Full Name: Chris Tu Answer Key
NetID:

Midterm Make-up Exam
CS 314, Fall 2024
Section 01-04

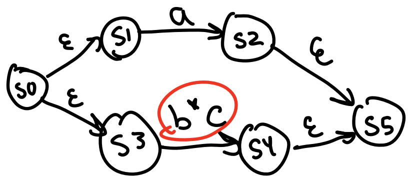
1 (20 pts) RE to NFA, and NFA to DFA

Give credit for matching the a, the OR, or the b*

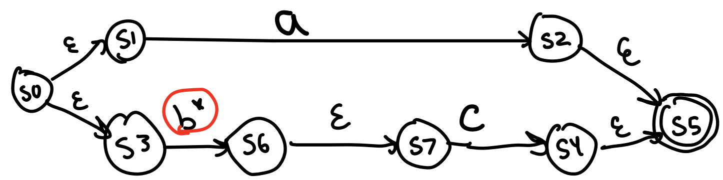
a | b^*c

(a) Use Thompson's construction to construct an NFA for the regular expression above.

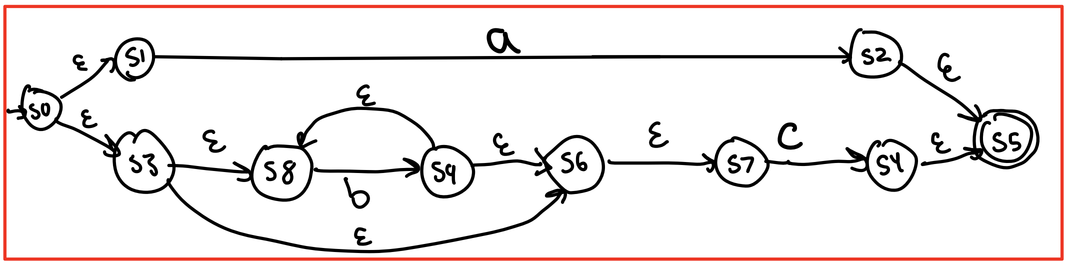
a|b*c =



=

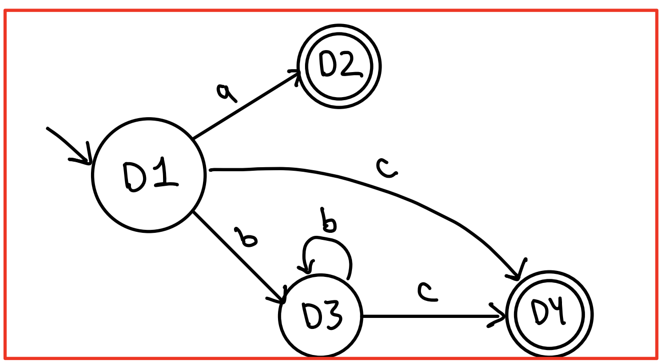


=



(b) Convert the NFA to a DFA using subset construction.

ε - closure (s0) = {s0, s1, s3, s8, s6, s7}



		a	b	c
D1	{s0, s1, s3, s8, s6, s7}	D2 {s2, s5}	D3 {s9, s8, s6, s7}	D4 {s4, s5}
D3	{s9, s8, s6, s7}	X	{s9, s8, s6, s7} D3	{s4, s4} D4

(c) Minimize the DFA. If the DFA is already minimal, justify it.

P1 = D1, D3 P2 = D2 D4

(D1, a) → D2 in P2
(D1, b) → D3 in P1
(D1, c) → D4 in P2

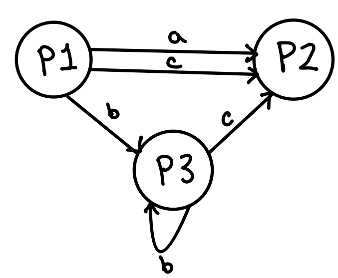
D1 and D3 must be split

(D3, a) → error
(D3, b) → D3 in P1
(D3, c) → D4 in P2

P1 = D1
P2 = D2 D4
P3 = D3

No further partitions can be made.

∴



is the minimal DFA.

2 (20 pts) Context Free Grammars

(a) Write a grammar for $a^x b^y$ where $y = x + 2$ and $x \geq 0$ (i.e., exactly 2 more b's than a's).

$\langle S \rangle = \langle A \rangle bb$

$\langle A \rangle = a \langle A \rangle b \mid \varepsilon$

(b) Consider the following grammar (where $< S > =$ start symbol):

$< S > \rightarrow (< A >) \mid \epsilon$

$< A > \rightarrow < A >; < S > \mid c$

Present a derivation for the string **(c;(c;))** and give the parse tree.

$\langle S \rangle$

$\Rightarrow_L (\langle A \rangle)$

$\Rightarrow_L (\langle A \rangle ; \langle S \rangle)$

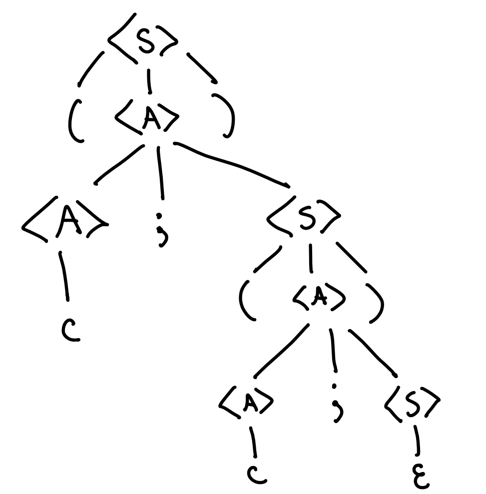
$\Rightarrow_L (c ; \langle S \rangle)$

$\Rightarrow_L (c;(\langle A \rangle))$

$\Rightarrow_L (c;(\langle A \rangle ; \langle S \rangle))$

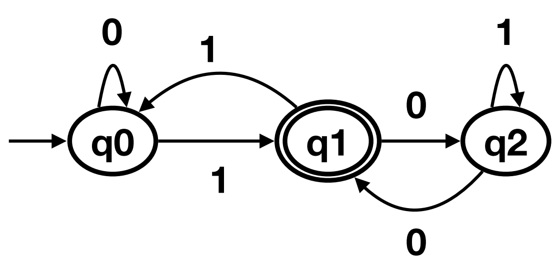
$\Rightarrow_L (c;(c;\langle S \rangle))$

$\Rightarrow_L (c;(c;))$



3 (20 pts) Simulating DFA

Consider the DFA below, determine whether the given strings can be accepted or rejected. $q0$ is the start state, and $q1$ is the accepting state.



(a) 10 (Accept or Reject?)

(b) 111 (Accept or Reject?)

(c) 1101 (Accept or Reject?)

Describe in English what the strings accepted by this DFA have in common.

- Binary numbers starting at 1 and incrementing by 3.
- All binary multiples of 3 plus 1

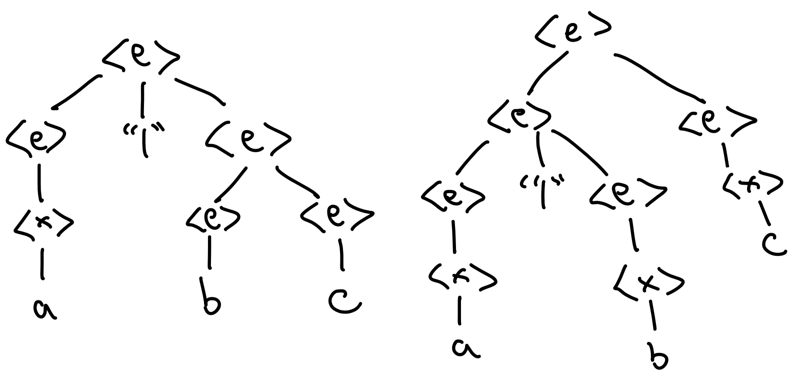
4 (40 pts) Unambiguous Grammar

Consider the following grammar that attempts to describe a regular expression:

1. $< e > \quad ::= < x > \mid \epsilon$
2. $< e > \quad ::= < e >< e >$ ✓
3. $< e > \quad ::= < e > *$ ✓
4. $< e > \quad ::= < e >$ "|" $< e >$ ✓
5. $< e > \quad ::= (< e >)$ ✓
6. $< x > \quad ::=$ a | b | c ... y | z

(a) Show that the above grammar is ambiguous.

a|bc



Two distinct parse trees for the same String

∴ ambiguous

(b) Recall that Kleen Closure (*) has highest precedence, followed by concatenation, and then alternation ("|"). Let's assume concatenation, alternation and Kleen closure are all left associative. Rewrite the grammar such that it is unambiguous and the precedence/associativity rules are enforced. Note that double quotes are added for the actual alternation operator to distinguish from the | sign that separates production rules.

Idea: Higher Precedence operations are nested deeper in grammar

precendence: ( ) > Kleene > concat > Union

<S> ::= <A>

lowest precedence: alternation/union – <A> ::= <A> "|" <C> | <C>

concat – <C> ::= <C><K> | <K>

Kleene – <K> ::= <K>* | <P>

highest precedence: parenthesis – <P> ::= (<S>) | <x>

<x> ::= a | b | c ... y | z