Finite Automata 06

Thumrongsak Kosiyatrakul
tkosiyat@cs.pitt.edu

Language

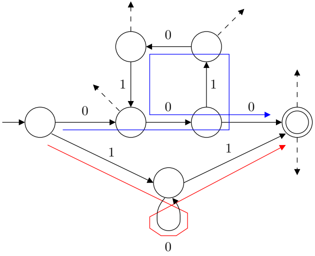
- A language is a set of strings
  - A set can be empty
  - A set can have a finite number of elements
  - A set can have an infinite number of elements
- Regular or not regular?
  - If L is the empty language,
    - L is regular since we can express it using the regular expression $\emptyset$
  - If L is finite
    $$L = \{s_1, s_2, s_3, \ldots, s_n\}$$
    for a number $n > 0$ and s_i is a string,
    - L is regular since we can express it using the regular expression
      $$s_1 \cup s_2 \cup s_3 \cup \cdots \cup s_n$$
- So, a non-regular language must be an infinite language

Regular Infinite Languages

- But an infinite language can be a regular language:
  - $\{w \mid w$ starts with a $1\}$
  - $\{w \mid w$ contains 011 as a substring$\}$
  - $\{w \mid w$ ends with $0110\}$
- Technically, there are infinite number of regular languages that contains infinite number of strings
- There must be something that can be used to distinguish between regular languages and non-regular languages
  - By definition, a language is regular if there are some finite state machines that recognize it
  - Recall that the number of states of a finite state machine must be finite
  - But a finite state machine can accept an infinite number of strings

Regular Infinite Languages

- What is the special property that makes a finite state machine accepts an infinite number of strings?
  - A loop **in a path to an accept state**



  - Let $L(M)$ be the language of the above machine M:
    - $10^*1 \subseteq L(M)$

$$10^*1 = \{11, 101, 1001, 10001, \dots\} \subseteq L(M)$$

In other words, $10^i1 \in L(M)$ for any $i \geq 0$

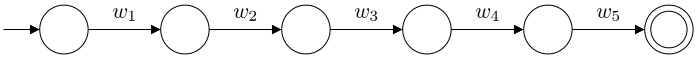
    - $00(1010)^*0 \subseteq L(M)$

$$00(1010)^*0 = \{000, 0010100, 00101010100, \dots\} \subseteq L(M)$$

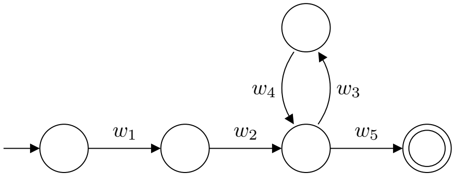
In other words, $00(1010)^i0 \in L(M)$ for any $i \geq 0$

Regular Infinite Languages

- Given a DFA M, how to detect that there is a loop in a path to an accept state?
- Suppose a DFA M has 5 states and it accepts the string $w = w_1w_2w_3w_4w_5$ of length 5



  - There are the total of 6 current states but there are only 5 states
    - At least two of them must be the same (Pigeonhole principle)
  - Suppose the third and the fifth are the same state



    - $w_1w_2(w_3w_4)^iw_5 \in L(M)$ for any $i \geq 0$

Regular Infinite Languages

- From previous example
  - Any strings of length **at least** 5 that is accepted by M will go through a loop
  - If we let $x = w_1w_2$, $y = w_3w_4$, and $z = w_5$, we can say that

$$xy^iz \in L(M) \text{ for any } i \geq 0$$

- Given an infinite regular language A, there is a finite state machine M that recognizes it
  - But we have no idea how many states it has
  - Suppose it has p states
    - Any string $s \in A$ of length **at least** p will go through a loop
    - s must be divided into $s = xyz$ where $y \neq \varepsilon$ such that

$$xy^iz \in A \text{ for any } i \geq 0$$

where y is the string that takes you around a loop

Example

- Consider the following language

$$A = \{w \mid w \text{ contains } 011 \text{ as a substring}\}$$

- We need at least a 4-states DFA to recognize A
- Let's find a string $s \in A$ of length at least 4
  - Let $s = 0111$
    - $x = 011$, $y = 1$, $z = \varepsilon$
    - $xy^0z = xz = 011 \in A$
    - $xy^1z = xyz = 0111 \in A$
    - $xy^2z = xyyz = 01111 \in A$
    - $\vdots$
    - $xy^iz \in A$ for $i \geq 0$
  - Let $s = 0101011$
    - $x = 0$, $y = 1$, $z = 01011$ and $xy^iz \in A$ for $i \geq 0$
    - $xy^0z = xz = 001011 \in A$
    - $xy^1z = xyz = 0101011 \in A$
    - $xy^2z = xyyz = 01101011 \in A$
    - $\vdots$
    - $xy^iz \in A$ for $i \geq 0$

Pumping Lemma

- The pumping lemma states that all regular languages has a special property
- If a language lack this property, it is not a regular language

Property

All strings in the language can be **pumped** if they are at least as long as a certain special value, called the **pumping length**. Each such strings contains a section that can be repeated any number of times with the resulting string remaining in the language.

- **pumped**: xy^iz for any $i \geq 0$
  - We can insert the string y in between x and z any number of times but the result string is still in the language

Pumping Lemma

Pumping Lemma

If A is a regular language, then there is a number p (the pumping length) where if s is any string in A of length at least p, then s must be divided into three pieces, $s = xyz$, satisfying the following conditions:

1. for each $i \geq 0$, $xy^iz \in A$,
2. $|y| > 0$, and
3. $|xy| \leq p$.

where

- $|s|$ represents the length of the string s
- y^i means that i copies of y are concatenated together
  - y^0 equals ε but it does not mean that $y = \varepsilon$
    - $(010)^0 = \varepsilon$ but $010 \neq \varepsilon$
  - $xy^0z = xz$, $xy^1z = xyz$, $xy^2z = xyyz$, $xy^3z = xyyyz$, and so on

Proof of the Pumping Lemma

- Let M be a DFA recognizing A and p (the pumping length) be the number of states of M.
- Let s be a string of length at least p.
  - $s = s_1 s_2 \dots s_n$ where $s_x \in \Sigma$ and $n \geq p$.
- Let $r_1, r_2, \dots, r_{n+1}$ be the sequence of states of M when processing s.
  - r_1 is the start state of M
  - When s_1 is processed, the state of M is changed to r_2, and so on.
    - $\delta(r_1, s_1) = r_2$
    - $\delta(r_2, s_2) = r_3$
    - $\vdots$
    - $\delta(r_i, s_i) = r_{i+1}$ for $1 \leq i \leq n$.
    - $\vdots$
    - $\delta(r_n, s_n) = r_{n+1}$
  - **Note** that there is no restriction that r_x and r_y cannot be the same state.

Proof of the Pumping Lemma

- The sequence $r_1, r_2, \ldots, r_{n+1}$ consists of $n + 1$ states
  - Since $n \geq p$, the above sequence has at least $p + 1$ states.
- Since the machine M has only p states, in the first $p + 1$ states of the sequence, at least two states r_j and r_l must be the same state.
  - Let r_j be the first occurrence of the repeated state
  - Let r_l be the second occurrence of the repeated state in the above sequence.
  - **Note** that $j < l$.
- Since r_l is in the first $p + 1$ states of the sequence $l \leq p + 1$.
- Let
  - $x = s_1 \ldots s_{j-1}$
  - $y = s_j \ldots s_{l-1}$
  - $z = s_l \ldots s_n$

Proof of the Pumping Lemma

- If M accepts $s = s_1 s_2 \dots s_n$ and $s = xyz$,
  - x takes M from r_1 to r_j,
  - y takes M from r_j to r_l, and
  - z takes M from r_l to r_{n+1}

  where r_{n+1} is an accept state.
- Let's check all conditions of the pumping lemma
  1. Thus M accept xy^iz for $i \geq 0$.
  2. Since $j < l$, $|y| > 0$.
  3. Since $l \leq p + 1$, $|xy| \leq p$.

How to use the Pumping Lemma

- To check whether a language B is **not** regular using the Pumping Lemma, we use prove by contradiction
  - Assume that B is regular
    - There exists a machine M with p states that recognizes B
  - Select a string $s \in B$ of length at least p so that the conditions 1, 2, and 3 of the pumping lemma lead to a contradiction
- **Notes**
  - The choice of s must involve p to ensure that s has length at least p (e.g., $s = 0^p011$, $s = a^pba^{2p}$ or $s = b^{p+1}a^pb$)
  - It is possible that some choices of s do not produce contradiction. **If we do not get a contradiction, we have not proved anything yet**
  - Once you pick an s, nothing tells us what x, y, and z should be. **We have to show that we must get a contradiction, no matter what x, y, and z are, as long as they satisfy conditions 1, 2, and 3**.

Example

Show that $B = \{0^n 1^n \mid n \geq 0\}$ is not regular.

- Assume that B is regular. In other words, there exists a machine M with p states that recognizes B.
  - There are infinite number of strings in B of length at least p
  - Just pick one (for now)
- Let $s = 0^p 1^p$. Note that $s \in B$ and $|s| = 2p \geq p$.
  - The pumping lemma says there are strings x, y, and z such that $s = xyz$ satisfying the conditions 1, 2, and 3
- Recall that there are multiple ways to divide s into x, y, and z such that $xyz = s = 0^p 1^p$
  - Examples:
    - $x = \varepsilon$, $y = 0$, and $z = 0^{p-1}1^p \rightsquigarrow xyz = \varepsilon 0 0^{p-1} 1^p = 0^p 1^p$
    - $x = 0^2$, $y = 0^3$, and $z = 0^{p-5}1^p \rightsquigarrow xyz = 0^2 0^3 0^{p-5} 1^p = 0^p 1^p$
    - $x = 0^p 1$, $y = 1$, and $z = 1^{p-2} \rightsquigarrow xyz = 0^p 1 1 1^{p-2} = 0^p 1^p$

Example: $B = \{0^n1^n \mid n \geq 0\}$

- Since there are multiple ways to divide s, we are going to focus on all possible way to divide s into x, y, and z satisfying only conditions 2 and 3 first
  - We will try to get a contradiction from the first condition
- The condition 3 says $|xy| \leq p$
  - Since s starts with p 0s, to satisfy this condition, x and y must be strings that contains only 0s
    - If x contains one 1, for $s = 0^p1^p = xyz$, $|x|$ is already $p + 1$

$$|xy| = |x| + |y| = (p + 1) + |y| > p$$

    - If y contains one 1, for $s = 0^p1^p = xyz$, $|xy|$ is already $p + 1$
- Formally, to satisfy conditions 3
  - $x = 0^j$ for some $j \geq 0$
  - $y = 0^k$ for some $k > 0$
    - $k > 0$ makes $|y| > 0$ (satisfying condition 2)

Example: $B = \{0^n1^n \mid n \geq 0\}$

- Now we have
  - $x = 0^j$ for some $j \geq 0$ and
  - $y = 0^k$ for some $k > 0$
- To make $xyz = s = 0^p1^p$, z must be $0^{p-(j+k)}1^p$

$$xyz = 0^j0^k0^{p-(j+k)}1^p = 0^{j+k+p-(j+k)}1^p = 0^p1^p$$

- Condition 1 says that $xy^iz \in B$ for any $i \geq 0$
  - We just need to find an i such that $xy^iz \notin B$
  - Let $i = 0$

$$\begin{aligned} xy^0z &= 0^j(0^k)^00^{p-(j+k)}1^p \\ &= 0^j0^{p-(j+1)}1^p \\ &= 0^{p-k}1^p \end{aligned}$$

  - For $0^{p-k}1^p$ to be in $B = \{0^n1^n \mid n \geq 0\}$
    - $p - k$ must be equal to p
    - k must be 0 to make $p - k = p$ but k cannot be 0
    - Contradiction $\rightsquigarrow$ B is not regular

Example: $B = \{0^n 1^n \mid n \geq 0\}$

- There are multiple is that can lead to a contradiction
  - But i should not be 1 since $xy^1z = xyz = s \in B$
- Let $i = 2$

$$\begin{aligned} xy^2z &= 0^j(0^k)^2 0^{p-(j+k)}1^p \\ &= 0^j 0^k 0^k 0^{p-(j+1)} 1^p \\ &= 0^{p+k}1^p \end{aligned}$$

- For $0^{p+k}1^p$ to be in B
  - $p + k$ must be equal to p
  - k must be 0 to make $p + k = p$ but k cannot be 0
  - Contradiction $\rightsquigarrow$ B is not regular
- In this example, any $i \neq 1$ will give you a contradiction

Example: $B = \{0^n 1^n \mid n \geq 0\}$

- There are multiple strings s of length at least p that work for this example
- Example: $s = 0^{2p}1^{2p}$
  - This this string s, use exact same proof where $x = 0^j$ for any $j \geq 0$, $y = 0^k$ for any $k > 0$, and $z = 0^{2p-(j+k)}1^{2p}$
- Example: $s = 0^{\frac{p}{2}}1^{\frac{p}{2}}$
  - This one is a little bit harder since condition 3 does not help much
  - There are three possibility for the string y
    - y contains nothing but 0s ($y = 0^k$ for some $k > 0$) $\rightsquigarrow$ contradiction because xy^2z will have more 0s than 1s
    - y contains some 0s and 1s ($y = 0^k1^m$ for some $k, m > 0$) $\rightsquigarrow$ contradiction because $xy^2z = 0^j0^k1^m0^k1^m1^{\frac{p}{2}-m} \notin B$
    - y contains nothing but 1s ($y = 1^k$ for some $k > 0$) $\rightsquigarrow$ contradiction because xy^2z will have more 1s than 0s

Rule of Thumb

- Pick a string s in the language of length at least p such that it starts with at least p of the same symbol
    - $0^p 1^p$
    - $0^{2p} 1^{2p}$
    - Condition 3 will help reducing the amount of proofs that you have to do

Some Incorrect Proofs: $B = \{0^n 1^n \mid n \geq 0\}$

- Let $s = 000111$
  - s does not have length at least p (p can be any positive number)
- Let $s = 0^p 1^{2p}$
  - $s \notin B$, cannot use the Pumping lemma
- Let $s = 0^p 1^p$ and $x = 0$, $y = 0^{p-1}$, $z = 1^p$
  - This only show one way of dividing s into x, y, and z such that $s = xyz$
  - There are multiple ways
  - Need to show them all by using variable (e.g., 0^j, 0^k, etc)
- Let $s = 0^p 1^p$ and $x = 0^j$, $y = 0^k$, and $z = 1^p$
  - $xyz = 0^j 0^k 1^p = 0^{j+k} 1^p \neq s = 0^p 1^p$
  - If you say $j + k = p$, it is still incorrect
    - You only show all possible way such that $s = xyz$ where $z = 1^p$
    - But z can have some 0s as well

Show that $B = \{0^n 1^n \mid n \geq 0\}$ is not regular

Assume that B is regular. Since B is regular, the Pumping lemma says that for any string $s \in B$ of length at least p, s can be divided into $s = xyz$ satisfying the following conditions:

1. $xy^i z \in B$ for any $i \geq 0$
2. $|y| > 0$
3. $|xy| \leq p$

Let $s = 0^p 1^p$. Since s starts with p 0s, to satisfy the third condition, x and y are strings that contain nothing but 0s. In other words, $x = 0^j$ for any $j \geq 0$, and $y = 0^k$ for any $k > 0$. Note that k must be greater than 0 because $|y| = |0^k| = k$, and the condition 2 says that $|y| > 0$. Since $x = 0^j$ and $y = 0^k$, $z = 0^{p-(j+k)} 1^p$. Let $i = 0$. We have

$$\begin{aligned} xy^i z &= xy^0 z \\ &= xz \\ &= 0^j 0^{p-(j+k)} 1^p \\ &= 0^{p-k} 1^p \end{aligned}$$

For the string $0^{p-k} 1^p$ to be in B, the number of 0s must be equal to the number of 1s. In other words, $p - k$ must be equal to p. This requires k to be 0. But since k must be greater than 0, $xy^0 z \notin B$ — contradiction. Therefore, B is not regular.