

CSE 311: Foundations of Computing I

Section 8: RegEx, CFG, DFA, NFA Solutions

1. Regular Expressions

- (a) Write a regular expression that matches base 10 non-negative numbers.
(Note that there should be no leading zeroes.)

Solution:

$$0 \cup ((1 \cup 2 \cup 3 \cup 4 \cup 5 \cup 6 \cup 7 \cup 8 \cup 9)(0 \cup 1 \cup 2 \cup 3 \cup 4 \cup 5 \cup 6 \cup 7 \cup 8 \cup 9)^*)$$

- (b) Write a regular expression that matches all binary strings that do not have any consecutive 0's or 1's.

Solution:

$$((01)^*(0 \cup \epsilon)) \cup ((10)^*(1 \cup \epsilon))$$

- (c) Write a regular expression that matches all binary strings that contain the substring "111", but not the substring "000".

Solution:

$$(01 \cup 001 \cup 1^*)^*(0 \cup 00 \cup \epsilon)111(01 \cup 001 \cup 1^*)^*(0 \cup 00 \cup \epsilon)$$

(If you don't want the substring 000, the only way you can produce 0s is if there are only one or two 0s in a row, and they are immediately followed by a 1 or the end of the string.)

2. CFGs

Give CFGs for each of the following languages.

- (a) All binary strings that end in 00.

Solution:

$$S \rightarrow 0S \mid 1S \mid 00$$

- (b) All binary strings that contain at least three 1's.

Solution:

$$S \rightarrow TTT \\ T \rightarrow 0T \mid T0 \mid 1T \mid 1$$

T generates all binary strings with at least one 1.

- (c) All binary strings with the same number of 1s and 0s.

Solution:

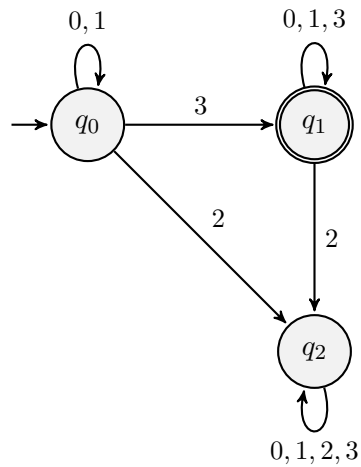
$$\mathbf{T} \rightarrow \mathbf{TT} \mid \mathbf{0T1} \mid \mathbf{1T0} \mid \varepsilon$$

3. DFAs

Construct DFAs to recognize each of the following languages.

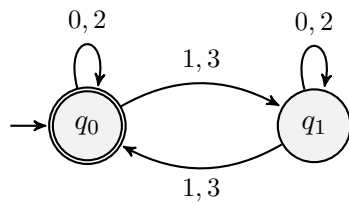
- (a) All strings over the alphabet $\Sigma = \{0, 1, 2, 3\}$ that contain at least one 3 but no 2.

Solution:



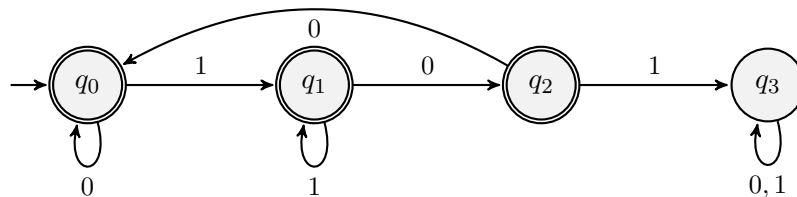
- (b) All strings over the alphabet $\Sigma = \{0, 1, 2, 3\}$ whose digits sum to an even number.

Solution:



- (c) Binary strings that do not contain the substring 101.

Solution:



q_3 : strings that contain 101.

q_2 : strings that don't contain 101 and end in 10.

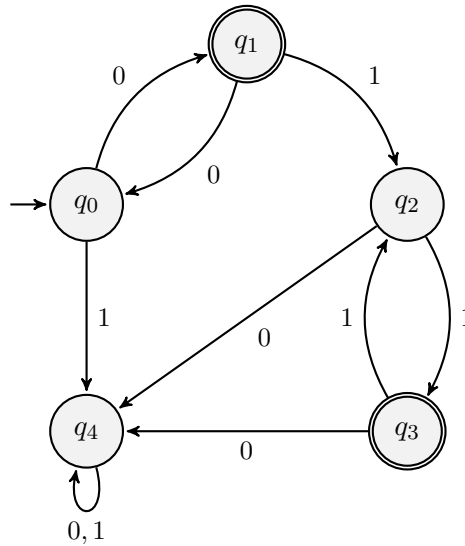
q_1 : strings that don't contain 101 and end in 1.

q_0 : $\varepsilon, 0$, strings that don't contain 101 and end in 00.

- (d) Binary strings that contain an even number of 1s and odd number of 0's and do not contain the substring 10.

Solution:

Note that a binary string not containing 10 as a substring must be of the form 0^*1^* . The additional constraints mean we are looking for numbers of the form 0^x1^y with x odd and y even.



q_4 : strings that are not of the form 0^*1^* or are of the form 0^x11^* with x even

q_3 : strings of the form 0^x1^y with x odd and y even

q_2 : strings of the form 0^x1^y with x and y odd

q_1 : strings of the form 0^x with x odd

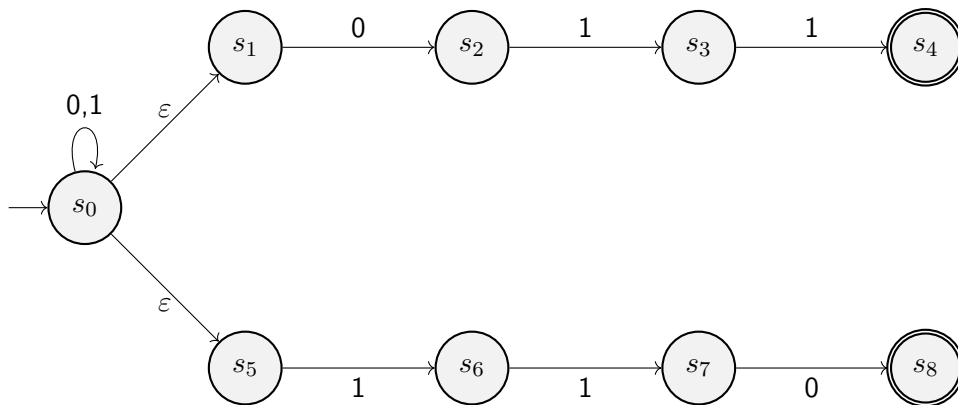
q_0 : strings of the form 0^x with x even

4. NFAs

Construct NFAs to recognize each of the following languages.

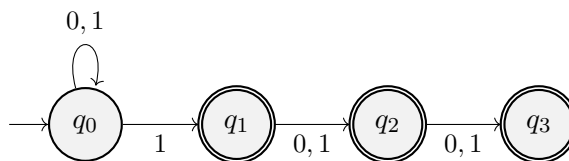
- (a) Binary strings ending in either 011 or 110.

Solution:



- (b) Binary strings that have a 1 as one of the last three digits.

Solution:



5. All the Machines!

Using the alphabet $\Sigma = \{0, 1, 2, 3, 4, 5\}$, define the language L as follows. If x is a string from Σ^* with characters x_0, \dots, x_n , then $x \in L$ iff: for every i between 0 and n , if x_i is an odd digit, then $x_k > x_i$ for every $k > i$. For example, if one of the digits is a 3, every digit after it must be a 4 or higher.

(a) List 5 strings in L and 5 strings not in L .

Solution:

Accepted:

- 145
- 135
- 12425
- 2004
- 2034

Rejected:

- 321
- 11
- 455
- 452
- 2010

(b) Construct a regular expression, a DFA, and a CFG for the language L .

Solution:

Regex: $(0 \cup 2 \cup 4)^*(\epsilon \cup 1)(2 \cup 4)^*(\epsilon \cup 3)4^*(\epsilon \cup 5)$

CFG:

- S** \rightarrow 0**S**|2**S**|4**S**|**A**
- A** \rightarrow 1**B**|**B**
- B** \rightarrow 2**B**|4**B**|**C**
- C** \rightarrow 3**D**|**D**
- D** \rightarrow 4**D**|**E**
- E** \rightarrow 5| ϵ

DFA:

