## Week4 monday

Recap so far: In DFA, the only memory available is in the states. Automata can only "remember" finitely far in the past and finitely much information, because they can have only finitely many states. If a computation path of a DFA visits the same state more than once, the machine can't tell the difference between the first time and future times it visits this state. Thus, if a DFA accepts one long string, then it must accept (infinitely) many similar strings.

**Definition** A positive integer p is a **pumping length** of a language L over  $\Sigma$  means that, for each string  $s \in \Sigma^*$ , if  $|s| \ge p$  and  $s \in L$ , then there are strings x, y, z such that

$$s = xyz$$

and

$$|y| > 0$$
, for each  $i \ge 0$ ,  $xy^i z \in L$ , and  $|xy| \le p$ .

**Negation**: A positive integer p is **not a pumping length** of a language L over  $\Sigma$  iff

$$\exists s \ ( \ |s| \ge p \land s \in L \land \forall x \forall y \forall z \ ( \ (s = xyz \land |y| > 0 \land |xy| \le p \ ) \rightarrow \exists i (i \ge 0 \land xy^iz \notin L)) \ )$$

Informally:

Restating **Pumping Lemma**: If L is a regular language, then it has a pumping length.

Contrapositive: If L has no pumping length, then it is nonregular.

The Pumping Lemma cannot be used to prove that a language is regular.

The Pumping Lemma can be used to prove that a language is not regular.

Extra practice: Exercise 1.49 in the book.

**Proof strategy**: To prove that a language L is **not** regular,

- Consider an arbitrary positive integer p
- Prove that p is not a pumping length for L
- Conclude that L does not have any pumping length, and therefore it is not regular.

Example:  $\Sigma = \{0, 1\}, L = \{0^n 1^n \mid n \ge 0\}.$ 

Fix p an arbitrary positive integer. List strings that are in L and have length greater than or equal to p:

 ${\rm Pick}\ s =$ 

Suppose s = xyz with  $|xy| \le p$  and |y| > 0.

Then when i =,  $xy^iz =$ 

**Example**:  $\Sigma = \{0, 1\}$ ,  $L = \{ww^{\mathcal{R}} \mid w \in \{0, 1\}^*\}$ . Remember that the reverse of a string w is denoted  $w^{\mathcal{R}}$  and means to write w in the opposite order, if  $w = w_1 \cdots w_n$  then  $w^{\mathcal{R}} = w_n \cdots w_1$ . Note:  $\varepsilon^{\mathcal{R}} = \varepsilon$ . Fix p an arbitrary positive integer. List strings that are in L and have length greater than or equal to p: Pick s =Suppose s = xyz with  $|xy| \le p$  and |y| > 0.  $, xy^iz =$ Then when i =**Example**:  $\Sigma = \{0, 1\}, L = \{0^j 1^k \mid j \ge k \ge 0\}.$ Fix p an arbitrary positive integer. List strings that are in L and have length greater than or equal to p: Pick s =Suppose s = xyz with  $|xy| \le p$  and |y| > 0.  $xy^iz =$ Then when i =**Example**:  $\Sigma = \{0, 1\}, L = \{0^n 1^m 0^n \mid m, n \ge 0\}.$ Fix p an arbitrary positive integer. List strings that are in L and have length greater than or equal to p: Pick s =Suppose s = xyz with  $|xy| \le p$  and |y| > 0.  $xy^iz =$ Then when i =

Extra practice:

Language	$s \in L$	$s \notin L$	Is the language regular or nonregular?
$\{a^nb^n\mid 0\leq n\leq 5\}$			
$\{b^na^n\mid n\geq 2\}$			
$\{a^mb^n\mid 0\leq m\leq n\}$			
$\{a^mb^n\mid m\geq n+3, n\geq 0\}$			
$\{b^ma^n\mid m\geq 1, n\geq 3\}$			
$\{w \in \{a,b\}^* \mid w = w^{\mathcal{R}}\}$			
$\{ww^{\mathcal{R}} \mid w \in \{a, b\}^*\}$			

## Week3 friday

**Definition and Theorem**: For an alphabet  $\Sigma$ , a language L over  $\Sigma$  is called **regular** exactly when L is recognized by some DFA, which happens exactly when L is recognized by some NFA, and happens exactly when L is described by some regular expression

We saw that: The class of regular languages is closed under complementation, union, intersection, set-wise concatenation, and Kleene star.

**Prove or Disprove**: There is some alphabet  $\Sigma$  for which there is some language recognized by an NFA but not by any DFA.

**Prove or Disprove**: There is some alphabet  $\Sigma$  for which there is some finite language not described by any regular expression over  $\Sigma$ .

**Prove or Disprove**: If a language is recognized by an NFA then the complement of this language is not recognized by any DFA.

Fix alphabet  $\Sigma$ . Is every language L over  $\Sigma$  regular?

Set	Cardinality
$\{0,1\}$	
$\{0,1\}^*$	
$\mathcal{P}(\{0,1\})$	
The set of all languages over $\{0,1\}$	
The set of all regular expressions over $\{0,1\}$	
The set of all regular languages over $\{0,1\}$	

Strategy: Find an **invariant** property that is true of all regular languages. When analyzing a given language, if the invariant is not true about it, then the language is not regular.

**Pumping Lemma** (Sipser Theorem 1.70): If A is a regular language, then there is a number p (a pumping length) where, if s is any string in A of length at least p, then s may be divided into three pieces, s = xyz such that

- $\bullet |y| > 0$
- for each  $i \ge 0$ ,  $xy^iz \in A$
- $|xy| \leq p$ .

## **Proof illustration**

True or False: A pumping length for  $A = \{0, 1\}^*$  is p = 5.