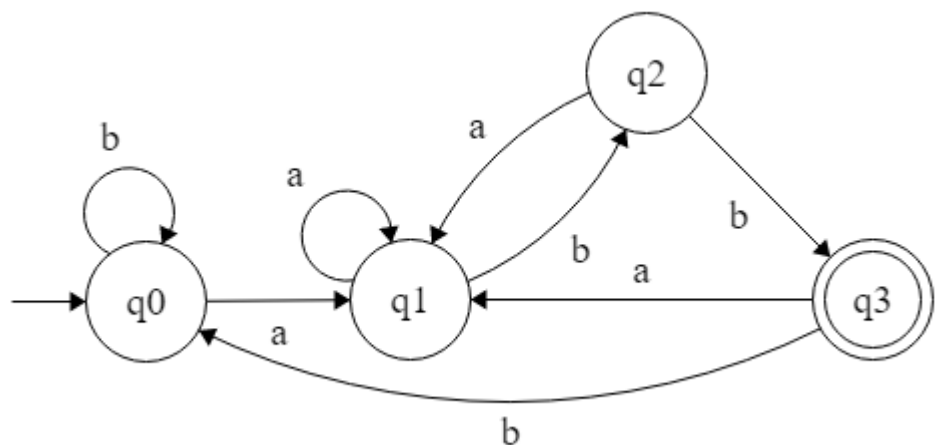


On my honor, I have not given, nor received, nor witnessed any unauthorized assistance on this work.

Print name and sign: _____

Question:	1	2	3	4	Total
Points:	13	8	6	3	30
Score:					



1. Consider the following DFA:

- (a) (4 points) Give two strings (including the shortest string) that this DFA accepts. Then give two strings which are **not** accepted by the DFA.

Solution: Yes: The shortest is abb . Many answers possible for the second string, but some are $aaaabb$, $abababb$, $abbbabb$.

No: Lots of possibilities. ϵ (the shortest, although this wasn't asked for), a , b , aa , ba , $abab$, $aaaa$, $bbbb$,...

- (b) (4 points) Give the formal 5-tuple definition for this DFA.

Solution: $M = \{\{q_0, q_1, q_2, q_3\}, \{a, b\}, \delta, q_0, \{q_3\}\}$

where the transition function δ is described by this transition table.

	a	b
q0	q1	q0
q1	q1	q2
q2	q1	q3
q3	q1	q0

- (c) (2 points) Informally describe the language this DFA recognizes.

Solution: Strings ending in “abb”

- (d) (3 points) Give a regular expression describing the language this DFA accepts.

Solution: $(a|b)^*abb$

2. Consider the alphabet $\Sigma = \{0, 1\}$ and the language, L , represented by the regular expression $0^*(01)^*1^*$. State whether or not each of the following strings is in L .

(a) (2 points) ϵ yes

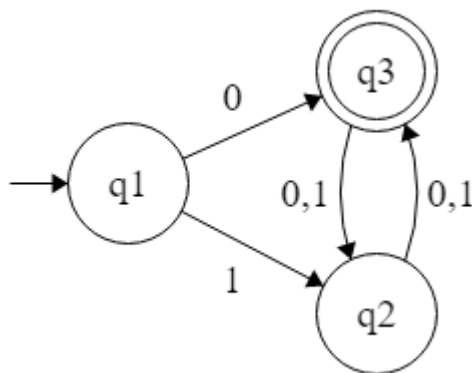
(b) (2 points) 1 yes

(c) (2 points) 01 yes

(d) (2 points) 10 no

3. (6 points) Consider the language of all strings that start with 0 and has odd length **or** start with 1 and has even length.

Draw a DFA for this language



Solution:

4. (3 points) List one similarity and one difference between the programmer version of “regular expressions” and the formal language description “regular expressions”.

Solution: There are some syntax similarities (obvs), but a big similarity is that they're both (fundamentally) based on set theory. E.g. "What is a subset of another set?". They also both use some of the same operators like `|` and `*`. Concatenation is also used, but doesn't have a symbol in programming regular expressions the way it does in (some versions of) formal language expression. There are some syntax differences (obvs), but the biggest is that programming regular expressions have memory (that is, they can remember previous states and results). This makes them significantly more powerful than the regular expressions we use as