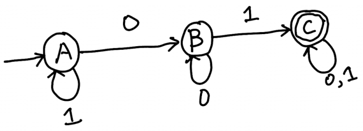


CMSC330 Fall 2025 Quiz 2

Proctoring TA: ______________________ **Name:** ______________________

Section Number: ______________________ **UID:** ______________________

Problem 1: FSM

[Total 1 pts]

Which of the following is not allowed in a DFA? (Select all that apply)

Ⓐ Epsilon transitions from one state to another

Ⓑ Multiple start states

Ⓒ Multiple transitions from a state on the same symbol to different states

Ⓓ Multiple accepting states

Problem 2: Regular Expressions, NFA & DFAs

[Total 19 pts]

(a) Write a regular expression for a strong password rule: [3 pts]

- at least 5 characters
- password length must be an odd number.
- The password must contain only lowercase letters, digits, and special characters #%^&.
- the first 2 characters must be one lowercase letter and one digit in any order.

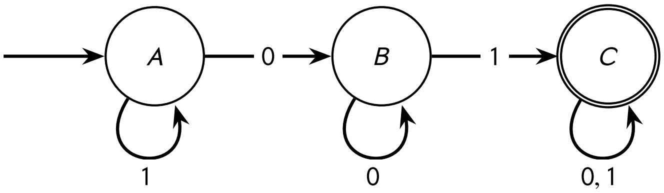
Examples:

```
Good passwords: a1&11    1a&11$1   2b&1abc1$   c4123    2bbbbbc
Bad passwords:  aaa1#234   123ab  a112 11abc a2cd34
```

(b) Draw an NFA that accepts the strings that end with "xxyy" over the alphabet $\Sigma = \{x, y\}$. Note that this must accept any string that **ends with** this rather than just that exact string. [5 pts]

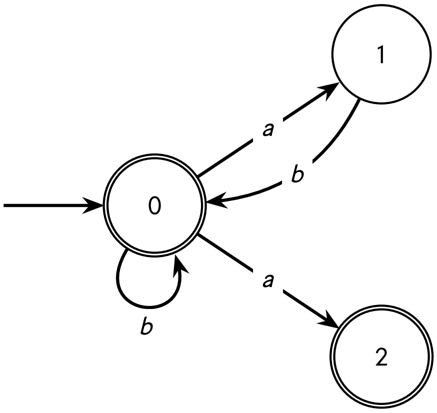
(c) Convert the following regular expression into a NFA: `(y|(z?))x*` [5 pts]

(d) Which regular expression represents the language accepted by this DFA? [1 pts]



Ⓐ `1*0+1(0|1)*` Ⓑ `0*1*` Ⓒ `10(0|1)*` Ⓓ `1*(0|1)*`

(e) Consider the following NFA: [5 pts]



(1 point each) For each string below, circle "accept" if the string is accepted by the NFA, and "reject" if it is rejected.

	Accept	**Reject**
bbbaa	Ⓐ	Ⓡ
ba	Ⓐ	Ⓡ
b	Ⓐ	Ⓡ
babb	Ⓐ	Ⓡ
ϵ	Ⓐ	Ⓡ

Regex

*	zero or more repetitions of the preceding character or group
+	one or more repetitions of the preceding character or group
?	zero or one repetitions of the preceding character or group
.	any character
$r_1 \mid r_2$	r_1 or r_2 (eg. a|b means ’a’ or ’b’)
`[abc]`	match any character in `abc`
[^r_1]	anything except r_1 (eg. [^abc] is anything but an ’a’, ’b’, or ’c’)
[r_1-r_2]	range specification (eg. [a-z] means any letter in the ASCII range of a-z)
{n}	exactly n repetitions of the preceding character or group
{n,}	at least n repetitions of the preceding character or group
{m,n}	at least m and at most n repetitions of the preceding character or group
^	start of string
$	end of string
(r_1)	capture the pattern r_1 and store it somewhere (match group in Python)
\d	any digit, same as [0-9]
\s	any space character like \n, \t, \r, \f, or space