Limits on the power of FA

How do you prove that there is no Finite Automaton recognizing a given language?

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 ∈ 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.

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 ∈ 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
      \[
      w = \underbrace{000000000...0111111111...1}_p
      \]
    – w = xyz, with |y| > 0 and |xy| \leq p.

Outline

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

** first problem set out today
Pumping Lemma Examples

- 3 possibilities:
  \[ w = \underbrace{00000000\ldots01111111111\ldots1}_{x\ y\ z} \]
  \[ w = \underbrace{00000000\ldots01111111111\ldots1}_{x\ y\ z} \]
  \[ w = \underbrace{00000000\ldots01111111111\ldots1}_{x\ y\ z} \]
  - in each case, pumping on \( y \) gives a string not in language \( L \).

Pumping Lemma Examples

- Theorem: \( L = \{w: \text{w has an equal # of 0s and 1s}\} \) is not regular.
- Proof:
  - let \( p \) be the pumping length for \( L \)
  - choose \( w = 0^p1^p \)
  \[ w = \underbrace{00000000\ldots01111111111\ldots1}_{x\ y\ z} \]
  - \( w = xyz \), with \( |y| > 0 \) and \( |xy| \leq p \).

Pumping Lemma Examples

- 3 possibilities:
  \[ w = \underbrace{00000000\ldots01111111111\ldots1}_{x\ y\ z} \]
  \[ w = \underbrace{00000000\ldots01111111111\ldots1}_{x\ y\ z} \]
  \[ w = \underbrace{00000000\ldots01111111111\ldots1}_{x\ y\ z} \]
  - 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:
  \[ w = \underbrace{00000000\ldots01111111111\ldots1}_{x\ y\ z} \]
  - 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 \( w = 0^p1^p \)
  \[ w = \underbrace{00000000\ldots01111111111\ldots1}_{x\ y\ z} \]
  - \( w = xyz \), with \( |y| > 0 \) and \( |xy| \leq p \).

- 1 possibility:
  \[ w = \underbrace{00000000\ldots01111111111\ldots1}_{x\ y\ z} \]
  - pumping on \( y \) gives strings in the language (?)
  - this seems like a problem...
  - Lemma states that for every \( i \geq 0, xy^iz \in L \)
  - \( xy^iz \) not in \( L \). So \( L \) not regular.
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

• A “problem” is a language
• A “computation” receives an input and either accepts, rejects, or loops forever.
• A “computation” recognizes a language (it may also decide the language).
• Finite Automata perform simple computations that read the input from left to right and employ a finite memory.

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.

FA Summary

• Regular expressions are languages built up from the operations union, concatenation, and star.
• Regular expressions describe exactly the same languages that FAs (and NFAs) recognize.
• Some languages are not regular. This can be proved using the Pumping Lemma.

Machine view of FA

input tape

0110011100101

\( q_0 \)
finite control

Machine view of FA

input tape

0110011100101

\( q_3 \)
finite control
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\}