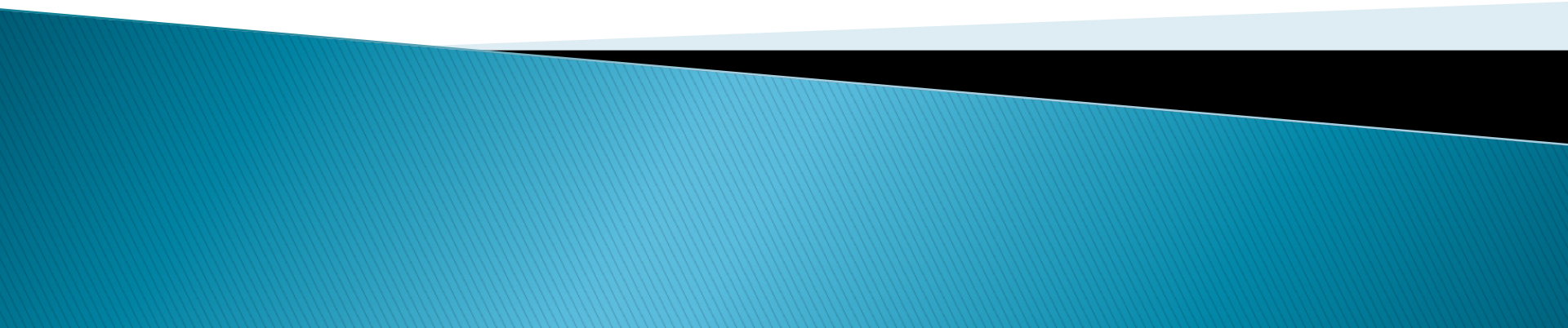
Non-Regular Languages: the Pumping Lemma



Non-Regular Languages

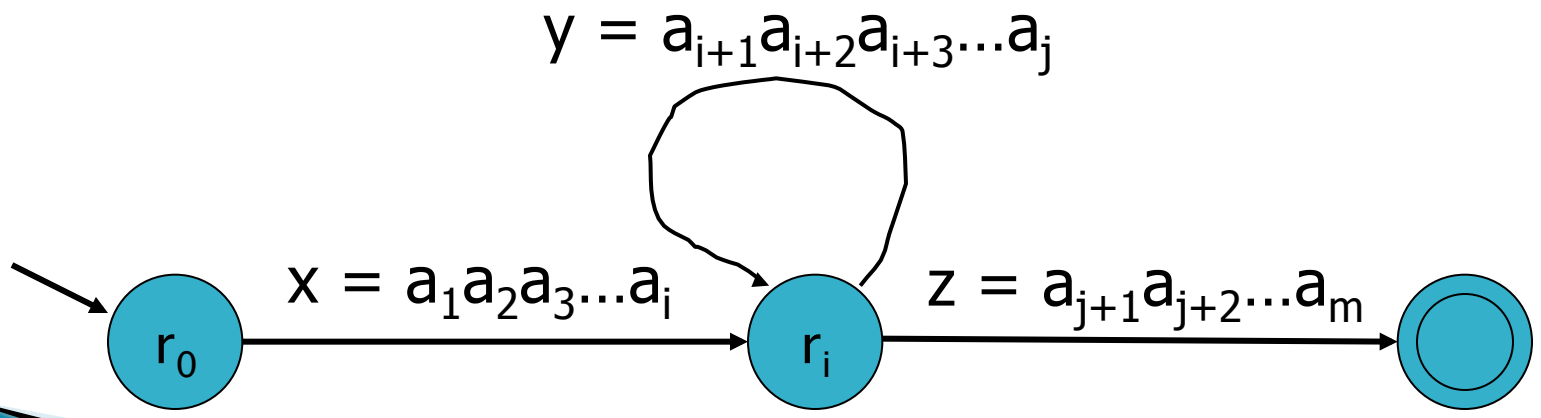
- Recall previously:
  - Used Myhill-Nerode Theorem as a tool to show that some languages are not regular
    - (Prove that there is an infinite set of pairwise distinguishable strings)
- We will discuss one additional method for showing that a language is not regular

The Pumping Lemma

- The pumping lemma formalizes the idea that if a string from a regular language is long enough, eventually at least one state from its DFA will have to be repeated on the path that accepts the string.
  - Implies that there is a Kleene star in there somewhere!
- Continually looping (pumping) on this state will produce an infinite number of strings in the language

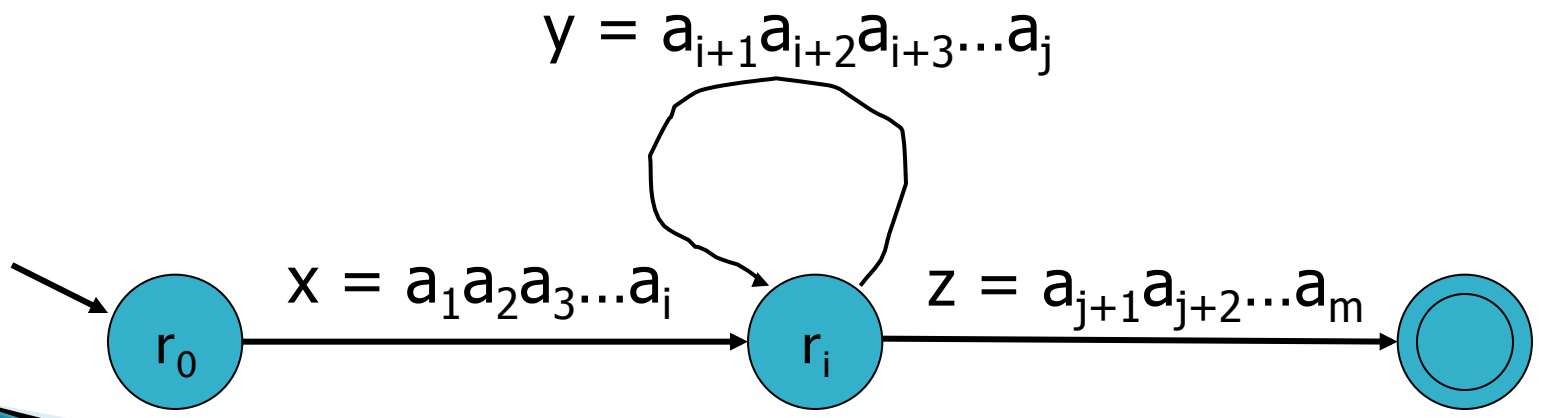
The Pumping Lemma Visually

- Consider a string $s = a_1a_2a_3\ldots a_m \in L$
- Suppose s is a long string – comprising at least as many symbols as the DFA for language L has states
  - At least one state must be visited twice



The Pumping Lemma Visually

- Let r_i be the first repeated state as the string s is processed
- Let a_i be the symbol that puts the DFA in state r_i for the first time
- We can divide the string into 3 pieces:
  - Symbols leading up to the first repeat state
  - Symbols that complete a loop back to the first repeat state
  - Symbols comprising the remainder of the string



The Pumping Lemma

- Let L be a regular language
- Then there exists a constant p (which varies for different languages), such that for every string $s \in L$ with $|s| \geq p$, s can be expressed as $s = xyz$ such that:
  - $|y| > 0$
  - $|xy| \leq p$
  - For all $k \geq 0$, the string xy^kz is also in L.

The Pumping Lemma

- What does p correspond to?
  - The number of states in the DFA
- What does y correspond to?
  - The symbols in the loop
- Why is $|y| > 0$?
  - The looping portion must involve processing at least one symbol
- Why is $|xy| \leq p$?
  - Because the first loop completes when a state is visited for the second time. If p is at least as large as the number of states of the DFA, a state must be visited twice after processing p symbols (which involves visiting p+1 total states, including the start state).
- Why is xy^kz also in L for all $k \geq 0$?
  - Because you can take the loop an arbitrary number of times, including 0!

The Pumping Lemma

- Can $|x| = 0$?
  - Yes – x can be the empty string
- What does this correspond to?
  - The loop beginning at the start state
- Can $|z| = 0$?
  - Yes – z can be the empty string
- What does this correspond to?
  - The loop occurring at the accepting state

The Pumping Lemma

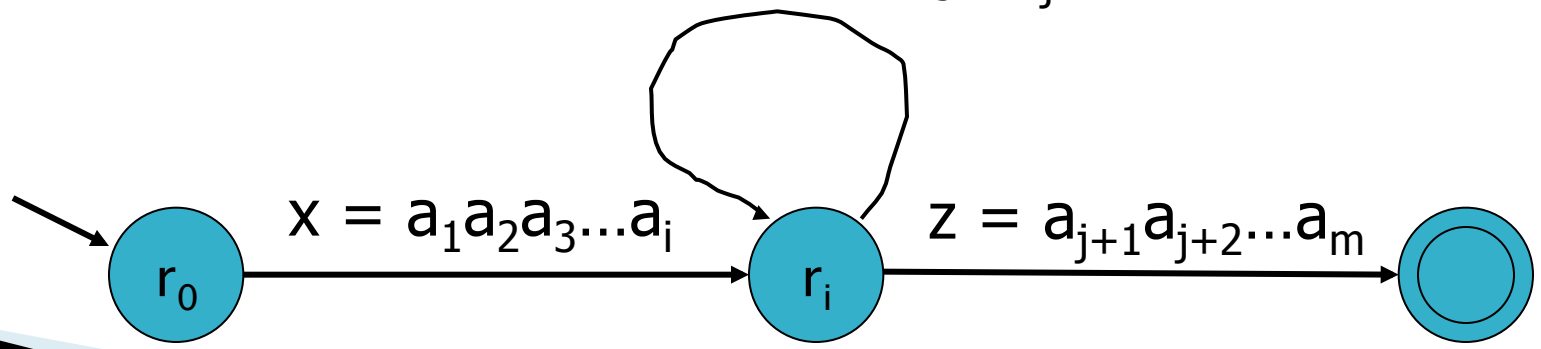
- What this means:
  - For a long enough string s in L:
    - We can express s as the concatenation of three smaller strings
    - The middle string can be “pumped” (repeated) any number of times (including 0 = deleting) and the resulting string will be in L.

The Pumping Lemma

- Proof of the pumping lemma
  - Since L is regular, there is a DFA $M=(Q, \Sigma, \delta, q_0, F)$ that accepts L.
    - Assume M has p states.
  - Consider a string $s \in L$ with $|s| = m \geq p$.
    - Express $s = a_1 a_2 a_3 \ldots a_m$ where each $a_i \in \Sigma$.
  - Define r_i to be the state M is in after reading i symbols:
    - $r_i = \delta^* (q_0, a_1 a_2 \ldots a_i)$
    - $r_0 = q_0$

The Pumping Lemma

- Proof of the pumping lemma
  - Since $|s| \geq p$, and there are only p states, one state on its path must be visited more than once.
    - There exist integers i and j, $0 \leq i < j \leq p \leq m$ such that $r_i = r_j$
    - Then $s = xyz$
      - $x = a_1 a_2 \ldots a_i$
      - $y = a_{i+1} a_{i+2} \ldots a_j$
      - $z = a_{j+1} a_{j+2} \ldots a_m$



The Pumping Lemma

- What good is the pumping lemma?
- The real strength of the pumping lemma is proving that languages are not regular
  - Proof by contradiction
    - Assume that the language to be tested is regular
    - Use the pumping lemma to come to a contradiction
    - Original assumption about the language being regular is false
- You cannot prove a language to be regular using the Pumping Lemma!

The Pumping Lemma

- To show that a language L is not regular
  - PROOF BY CONTRADICTION
  - Assume L is regular
  - Choose an “appropriate” string s in L
    - In terms of p (number of states in DFA)
  - Express s = xyz following rules of pumping lemma
  - Show that xy^kz is not in L, for some k
  - The above contradicts the Pumping Lemma
  - The assumption that L is regular is wrong
  - L must not be regular

Pumping Lemma

- Example:
  - $L = \{a^k b^k \mid k \geq 0\}$
  - Assume that L is regular.
    - Then there is a DFA, M that accepts L.
    - Let p be the number of states in M
    - Choose an appropriate string $s \in L$
      - Let $s = a^p b^p$
    - Apply Pumping Lemma to s
      - $s = xyz$
      - $|xy| \leq p$
      - $|y| > 0$

Pumping Lemma

- $s = xyz = a^p b^p$
  - aa … a bb … b
  - Since $|xy| \leq p$, xy must consist entirely of a's and, as such, y must also consist entirely of a's.
    - $y = a^j$ for some $0 < j \leq p$

Pumping Lemma

- $s = xyz = a^p b^p = a^i a^j a^k b^p$ where $i+j+k = p$
  - By the Pumping Lemma
    - xy^2z is also in L
    - $xy^2z = a^i a^{2j} a^k b^p$
    - But $i + 2j + k \neq p$ (since $j > 0$)
    - xy^2z has more a's that b's
    - Thus $xy^2z \notin L$ CONTRADICTION!

Pumping Lemma

- We arrived at a contradiction,
  - Thus our original assumption that L is regular must be incorrect
  - Thus L is not regular.
- Note that we need to find only 1 string s that fails in order for the proof by contradiction to work.
  - The key is finding the s that does the job

The Pumping Lemma

- Remember the wording of the lemma:
  - FOR ALL strings $s \in L$ that are long enough,
  - THERE EXISTS a decomposition $s = xyz$
  - Such that xy^kz is in the language FOR ALL $k \geq 0$ (and other properties hold too)
- So to show that a language is NOT regular by contradiction, we assume that it is regular and then show that:
  - THERE EXISTS string $s \in L$ that is long enough (just need 1!)
  - NO MATTER how we decompose $s = xyz$ (we have to cover ALL legitimate decompositions!)
  - xy^kz is NOT in the language FOR SOME $k \geq 0$ (we just have to show 1 case that isn't in the language!)

Pumping Lemma

- Another Example:
  - $L = \{0^i w \mid |w| \leq i, w \in \{0,1\}^*\}$
  - Ex: 0001, 0010, 0000101

  - Assume that L is regular.
    - Then there is a DFA, M, that accepts L.
    - Let p be the number of states in M

Pumping Lemma

- Another Example:
  - $L = \{0^i w \mid |w| \leq i, w \in \{0,1\}^*\}$
  - Choose an appropriate string $s \in L$
    - Let $s = 0^p 1^p$
  - Apply Pumping Lemma to s
    - $s = xyz$
    - $|xy| \leq p$
    - $|y| > 0$

Pumping Lemma

- $s = xyz = 0^p1^p$
  - 00 … 0 11 … 1
  - Since $|xy| \leq p$, xy must consist entirely of 0's and, as such, y must also consist entirely of 0's.
    - $y = 0^j$ for some $j \leq p$

Pumping Lemma

- $s = xyz = 0^p1^p = 0^i0^j0^k1^p$ where $i+j+k = p$
  - By the Pumping Lemma
    - xy^0z is also in L
    - $xy^0z = xz = 0^i0^k1^p$
    - But $p > i + k$
    - The length of the prefix of 0's is less than the suffix w
    - Thus $xy^0z \notin L$ CONTRADICTION!

Pumping Lemma

- We arrived at a contradiction,
  - Thus our original assumption that L is regular must be incorrect
  - Thus L is not regular.
- Note that we need to find only 1 string s that fails in order for the proof by contradiction to work.
  - We can show s fails when pumping 0 times

Pumping Lemma

- Another example:
  - L = set of palindromes over {a,b}
    - Strings that read the same forwards and backwards
  - Ex: aa, abba, abbbbba, ϵ

  - Assume that L is regular.
    - Then there is a DFA, M that accepts L.
    - Let p be the number of states in M

Pumping Lemma

- Another Example:
  - L = set of palindromes over {a,b}
  - Choose an appropriate string $s \in L$
    - Let $s = a^p b a^p$
  - Apply Pumping Lemma to s
    - $s = xyz$
    - $|xy| \leq p$
    - $|y| > 0$

Pumping Lemma

- $s = xyz = a^p b a^p$
  - aa … a b aa … a
  - Since $|xy| \leq p$, xy must consist entirely of a's and, as such, y must also consist entirely of a's.
    - $y = a^j$ for some $j \leq p$

Pumping Lemma

- $s = xyz = a^p b a^p = a^i a^j a^k b a^p$ where $i+j+k = p$

- By the Pumping Lemma
  - xy^2z is also in L
  - xy^2z has more than p a's at the start
  - Number of a's following b is still p
  - Thus xy^2z cannot be a palindrome
    - Thus $xy^2z \notin L$ CONTRADICTION!

Pumping Lemma

- Summary
  - The pumping lemma formalizes the idea that for a regular language, if an accepted string is long enough, eventually at least one state from the DFA will be have to be repeated on the path that accepts the string.
  - Continually looping on this state will produce an infinite number of strings in the language
  - Used to show that languages are not regular