BU CS 332 – Theory of Computation

Lecture 9:

- Midterm I review

Reading:

Sipser Ch 0-2.3

Mark's new office hours:
Th 5-6
Fr 9:30-10:30 (MCS 114)

Mark Bun

February 19, 2020

Midterm I Topics

Deterministic FAs (1.1)

- Given an English or formal description of a language L, draw the state diagram of a DFA recognizing L (and vice versa)
- Know the formal definition of a DFA (A DFA is a 5 tuple…) and convert between state diagram and formal description
- Know the formal definition of how a DFA computes
- Regular operations: Union, concatenation, star and closure of regular languages under regular operations, construction for closure under complement
  - Cross-product construction for union/intersection

Nondeterministic FAs (1.2)

- Given an English or formal description of a language L, draw the state diagram of an NFA recognizing L (and vice versa)
- Know the formal definition of an NFA
- Know the power set construction for converting an NFA to a DFA
- Proving closure properties: Know the constructions for union, concatenation, star
- Recall other closure properties: reverse, intersection, complement

Regular Expressions (1.3)

- Given an English or formal description of a language L, construct a regex generating L (and vice versa)
- Formal definition of a regex
- Know how to convert a regex to an NFA
- Know how to convert a DFA/NFA to a regex

Non-regular Languages (1.4)

- Know the proof ideas for the pumping lemma for regular languages
- Understand the statement of the pumping lemma and how to apply it
- Beyond the pumping lemma: Showing languages are non-regular by combining pumping lemma with closure properties

Context-free Grammars (2.1)

- Given an English or formal description of a language L, give a CFG (in Backus-Naur form) generating L (and vice versa)
- Formal definition of a CFG (A CFG is a 4-tuple…), context-free languages
- Parse trees, derivations

- You are not responsible for the material on ambiguity in parsing and Chomsky Normal Form

  But these are interesting! Read about them if you have time

Pushdown Automata (2.2)

- Given an English or formal description of a language L, describe a PDA recognizing L in terms of:
  - An algorithmic description of the machine
  - A state diagram for the machine
  - (and vice versa)

- Formal definition of a PDA
- Know that PDAs recognize the context-free languages. You are not responsible for knowing the proof.
- Closure properties of CFLs: Regular operations and intersection with regular languages, but not complement or intersection

Non-context-free Languages (2.3)

- Know the proof ideas for the pumping lemma for CFLs
- Understand the statement of the pumping lemma and how to apply it
- Beyond the pumping lemma: Showing languages are non-context-free by combining pumping lemma with closure properties

You are not responsible for Chapter 2.4 on deterministic CFLs (But read this if you're interested in how CFLs are parsed in real compilers, etc.)

Exam Tips

Study Tips

- Review problems from HW 0-3, discussion sections 1-3, solved exercises/problems in Sipser, and suggested exercises on the homework
  - We will literally ask you a question from the homework exercises, so make sure you understand these

Exercises Please practice on exercises and solved problems in Chapter 1 and on the exercise below. The material they cover may appear on exams.

1. (**Conversion procedures**) Use asymptotic (big-O) notation to answer the following questions. Provide brief explanations.
   (a) Let N be an NFA that has n states. If we convert N to an equivalent DFA M using the procedure we described, how many states would M have?
   (b) Let M be a DFA that has n states. If we convert M to an equivalent regular expression R using the procedure we described, how many symbols would R have in the worst case?

- You may bring a page of notes to the exam. Preparing this note sheet is a great way to study. Name on note sheet

Study Tips

- Make sure you know how to solve the problems on the practice midterm and are familiar with the format. The format/length of the real midterm will be very similar.

1) T/F questions
3) CFLs
2) Regular languages
4) Short-Answers

- If you need more practice, there are lots of problems in the book. We're happy to talk about any of these problems in office hours.

For the exam itself

- You may cite without proof any result…
  - Stated in lecture
  - Stated and proved in the main body of the text (Ch. 0-2.3)
  - These include worked-out examples of state diagrams, regexes, CFGs, non-regular/non-CF languages

- Not included above: homework problems, discussion problems, (solved) exercise/problems in the text

  Don't use the Myhill-Nerode theorem

- Showing your work / explaining your answers will help us give you partial credit

Practice Problems

How do we combine pumping w/ closure properties?

Goal: show language L is non-regular

(But boo! L satisfies PL. what do??)

Assume for the sake of contradiction that L is regular

1) Apply closure properties

$$L \xrightarrow{\text{star}} L_1 \xrightarrow{\text{complement}} L_2 \xrightarrow{\text{intersection w/ regular lA}} L_3$$

L_3 is regular

2) Apply PL to show L_3 not regular ✱

Example $L = \{0^m 1^n \mid m \neq n\}$ $B = L \cup A$

know: $A = \{0^n 1^n \mid n \geq 0\}$ not regular
$B = \{w \mid$ all 0s in w come before all 1s$\} \subseteq \{0,1\}^*$

Assume for contradiction that L is regular

1) Use regular operations to turn L into A

$\overline{L} = \{0^n 1^n \mid n \geq 0\} \cup \overline{B}$ (regular)

$A = \overline{L} \cap B$ $= B \setminus L$

(Can obtain by from L using closure properties + fact that B has a DFA (and is hence regular))

2) Use PL (or cite that we did in class) to conclude A is not regular. ⚡

Could it be the case? A non-regular, B non-regular

L regular

$A = \overline{L} \cap B$

$L = \emptyset$ $A = B = \{0^n 1^n \mid n \geq 0\}$

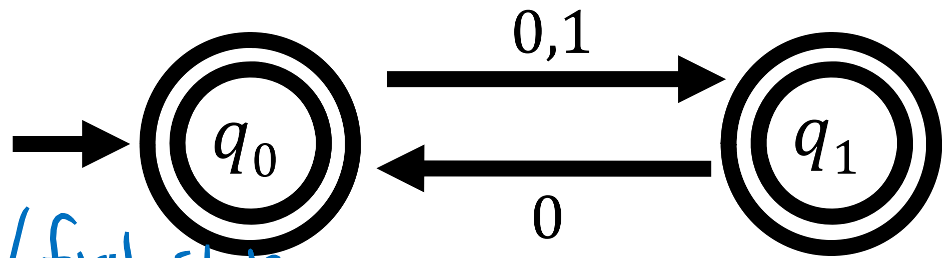
Regular Languages

Name six operations under which the regular languages are closed

Prove or disprove: The non-regular languages are closed under union

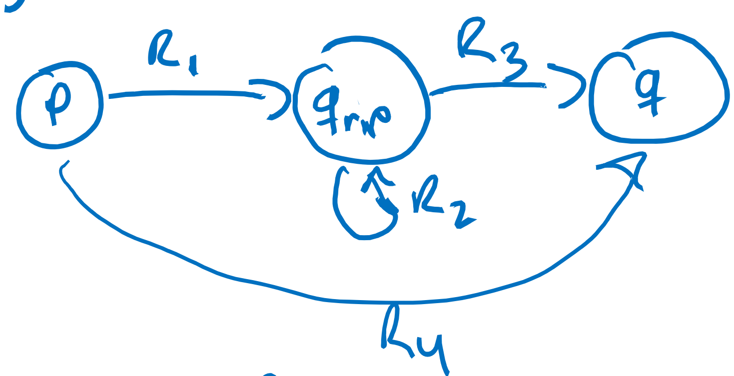
Give the state diagram of an NFA recognizing the language (01 U 10)*

Give an equivalent regular expression for the following NFA

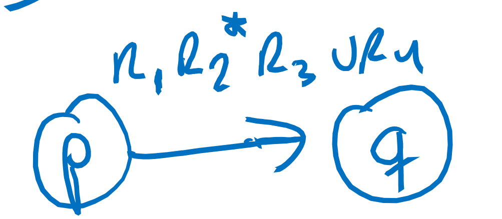


1) Convert to a GNFA by adding new start/final state

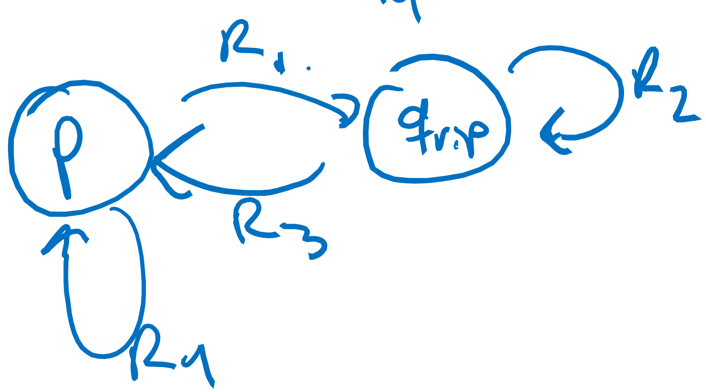
2) Repeatedly remove states and replace w/ regexes



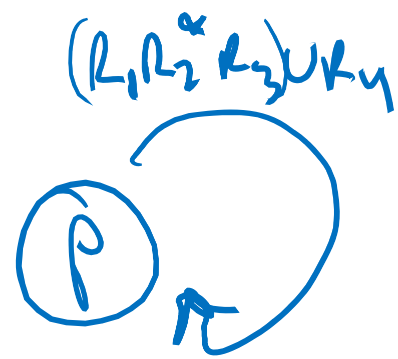
replace with



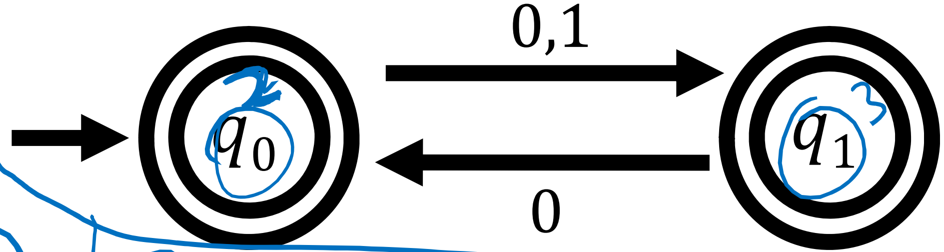
p & q might be the same

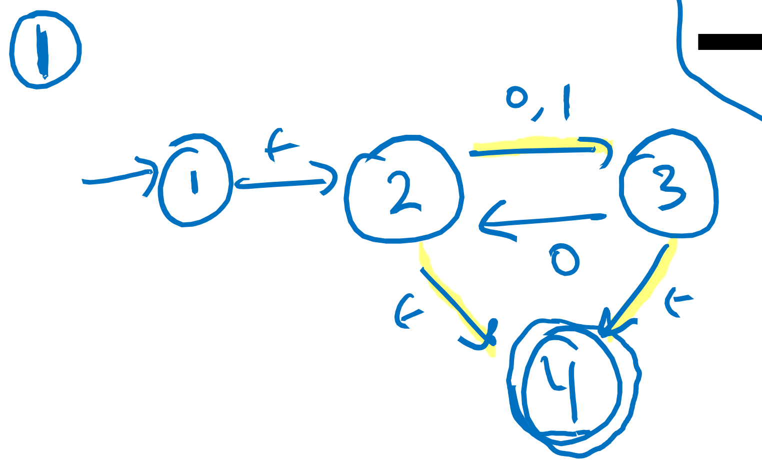


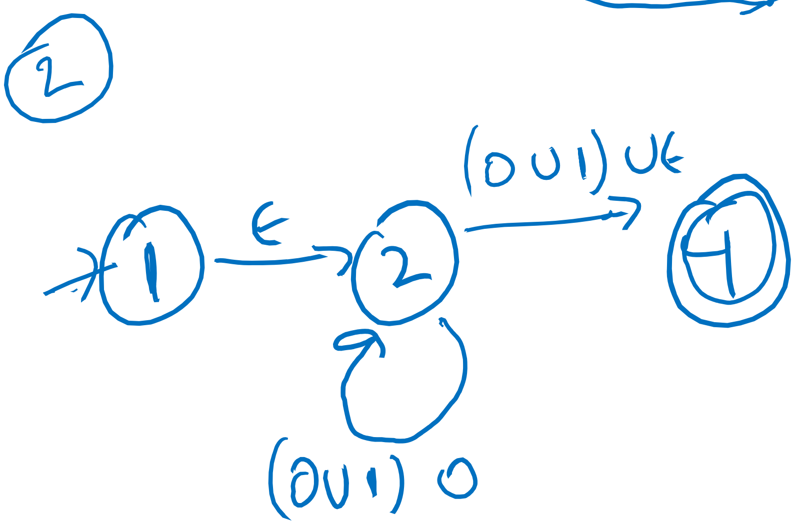
replace with



Give an equivalent regular expression for the following NFA







③ remove state 2

$\epsilon((0\cup 1)0)^*((0\cup 1)\cup\epsilon)$

$= ((0\cup 1)0)^*(0\cup 1\cup\epsilon)$

Final regex:

$((0\cup 1)0)^*(0\cup 1\cup\epsilon)$

"0-1 strings where every even position is 0"

a0b0c0 ... z ← can be empty

Let R be a regular expression with n symbols. If we convert R into an NFA using the procedure described in class, how many states could it have in the worst case?

Is the following language regular?

$$\{a^n a^n | n \geq 0\}$$

Is the following language regular?

$$\{0^n1^n | 0 \leq n \leq 2020\}$$

Let $L = \{w \in \{0,1\}^* \mid w \text{ has the same number of 0s and 1s}\}$.
Let p be a pumping length and $s = (01)^p$.
Give a decomposition of $s = xyz$ which **can** be pumped in L.
Is L regular?

Context-Free Languages

Name three operations under which the context-free languages are closed.

Name two operations under which the CFLs are *not* closed

What language is generated by the CFG

$$S \rightarrow aSb \mid bY \mid Ya$$
$$Y \rightarrow bY \mid aY \mid \varepsilon \quad ?$$

See Piazza for the conclusion

$\{a^n c b^n \mid n \geq 0\}$

$\{\varepsilon, b(a \cup b)^* \cup (a \cup b)^* a\}$ Generates every $\{a,b\}$-string

abab

$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaaSbbb \ldots$

$\overbrace{aaa\ldots a}^{n}$ b ababbb $\underbrace{b\ldots b}_{n}$ $\Rightarrow$ aaa bY bbb

Starts w/ a's, ends w/ b's, # trailing b's ≥ # leading a's

or

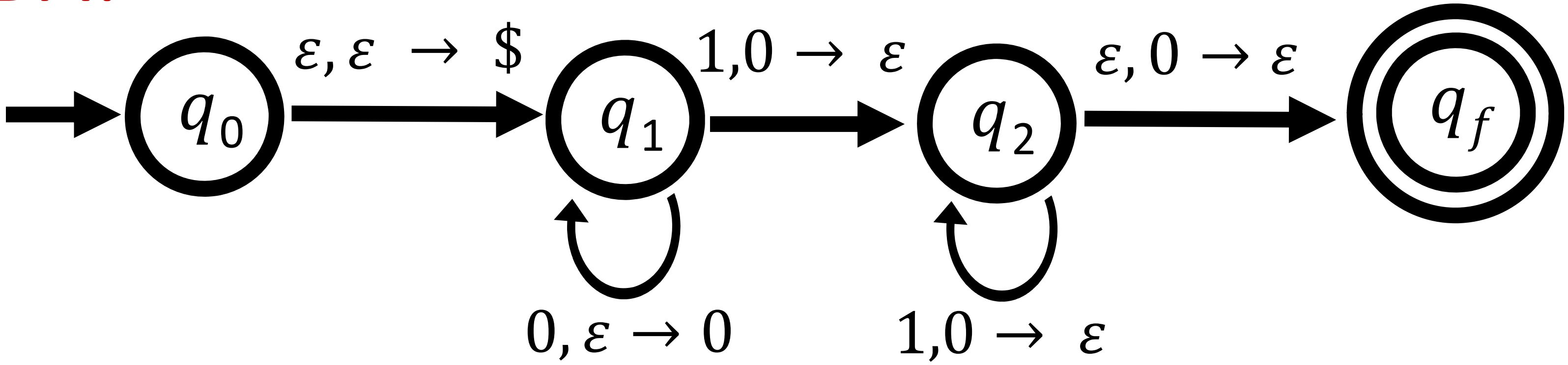
$\overbrace{aaa\ldots a}^{n}$ aaa bbab $\underbrace{b\ldots b}_{n}$ $\Rightarrow$ aaa Ya bbb

Starts w/ a's, ends w/ b's, # leading a's ≥ # trailing b's

$\overset{*}{\Rightarrow}$ aaab (any string) bbb | a^n b (any string) b^n $(n \geq 0)$

or $\Rightarrow$ aaa (any string) a bbb | or a^n (any string) a b^n $(n \geq 0)$

What language is recognized by the following PDA?



Give a CFG for the language

$$\{w\ \#0^n \mid n \geq 0, |w| = n\}$$

Give a PDA recognizing the language $\{0^n 1^n | \, n \geq 0\}$

Prove that $\{w \in \{0,1\}^* \mid w \text{ is a palindrome with the same number of 0s and 1s}\}$ is not context-free