BU CS 332 – Theory of Computation

Lecture 9:

Test 1 Review

1 8½ x 11 sheet of notes, double sided

Mark Bun

October 4, 2022

Test 1 Topics

Sets, Strings, Languages (0)

- Know the definition of a string and of a language (and the difference between them)
- Understand operations on strings: Concatenation, reverse
- Understand operations on languages: Union, intersection, concatenation, reverse, star, complement
- Know the difference between $\emptyset$ and ε

$L^R = \{w^R \mid w \in L\}$

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
- Construction for closure of regular languages under complement

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 a.k.a. subset construction
- Proving closure properties: Know the constructions for union, concatenation, star
- Know how to prove your own closure properties

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

Via GNFAs "generalized NFAs"
i.e. NFAs that have regexes labeling transitions

~~Non-regular Languages~~ (Myhill-Nerode Note)

Negative results for small DFAs

- Understand the statements of the distinguishing set method for proving DFA size lower bounds / ~~non-regularity~~
- Understand the proof of why the distinguishing set method works, and be able to use it to prove similar statements
- Know how to apply the method to specific languages
- Note: I won't ask you to show anything is non-regular, since you didn't have any homework problems on this yet

$\cup, \circ, *, +, \Sigma'$

$|$

Test format

Problem 1: "Check your type checker"

E.g., Is aabba a string, language, or a regex?

$= \{ab, aab\}$

language

How about $\underbrace{\{ab\}}_{\text{language}} \cup \underbrace{\{aab\}}_{\text{language}}$?

regex generating this language.

$ab \cup aab$

Problem 2: True/false with **justification**

Either provide a convincing explanation or a specific counterexample

Problems 3-5(?) Homework-style problems

Test tips

- You may cite without proof any result…
  - Stated in lecture
  - Stated and proved in the main body of the text (Ch. 0-1.3)
  - These include worked-out examples of state diagrams, regexes
- Not included above: homework problems, discussion problems, (solved) exercises/problems in the text

- Showing your work / explaining your answers will help us give you partial credit
- Make sure you're interpreting quantifiers (for all / there exists) correctly and in the correct order

Practice Problems

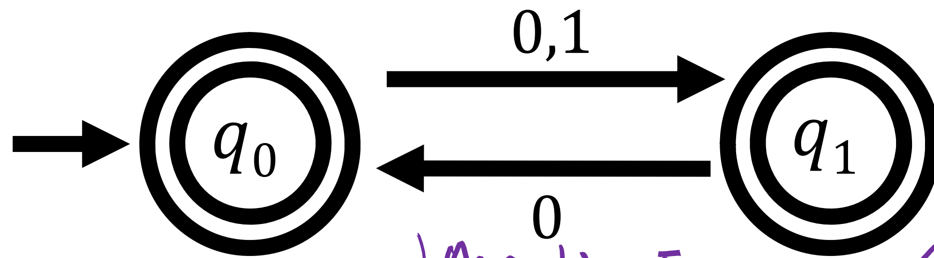
Name six operations under which the regular languages are closed

Prove or disprove: All finite languages are regular

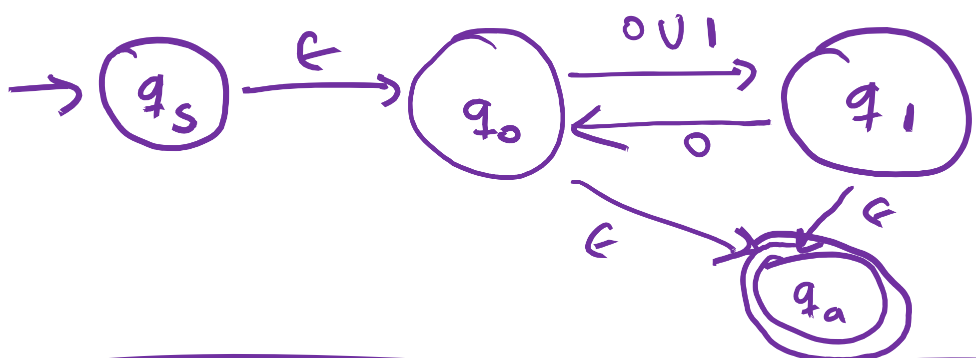
Prove or disprove: The **non-**regular languages are closed under union

Give the state diagram of an NFA recognizing the language $(01 \cup 10)^* \circ 1$

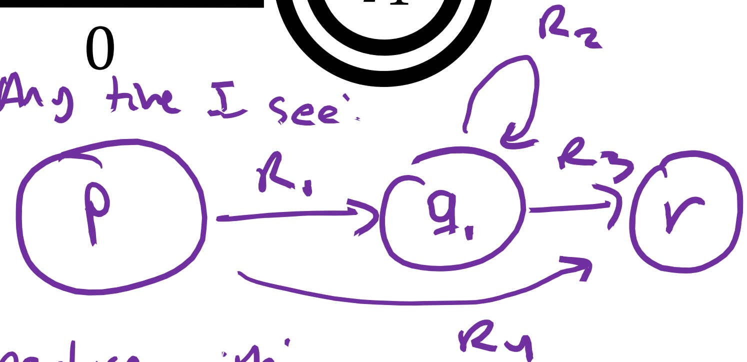
Give an equivalent regular expression for the following NFA



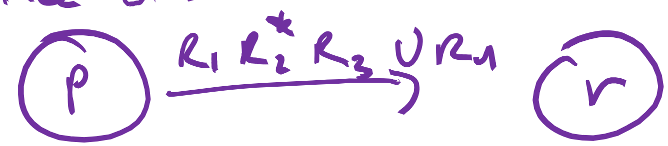
1| Convert to GNFA



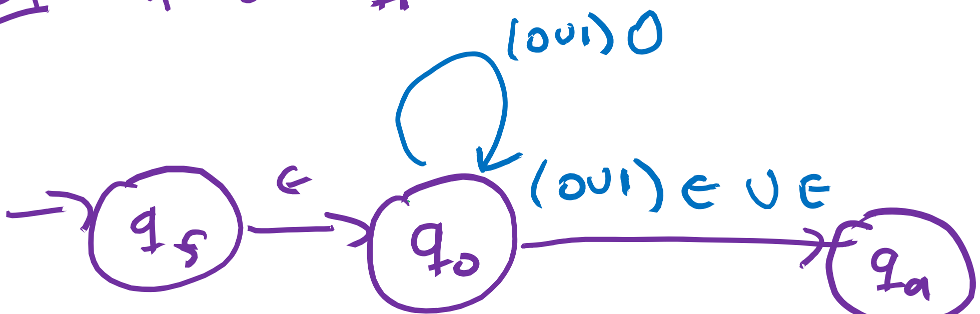
Any time I see:



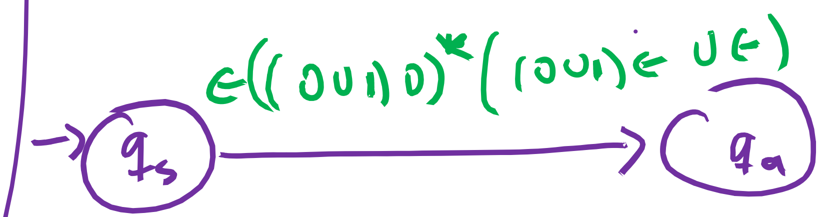
replace with:



2| Rip out q_1



3| Rip out q_0. $(0 \cup 1)\epsilon = 0 \cup 1$



Final regex: $((0 \cup 1) 0)^* (0 \cup 1 \cup \epsilon)$.

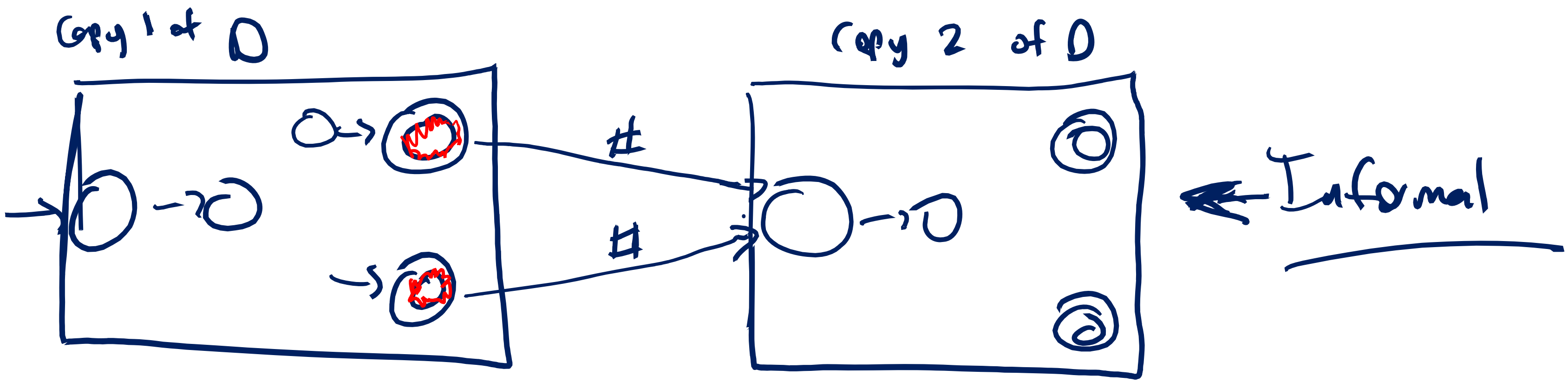
For a language L over $\{0,1\}$, define the operation $\textbf{split}(L) = \{x\#y \mid x, y \in L\}$. Show that the regular languages are closed under split

Alphabet for split(L) is $\{0, 1, \#\}$

WTS: For every regular L, split(L) is regular

Strategy 1 express split(L) in terms of other operations

Strategy 2: Given a DFA D for L, transform it into a DFA / NFA for split(L)



Is the following language regular? $\{a^n a^n \mid n \geq 0\}$

Is the following language regular?

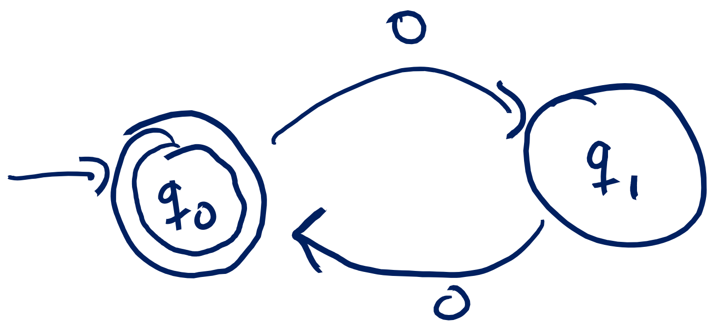
$\{0^n 1^n \mid 0 \le n \le 2022\}$

How many states does a DFA recognizing $\{0^n 1^n \mid 0 \le n \le 2022\}$ require?

Sample T/F problem 2c

$\exists$ a DFA recognizing $A = \{ww^R \mid w \in \{0\}^*\}$ using ≤ 4 states

what is A? $A = \{ w \in \{0\}^* \mid |w| \text{ is even} \}$



$2 \leq 4$ so there is a DFA recognizing using ≤ 4 states $\boxed{T}$

Alternately: construct an NFA w/ 1 state?

Then using subset construction, $\exists$ an equivalent DFA w/ $\leq 2^1 = 2 \leq 4$ states

If A is recognized by an NFA w/ 3 states, then there does not exist a pairwise dist. set for A of size 10

Let A be recognized by 3-state NFA N

Then by subset construction, $\exists$ 8-state DFA D recognizing A

$\Rightarrow$ Every pairwise dist. set for A has size ≤ 8

Thm: If $\exists$ a PD set S for A, then every DFA for A needs $\geq |S|$ states

Contrapositive: $\neg$(every DFA for A needs $\geq k$ states) $\Rightarrow$

$\neg$($\exists$ a PD set of size k for A)

$\exists x\, P(x) \Rightarrow \forall y\, Q(y)$

i.e. $\exists$ a DFA for A using $< k$ states $\Rightarrow$ every PD set for A has size $< k$

$\neg \forall y\, Q(y) \Rightarrow \neg \exists x\, P(x)$

$\exists y\, \neg Q(y) \Rightarrow \forall x\, \neg P(x)$

Alternative argument:

- know $\exists$ a DFA for A using ≤ 8 states (*)
- Assume FTSoC $\exists$ a PD set for A of size 10

$\implies$ every DFA for A needs ≥ 10 states

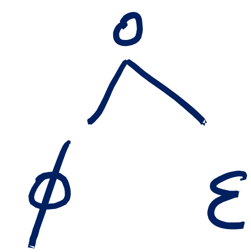
Contradicts (*)

$L(\emptyset \circ \varepsilon)$?

$= L(\emptyset) \circ L(\varepsilon)$

$= \underbrace{\emptyset}_{A} \circ \underbrace{\{\varepsilon\}}_{B}$

$= \emptyset$



Given languages A & B.
To form $A \circ B$:

Initialize $S = \emptyset$

For each $x \in \emptyset$:
For each $y \in \{\varepsilon\}$:
Add xy to S

For each $x \in A$:
For each $y \in B$:
Add xy to S

Return S

$\{\varepsilon\} \circ \emptyset = \emptyset$

For each $x \in \{\varepsilon\}$:
For each $y \in \emptyset$:
Add xy to S

$\emptyset^* = \{\varepsilon\}$

$A^* = \{ w_1 w_2 \dots w_n \mid \begin{array}{l} n \geq 0 \\ w_i \in A \end{array} \}$

$= \{\varepsilon\} \cup A \cup \underline{A \circ A} \cup \dots$

$A_\varepsilon = A \cup \{\varepsilon\}$

$A^* = \{ w_1 w_2 \dots w_n \mid \begin{array}{l} n \geq 0 \\ w_i \in A_\varepsilon \end{array} \}$