinite distinctive set for L. Thus L is not regular.

1.41/1.76: Try to work this out with $S = 1^*$. The key is that for any two numbers m and n, there is always a number p such that $m + p$ is a perfect square but $n + p$ is not.

1.42/1.77: Here you can have $S = 0^*$, and it helps to use the phrase "Without loss of generality, we may suppose $m < n$" before "Take $z = \ldots$.