Last time:
- NFAs to regular expressions
- Pumping lemma

Today:
- Proving a language is not regular
- Pushdown automata (PDAs)

On Friday:  Homework 2 due
Homework 3 out

Sofya Raskhodnikova
THE PUMPING LEMMA

Let \( L \) be a regular language with \( |L| = \infty \)

Then **there exists a length** \( p \) such that

**if** \( w \in L \) and \( |w| \geq p \) **then**

\( w \) can be split into three parts \( w=xyz \) where:

1. \( |y| > 0 \)
2. \( |xy| \leq p \)
3. \( xy^iz \in L \) for all \( i \geq 0 \)
Proof by **contradiction**: assume $L$ is regular.

Then there is a pumping length $p$.

Find a string $w \in L$ with $|w| \geq p$.

Show that no matter how you choose $xyz$, $w$ cannot be pumped!

Conclude that $L$ is not regular.
Pumping lemma as a game

1. **YOU** pick the language $L$ to be proved nonregular.

2. **ADVERSARY** picks $p$, but doesn't reveal to **YOU** what $p$ is; **YOU** must devise a play for all possible $p$'s.

3. **YOU** pick $w \in L$, which should depend on $p$ and which must be of length at least $p$.

4. **ADVERSARY** divides $w$ into $x, y, z$, obeying PL conditions: $y$ is not empty and falls within the first $p$ characters of $w$. Again, **ADVERSARY** does not tell **YOU** what $x, y, z$ are.

5. **YOU** win by picking $i$, which may be a function of $p, x, y, z$, such that $xy^i z$ is not in $L$. 

9/19/2017
Exercise

Prove $C = \{ 0^i1^j \mid i > j \geq 0 \}$ is not regular.

Proof: Assume ... pumping length $p$

What string can you choose in your next move?

A. 00011
B. $0^p1^p$
C. $0^{p/2}1^{p/2-1}$
D. $0^{p+1}1^p$
E. $0^{p+2}1^p$
F. More than one choice above works.
Prove $C = \{ 0^i1^j \mid i > j \geq 0 \}$ is not regular.

Proof: Assume ... pumping length $p$
Find a $w \in C$ of length at least $p$

$0^{p+1}1^p$

Show that $w$ cannot be pumped:

$w = 00...0011...11$

$y$ must be in this part

$x y y z = 00...00011...11$

$> p+1$

$x z = 0...0011...11$

$\leq p$
Prove $C = \{ 0^i1^j \mid i > j \geq 0 \}$ is not regular.

Proof: Assume … pumping length $p$

Find a $w \in C$ of length at least $p$

$0^{P+1}1^P$

If $w = xyz$ with $|xy| \geq p$ then $y = 0^J$ for some $J \geq 1$.

Then $xy^0z = xz = 0^{P+1-J}1^P \notin C$

Contradiction!
Let \( \text{BALANCED} = \{ w \mid w \text{ has an equal } \# \text{ of } 1\text{s and } 0\text{s} \} \)

Assume … there is a \( p \)

Find a \( w \in \text{BALANCED} \) of length at least \( p \)

\( (01)^p \) \( \not\in \text{BALANCED} \)

Show that \( w \) cannot be pumped:

If \( w = xyz \) with \( |xy| \leq p \) then
\( y = 0^j \) for some \( j>0 \).

Then \( xyyz = 0^{p+j}1^p \not\in \text{BALANCED} \)
Pumping a language can be lots of work…
Let’s try to reuse that work!

\[ \{0^n1^n \mid n \geq 0\} = \text{BALANCED} \cap 0^*1^* \]

If BALANCED is regular then so is \( \{0^n1^n \mid n \geq 0\} \)
Prove: A is not regular

If A is regular, then $A \cap C (= B)$ is regular.

But B is not regular so neither is A.
Prove $A = \{0^i1^j \mid i \neq j\}$ is not regular using $B = \{0^n1^n \mid n \geq 0\}$

$\neg A = B \cup \{ \text{strings that mix 0s and 1s} \}$

$\neg A \cap 0^*1^* = B$
MODEL OF A PROBLEM: LANGUAGE

MODEL OF A PROGRAM: DFA

EQUIVALENT MODELS: NFA, REGEXP

PROBLEMS THAT A DFA CAN’T SOLVE

ARE WE DONE?
NONE OF THESE ARE REGULAR

- $\Sigma = \{0, 1\}$, $L = \{0^n1^n \mid n \geq 0\}$
- $\Sigma = \{a, b, c, \ldots, z\}$, $L = \{w \mid w = w^R\}$
- $\Sigma = \{(, )\}$, $L = \{\text{balanced strings of parens}\}$

We can write a C or JAVA program for any of them!
PUSHDOWN AUTOMATA (PDA)

FINITE STATE CONTROL

STACK (Last in, first out)

INPUT
PDA to recognize \( L = \{ 0^n1^n \mid n \geq 0 \} \)
A **PDA** is a 6-tuple \( P = (Q, \Sigma, \Gamma, \delta, q_0, F) \)

- \( Q \) is a finite set of states
- \( \Sigma \) is the alphabet
- \( \Gamma \) is the stack alphabet
- \( \delta : Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon \rightarrow P(Q \times \Gamma_\varepsilon) \) is the transition function
- \( q_0 \in Q \) is the start state
- \( F \subseteq Q \) is the set of accept states

\( P(Q \times \Gamma_\varepsilon) \) is the set of subsets of \( Q \times \Gamma_\varepsilon \) and \( \Sigma_\varepsilon = \Sigma \cup \{\varepsilon\} \)

**Note:** A PDA is defined to be nondeterministic.
Formal Definition

A PDA is a 6-tuple $P = (Q, \Sigma, \Gamma, \delta, q_0, F)$

- $Q$ is a finite set of states
- $\Sigma$ is the alphabet
- $\Gamma$ is the stack alphabet
- $\delta : Q \times \Sigma \varepsilon \times \Gamma \varepsilon \rightarrow P(Q \times \Gamma \varepsilon)$ is the transition function
- $q_0 \in Q$ is the start state
- $F \subseteq Q$ is the set of accept states

A PDA starts with an empty stack.
It accepts a string if at least one of its computational branches reads all the input and gets into an accept state at the end of it.
An example PDA

\[ Q = \{ q_0, q_1, q_2, q_3 \} \quad \Sigma = \{ 0, 1 \} \quad \Gamma = \{ \$, 0 \} \]

\[ \delta : Q \times \Sigma \times \Gamma \rightarrow P(Q \times \Gamma) \]

\[ \delta(q_1, 1, 0) = \{ (q_2, \varepsilon) \} \quad \delta(q_2, 1, 1) = \emptyset \]
1. Place the marker symbol \( \$ \) onto the stack.

2. Keep reading a 0 and pushing it onto the stack. If you read a 1, pop a 0 and go to the next step.

3. Nondeterministically keep reading a 1 and popping a 0 or go to the next step.

4. If the top of the stack is \( \$ \), enter the accept state. (Then PDA accepts if the input has been read).
Exercise

What strings are accepted by this PDA?

\[ \Sigma = \{a, b, c, \ldots, z\} \]

A. Only \( \varepsilon \)
B. Palindromes
C. Even-length palindromes
D. All strings that start and end with the same letter
E. None of the above
1. Place the marker symbol $ onto the stack.
2. Nondeterministically keep reading a character and pushing it onto the stack or go to the next step.
3. Nondeterministically keep reading and popping a matching character or go to the next step.
4. If the top of the stack is $, enter the accept state.
Build a PDA to recognize
\[ L = \{ a^i b^j c^k \mid i, j, k \geq 0 \text{ and } (i = j \text{ or } i = k) \} \]