TESTS FOR LANGUAGE REGULARITY

JOEL FRIEDMAN

Contents

Disclaimer: The material may sketchy and/or contain errors, which I will elaborate upon and/or correct in class. For those not in CPSC 421/501: use this material at your own risk...

The reference [Sip] is to the course textbook, *Introduction to the Theory of Computation* by Michael Sipser, 3rd Edition.

1. Introduction

This year in CPSC 421 we have seen a number of ways to prove that a language $L \subset \Sigma^*$ is not regular, namely:

(1) Walk-counting tests: if L regular, then

$$f(n) \stackrel{\text{def}}{=} \Big|\big\{s \in L \;\;\big|\;\; |s| = n\big\}\Big|$$

is the a *walk-counting function* as we desribed in the article "Directed Graphs and Asymptotic Tests[1];" in this article we gave a number ways to show that some functions cannot be walk-counting functions.

(2) The Pumping Lemma of Section 1.4 of [Sip].

(3) The Myhill-Nerode Theorem: if for each $s \in \Sigma^*$ we set

$$\text{A.F.}(L,s) = \text{AcceptingFutures}(L,s) \stackrel{\text{def}}{=} \{t \in \Sigma^* | st \in L\},$$

then if there are infinitely many values of $\text{A.F.}(L,s)$ as s varies over Σ^*, then L is not regular.

Research supported in part by an NSERC grant.

[1] In more detail, if L is accepted by a DFA with p states, we may associate this DFA to its underlying directed graph, which has p vertices—representing the states—and $|\Sigma|$ edges leaving each vertex—respresenting the transitions. Then $f(n)$ counts the number of walks of length n in a directed graph that begin in the vertex corresponding to the initial vertex and that end in a vertex which corresponds to an accepting state.

Furthermore, if L is regular, then all the above methods have variants that can give lower bounds on the minimum number of states in a DFA recognizing L. In fact, the Myhill-Nerode Theorem gives an algorithm for building the DFA.

Generally speaking, walk-counting tests are the simplest and most direct method to use. The Pumping Lemma is trickier to use, in that it requires a judicious choice of string to apply to the lemma and requires you to prove that this string cannot be pumped; the pedagogical advantage of the Pumping Lemma is that its proof doesn't require anything not covered in [Sip] (such as the linear algebra needed for walk-counting tests). However, neither walk-counting methods nor the Pumping Lemma always determine whether or not a language is regular. The Myhill-Nerode Theorem is the most powerful method, in the sense that it works for all languages, determining whether or not they are regular and giving an algorithm for building the DFA; walk-counting tests and the Pumping Lemma don't always determine this information; the difficulty in the Myhill-Nerode Theorem is that you have to be able to determine how many sets there are of the form A.F.(L, s) for a number of values of s.

One can combine the above methods with various properties of regular languages to get more results regarding regularity. Here are some examples.

(1) If L' is a regular language, and $L \cap L'$ is not regular, then L is not regular; this follows from the contrapositive: if L, L' are both regular, then so is $L \cap L'$ (see [Sip], Sections 1.1–1.2).
(2) A similar statement holds for $L \cup L'$, for $L \circ L'$, for L^*, and for the complement of L.
(3) Substitution: if Σ' is another alphabet, and if $f\colon \Sigma \to (\Sigma')^*$ is a function (mapping the symbols or letters of Σ to strings over Σ), then f determines a map $\Sigma^* \to (\Sigma')^*$. If

$$f(L) = \{f(s) \mid s \in L\}$$

is not regular, then L is not regular. This follows from the contrapositive: if L is regular, then L is described by a regular expression, and f maps this regular expression to a regular expression in the symbols in Σ'; hence $f(L)$ is regular.

Since the Pumping Lemma is covered extensively in [Sip], Section 1.4, we now make some remarks regarding the other two methods, namely walk-counting methods and the Myhill-Nerode Theorem. The Myhill-Nerode Theorem is briefly covered in [Sip], Exercises 1.51 and 1.52.

2. Walk-Counting Functions

Recall that a walk-counting function is a function $f\colon \mathbb{N} \to \mathbb{Z}_{\geq 0}$ where $f(n)$ is the number of walks of length n in a fixed (finite) directed graph, $G = (V, E, t, h)$, that begin in a fixed subset of vertices $V_1 \subset V$ and end in a fixed subset of vertices $V_2 \subset V$. Here are some things that hold for walk-counting functions in such a graph, G:

(1) if G has at most p vertices, then f satisfies the recurrence relation

$$f(n) = c_1 f(n-1) + \cdots + c_p f(n-p)$$

for all $n \geq p+1$, for some $c_i \in \mathbb{Z}$ (such c_i can be found from the *characterisitic polynomial* of the *adjacency matrix* of G; this recurrence also holds for $n = p$ since it makes sense to speak of $f(0)$).

(2) if f has asymptotic ratio ρ, i.e.,

$$\lim_{n\to\infty} \frac{f(n+1)}{f(n)}$$

exists and equals ρ, then

(a) if ρ is a rational number, then ρ must be an integer (more generally, ρ must be an algebraic integer[2].

(b) $f(n) \sim c\rho^n n^q$ for some $c > 0$ and some positive integer $q \leq p-1$. From part (b) it follows that $f(n)$ cannot be $o(\rho^n)$, and cannot be $\Theta(\rho^n n^\alpha)$ for some α that is not an integer.

There are many variants of the above tests. For example, if $g(n) = f(nd)$ for some integer $d \geq 2$ has an asymptotic ratio, then g must satisfy the above conditions on f (since $g(n)$ is a walk counting function on the directed graph whose vertex set is V and whose edges represent walks of length d in the G). For example, if

$$L = \big\{ s \in \{0,1\}^* \;\big|\; s \text{ has the same number of 0's and 1's} \big\},$$

then using $f(n)$ to denote the number of words of length n in L, we have that (1) $f(n) = 0$ if n is odd, and (2) otherwise, $f(n) = \binom{n}{n/2}$. It follows that the asymptotic ratio of f does not exist, but that of $g(n) = f(2n)$ exists and equals 4. Furthermore Stirling's formula shows that

$$g(n) = \binom{2n}{n} = O\Big(2^{2n}/\sqrt{n}\Big) = o(4^n),$$

and hence L is not regular.

3. The Myhill-Nerode Theorem

If L is regular, then the Myhill-Nerode Theorem gives an algorithm for constructing the DFA for L with the smallest number of states: namely, the initial state of the DFA, q_0, is labelled with the set $\text{A.F.}(L, \epsilon)$. To see what are the transitions out of q_0, we compute $\text{A.F.}(L, \sigma)$ for all $\sigma \in \Sigma$, and for each new value of $\text{A.F.}(L, \sigma)$ we construct a new state. More generally, once we construct a new state $\text{A.F.}(L, t)$ for some $t \in \Sigma^*$, the transition out of this state upon reading $\sigma \in \Sigma$ is the state $\text{A.F.}(L, t\sigma)$ for $\sigma \in \Sigma$. If L is regular, then this procedure will eventually lead to a situation where for every state labelled $\text{A.F.}(L, t)$, the values of $\text{A.F.}(L, t\sigma)$ already appear in some previously constructed state.

The difficulty with the above procedure is that we are required to "compute" $\text{A.F.}(L, s)$ for a number of $s \in \Sigma^*$, which really means that for a finite number of pairs, (s, s'), we must be able to determine whether or not $\text{A.F.}(L, s) = \text{A.F.}(L, s')$. This may not be easy to do in practice. Furthermore, if L is not regular, then proving this by the Myhill-Nerode Theorem requires us to demonstrate an infinite number of elements, $s \in \Sigma^*$, such that $\text{A.F.}(L, s)$ are all distinct. Again, depending on how L is specified, this may not be easy to do in practice.

[2] i.e., ρ is the root of an equation $x^n + a_1 x^{n-1} + a_n = 0$ where $a_1, \ldots, a_n \in \mathbb{Z}$.

4. Examples

(1) Let L be the langauge of binary strings that represent prime numbers written in base 2. The Prime Number Theorem implies that the number of elements in L of length n is $\sim 2^n/(n\log_e(2))$ which has asymptotic ratio 2 but is $o(2^n)$. Hence n is not regular.
(2) One can construct examples where any of the three tests produces stronger or more easily obtained results than the other two.
(3) For any integer $p \geq 1$ and the language L described by the regular expression $(1^p)^*$, all three tests easily prove that the minimal number of states in a DFA recognizing L is p.

Department of Computer Science, University of British Columbia, Vancouver, BC V6T 1Z4, CANADA, and Department of Mathematics, University of British Columbia, Vancouver, BC V6T 1Z2, CANADA.

E-mail address: jf@cs.ubc.ca or jf@math.ubc.ca

URL: http://www.math.ubc.ca/~jf