

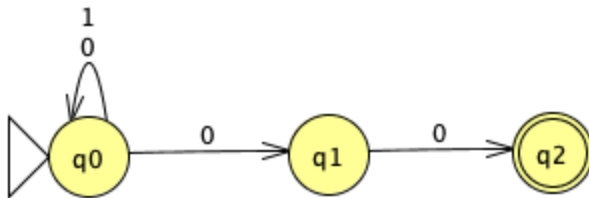
## CSE 355: Intro to Theoretical Computer Science

### Recitation #3 (20 pts)

• Due: Tuesday, Feb. 2, 2021 at 11:59pm Arizona time on Canvas.

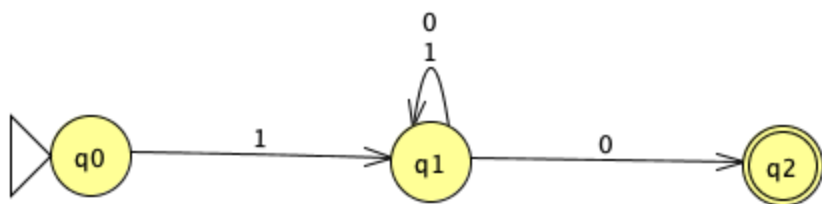
1. [5 pts] Use [JFLAP](http://www.jflap.org) (<http://www.jflap.org>) to draw the state diagram of NFA with **three** states that recognize the following language, assume alphabet  $\Sigma = \{0, 1\}$ . (Save the diagram as .jpg file and paste it here)

$$L = \{\omega \mid \omega \text{ ends in } 00\}$$

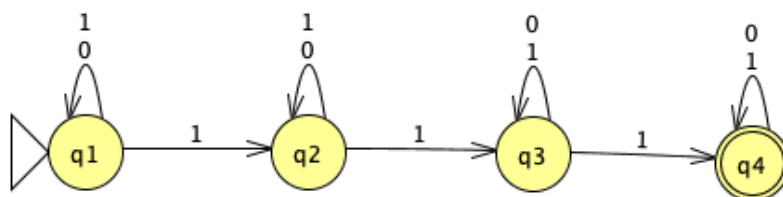


2. [5 pts] Use the construction in the proof of Theorem 1.45 (textbook pp.80) to give the state diagram of NFA that recognize the **union** of the following two languages ( $L_1 \cup L_2$ ). First construct NFAs state diagram using [JFLAP](http://www.jflap.org) (<http://www.jflap.org>) for each individual language, then combine them. Assume alphabet  $\Sigma = \{0, 1\}$ .

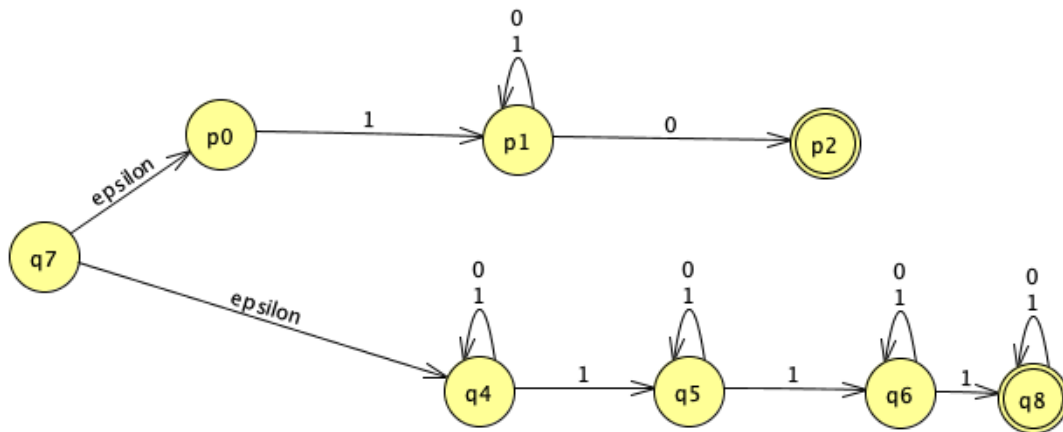
$$L_1 = \{\omega \mid \omega \text{ begins with a 1 and ends with a 0}\}$$



$L_2 = \{\omega \mid \omega \text{ contains at least three 1s}\}$

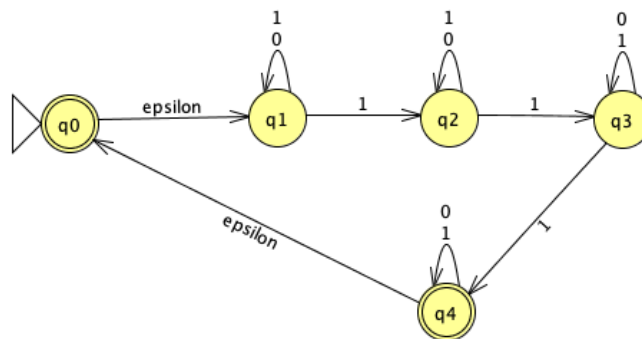


Combine:

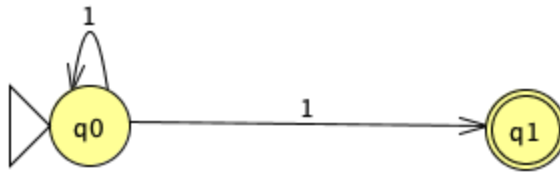


3. [5 pts] Use the construction in the proof of Theorem 1.49 (textbook pp.83) to give the state diagram of NFA that recognize the **star** of the following language ( $L_1^*$ ). Assume alphabet  $\Sigma = \{0, 1\}$ .

$L_1 = \{\omega \mid \omega \text{ contains at least three 1s}\}$



4. [5 pts] Let alphabet  $\Sigma = \{0, 1\}$  and  $L = \{\omega \mid \omega = 1^m 0^n \text{ such that } m \geq 0, n \geq 1\}$  A) [2pts] Draw the state diagram of an NFA with no more than two states that recognize  $L$



B) [3 pts] Use the powerset construction method discussed in class to convert above NFA into a DFA (no need to simplify your DFA), draw its state diagram.

```

graph LR
    start(( )) --> q0((q0))
    q0 -- 0 --> q1((q1))
    q0 -- 1 --> q2(((q2)))
    q1 -- 0 --> q1
    q1 -- 1 --> q2
    q2 -- 1 --> q2
    q2 -- 0 --> q3((q3))
    q3 -- 0 --> q3
    q3 -- 1 --> q2
  
```

**Table Text Size**

Input	Result
10	Accept
110	Accept
1111110	Accept
1100000	Accept
1111110000	Accept
1	Accept
1111	Accept
01	Reject
00000	Reject
0000011	Reject
11100001	Reject
10101010	Reject

||
Load Inputs
Run Inputs
Clear
Enter Lambda
View Trace