Pumping Lemma (Continued from last lecture)

Lecture 7

Binghui Peng

What kind of language is NOT regular?

Two Methods of Proof

- **Method 1:** Run the DFA on many small words. By the **Pigeonhole Principle (PHP)**, two of the words must finish in the same state. Then use this to derive a contradiction.

- **Method 2 — Pumping Lemma:** Run the DFA on one long word. By the **PHP**, the word must visit the same state twice (creating a loop). Then use this to derive a contradiction.

The Pumping Lemma (Formal)

Pumping Lemma: If L is regular, then there exists a constant n (the pumping length) such that for all $w \in L$ with $|w| \geq n$, there exist x, y, z such that:

1. $w = xyz$ and $y \neq e$.
2. $|xy| \leq n$.
3. For all $i \geq 0$, $xy^i z \in L$.

Proof of pumping Lemma on whiteboard

How We Use the Pumping Lemma

To prove L is not regular:

1. Assume L is regular.
2. Let n be the pumping length.
3. Choose a specific string $w \in L$ with $|w| \geq n$.
4. Consider all possible ways to split $w = xyz$ with $|xy| \leq n$ and $y \neq e$.
5. Derive contradiction and show L is not regular.

Four examples we talked in the last lecture

- $L_1 = \{a^n b^n : n \geq 0\}$ revisited
- $L_3 = \{w : \#_a(w) \neq \#_b(w)\}$
- $L_4 = \{a^{n^2} : n \in \mathbb{N}\}$

$L_5 = \{a^p : p \text{ is prime}\}$ is Not Regular

$L_5 = \{a^p : p \text{ is prime}\}$ is Not Regular

Proof: Assume L_5 is regular. Let $a^p \in L_5$ where p is a large prime.

$L_5 = \{a^p : p \text{ is prime}\}$ is Not Regular

Proof: Assume L_5 is regular. Let $a^p \in L_5$ where p is a large prime. Split $a^p = xyz$ where $|y| = k \geq 1$.

$L_5 = \{a^p : p \text{ is prime}\}$ is Not Regular

Proof: Assume L_5 is regular. Let $a^p \in L_5$ where p is a large prime. Split $a^p = xyz$ where $|y| = k \geq 1$. For all i, $|xy^i z| = p + (i-1)k$ must be prime.

$L_5 = \{a^p : p \text{ is prime}\}$ is Not Regular

Proof: Assume L_5 is regular. Let $a^p \in L_5$ where p is a large prime. Split $a^p = xyz$ where $|y| = k \geq 1$. For all i, $|xy^i z| = p + (i-1)k$ must be prime. Let $i = p + 1$. Then:

$$|xy^{p+1}z| = p + pk = p(1 + k)$$

$L_5 = \{a^p : p \text{ is prime}\}$ is Not Regular

Proof: Assume L_5 is regular. Let $a^p \in L_5$ where p is a large prime. Split $a^p = xyz$ where $|y| = k \geq 1$. For all i, $|xy^i z| = p + (i-1)k$ must be prime. Let $i = p + 1$. Then:

$$|xy^{p+1}z| = p + pk = p(1 + k)$$

Since $p > 1$ and $1 + k > 1$, the result is composite. **Contradiction.**

$L_6 = \{w : \#_a(w) > \#_b(w)\}$ is Not Regular

$L_6 = \{w : \#_a(w) > \#_b(w)\}$ is Not Regular

Choose $w = b^n a^{n+1}$.

$L_6 = \{w : \#_a(w) > \#_b(w)\}$ is Not Regular

Choose $w = b^n a^{n+1}$. Since $|xy| \leq n$, the part y must consist only of b's ($y = b^k, k \geq 1$).

$L_6 = \{w : \#_a(w) > \#_b(w)\}$ is Not Regular

Choose $w = b^n a^{n+1}$. Since $|xy| \leq n$, the part y must consist only of b's ($y = b^k, k \geq 1$). Pump y with $i = 2$: $xy^2z = b^{n+k}a^{n+1}$.

$L_6 = \{w : \#_a(w) > \#_b(w)\}$ is Not Regular

Choose $w = b^n a^{n+1}$. Since $|xy| \leq n$, the part y must consist only of b's ($y = b^k, k \geq 1$). Pump y with $i = 2$: $xy^2z = b^{n+k}a^{n+1}$. If we pick $k \geq 2$, we already have more b's than a's. If we pick i large enough, we will always have more b's than a's.

$L_6 = \{w : \#_a(w) > \#_b(w)\}$ is Not Regular

Choose $w = b^n a^{n+1}$. Since $|xy| \leq n$, the part y must consist only of b's ($y = b^k, k \geq 1$). Pump y with $i = 2$: $xy^2z = b^{n+k}a^{n+1}$. If we pick $k \geq 2$, we already have more b's than a's. If we pick i large enough, we will always have more b's than a's. Thus $xy^iz \notin L_6$ for large i. **Contradiction.**

$L_7 = \{a^n b^m : n > m\}$ is Not Regular

$L_7 = \{a^n b^m : n > m\}$ is Not Regular

Problem: Pumping a's (x and y at the start) only keeps us in the language.

$L_7 = \{a^n b^m : n > m\}$ is Not Regular

Problem: Pumping a's (x and y at the start) only keeps us in the language.

Observation: The Pumping Lemma also allows pumping "down" ($i = 0$).

$L_7 = \{a^n b^m : n > m\}$ is Not Regular

Problem: Pumping a's (x and y at the start) only keeps us in the language.

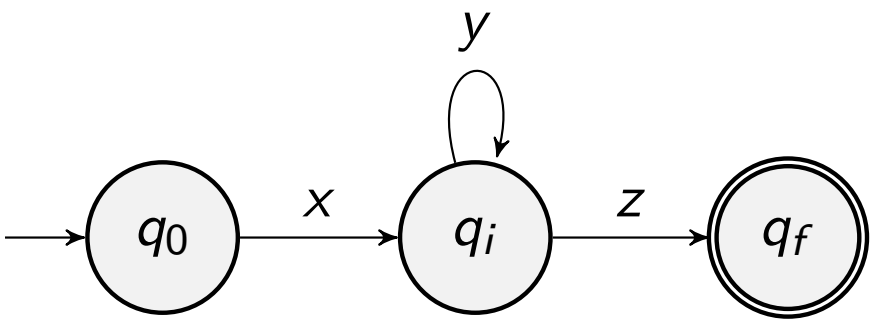
Observation: The Pumping Lemma also allows pumping "down" ($i = 0$). Choose $w = a^{n+1} b^n$. Let $y = a^k$ ($k \geq 1$).

$L_7 = \{a^n b^m : n > m\}$ is Not Regular

Problem: Pumping a's (x and y at the start) only keeps us in the language.

Observation: The Pumping Lemma also allows pumping "down" ($i = 0$). Choose $w = a^{n+1}b^n$. Let $y = a^k$ ($k \geq 1$). $xy^0z = a^{n+1-k}b^n$. Since $k \geq 1$, $n + 1 - k \leq n$, so the number of a's is no longer greater than the number of b's. **Contradiction.**

The $i = 0$ case: Pumping Down



xy^0z means we bypass the loop entirely and go straight from x to z.

$L_8 = \{a^{n_1} b^m c^{n_2} : n_1, n_2 > m\}$ is Not Regular

$L_8 = \{a^{n_1} b^m c^{n_2} : n_1, n_2 > m\}$ is Not Regular

Choose $w = a^n b^{n-1} c^n$.

$L_8 = \{a^{n_1} b^m c^{n_2} : n_1, n_2 > m\}$ is Not Regular

Choose $w = a^n b^{n-1} c^n$. As $|xy| \leq n$, $y = a^k$ for some $k \geq 1$.

$L_8 = \{a^{n_1} b^m c^{n_2} : n_1, n_2 > m\}$ is Not Regular

Choose $w = a^n b^{n-1} c^n$. As $|xy| \leq n$, $y = a^k$ for some $k \geq 1$. Consider $i = 0$: $xy^0 z = a^{n-k} b^{n-1} c^n$.

$L_8 = \{a^{n_1} b^m c^{n_2} : n_1, n_2 > m\}$ is Not Regular

Choose $w = a^n b^{n-1} c^n$. As $|xy| \leq n$, $y = a^k$ for some $k \geq 1$. Consider $i = 0$: $xy^0 z = a^{n-k} b^{n-1} c^n$. If $k \geq 1$, then $n - k \leq n - 1$, which is not greater than $m = n - 1$.

$L_8 = \{a^{n_1} b^m c^{n_2} : n_1, n_2 > m\}$ is Not Regular

Choose $w = a^n b^{n-1} c^n$. As $|xy| \leq n$, $y = a^k$ for some $k \geq 1$. Consider $i = 0$: $xy^0 z = a^{n-k} b^{n-1} c^n$. If $k \geq 1$, then $n - k \leq n - 1$, which is not greater than $m = n - 1$. Thus $xy^0 z \notin L_8$. **Contradiction.**

Lower Bounds: Looking Ahead

1. DFAs are simple enough that we can prove certain languages are not regular.
2. Context-Free Grammars are also simple enough to prove some languages are not context-free.
3. Proving P $\neq$ NP is hard because Turing Machines are very complex.
4. Proving undecidability is often easier because it depends on the fundamental logic of computation rather than model details.