

# Homework 3

## Problem 1: [10 points]

For each of the language below, construct a regular expression that represents this language. All languages are over alphabet  $\Sigma = a, b$ .

In addition to the regular expression itself, provide a brief explanation (1-3 sentences) of the idea behind your construction.

**Hint:** Sometimes you might need to deal with  $\epsilon$  and single-character strings ( $a$  and  $b$ ) as special cases...

(a) The language is  $\{w \mid w \text{ does NOT end with } aa\}$ . For example,  $\epsilon$ ,  $a$ , and  $aaba$  are in the language, while  $baa$  and  $babaa$  are not.

**Solution:**

Answer

**Explanation:**

Answer

(b) The language is  $\{w \mid \text{the first and last characters of } w \text{ are different}\}$ . For example,  $abab$ , and  $bbbbaa$  are in the language, while  $\epsilon$ ,  $a$  and  $bbbbab$  are not.

**Solution:**

Answer

**Explanation:**

Answer

## Problem 2: [5 points]

Let  $L = \{a^m b a^n \mid m \geq n\}$ . For example,  $b$ ,  $aaba$ , and  $aaabaaa$  are in the language, while  $\epsilon$ ,  $aabba$ , and  $abaaa$  are not. Use the pumping lemma to prove that  $L$  is not a regular language over alphabet  $\Sigma = \{a, b\}$ .

**Hint:** If you get stuck, reviewing the last example in Lec 13 might help.

**Solution:**

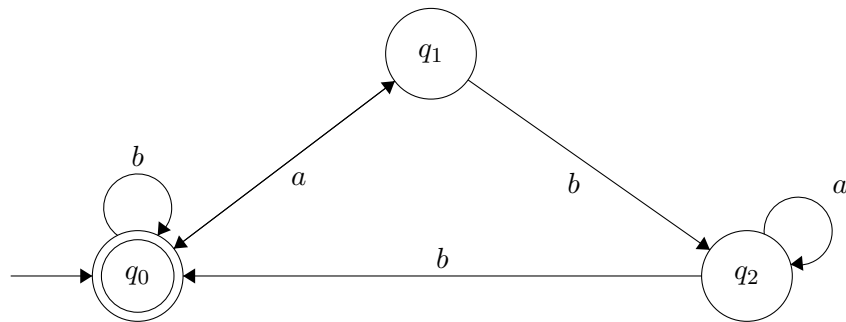
Answer

### Problem 3: [5 points]

Give a regular expression equivalent to the following NFA (in fact it is a DFA).

I encourage you to write down the intermediate steps in addition to your final answer, so that you might get partial credits even if the final answer is wrong. (If your submission only includes the final answer, then it's either 5 points or 0 points.)

**Hint:** As discussed in class, the very first step should be adding a "dummy" starting state and a separate "dummy" accepting state; this is not strictly necessary but it's error-prone if you don't do this.



**Solution:**

Answer