HKN CS 374 Midterm 1 Review

Tim Klem
Noah Mathes
Mahir Morshed

Midterm topics

It's all about recognizing sets of strings!

1. String Induction
2. Regular languages
   a. DFA
   b. NFA
   c. Regular expressions
3. Context-free languages
   a. Context-free grammar

String Induction

- Proof standards are strictest here - stick to the definitions!
- Be careful of the order of the if-else branches

$$|w| := \begin{cases} 0 & \text{if } w = \varepsilon \\ 1 + |x| & \text{if } w = ax \text{ for some symbol } a \text{ and some string } x \end{cases}$$

$$w \bullet z := \begin{cases} z & \text{if } w = \varepsilon \\ a \cdot (x \bullet z) & \text{if } w = ax \text{ for some symbol } a \text{ and some string } x \end{cases}$$

Lemma 1: $w \bullet \varepsilon = w$ for all strings w.

Lemma 2: $|w \bullet x| = |w| + |x|$ for all strings w and x.

Lemma 3: $(w \bullet x) \bullet y = w \bullet (x \bullet y)$ for all strings w, x, and y.

Language Definitions

- A DFA *recognizes* (or accepts) a language L iff every string in L is accepted by the DFA, and every string not in L is rejected by the DFA.
    - Likewise for NFAs, regular expressions
- A context-free language L is *generated by* a CFG iff (almost copy and paste from above)

Regular Languages & Expressions

- Formally, one of
  a. The empty set
  b. The set containing one string
  c. The union, concatenation, or Kleene closure of regular language(s)
- Additionally, closed under intersection, set difference, reversal, homomorphisms, inverse homomorphisms
- Regular expressions follow the structure of the formal definition

Deterministic Finite Automata

- Formally,
  - An arbitrary finite set Σ, called the ***input alphabet***.
  - Another arbitrary finite set Q, whose elements are called ***states***.
  - An arbitrary ***transition*** function $\delta : Q \times \Sigma \to Q$.
  - A ***start state*** $s \in Q$.
  - A subset $A \subseteq Q$ of ***accepting states***.

- In drawings, Q is the set of circles, δ is the set of arrows, s is the circle with an arrow leading into it without any source, and A is the set of doubly-circled circles

Non-deterministic Finite Automata

- Formally,

Definition

A non-deterministic finite automata (NFA) $N = (Q, \Sigma, \delta, s, A)$ is a five tuple where

- Q is a finite set whose elements are called states,
- Σ is a finite set called the input alphabet,
- $\delta : Q \times \Sigma \cup \{\varepsilon\} \to \mathcal{P}(Q)$ is the transition function (here $\mathcal{P}(Q)$ is the power set of Q),
- $s \in Q$ is the start state,
- $A \subseteq Q$ is the set of accepting/final states.

- Drawn similarly like DFAs, but with the extra epsilon transitions
- An NFA accepts a string s can reach *at least one* of the accepting states

Regular Languages: Useful Theorems

Theorem (DFA=NFA=ε-NFA)

A language is accepted by a deterministic finite automaton if and only if it is accepted by a non-deterministic finite automaton.

Theorem (REGEX=DFA)

A language is accepted by a deterministic finite automaton if and only if it is accepted by a regular expression.

Source: CSCI

Is it regular?

- First, determine if the language definition can be simplified
- Then, check if you need an unbounded counter
- If so, try to create a fooling set
  - If possible, choose the simplest prefixes you can find
  - Be careful of generalized string splits - a string belongs to a presented language if *any* permitted combo of strings satisfies the conditions within the set condition
  - Use letter changes to guard against these splits
- If not, try to create a DFA/NFA/Regex

Regular Expressions: Examples

Write regular expressions for the following languages over the alphabet $\Sigma = \{a, b\}$:

(a) All strings that do not end with aa
(b) All strings that contain an even number of b's
(c) All strings which do not contain ba

Source: [Stanford]

Regular Expressions: Examples

Write regular expressions for the following languages over the alphabet Σ = {a, b}:

(a) All strings that do not end with aa

$\in + a + b + (a + b)^{*} (ab + ba + bb)$

(b) All strings that contain an even number of b's

$a^{*} (ba^{*} ba^{*})^{*}$

(c) All strings which do not contain ba

$a^{*} b^{*}$

Source: Stanford

Regular Languages: Example

Is the language $\{ xyx^r \mid x, y \in \{0,1\}^+ \}$ regular?

Is the language $\{ xyxyxy \mid x, y \in \{0,1\}^+ \}$ regular?

Regular Languages: Example

Is the language $\{ xyx^r \mid x, y \in \{0,1\}^+ \}$ regular?

Answer: Yes - equivalent to $0(0+1)^+0 + 1(0+1)^+1$

Is the language $L = \{ xyxyxy \mid x, y \in \{0,1\}^+ \}$ regular?

Answer: No - use 1 to indicate end of string xy in fooling set

Fooling set: $\mathbf{F} = \{0^n1^n \mid n >= 0\}$

Suffix ($\mathbf{z}$) : $0^n1^n0^n1^n$

$\mathbf{xz} \in L$ but $\mathbf{yz} \notin L$

$\mathbf{x} = 0^n1^n$

$\mathbf{y} = 0^i1^i \quad i \neq n$

Deterministic Finite Automata: Examples

Draw DFAs for the following languages over the alphabet $\Sigma = \{a, b\}$:

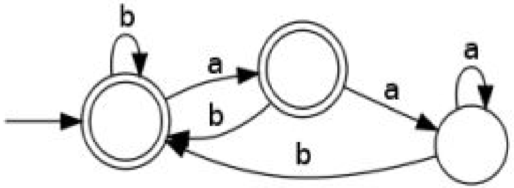
(a) All strings that do not end with aa
(b) All strings that contain an even number of b's
(c) All strings which do not contain ba

Source: Stanford

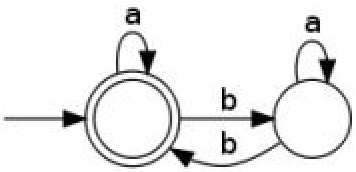
Deterministic Finite Automata: Examples

Draw DFAs for the following languages over the alphabet Σ = {a, b}:

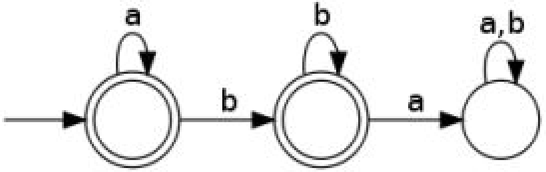
(a) All strings that do not end with aa
(b) All strings that contain an even number of b's
(c) All strings which do not contain ba



(a)



(b)



(c)

Source: [Stanford]

Context-free Grammars and Languages

- Formally,
  - A finite set Σ, whose elements are called ***symbols*** or ***terminals***.
  - A finite set Γ disjoint from Σ, whose elements are called ***non-terminals***.
  - A finite set R of ***production rules*** of the form $A \to w$, where $A \in \Gamma$ is a non-terminal and $w \in (\Sigma \cup \Gamma)^*$ is a string of symbols and variables.
  - A ***starting*** non-terminal, typically denoted S.
- *Not* closed under intersection or set difference
- Design:
  a. Pair letters - identify their positions and guarantee that substrings hold certain properties
  b. Drop letters - useful for inequalities or early termination
  c. Cumulative counting - does the difference operator Δ hit zero?

Context-free Grammars: Examples

Construct context-free grammars for each of the following languages:

a) $\{a^m b^n \mid m >= n\}$
b) $\{a^m b^n c^p d^q \mid m + n = p + q\}$
c) $\{w \in (a+b)^* \mid$ w has twice as many b's as a's$\}$
d) $\{uawb \mid u, w \in (a+b)^*, |u| = |w|\}$

Source: [UT-Austin]

Context-free Grammars: Examples

Construct context-free grammars for each of the following languages:

a) $\{a^m b^n \mid m >= n\}$

Answer: S → aSb | aCb | ε, C → aC | ε

b) $\{a^m b^n c^p d^q \mid m + n = p + q\}$

Answer: S → aSd | T | U, T → aTc | V, U → bUd | V, V → bVc | ϵ

c) {w ∈ (a+b)* | w has twice as many b's as a's}

Answer: S → SaSbSbS | SbSaSbS | SbSbSaS | ϵ

d) {uawb | u, w ∈ (a+b)*, |u| = |w|}

Answer: S → Tb, T → aTa | aTb | bTa | bTb | a

Source: UT-Austin

Language Transformation

Let L_1 and L_2 be regular languages that share a common alphabet $\Sigma = \{0, 1\}$. Define the bitwise-xor of L_1 and L_2 as

$$xor(L_1, L_2) = \{xor(x, y) \mid x \in L_1 \text{ and } y \in L_2\},$$

where $xor(x, y)$ is the bitwise-xor of two binary strings x and y. Prove that $xor(L_1, L_2)$ is regular.

Language Transformation

$$Q_3 = Q_1 \times Q_2$$
$$\delta_3((q_1, q_2), 0) = \{(\delta_1(q_1, 0), \delta_2(q_2, 0)), (\delta_1(q_1, 1), \delta_2(q_2, 1))\}$$
$$\delta_3((q_1, q_2), 1) = \{(\delta_1(q_1, 0), \delta_2(q_2, 1)), (\delta_1(q_1, 1), \delta_2(q_2, 0))\}$$
$$s_3 = (s_1, s_2)$$
$$A_3 = A_1 \times A_2$$

True/False Examples

G.4 Let M be an **NFA** over the alphabet Σ. Let M' be identical to M, except that accepting states in M are non-accepting in M' and vice versa. Each string in Σ^* is accepted by exactly one of M and M'.

True/False Examples

B.36 If $L \subseteq L'$ and L is not regular, then L' is not regular.

True/False Examples

H.9 For every context-free language L, the language $\{w^R \mid w \in L\}$ is also context-free.

True/False Examples

Let $S = \{L_1, L_2, L_3, \ldots\}$ be an *infinite* set of regular languages. Then, the union of all of the elements of S is a regular language.

True/False Examples

B.12 For all languages L, if L^* is regular, then L is regular.

True/False Examples

C.15 For all context-free languages A, B, and C, the language $(A \cup B)^* \bullet C$ is also context-free.