Outline
- Non-regular languages: Pumping Lemma
- Pushdown Automata
- Context-Free Grammars and Languages

Limits on the power of FA
- Is every language describable by a sufficiently complex regular expression?
- If someone asks you to design a FA for a language that seems hard, how do you know when to give up?

Is this language regular?
\{w : w has an equal # of "01" and "10" substrings\}

Limits on the power of FA
- Intuition:
  - FA can only remember finite amount of information. They cannot count
  - languages that "entail counting" should be non-regular...
- Intuition not enough:
  \{w : w has an equal # of "01" and "10" substrings\} = 0\Sigma^*0 \cup 1\Sigma^*1
  but \{w : w has an equal # of "0" and "1" substrings\} is not regular!

Non-regular languages
- **Pumping Lemma**: Let L be a regular language. There exists an integer p ("pumping length") for which every \(w \in L\) with \(|w| \geq p\) can be written as
  \(w = xyz\) such that
  1. for every \(i \geq 0\), \(xy^iz \in L\), and
  2. \(|y| > 0\), and
  3. \(|xy| \leq p\).
Non-regular languages

- Using the Pumping Lemma to prove L is not regular:
  - assume L is regular
  - then there exists a pumping length p
  - select a string \( w \in L \) of length at least p
  - argue that for every way of writing \( w = xyz \) that satisfies (2) and (3) of the Lemma,
    pumping on y yields a string not in L.
  - contradiction.

Pumping Lemma Examples

- Theorem: \( L = \{0^n1^n : n \geq 0\} \) is not regular.
  - Proof:
    - let p be the pumping length for L
    - choose \( w = 0^p1^p \)
      \[ w = \underbrace{00000000\ldots0}^p_0 \underbrace{11111111\ldots1}^p_1 \]
    - \( w = xyz \), with \(|y| > 0 \) and \(|xy| \leq p\).

Pumping Lemma Examples

- 3 possibilities:
  \[ w = \underbrace{00000000\ldots0}^p_0 \underbrace{11111111\ldots1}^p_1 \]
  \[ w = \underbrace{00000000\ldots0}^p_0 \underbrace{11111111\ldots1}^p_1 \]
  \[ w = \underbrace{00000000\ldots0}^p_0 \underbrace{11111111\ldots1}^p_1 \]
  - in each case, pumping on y gives a string not in language L.

Pumping Lemma Examples

- Theorem: \( L = \{w : w \text{ has an equal # of 0s and 1s}\} \) is not regular.
  - Proof:
    - let p be the pumping length for L
    - choose \( w = 0^p1^p \)
      \[ w = \underbrace{00000000\ldots0}^p_0 \underbrace{11111111\ldots1}^p_1 \]
    - \( w = xyz \), with \(|y| > 0 \) and \(|xy| \leq p\).

Pumping Lemma Examples

- 3 possibilities:
  \[ w = \underbrace{00000000\ldots0}^p_0 \underbrace{11111111\ldots1}^p_1 \]
  \[ w = \underbrace{00000000\ldots0}^p_0 \underbrace{11111111\ldots1}^p_1 \]
  \[ w = \underbrace{00000000\ldots0}^p_0 \underbrace{11111111\ldots1}^p_1 \]
  - first 2 cases, pumping on y gives a string not in language L; 3rd case a problem!

Pumping Lemma Examples

- recall condition 3: \(|xy| \leq p\)
  - since \( w = 0^p1^p \) we know more about how it can be divided, and this case cannot arise:
    \[ w = \underbrace{00000000\ldots0}^p_0 \underbrace{11111111\ldots1}^p_1 \]
  - so we do get a contradiction.
  - conclude that L is not regular.
Pumping Lemma Examples

• Theorem: \( L = \{0^i1^j : i > j \} \) is not regular.
• Proof:
  – let \( p \) be the pumping length for \( L \)
  – choose \( w = 0^{p+1}1^p \)
  \( w = 00000000...01111111...1 \)
  \( y \)
  \( x \)
  \( z \)
  \( M \)
  \( p+1 \)
  \( p \)

  – \( w = xyz \), with \( |y| > 0 \) and \( |xy| \leq p \).

Proof of the Pumping Lemma

– Let \( M \) be a FA that recognizes \( L \).
– Set \( p = \) number of states of \( M \).
– Consider \( w \in L \) with \( |w| \geq p \). On input \( w \), \( M \) must go through at least \( p+1 \) states. There must be a repeated state (among first \( p+1 \)).

FA Summary

• The languages recognized by FA are the regular languages.
• The regular languages are closed under union, concatenation, and star.
• Nondeterministic Finite Automata may have several choices at each step.
• NFAs recognize exactly the same languages that FAs do.
A more powerful machine

- Limitation of FA related to fact that they can only "remember" a bounded amount of information

- What is the simplest alteration that adds unbounded "memory" to our machine?

- Should be able to recognize, e.g., \( \{0^n1^n: n \geq 0\} \)