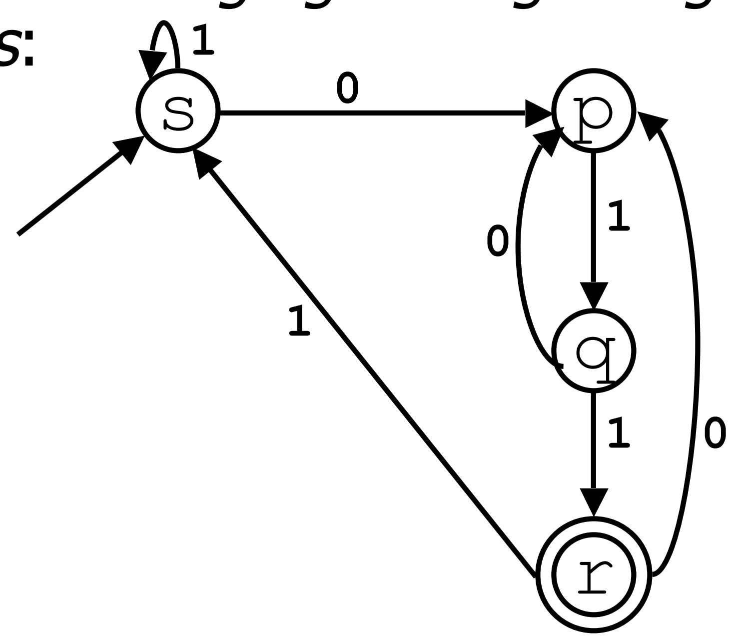
Pumping Lemma & Distinguishability

Jim Hook
Tim Sheard
Portland State University

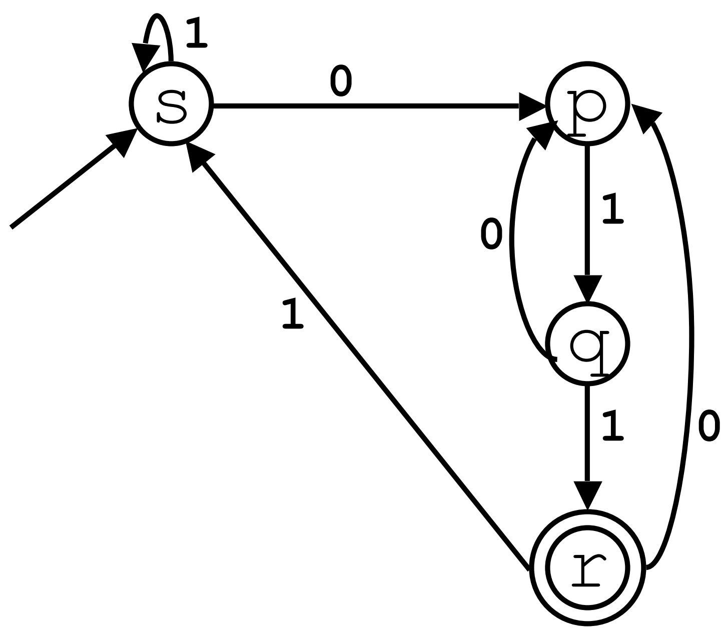
Importance of loops

Consider this DFA. The input string `01011` gets accepted after an execution that goes through the state sequence s → p → q → p → q → r. This path contains a loop (corresponding to the substring `01`) that starts and ends at p. There are two simple ways of modifying this path *without changing its beginning and ending states*:



(1) delete the loop from the path;

(2) instead of going around the loop once, do it several times. As a consequence, we see that all strings of the form `0(10)`^{i}`11` (where i ≥ 0) are accepted.



Long paths must contain a loop

Suppose *n* is the number of states of a DFA. Then every path of length *n* or more visits at least *n+1* states, and therefore must visit some state twice. Thus, *every path of length n or longer must contain a loop*.

The pumping lemma

Suppose L is a regular language, w is a string in L, and u is a non-empty substring of w. Thus, $w=xuy$, for some strings x, y . We say that u is a *pump* in w if all strings xu^iy (that is, xy, xuy, $xuuy$, $xuuuy$, …) belong to L.

Pumping Lemma. Let L be a regular language. Then there exists a number n, such that for all $w \in L$ such that $|w| \geq$ n, there exists a prefix of w whose length is less than n which contains a pump. Formally: If $w \in L$ and $|w| \geq n$ then $w = xyz$ such that

1. $y \neq \varepsilon$ (y is the pump)
2. $|xy| \leq n$ (xy is the prefix)
3. $xy^iz \in L$

Definition. The number n associated to the regular language L as described in the Pumping Lemma is called the *pumping constant* of L.

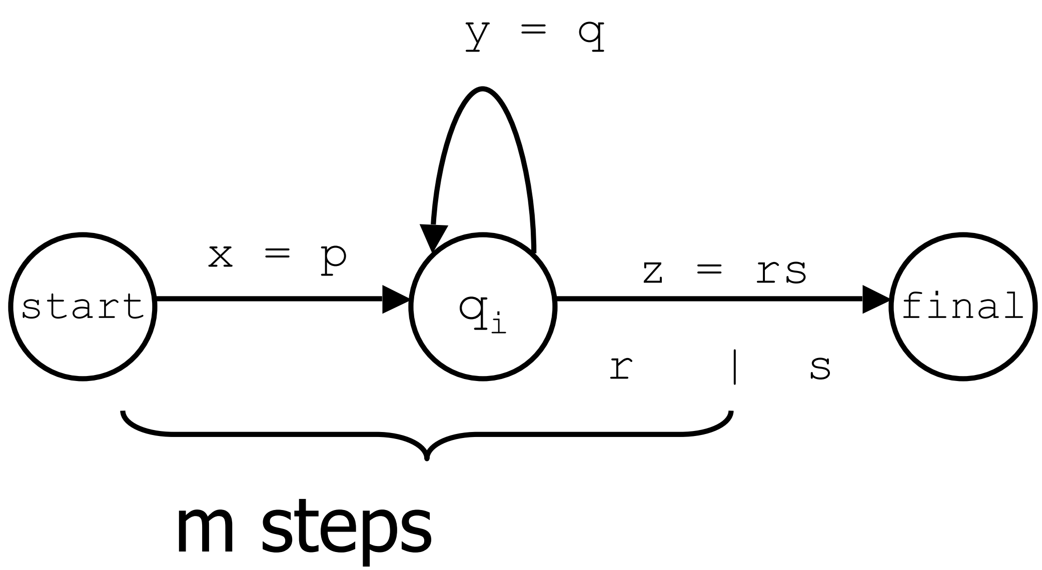
Proof

$w \in L$, $|w| \geq n$, $w = xyz$ such that 1. $y \neq \varepsilon$ 2. $|xy| \leq n$ 3. $xy^iz \in L$

Let the DFA have m states. Let $|w| \geq m$. Consider the path from the start state s to the (accepting) state $\underline{\delta}(s,w)$. Just following the first m arcs, we make $m+1$ total visits to states, so there must be a loop formed by some of these arcs.

We can write $w=opqr$, where p corresponds to that loop, and $|opq| = m$ (the prefix of size m). Thus let n=|op|, x=o, y = p, and z = qr.

1) Since every loop has at least one arc, we know |p| >0, thus $y \neq \varepsilon$
2) $|xy| \leq n$ because $xy = op$ and $n = |op|$
3) $xy^iz \in L$ because If p is a loop, its starts at state s_i and $\underline{\delta}(s_i,p) = s_i$, and we know that $\underline{\delta}(s_i,qr) = s_{final.}$. Thus $\underline{\delta}(s_{start},x) = s_{i,}$ Thus for each i $\underline{\delta}(s_i,y^i) = s_{i,}$ and were done.


y = q
x = p
start
q_i
z = rs
final
r | s
m steps

Proving non-regularity

To prove that a given language is not regular, we use the Pumping Lemma as follows.

Assuming L is regular (we are arguing by contradiction!), let n be the pumping constant of L. Making no other assumptions about n (we don't know what it is exactly), we need to produce a string $w \in L$ of length $\geq$ n that does not contain a pump in its n-prefix. This w depends on n; we need to give w for any value of n.

There are many substrings of the n-prefix of our chosen w and we must demonstrate that *none of them is a pump*. Typically, we do this by writing w=xuy, a decomposition of w into three substrings about which we can only assume that $u \neq \varepsilon$ and $|xu| \leq n$. Then we must show that *for some concrete i* (zero or greater) the string xu^iy does not belong to L.

Skill required

Notice the game-like structure of the proof. Somebody gives us n. Then we give w of length ≥ n. Then our opponent gives us a non-empty substring u of the n-prefix of w (and with it the factorization w =xuy of w). Finally, we choose i such that $xu^iy \notin L$.

Our first move often requires ingenuity: We must find w so that we can successfully respond to whatever our opponent plays next.

Example 1

We show that $L=\{0^k1^k \mid k=0,1,2, \ldots\}$ is not regular. Assuming the Pumping Lemma constant of L is n, we take $w=0^n1^n$. We need to show that there are no pumps in the n-prefix of w, which is 0^n. If u is a pump contained in 0^n then $0^n = xuz$, and xuuz must also be in the language. But since $|u| > 0$, if $|xuz| = n$ then $|xuuz| = m$ where $m > n$. So we obtain a string 0^m1^n with $m>n$, which is obviously not in L, so a contradiction is obtained, and are assumption that 0^K1^K is regular must be false.

Note. The same choice of w and i works to show that the language:

$L=\{w \in \{0,1\}^* \mid w \text{ contains equal number of 0s and 1s}\}$

is not regular either.

Example 2

We show that $L = \{ uu \mid u \in \{a,b\}^* \}$ is not regular. Let n be the pumping constant. Then we choose $w=a^nba^nb$ which clearly has length greater than n.

The initial string a^n must contain the pump, u. So $w = xuyba^nb$, and $xuyb = a^nb$. But pumping u 0 times it must be the case that $xyba^nb$ is in L too. But since u is not ε, we see that $xyb \neq a^nb$, since it must have fewer a's. Which leads to a contradiction. Thus our original assumption that L was regular must be false.

Question. If in response to the given n we play $w=a^na^n$, the opponent has a chance to win. How?

Example 3

The language $L = \{ w \in \{a,b,c\}^* \mid$ the length of w is a perfect square$\}$ is not regular.

In response to n, we play any string w of length n^2 (which clearly has length greater than n). The opponent picks a pump u such that $w = xuy$; let $k=|u|$ and we have

$$|xu^iy| = |xuy| + (i-1)\,|u| = n^2 + (i-1)k.$$

If we can find i such that $n^2+(i-1)k$ is not a perfect square, then we are led to a contradiction. A good choice is $i=kn^2+1$. In that case

$n^2+(i-1)k =$

$n^2+(kn^2+1 - 1)k =$

$n^2+k^2n^2 =$

$n^2(k^2+1)$, which is not a perfect square.

Distinguishability

Myhill Nerode

The Myhill Nerode theorem is another characterization of the regular languages

It uses a language to carve up the set of all strings into equivalence classes

Intuitively these equivalence classes will correspond to states in a minimal DFA

Definition

x and y are *distinguishable* with respect to L if there is a z such that either xz is in L and yz is not in L or xz is not in L and yz is in L

in other words

x and y are *indistinguishable* wrt L if for all z, xz in L iff yz in L

Example

A = {a,b}

a and b are indistinguishable

epsilon is distinguishable from all other strings

all strings other than epsilon, a and b are indistinguishable

In other words, there are three equivalence classes for A: [epsilon], [a], [aa].

The number of equivalence classes induced by a language is called the *index* of the language (A is of index 3)

Homework

In homework you will show:

- indistinguishable by L is an equivalence relation
- if L is recognized by a DFA with k states then L has index at most k
- If L has finite index k, then it is recognized by a DFA with k states
- L is regular iff it has finite index. The index is the size of the smallest DFA recognizing L

Examples

What is the index of $a^n b^n$?

What are the equivalence classes of a*b*?

What is the index of a*b*?