BU CS 332 – Theory of Computation

Lecture 8:

Test 1 Review

Reading:
"Myhill-Nerode" note
Sipser Ch 1.4 (optional)
Sipser Ch 2.1

Mark Bun

February 16, 2021

Mea Culpa

What I wrote: 010 010

Let $L = \{ww \mid w = w^R\}$ and consider the distinguishing set $S = \{0^n \mid n \geq 0\}$. For $x = 0^n$ and $y = 0^m$, $m \neq n$, which of the following is a distinguishing extension for x and y?

a) $z = 0^n$

b) $z = 1^n$

c) $z = 10^n$

d) $z = 01^n$

$xz = 0^n 0^n \in L$

$yz = 0^m 0^n$

If $m+n$ even: $\in L$

$m+n$ odd: $\notin L$

(b–d): $xz, yz \notin L$

None of these always work

Mea Culpa

What I meant to write:

Let $L = \{w \mid w = w^R\}$ and consider the distinguishing set $S = \{0^n \mid n \geq 0\}$. For $x = 0^n$ and $y = 0^m, m \neq n$, which of the following is a distinguishing extension for x and y?

a) $z = 0^n$

b) $z = 1^n$

c) $z = 10^n$

d) $z = 01^n$

$xz = 0^n 0^n \in L$ $\quad$ $yz = 0^m 0^n \in L$

$z = 0^n$ is never a dist. ext.

Reusing a Proof

Finding a distinguishing set can take some work…

Let's try to reuse that work!

How might we show that

$BALANCED$ = $\{w \mid w$ has an equal # of 0s and 1s$\}$

is not regular?

non-regular regular

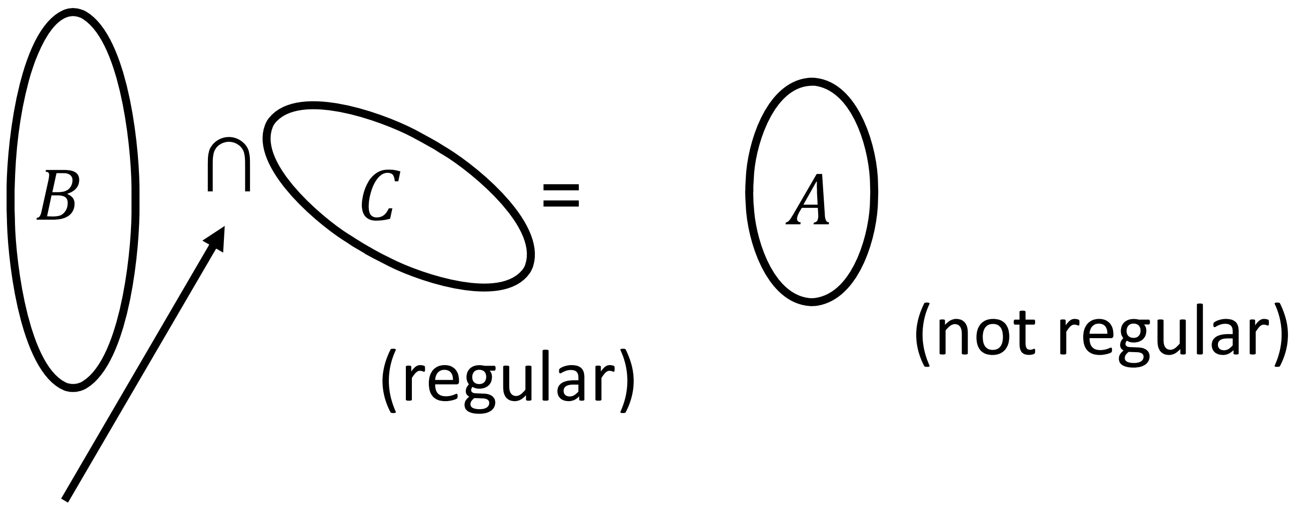
$\{0^n1^n \mid n \geq\ 0\}$ = $BALANCED \cap \{w$ | all 0s in w appear before all 1s$\}$

Suppose for contradiction BALANCED regular

By closure under $\cap$: $\{0^n1^n \mid n \geq 0\}$ regular ↯

Using Closure Properties

If A is not regular, we can show a related language B is not regular

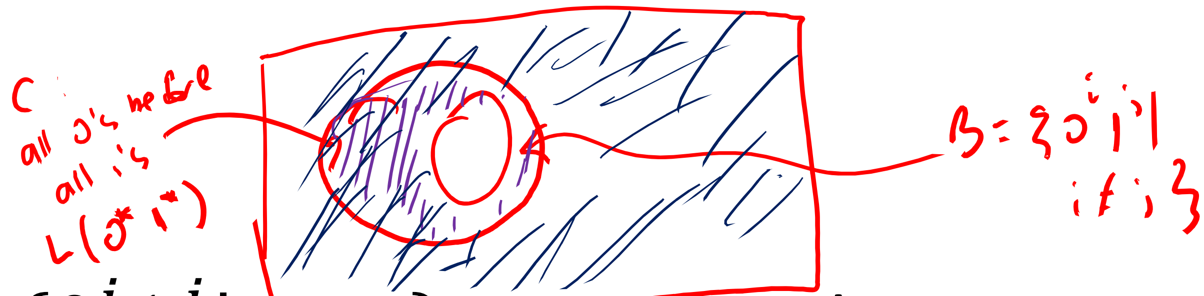


any of $\{\circ, \cup, \cap\}$ or, for one language, $\{\neg, ^{R}, *\}$

<u>By contradiction:</u> If B is regular, then $B \cap C$ $(= A)$ is regular.

But A is not regular so neither is B!

Example



Prove $B = \{0^i 1^j | i \neq j\}$ is not regular using

- nonregular language

  $A = \{0^n 1^n | n \geq 0\}$ and

- regular language

  $C = \{w$ | all 0s in w appear before all 1s$\}$

Which of the following expresses A in terms of B and C?

a) $A = B \cap C$

b) $A = \bar{B} \cap C$

c) $A = B \cup C$

d) $A = \bar{B} \cup C$

Assume for contradiction: B is regular

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

(A ← nonregular; C ← regular)

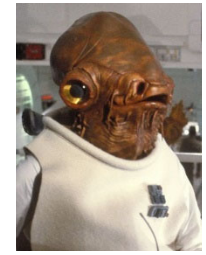
B regular $\Rightarrow \overline{B}$ regular (closure under ¬)

$\Rightarrow \overline{B} \cap C$ regular (closure under $\cap$)

$\Rightarrow A$ regular ✱

Conclude B nonregular

!DANGER!



Let $B = \{0^i 1^j | i \neq j\}$ and write $B = A \cup C$ where

- nonregular language

$$A = \{0^i 1^j | i > j \geq 0\} \text{ and}$$

- nonregular language

$$C = \{0^i 1^j | j > i \geq 0\} \text{ and}$$

Does this let us conclude B is nonregular?

Non-regular languages are not closed under union

$B = \{0^i 1^j | i \neq j\}$, $A = \{0^n 1^n | n \geq 0\}$ $B \cup A = L(0^* 1^*)$ regular

L nonregular $\Rightarrow L \cup \overline{L} = \Sigma^*$

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

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
  subset construction
- Proving closure properties: Know the constructions for union, concatenation, star
- Know how to prove your own closure properties
  - Recognize operation as combination of other operations
  - Building an NFA

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 (Myhill-Nerode Note)

- 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
- Know how to show languages are non-regular by combining distinguishing set method with closure properties

Test tips

1) T/F w/ justification
... the rest ≈ homework

- You may cite without proof any result…
  - Stated in lecture
  - Stated and proved in the main body of the text (Ch. 0-1.4)
  - These include worked-out examples of state diagrams, regexes
- Not included above: homework problems, discussion problems, (solved) exercises/problems in the text

  A 1.28

- 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

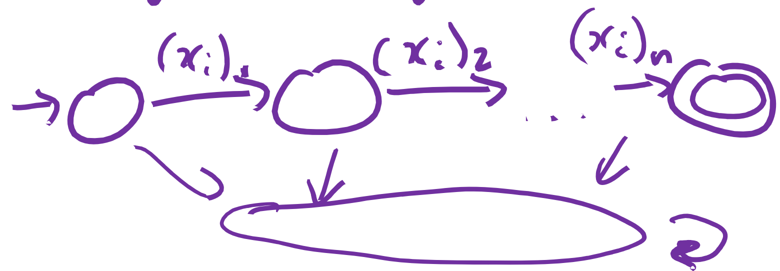
True

← all strings, $m \geq 0$

L finite, write $L = \{x_1, \ldots, x_m\}$

1) $L_i = \{x_i\}$ claim: Each L_i is regular

a) Construct a DFA



b) $L_i = L(x_i)$ (regular expression)

$L = L_1 \cup L_2 \cup \ldots \cup L_m$

L_1 regular, $L_1 \cup L_2$ regular, ..., $L_1 \cup \ldots \cup L_i$ regular $\forall i$

$\Rightarrow L$ regular

2) Converse to dist. set method: If L has no infinite dist. set, then L is regular

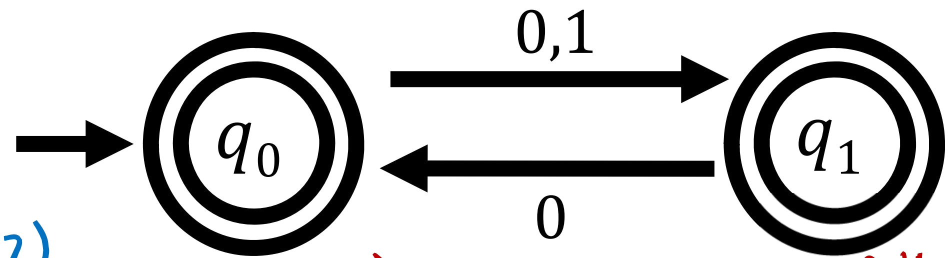
L finite $\Rightarrow$ L has no infinite dist set

↖ takes an argument

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

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

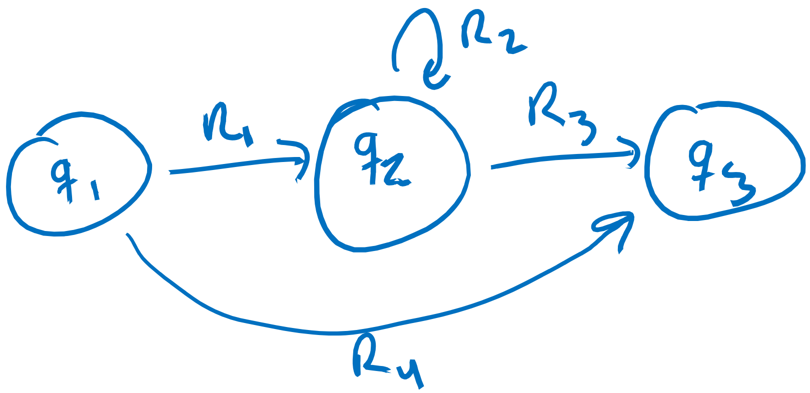
Give an equivalent regular expression for the following NFA



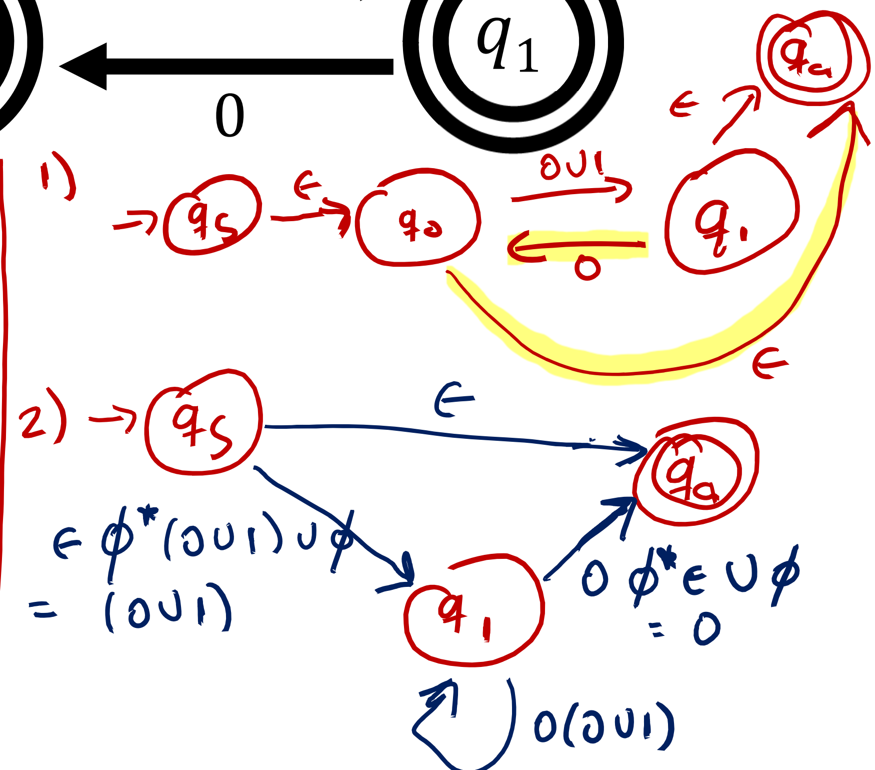
In general:

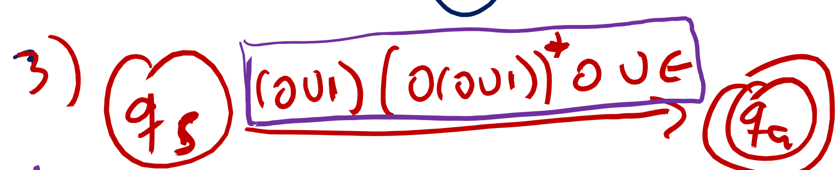
1) NFA → GNFA
(may increase # states by 2)

2) while GNFA has > 2 states:
- choose a state, rip it out, replace all transitions



replace w/





A GNFA is an NFA but
- transitions labeled by regexes
- start state only has out transitions
- only 1 accept state w/ only in transitions

Is the following language regular?

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

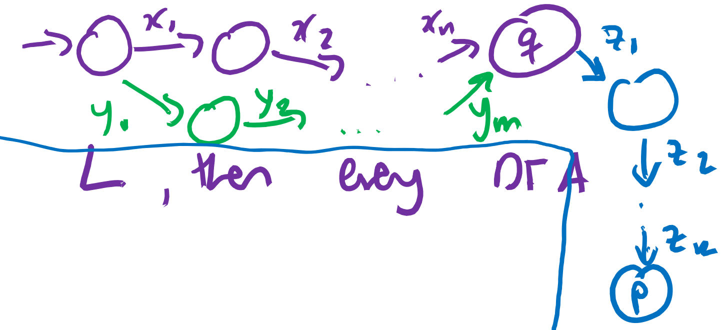
Is the following language regular?

$$\{0^n 1^n | 0 \leq n \leq 2021\}$$

How many states does a DFA recognizing $\{0^n 1^n | 0 \leq n \leq 2021\}$ require?

Distinguishing set method:

If S is pairwise distinguishable by L, then every DFA for L needs $\geq |S|$ states



Proof. S pairwise dist means $\forall\, x, y \in S$ $\exists\, z$ s.t. exactly one of xz, yz $\in L$

Assume (for contradiction) L recog. by a DFA M w/ $|S|-1$ states

Pigeons: strings $x \in S$

Holes: states $q \in Q$ of DFA

Assign pigeon x to hole q if M ends in state q when reading x

Pigeonhole principle: $\exists\, q \in Q$ s.t. $\exists\, x \neq y \in S$, both assigned to q

Extend to xz, yz for z a dist. extension

M ends in same state p upon reading both xz and yz

p accept state $\Rightarrow$ both $xz, yz \in L$ or p reject state $\Rightarrow$ both $xz, yz \notin L$ ✗