B.U. CS 332 – Theory of Computation

Lecture 9:
Test 1 Review

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 $\varepsilon$
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 

• 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

GNFAs  "generalized NFA's"
\[ V \in GNFAs \text{  "generalized NFA's" } \]
\[ i.e. \text{ NFA's that have regular expressions labeling transitions} \]
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.
- Note: I won’t ask you to show anything is non-regular, since you didn’t have any homework problems on this yet.
Test format

Problem 1: “Check your type checker”

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

How about $\{ab\} \cup \{aab\}$?

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 NFA

2. Rip out q₄

3. Rip out q₀

Final regex: \((\text{ou1})^*\text{ou1}\text{ue}\)
For a language $L$ over $\{0, 1\}$, define the operation $\text{split}(L) = \{x \# y \mid x, y \in L\}$. Show that the regular languages are closed under split

**Alphabet for \text{split}(L) \in \{0, 1, \#\}**

**WTS:** For every regular $L$, $\text{split}(L)$ is regular

**Strategy 1:** express $\text{split}(L)$ in terms of other operations

**Strategy 2:**

Given a DFA $D$ for $L$, transform it into a DFA/NFA for $\text{split}(L)$

![Diagram of DFA transformation](image)
Is the following language regular? \( \{a^n a^n \mid n \geq 0\} \)
Is the following language regular?
\{0^n1^n \mid 0 \leq n \leq 2022\}
How many states does a DFA recognizing \( \{0^n1^n \mid 0 \leq n \leq 2022\} \) require?
Sample T/F problem 2c

\[
\exists \text{ a DFA recognizing } A = \{ w \in \{0,1\}^* \mid |w| \text{ is even} \}\] using \( \leq 4 \) states

What is \( A' \)?

\[ A' = \{ w \in \{0,1\}^* \mid |w| \text{ is even} \} \]

![DFA diagram]

2 \( \leq 4 \) so there is a DFA recognizing using \( \leq 4 \) states

Alternatively: Construct an NFA with 1 state.

Then using subset construction, \( 2^1 = 2 \leq 4 \) 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 a 3-state NFA $N$

Then by subset construction, $\exists$ 8-state NFA $O$ recognizing $A$

$\implies$ every pairwise dist. set for $A$ has size $\leq 8$

Then: if $\exists$ a PO set $S$ for $A$, then every NFA for $A$ needs $\geq 151$ states

(contrapositive): $\neg (\forall y \forall x \neg p(y))$ $\implies$ $\neg (\exists x p(x))$

i.e. $\exists x \neg p(x)$

$\neg (\forall y \forall x \neg p(y))$ $\implies$ $\forall x \neg p(x)$

$\neg (\exists x p(x))$ $\implies$ $\forall x \neg p(x)$

There exists no NFA for $A$ using $< k$ states $\implies$

every PO set for $A$ has size $< k$
Alternative approach:

- Assume \( \exists \text{ a DTM for } A \) using \( \leq 8 \) states (\( \ast \))

\( \Rightarrow \) every DTM for \( A \) needs \( \geq 10 \) states

Contradicts (\( \ast \))
Given languages $A$ & $B$.

To form $A \cap B$:

1. Initialize $S = \emptyset$
2. For each $x \in A$:
   - For each $y \in \epsilon_f$:
     - Add $xy$ to $S$
3. For each $y \in \epsilon_f$:
   - For each $x \in \emptyset$:
     - Add $xy$ to $S$

Return $S$
\[ \phi^* = \varepsilon \varepsilon^3 \]

\[ A^* = \{ w_1, w_2, \ldots, w_n \mid \forall \omega \in A \exists \zeta \in A \} \]

\[ = \varepsilon^3 \cup A \cup \lambda \varepsilon^3 \]

\[ A_\varepsilon = A \cup \varepsilon \varepsilon^3 \]

\[ A^*_\varepsilon = \{ w_1, w_2, \ldots, w_n \mid \forall \omega \in A \varepsilon \exists \zeta \in A_\varepsilon \} \]