CSE 311 Section 09

Models of Computation

Administrivia

Announcements & Reminders

- HW6 Regrade Requests
  - Submit a regrade request if something was graded incorrectly

- HW7
  - Due **tomorrow** Friday 11/22 @11:59pm
  - Late due date 11/25 @ 11:59pm

- HW8
  - Due Wednesday 12/04 @ 11:59pm

- Final Exam
  - Monday 12/09 @ 12:30pm-2:20
  - Keep an eye out for conflict exam form on Ed

Regular Expressions

Regular Expressions

Basis:

- $\boldsymbol{\varepsilon}$: The empty string itself matches the pattern (and nothing else does).
- $\varnothing$: No strings match this pattern
- $\boldsymbol{a}$ for any $\boldsymbol{a} \in \boldsymbol{\Sigma}$: The character itself matching this pattern

Recursive:

- If $\boldsymbol{A}, \boldsymbol{B}$ are regular expressions then $\boldsymbol{(A \cup B)}$ is a regular expression
  - matched by any string that matches A or that matches B [or both]
- If $\boldsymbol{A}, \boldsymbol{B}$ are regular expressions then $\boldsymbol{AB}$ is a regular expression
  - matched by any string x such that $x = yz$, y matches A and z matches B
- If $\boldsymbol{A}$ is a regular expression, then $\boldsymbol{A}^*$ is a regular expression
  - matched by any string that can be divided into 0 or more strings that match A

Regular Expressions

A regular expression is a recursively defined set of strings that form a language.

A regular expression will generate all strings in a language, and won't generate any strings that ARE NOT in the language

Hints:

- Come up with a few examples of strings that ARE and ARE NOT in your language
- Then, after you write your regex, check to make sure that it CAN generate all of your examples that are in the language, and it CAN'T generate those that are not

Problem 1 – Regular Expressions

a) Write a regular expression that matches base 10 numbers (e.g., there should be no leading zeroes).

b) Write a regular expression that matches all base-3 numbers that are divisible by 3.

c) Write a regular expression that matches all binary strings that contain the substring "111", but not the substring "000".

d) Write a regular expression that matches all binary strings that do not have any consecutive 0's or 1's.

e) Write a regular expression that matches all binary strings of the form $1^k y$, where $k \geq 1$ and $y \in \{0,1\}^*$ has at least k 1's.

Work on this problem with the people around you.

Context-Free Grammars

Context-Free Grammars

A context free grammar (CFG) is a finite set of production rules over:

- An alphabet Σ of “terminal symbols”
- A finite set V of “nonterminal symbols”
- A start symbol (one of the elements of V) usually denoted S

A production rule for a nonterminal $A \in V$ takes the form

- $A \to w1 \mid w2 \mid \ldots \mid wk$

Where each $wi \in V \cup \Sigma^*$ is a string of nonterminals and terminals.

Problem 2 – CFGs

Write a context-free grammar to match each of these languages.

a) All binary strings that start with 11.

b) All binary strings that contain at most one 1.

c) All strings over 0, 1, 2 with the same number of 1s and 0s and exactly one 2.

Work on this problem with the people around you.

Deterministic Finite Automata

Deterministic Finite Automata

- A DFA is a finite-state machine that accepts or rejects a given string of symbols, by running through a state sequence uniquely determined by the string.

- In other words:
    - Our machine is going to get a string as input. It will read one character at a time and update "its state."
    - At every step, the machine thinks of itself as in one of the (finite number) vertices. When it reads the character, it follows the arrow labeled with that character to its next state.
    - Start at the "start state" (unlabeled, incoming arrow).
    - After you've read the last character, accept the string if and only if you're in a "final state" (double circle).

- Every machine is defined with respect to an alphabet Σ
- Every state has exactly one outgoing edge for every character in Σ
- There is exactly one start state; can have as many accept states (aka final states) as you want – including none.

Problem 3 – DFAs, Stage 1

Construct DFAs to recognize each of the following languages.
Let $\Sigma = \{0, 1, 2, 3\}$.

a) All binary strings.

b) All strings whose digits sum to an even number.

c) All strings whose digits sum to an odd number.

Work on this problem with the people around you.

All The Models

Problem 9 – All The Models

Construct a valid regular expression, CFG, and DFA for the following languages.

a) All strings whose base-6 representation is divisible by 3 (leading zeros are ok). Let $\Sigma = \{0, 1, 2, 3, 4, 5\}$.

b) All binary strings of 0s capped by a 1 on either side

Work on this problem with the people around you.

That’s All, Folks!

Thanks for coming to section this week!
Any questions?