CS5371
Theory of Computation

Lecture 5: Automata Theory III (Non-regular Language, Pumping Lemma, Regular Expression)

Announcement

- Homework 1 is released!
- Due: 10/13, 3:20 pm (before class)
  - Late submission will not be marked
- Difficulty: Quite difficult
  - Start early >=)

Objectives

- Prove the Pumping Lemma, and use it to show that there are non-regular languages
- Introduce Regular Expression
  - which is one way to describe a language (or a set of strings)

Non-Regular Language?

- To understand the power of DFA, apart from knowing what it can do, we need to know what it cannot do
- Let's look at the language B = $\{0^n1^n \mid n \geq 0\}$
- If we try to find a DFA to recognize B, the DFA needs to keep track of the number of 0's we have seen so far
- However, number of 0's is unlimited… there are unlimited number of possibilities
- So, it is NOT POSSIBLE if the DFA just has FINITE number of states!

Pumping Lemma

Theorem: If *A* is a regular language, then there is a number p (called the pumping length) such that:

if s is a string in *A* of length at least p, then s can be divided into three pieces, s = xyz, satisfying the following three conditions:

- For each $k \geq 0$, $xy^kz \in A$
- $|y| > 0$, and
- $|xy| \leq p$

Pumping Lemma (Proof)

- Let us assign the pumping length p to be the number of states in the DFA that recognize A
- Consider the sequence of states that the DFA goes through when reading

$$s = s_1s_2 \ldots s_n$$

- At the beginning, it is at state $r_0 = q_{start}$
- Then, it goes to r_1 after reading s_1, then goes to r_2, then goes to r_3 …

Pumping Lemma (Proof)

- When it has finished reading s_p, one of the state has been visited at least two times (why?)
- That is, $r_i = r_j$, for some $0 \leq i < j \leq p$
- Now, let $x = s_1 s_2 \ldots s_i$,

  $y = s_{i+1} s_{i+2} \ldots s_j$, and

  $z = s_{j+1} s_{j+2} \ldots s_n$

- We can check that $xy^k z \in A$ for all $k \geq 0$ (why?)
- Also, $|y| > 0$ and $|xy| \leq p$ (why?)

Use of Pumping Lemma
(Example 1)

- Lemma: The language $B = \{0^n1^n \mid n \geq 0\}$ is not regular.

- How to prove?
  - Use Pumping Lemma
  - By contradiction
- Proof: Assume that B is regular. Then…

Use of Pumping Lemma
(Example 1)

- Then, let p be the pumping length
- We know that 0^p1^p is in B
- By pumping lemma, we know that 0^p1^p can be divided into three parts, xyz, such that $|y| > 0$, $|xy| \leq p$, and xy^kz is in B for all $k \geq 0$
- In this case, y consists of all 0's and at least 1 zero (why??)
- xyyz is in B, but xyyz has more 0's than 1's
- Contradiction occurs!

Use of Pumping Lemma
(Example 2)

- Lemma: The language C = { w | w has an equal number of 0s and 1s } is not regular.

- How to prove?

Use of Pumping Lemma
(Example 2)

- Proof 1: Similar to Example 1. Let s = 0^p1^p and apply pumping lemma.
- Proof 2: We use the fact: the class of regular languages is closed under intersection (will be proved in tutorial next Tue). That is,

> If A and B are regular languages,
> then A ∩ B is also a regular language.

Use of Pumping Lemma
(Example 2)

- A = { 0^m1^n | m, n $\geq$ 0} is regular
- If C is regular, $C \cap A$ is regular
- However, $C \cap A$ = { 0^n1^n | n $\geq$ 0} is not regular
- Thus, contradiction occurs (where?). So, C is not regular

Use of Pumping Lemma
(Example 2)

- In Proof 1, we choose $s = 0^p1^p$, we can apply pumping lemma successfully and prove that C is not regular
- However, if we 'unluckily' choose $s = (01)^p$, using pumping lemma may not give contradiction… (E.g., $|x| = \varepsilon$, $y = 01$, $z = (01)^{p-1}$, then every xy^kz is in C)
- So, if you fail on first attempt, don't give up, try another one!

Use of Pumping Lemma
(Example 3)

- Lemma: The language $F = \{ ww \mid w \in \{0,1\}^* \}$ is not regular.

- How to prove?

Use of Pumping Lemma
(Example 4)

- Lemma: The language $\{ 1^{n^2} \mid n \geq 0\}$ is not regular.
- Proof:
  - Let p be the pumping length.
  - Let $s = 1^{p^2}$.
  - By pumping lemma, we have $|xyz| = p^2$. Also, $0 < |y| \leq |xy| \leq p$.
  - $p^2 < |xyyz| \leq p^2 + p < (p+1)^2$
  - Contradiction occurs (where??)

Use of Pumping Lemma
(Example 5)

- Lemma: The language $E = \{ 0^i1^j \mid i > j \}$ is not regular.

- Proof: Let $s = 0^{p+1}1^p$. By pumping lemma, we can divide s into xyz such that y consists of all 0's and $|y| > 0$.
- Then, $xz \in E$ but xz does not have more 0s than 1s
- Contradiction occurs

Regular Expression

- In arithmetic, we can use the operations + and x to build up expressions

  such as (5+3) x 4

  - The value of this expression is 32
- Similarly, we can use regular operations to build up regular expressions

  such as $(0 \cup 1)0^*$

  - The value of this expression is a language

What does $(0 \cup 1)0^*$ mean?

- The symbols 0 and 1 are shorthand for the set {0} and {1}
  - So, $(0 \cup 1)$ means $(\{0\} \cup \{1\})$
  - 0^* means $\{0\}^*$, whose value is the language consisting of all strings with any number of 0s
- Just like x in arithmetic expression, the concatenation symbol $\circ$ is often omitted
  - So, $(0 \cup 1) 0^*$ means $(0 \cup 1) \circ 0^*$
- This expression describes the set of strings that start with a 0 or a 1, which is followed by any number of 0s

What does the following regular expressions mean?

0^*10^*	Binary strings containing exactly one 1
$\Sigma^*1\Sigma^*$	Any strings containing 1
$\Sigma^*001\Sigma^*$	Any strings containing 001
$1^*(01^+)^*$	Binary strings with 1 following each 0
$(\Sigma\Sigma)^*$	Any strings with even length
$(0 \cup \varepsilon) 1^*$	$01^* \cup 1^*$
$1^* \emptyset$	Empty set (no strings)

Note: The notation R^+ means RR^*

Next time

- Formally define regular expression
- We will also show that
  - (1) Language recognized by DFA can be described by Regular Expression
  - (2) Language described by Regular Expression can be recognized by DFA