Nonregular Languages – the Pumping Lemma

Consider

$$\begin{aligned} C &= \{a^n b^n \mid n \geq 0\} \\ &= \{\epsilon, ab, aabb, aaabbb, aaaaabbbbb\} \end{aligned}$$

Intuitively: must remember *how far* the center point is from the start to accept the string:

$aaaaaaaaaaaaaaaaaaabbbbbbbbbbbbbbbbbbb$

But FAs has only finite memory, and the center can be arbitrarily far from the start.

Pigeonhole Principle

To prove formally that there is no DFA that accepts we need:

The Pigeonhole Principle: If A and B are finite sets and $|A| > |B|$ then there is no 1-1 function from A to B, i.e., if we assign each element of A (the "pigeons") to an element of B (the "pigeonholes") eventually we must put more than one pigeon in the same hole.

Proof that C is not regular

The proof is by contradiction. Suppose C is regular. Then there is a DFA M such that $C = L(M)$.

- Let $s =$ number of states in M.
- Given $a^n b^n$ for $n > s$, M must be in some state p more than once while the a's are scanned, by the pigeonhole principle.
- Partition $a^n b^n$ into x, y, and z, where y is the string of a's scanned between two times state p is entered. Let $i = |y|$.

Observe: We can leave out y or repeat y any number of times and end up in the same state. But then for any $k \geq 0$, $a^{n+(k-1)i} b^n \in L(M)$! E.g., $a^{n-i} b^n \in L(M)$.

The Pumping Lemma

Theorem: Let A be a regular language. Then there is a number p (the "pumping length" of A) such that for every string w in A such that $|w| \geq p$, we can break w into three strings, $w = xyz$, such that:

1. $|xy| \leq p$
2. $y \neq \epsilon$
3. $xy^k z \in A$ for each $k \geq 0$.

Proof of the Pumping Lemma

- Let p be the number of states in the finite automaton $M = (Q, \Sigma, \delta, q_0, F)$ which accepts A. Let $w = w_1 w_2 ... w_n$ be a string of length $n \geq p$. Let $r_1 ... r_{n+1}$ be the sequence of states M enters into while processing w.

- By the Pigeonhole Principle, two of the states among the first $p + 1$ states are the same. Call the first r_j and the second r_l.

- – Let $x = w_1 ... w_{j-1}$, $y = w_j ... w_{l-1}$, $z = w_l ... w_n$.
  – We can easily verify each of the conditions of the lemma.

Proving a language L is not regular

The Pumping Lemma gives a condition that must be satisfied by every regular language. How can we use it to show a language is *not* regular?

Contrapositive: L is *not* regular if for every $n \geq 0$, there exists a string $w \in L$, $|w| \geq n$, such that for $every$ decomposition of w into xyz with $|xy| \leq n$, there is some $k \geq 0$ such that $xy^kz \notin L$.

Example 1

$A = \{a^r b^s \mid r \geq s\}$

We are given $n \geq 0$.

We pick $w = a^n b^n$.

Now we are given a decomposition xyz of w with the following properties: for $|xy| \leq n$ and $y \neq \epsilon$.

- Since $|xy| \leq n$, it *must* be the case that $xy = a^j$ for some $j \geq 0$. Since $y \neq \epsilon$, it *must* be the case that $y = a^i$ with $i > 0$.

We pick $k = 0$. The string $xy^0z = a^{n-i}b^n \notin L$ since there are more b's than a's. (Pumping down)

Using closure properties

Theorem: The class of languages accepted by finite automata is closed under

1. union;
2. concatenation;
3. star;
4. complementation;
5. intersection.
6. reversal

Example 2

$L = \{w \in \{a,b\}^* \mid w$ has an equal number of $a's$ and $b's\}$

If L is regular then $L \cap L(a^*b^*)$ is regular, since the regular languages are closed under intersection. But $L \cap L(a^*b^*) = \{a^nb^n \mid n \geq 0\}$. which we already showed is not regular, giving a contradiction.

Or using the pumping lemma

How to pick a string:

Example 3

$L = \{ww \mid w \in \{0,1\}^*\}$

Example 4

$L = \{010^n1^n \mid n \geq 0\}$

More than one case for the decomposition.