COT4210 Exam I

Noah Fuhrman

March 7, 2025

Solution to Problem 1

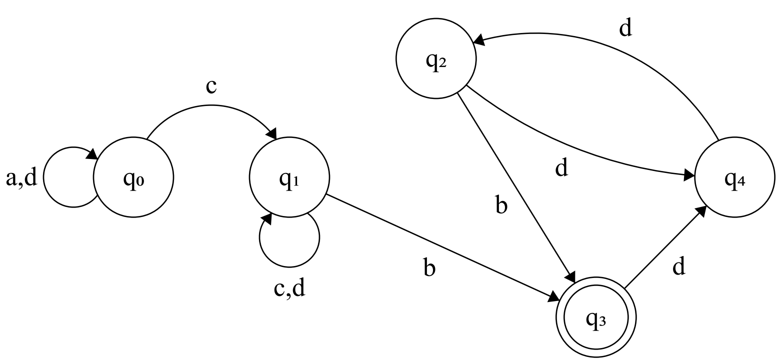
Step 1: Construct an NFA for the language L

- Let $M = (Q_c, \Sigma, \delta, q_0, F)$ be an NFA that recognizes language L where...
  - $Q = \{q_0, q_1, q_2, q_3, q_4\}$ is the set of states
  - $\sum = \{a, b, c, d\}$
  - δ :

	a	b	c	d
q_0	q_0	$\emptyset$(no transition)	q_1	q_0
q_1	$\emptyset$(no transition)	q_3	q_1	q_1
q_2	$\emptyset$(no transition)	q_3	$\emptyset$(no transition)	q_4
q_3	$\emptyset$(no transition)	$\emptyset$(no transition)	$\emptyset$(no transition)	q_4
q_4	$\emptyset$(no transition)	$\emptyset$(no transition)	$\emptyset$(no transition)	q_2

  - q_0 is the start state
  - $F = \{q_3\}$ is the set of accept states

Step 2: Visualize the NFA



Step 3: Prove that the machine accepts L

- By definition, a language is called a regular language if some finite automaton recognizes it.
- Since we have already constructed an NFA that recognizes L, we have shown that L is regular. (end of solution)

Solution to Problem 2

Step 1: Establish Conditions

- L is a regular language
- h is a homomorphism on the alphabet of L
- Since L is regular, there must exist an NFA $M = (Q, \sum, \delta, q_0, F)$ that recognizes L

Step 2: Construct an NFA that accepts $h(L)$

- Let $M' = (Q', \sum', \delta', q_0', F')$ be the NFA that recognizes $h(L)$
  - Q' includes all states from Q plus new intermediate states needed for longer transitions (the strings)
  - $\sum'$ is the alphabet of $h(L)$
  - $q_0' = q_0$ (same start state)
  - $F' = F$ (same accept states)
  - For each transition $\delta(p, a) = q$ in the original NFA:
    - If $h(a) = \epsilon$ (empty string): Add an ϵ-transition from p to q
    - If $h(a) = w_1 w_2 ... w_k$ (k ≥ 1): Create a path of k transitions with k-1 new intermediate states that reads the string $h(a)$ character by character.
- The resulting NFA M' accepts a string y if and only if y = h(s) for some $s \in L$

Step 3: Conclusion

- Since M' is an NFA and NFAs recognize regular languages, $h(L)$ must be regular
- Therefore, the class of regular languages is closed under homomorphism

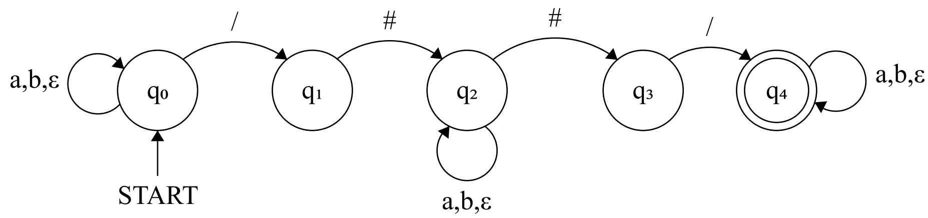
Solution to Problem 3

Step 1: Construct an NFA with at most five states that accepts any string containing a comment (L)

- Let $M = (Q, \Sigma, \delta, q_0, F)$ be an NFA that recognizes language L where...
  - $Q = \{q_0, q_1, q_2, q_3, q_4\}$ is the set of states
  - $\sum = \{a, b, /, \#\}$
  - δ :

	a	b	/	#	ϵ
q_0	q_0	q_0	q_1	$\emptyset$(no transition)	q_0
q_1	$\emptyset$(no transition)	$\emptyset$(no transition)	$\emptyset$(no transition)	q_2	$\emptyset$(no transition)
q_2	q_2	q_2	$\emptyset$(no transition)	$\emptyset$(no transition)	q_2
q_3	$\emptyset$(no transition)	$\emptyset$(no transition)	q_4	$\emptyset$(no transition)	$\emptyset$(no transition)
q_4	q_4	q_4	$\emptyset$(no transition)	$\emptyset$(no transition)	q_4

Step 2: Visualize the NFA



Step 3: Prove that the machine accepts L

- By definition, a language is called a regular language if some finite automaton recognizes it.
- Since we have already constructed an NFA that recognizes L, we have shown that L is regular. (end of solution)

Solution to Problem 4

Step 1: Understand the Language

- The language L is defined as $L = \{ccc\#cc\#c$ with c $\in \{a, b\}^*\}$
- This means that the language consists of strings that follow the pattern $ccc\#cc\#c$ where each c is the same string from $\{a, b\}^*$ (including the empty string).

Step 2: Assume L is regular so that we can use the pumping lemmma

- Let p be the pumping length given by the pumping lemma
- Let $c = a^p$ (a string of p consecutive 'a's)
- Let $s = a^p a^p a^p \# a^p a^p \# a^p$
- $s \in L$ since it follows the pattern $ccc\#cc\#c$ where each $c = a^p$
- $|s| = 6p + 2 \geq p$, so the pumping lemma applies

Step 3: Apply Rule 1 and 3 of the pumping lemma

- Since $|s| \geq p$, we can split s into three substrings $s = xyz$ where $|y| > 0$, $|xy| \leq p$, and x, y, z are substrings of s
- Since $|xy| \leq p$, both x and y must be contained within the first block of a's

Step 4: Apply and Evaluate Rule 2 of the pumping lemma

- By rule 2, the middle portion y must be repeatable $\rightarrow xy^i z \in L$ for all $i \geq 0$
- Let $i = 2$
- Let $s' = xy^2 z$

- $x = a^j$, $0 \le j < p$
- $y = a^k$, $k > 0$, and $j + k \le p$
- $z = a^{p-j-k} a^p a^p \# a^p a^p \# a^p$
  - xyz still follows the pattern $ccc\#cc\#c$ where xyz = $a^p a^p a^p \# a^p a^p \# a^p$
- $s' = a^j \cdot a^k \cdot a^k \cdot a^{p-j-k} a^p a^p \# a^p a^p \# a^p$
- $= a^{p+k} a^p a^p \# a^p a^p \# a^p$

- For s' to be in L, it must follow the pattern $ccc\#cc\#c$ where each c is the same string
- However, the first $c = a^{p+k}$ while all other c's are still a^p
- Since $k > 0$, we have $a^{p+k} \neq a^p$
- Therefore, $s' \notin L$

Step 5: Conclude

- We've shown that $xy^2z \notin L$, which contradicts the pumping lemma's requirement that $xy^iz \in L$ for all $i \ge 0$
- Therefore, L is not regular

Solution to Problem 5

Step 1: Determine the quantity of states of the DFA

- There are 3 states in the NFA, so there will be $2^3 = 8$ states in the DFA.
- We would have the following states in the DFA:
  - 1: state 1
  - 2: state 2
  - 3: state 3
  - 12: state 1, 2
  - 13: state 1, 3
  - 23: state 2, 3
  - 123: state 1, 2, 3
  - $\emptyset$: empty set

Step 2: Determine the Accept States of the DFA

- Given that the NFA accepts at states 1 and 3, the DFA accepts at any of the states that include 1 or 3
- Therefore, the accept states of the DFA are:
  - 1: state 1
  - 3: state 3
  - 13: state 1,3
  - 123: state 1,2,3

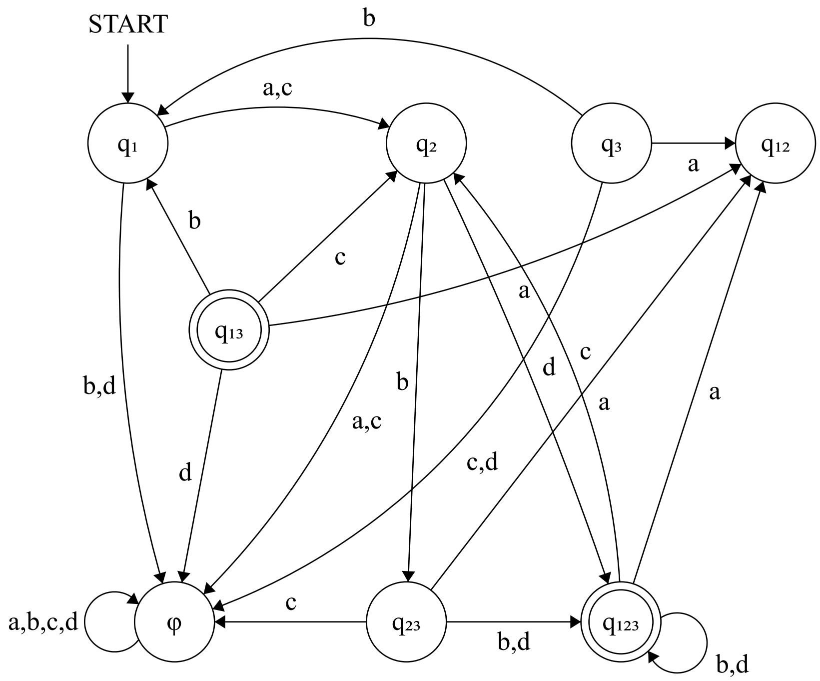
Step 3: Determine the starting state and transitions of the DFA

- Given that state 1 is the starting state and there are no ϵ-transitions from state 1, the starting state of the DFA is {1}
- The transitions of the DFA are as follows:

	a	b	c	d
{1}	{2}	$\emptyset$	{2}	$\emptyset$
{2}	$\emptyset$	{23}	$\emptyset$	{123}
{3}	{12}	{1}	$\emptyset$	$\emptyset$
{1,2}	{2}	{23}	{2}	{123}
{1,3}	{12}	{1}	{2}	$\emptyset$

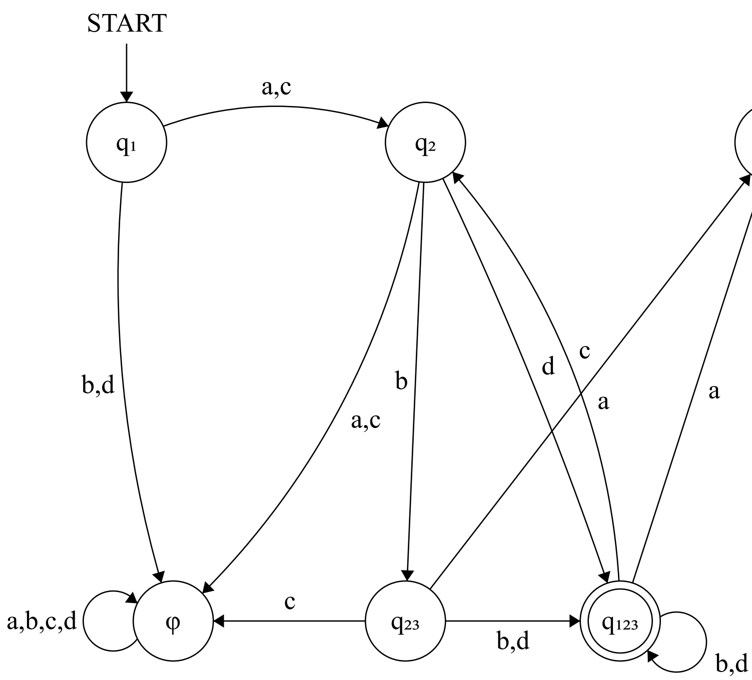
{2,3}	{12}	{123}	∅	{123}
{1,2,3}	{12}	{123}	{2}	{123}
∅	∅	∅	∅	∅

Step 4: Visualize the DFA



Solution to Problem 6

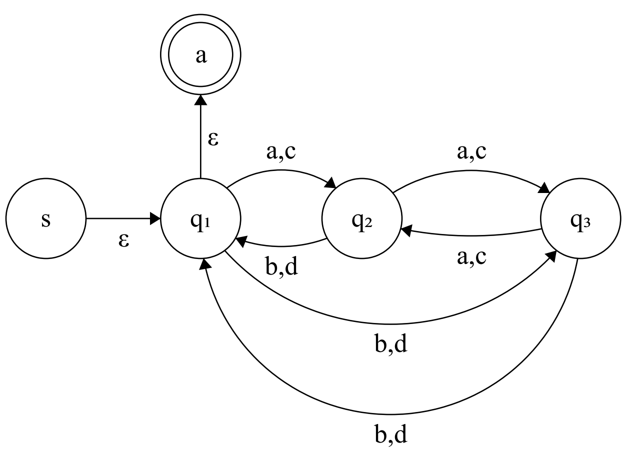
- To reduce the DFA, we need to remove states that are not reachable from the starting state.
- The only states that cannot be reached from the starting state are q_3 and q_{13}



Solution to Problem 7

Step 1: Determine start and accept states of GFNA

- The start state s will come before q_1 (old start state) with a ϵ-transition in-between them
- accepting state a will be above q_1 (old accepting state) with a ϵ-transition in-between them



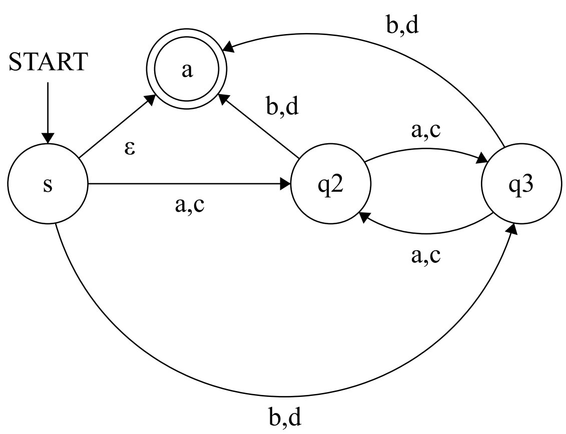
Step 2: Choose a state to rip and construct the ripping table

- Given that q_1 has the most transitions, we'll rip it (q_1 will be our r)

- There are five transitions coming in and out of q_1 (including the ϵ-transitions between s and a)

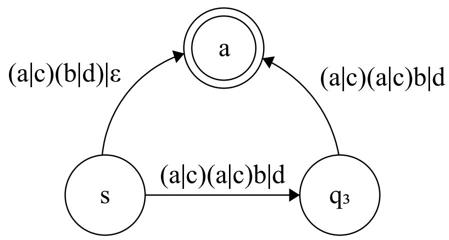
a	r	b	R_{ar}	R_r	R_{rb}	R_{ab}	$(R_{ar})(R_r)^*(R_{rb}) \mid (R_{ab})$	Reduced
s	1	a	ϵ	$\emptyset$	ϵ	$\emptyset$	$(\epsilon)(\emptyset)^*(\epsilon) \mid (\emptyset)$	ϵ
s	1	2	ϵ	$\emptyset$	$a\|c$	$\emptyset$	$(\epsilon)(\emptyset)^*(a\|c) \mid (\emptyset)$	$a\|c$
s	1	3	ϵ	$\emptyset$	$b\|d$	$\emptyset$	$(\epsilon)(\emptyset)^*(b\|d) \mid (\emptyset)$	$b\|d$
2	1	a	$b\|d$	$\emptyset$	ϵ	$\emptyset$	$(b\|d)(\emptyset)^*(\epsilon) \mid (\emptyset)$	$b\|d$
3	1	a	$b\|d$	$\emptyset$	ϵ	$\emptyset$	$(b\|d)(\emptyset)^*(\epsilon) \mid (\emptyset)$	$b\|d$

Step 3: Repair Machine After Ripping q_1



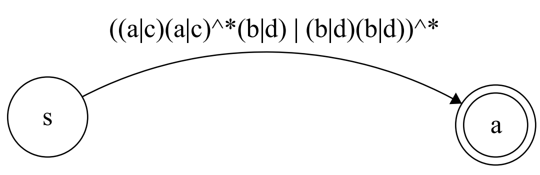
Step 4: Rip/Repair q_2

a	r	b	R_{ar}	R_r	R_{rb}	R_{ab}	$(R_{ar})(R_r)^*(R_{rb}) \mid (R_{ab})$	Reduced
s	2	a	$a\|c$	$\emptyset$	$b\|d$	ϵ	$(a\|c)(\emptyset)^*(b\|d) \mid (\epsilon)$	$(a\|c)(b\|d)\|\epsilon$
s	2	3	$a\|c$	$\emptyset$	a|c	$b\|d$	$(a\|c)(\emptyset)^*(a\|c) \mid (b\|d)$	$(a\|c)(a\|c)\|b\|d$
3	2	a	$a\|c$	$\emptyset$	$b\|d$	$b\|d$	$(a\|c)(\emptyset)^*(b\|d) \mid (b\|d)$	$(a\|c)(b\|d)\|b\|d$



Step 5: Rip/Repair q_3

a	r	b	R_{ar}	R_r	R_{rb}	R_{ab}	$(R_{ar})(R_r)^*(R_{rb}) \mid (R_{ab})$	Reduced
s	3	a	$(a \mid c)(a \mid c) \mid b \mid d$	$\emptyset$	$(a \mid c)(b \mid d) \mid b \mid d$	$(a \mid c)(b \mid d) \mid \epsilon$	$((a \mid c)(a \mid c) \mid b \mid d)(\emptyset)^*((a \mid c)(b \mid d) \mid b \mid d) \mid ((a \mid c)(b \mid d) \mid \epsilon)$	$((a \mid c)(a \mid c)^*(b \mid d) \mid (b \mid d)(b \mid d))^*$



Step 6: Convert to Regular Expression (Final Answer)

- The regular expression is: $((a|c)(a|c)^*(b|d) \mid (b|d)(b|d))^*$