

## Monday April 18

Extra question: What if a DFA does not have a loop? What is the pumping length of the language recognized by such a DFA?

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  $|s| \geq p$  and  $s \in L$ , then there are strings  $x, y, z$  such that

and  $s = xyz$  (loop happens quickly)  
 $|y| > 0$ , for each  $i \geq 0$ ,  $xy^iz \in L$ , and  $|xy| \leq p$ .

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

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

Informally: there is a long string in the language that is not pumpable.

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. (pos integer)

|| To prove  $L$  is nonregular, we try to show it has no pumping length.

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.

Extra practice: Give general description of the pumping length & a finite set.

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

1. Consider an arbitrary positive integer  $p$  (fixed but unknown).
2. Prove that  $p$  is not a pumping length for  $L$ .
3. Conclude that  $L$  does not have *any* pumping length, and therefore it is not regular.

$$L = L(12 \Sigma^* 34)$$

$$\Sigma = \{1, 2, 3, 4\}$$

regular



$S \in L$  and  $S$  is long ( $\overset{\text{length}}{\geq} 5$ )

$$S = \begin{matrix} 12 & & 34 \\ x & y & z \end{matrix}$$

where

$$\left. \begin{array}{l} xz \\ xy z \\ xyyz \\ xy yz \\ xyyyz \\ xyyyyz \end{array} \right\} \text{all in } L$$

Balance!

Examples of strings in  $L$   
 $\epsilon$     01    0011    000111

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

Examples of strings not in  $L$

10    0    11111

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

$0^p 1^p$  length  $2p$

$0^{2p} 1^{2p}$  length  $4p$

$0^{p+1000} 1^{p+1000}$  length  $2p+2000$

Goal: Show that  $p$  is not a pumping length for  $L$ .

Pick  $s =$

$0^p 1^p$

NOTE:  $|s| \geq p$  because  $|s| = 2p$ .

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

arbitrary  $x, y, z$  satisfying constraints

$00 \dots 00 11 \dots 11$

$m > 0$

$r = p - m - k$

$x = 0^k$   $y = 0^m$   $z = 0^r 1^p$  where

Then when  $i =$

2

$xy^i z =$

$x y y z = 0^k 0^m 0^m 0^r 1^p$

$= 0^{k+m+m+r} 1^p$

$= 0^{p+m} 1^p$

Show  $s$  is not pumpable.

Since we've found a string in  $L$  that should have been pumpable but wasn't, we've shown that  $p$  is not a pumping length for  $L$ . But  $p$  was arbitrary, so no pos integer is a pumping length for  $L$ , so  $L$  is not regular  $\square$

Notice: we could have chosen  $i$  to be 0 or 3, 105.

**Example:**  $\Sigma = \{0, 1\}$ ,  $L = \{ww^R \mid w \in \{0, 1\}^*\}$ .

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

Pick  $s = 0^p 110^p$

Suppose  $s = xyz$  with  $|xy| \leq p$  and  $|y| > 0$ . Then  $x = 0^k$ ,  $y = 0^m$ ,  $z = 0^r 110^p$  for some  $k \in \mathbb{Z}^{\geq 0}$ ,  $m \in \mathbb{Z}^+$ ,  $r \in \mathbb{Z}^{\geq 0}$  with  $k+m+r=p$ .

$(00^x)(0^y)(001100 \dots 00)$

Then when  $i = 0$ ,  $xy^iz = xz = 0^k 0^r 110^p = 0^{p-m} 110^p$  which is not in  $L$  because if split into two pieces, either each piece has single 1 but then different length pieces (since  $m > 0$ ) so not reversals of each other, or one piece has both 1s and the other none, so not reversals of each other.

**Example:**  $\Sigma = \{0, 1\}$ ,  $L = \{0^j 1^k \mid j \geq k \geq 0\}$ . extra practice

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  $|xy| \leq p$  and  $|y| > 0$ .

Then when  $i =$  ,  $xy^iz =$

**Example:**  $\Sigma = \{0, 1\}$ ,  $L = \{0^n 1^t 0^n \mid t, n \geq 0\}$ . extra practice

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  $|xy| \leq p$  and  $|y| > 0$ .

Then when  $i =$  ,  $xy^iz =$

## **Review: Week 4 Monday**

Recall: Review quizzes based on class material are assigned each day. These quizzes will help you track and confirm your understanding of the concepts and examples we work in class. Quizzes can be submitted on Gradescope as many times (with no penalty) as you like until the quiz deadline: the three quizzes each week are all due on Friday (with no penalty late submission open until Sunday).

Please complete the review quiz questions on Gradescope about pumping lemma and nonregular sets.

**Pre class reading for next time:** Page 112

office hours today after class in same Zoom room.  
Friday's class back in WLH 2001

Wednesday April 20

Language	$s \in L$	$s \notin L$	Is the language regular or nonregular?
$\{a^n b^n \mid 0 \leq n \leq 5\}$ <small><math>\{ \epsilon, ab, aabb, aaabbb, aaaabbbb, aaaaabbbbb \}</math></small>	aabb	abbb aaaaaabbbbbbb	Regular because finite
$\{b^n a^n \mid n \geq 2\}$	bbaa	ba aa bb	Nonregular using pumping lemma.
$\{a^i b^j \mid 0 \leq i \leq j\}$			
$\{a^i b^j \mid i \geq j + 3, j \geq 0\}$			
$\{b^i a^j \mid i \geq 1, j \geq 3\}$			
$\{w \in \{a, b\}^* \mid w = w^R\}$ The set of palindromes			Nonregular using pumping lemma
$\{ww^R \mid w \in \{a, b\}^*\}$	<u>aab</u> <u>baa</u> $w \quad w^R$	aabb abbb	Nonregular using pumping lemma

① The set of strings of even length that are palindromes

Consider arbitrary  $x, y, z$  satisfying  $s = xyz$   
 $|xy| \leq p$ ,  $|y| > 0$ ,  
 and wts there is some nonneg int  $i$  with  
 $xy^iz \notin \{ww^R \mid w \in \{a, b\}^*\}$

From the constraints on  $x, y, z$  we know we can write

$x = 0^k \quad y = 0^m \quad z = 0^r 110^p$   
 where  $k+m+r = p \quad m > 0 \quad k \geq 0 \quad r \geq 0$ .

Choose  $i = 0$  or  $2$  or  $\dots$

and show  $xy^iz \notin \{ww^R \mid w \in \{a, b\}^*\}$   
 $xyyz = 0^{k+2m} 0^r 110^p = 0^{p+m} 110^p$

② Using pumping lemma  
 - Consider arbitrary pos int,  $p$   
 - wts it is not a pumping length for  $\{ww^R \mid w \in \{a, b\}^*\}$   
 By giving a specific value  
 $s = 0^p 110^p$   
 and show that  $s$  is not pumpable  
 conclude this language has no pumping length so it's not regular.

To prove a set is nonregular

- ① Using pumping lemma
- ② Using closure properties

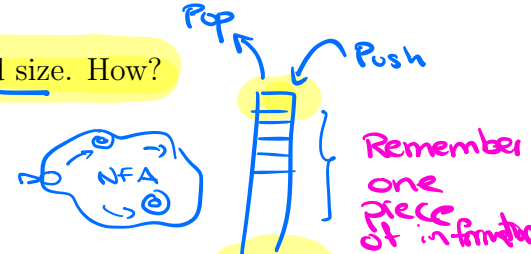
Regular sets are not the end of the story

## Parsing code

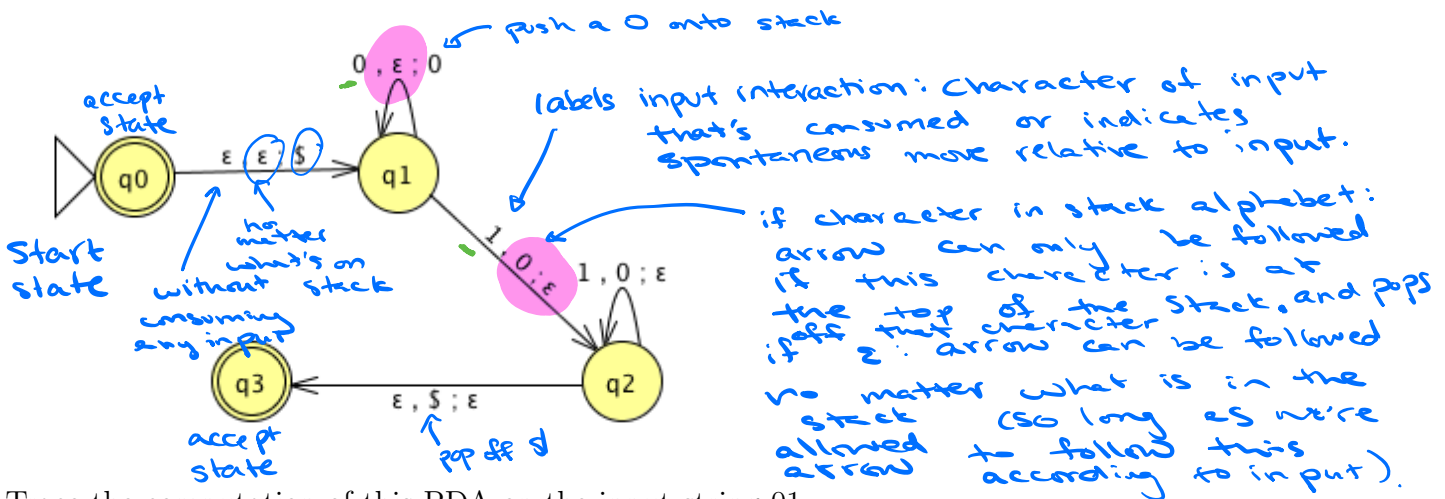
- Many nice / simple / important sets are not regular
- Limitation of the finite-state automaton model: Can't "count", Can only remember finitely far into the past, Can't backtrack, Must make decisions in "real-time"
- We know actual computers are more powerful than this model...

The next model of computation. Idea: allow some memory of unbounded size. How?

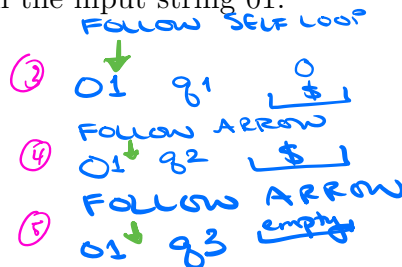
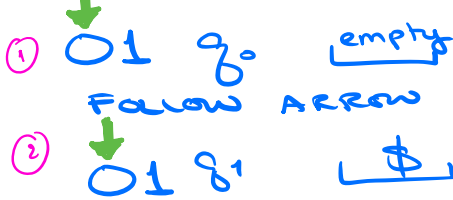
- To generalize regular expressions: context-free grammars



- To generalize NFA: **Pushdown automata**, which is like an NFA with access to a stack: Number of states is fixed, number of entries in stack is unbounded. At each step (1) Transition to new state based on current state, letter read, and top letter of stack, then (2) (Possibly) push or pop a letter to (or from) top of stack. Accept a string iff there is some sequence of states and some sequence of stack contents which helps the PDA processes the entire input string and ends in an accepting state.

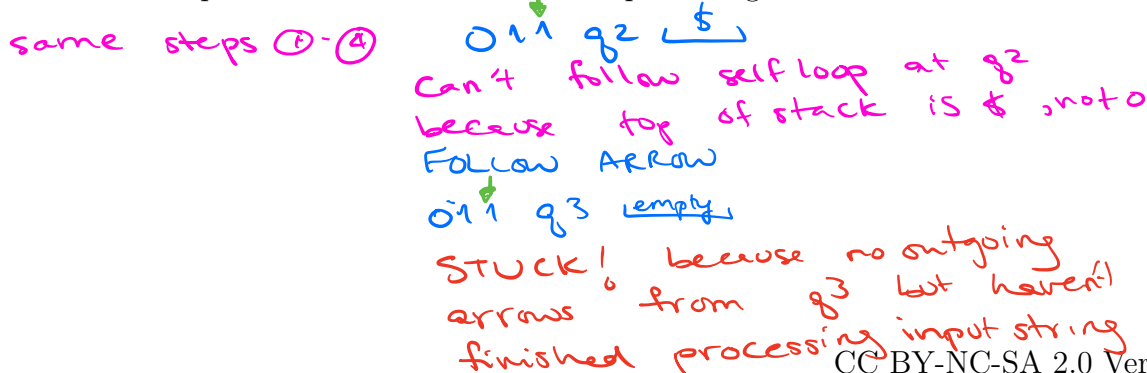


Trace the computation of this PDA on the input string 01.



01 is accepted by PDA

Trace the computation of this PDA on the input string 011.



011 is not accepted by this PDA

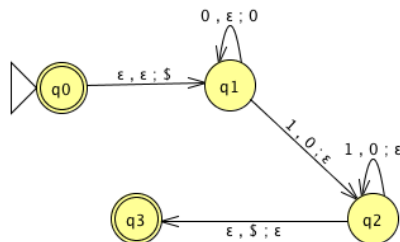
Friday :

**Definition** A **pushdown automaton** (PDA) is specified by a 6-tuple  $(Q, \Sigma, \Gamma, \delta, q_0, F)$  where  $Q$  is the finite set of states,  $\Sigma$  is the input alphabet,  $\Gamma$  is the stack alphabet,

$$\delta : Q \times \Sigma_{\epsilon} \times \Gamma_{\epsilon} \rightarrow \mathcal{P}(Q \times \Gamma_{\epsilon})$$

is the transition function,  $q_0 \in Q$  is the start state,  $F \subseteq Q$  is the set of accept states.

*Formal definition*



Draw the state diagram of a PDA with  $\Sigma = \Gamma$ .

Draw the state diagram of a PDA with  $\Sigma \cap \Gamma = \emptyset$ .

A PDA recognizing the set  $\{ \quad \}$  can be informally described as:

Read symbols from the input. As each 0 is read, push it onto the stack. As soon as 1s are seen, pop a 0 off the stack for each 1 read. If the stack becomes empty and there is exactly one 1 left to read, read that 1 and accept the input. If the stack becomes empty and there are either zero or more than one 1s left to read, or if the 1s are finished while the stack still contains 0s, or if any 0s appear in the input following 1s, reject the input.

State diagram for this PDA:

$$\Sigma = \{a, b, c\}. \quad L = \{a^i b^j c^k \mid i = j \text{ or } i = k, \ i \geq 0, j \geq 0, k \geq 0\}$$

## Review: Week 4 Wednesday

Please complete the review quiz questions on Gradescope about PDA definitions.

**Pre class reading for next time:** Page 102



## Friday April 22

Consider the state diagram of a PDA with input alphabet  $\Sigma$  and stack alphabet  $\Gamma$ .

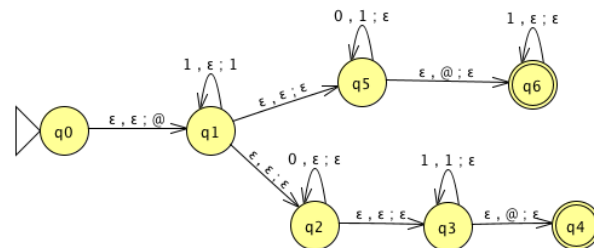
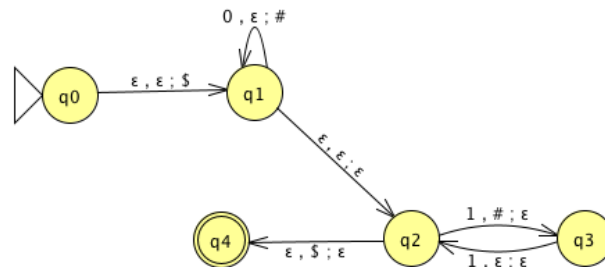
Label	means
$a, b \rightarrow c$ when $a \in \Sigma, b \in \Gamma, c \in \Gamma$	
$a, \varepsilon \rightarrow c$ when $a \in \Sigma, c \in \Gamma$	
$a, b \rightarrow \varepsilon$ when $a \in \Sigma, b \in \Gamma$	
$a, \varepsilon \rightarrow \varepsilon$ when $a \in \Sigma$	

How does the meaning change if  $a$  is replaced by  $\varepsilon$ ?

For the PDA state diagrams below,  $\Sigma = \{0, 1\}$ .

Mathematical description of language

State diagram of PDA recognizing language



$$\{0^i 1^j 0^k \mid i, j, k \geq 0\}$$

Assume  $a \in \Sigma$  and  $L$  is a language over  $\Sigma$ . Recall that

$$aL = \{aw \mid w \in L\}$$

If  $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$  is a PDA with  $L(M) = L$ , a PDA  $M_1$  that recognizes  $aL$  is

$$M_1 = ( \quad , \Sigma, \Gamma, \delta_1, \quad , \quad )$$

with

## Review: Week 4 Friday

Please complete the review quiz questions on Gradescope about PDA construction.