Dang Nhi Ngo
01553277

95

Department of Computer Science
University of Massachusetts Lowell
COMP.3040 Foundations of Computer Science
Fall 2018
Quiz 1 [5%]
9/27/2018

1. *[30 points]* Differentiate between
   - alphabet, string, language
   - Domain, Range
   - Empty set, Empty string
   - States, start state, accept state
2. *[15 points]* Write a formal description for the following sets
   - set containing 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024 $A = \{w \mid w = 2^k, 0 \leq k \leq 10\}$
   - set containing 0-9, a-f $A = \{w \mid w$ is the hexadecimal value from 0 to 15$\}$
   - set containing 0, 1, 1, 2, 3, 5, 8, 13, ... $A = \{w_0, w_1, \ldots, w_k \mid k \geq 0, w_k = w_{k-2} - w_{k+1}\}$

   X
   -5 (fibonachi)
3. *[20 points]* Give the formal definition for
   - DFA
   - NFA
   - Regular Language
   - Regular Expression
4. *[5 points]* A NFA is more powerful than a DFA.
   - True
   - (•) False
5. *[5 points]* Every NFA can be converted to a DFA.
   - (•) True
   - False
6. *[5 points]* A language is Regular if a DFA or NFA exists which can recognize it.
   - (•) True
   - False
7. *[5 points]* All regular languages are infinite–contain infinite number of strings }
   - True

◉ False

8. *[15 points]* The class of regular languages is closed under the following operations:

union	$A \cup B$
concatenation	$A \circ B$
star	A^*

1/ a/ Alphabet is a set of finite, nonzero objects called symbols
String is a set of symbols from an alphabet
Language is a set containing all the strings or character

b/ Domain is a set of all possible inputs to a function
Range is a set of outputs of a function

c/ Empty set is the set of zero members, $\emptyset$
Empty string is the string with length of zero, ε

d/ States are all the sets of state that has the exits
Start state is the state that starts the sequence
Accept state is the state that can end with valuable sets of input

3/ – DFA = $\{Q, \Sigma, \delta, q_0, F\}$
where Q: finite set of states
Σ: finite set of symbols from alphabet
δ: $Q \times \Sigma \to Q$ (transition function)
q_0: start state
F: set of accept states

– Regular language consists of all strings that a finite automaton recognizes
$A = L(M)$ M recognizes A

– Regular expression: regular operations used to build up expression describing language

– NFA = $\{Q, \Sigma, \delta, q_0, F\}$
where Q: finite set of states
Σ: finite set of symbols from alphabet and ε (empty string)
δ: $Q \times \Sigma_\varepsilon \to \mathcal{P}(Q)$ (transition function)
q_0: start state
F: set of accept states

Dang Nhi Ngoc Ngo

90

Department of Computer Science
University of Massachusetts Lowell
COMP.3040 Foundations of Computer Science
Fall 2018
Quiz 2 [5%]
11/29/2018

1. Give the formal definition of a Turing Machine (TM)

   TM is a 7 tuple $(Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$
   where Q, Σ, Γ are all finite sets
   Q : is a set of states
   Σ : is an input alphabet not containing blank input $\sqcup$
   Γ : is a tape input, which $\sqcup \in \Gamma$ and $\Sigma \in \Gamma$
   $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$ transition function
   $q_0 \in Q$: start state
   $q_{accept} \in Q$: accept state
   $q_{reject} \in Q$: reject state

2. What is an Enumerator? Give an example.

   Enumerator is a Turing machine with attached printer.
   It starts with blank input on its work tape. -2

3. What is Church-Turing Thesis?

   Church-Turing Thesis is the connection between the informal notion of algorithm and the precise definition.
   Intuitive notion of algorithms equals Turing Machine algorithms

4. A multi-tape TM is not more powerful than a single tape TM.

   - True
   - (•) False -8

5. A Non-deterministic TM is more powerful than a Deterministic TM.

   - True
   - (•) False

6. Order the class of languages in increasing order of power

   - Context Free 2
   - Regular 1
   - Turing Recongnizable 4
   - Turing Decidable 3

7. A = { $0^{2^n}, n >= 0$, is a decidable/recognizable language }

   - Decidable
   - (•) Recognizable

8. A_{DFA} = { (B, w) | B is a DFA that accepts input string w, is a decidable/recognizable language }

   (•) Decidable
   - Recognizable

9. A_{REX} = { (R, w) | R is a Regular Expression that generates string w, is a decidable/recognizable language }

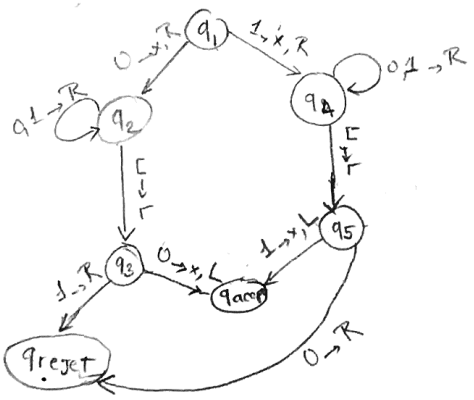
   (•) Decidable
   - Recognizable

10. A_{CFG} = { (G, w) | G is a CFG that generates string w, is a decidable/recognizable language }

    (•) Decidable
    - Recognizable

11. *[20 points]* Sketch an algorithm using a single tape TM to recognize the language L | ww^R, where w is a string and w^R is the reverse of the string. Assume $\sum = \{0, 1\}$

01100.00110



- Scan the input from left to right till the end of the input
- Determine if the input is the combination of w(string) and reverse of the string (w^R). If the length is even, accept it. Otherwise, reject it.
- Place an x to mark the first character. Then, move the head to the right till the end of the input for the last character. If the last character is not the same with the first character, reject it. Otherwise, move left and place an x to mark the character
- Keep moving to the left and place an x to mark if the character is not different with the character of w, If so, accept it. Otherwise, reject it.

Chapter 0

- Set is a group of objects represented as a unit
- The objects in a set are called as its elements or members
- A is a subset of B ($A \subseteq B$) if every member of A also is a member of B
- A is a proper subset of B ($A \subsetneq B$) if A is a subset of B and not equal to B
- An infinite set contains infinitely many elements
- Power set of A is the set of all subsets of A
- Empty set is the set with zero members, $\emptyset$
- Singleton set (1 member) / Unordered pair (2 members)
- The union of A and B is the set we get by combining all the elements in A and B into a single set $A \cup B$
- The intersection of A and B, $A \cap B$, is the set of elements that are in both A and B
- The complement of A, $\overline{A}$ is the set of all elements that are not in A
- A sequence of objects is a list of these objects in some order
  The order doesn't matter in a set, but in a sequence it does
- Finite sequences are called tuples
- Function is an object that sets up an input-output relationship
  ↳ or mapping
- Domain is the set of possible inputs to the function
- Range is the set of outputs of a function

$$f: D \to R$$

⇒ A function that does use all the elements of a range is said to be onto the range

- k-tuple: If $k = 1$, f has single argument → f is unary function
  $k = 2$ → f is binary function
- A predicate or property is a function whose range is {TRUE, FALSE}
- A relation is a property whose domain is a set of k-tuples $A \times \dots \times A$, k-ary relation, or a k-ary relation on A

infix notation ≠ prefix notation

- A binary relation R is an equivalence relation if R satisfies 3 conditions:
  + R is reflexive if for every x, xRx
  + R is symmetric if _____ and if, xRy implies yRx
  + R is transitive if _____, z, xRy and yRz implies xRz
- Undirected graph (or graph) is a set of points with lines connecting some of the points
  ≠ Directed graph has arrows instead of lines
- Nodes / vertices
  Edges
- Degree of a node: the number of edges at that node
  self-loop: an edge from a node to itself
- Labeled graph
- G is a subgraph of H ⇔ the nodes of G are a subset of the nodes of H
  & _ edges _____ edges _____ on the corresponding nodes
- A path in a graph is a sequence of nodes connected by edges
  A simple path is a path that doesn't repeat any nodes
  A graph is connected if every two nodes have a path between them
  A path is a cycle if it starts & ends in the same node
  A graph is a tree if it is connected & has no simple cycles
- Outdegree ≠ Indegree
  ↓ Σ arrows pointing from a node
  ↘ Σ arrows pointing to a particular node
- Empty string is the string of length zero, ε

Chapter 1 : Regular Languages

1/ Finite Automata

- State diagram
- Start state, Accept state, Transitions
- The output is either accept or reject
- $M_1 = (Q, \Sigma, \delta, q_0, F)$

  Q : states $\quad Q = \{q_1, q_2, q_3\}$

  Σ : alphabet $\quad \Sigma = \{0, 1\}$

  δ : $Q \times \Sigma \to Q$ transition function

	0	1
q_1	q_1	q_2
q_2	q_3	q_2
q_3	q_2	q_2

  q_0 : start state $\quad q_1$ is the start state

  F : set of accept states $\quad F = \{q_2\}$

- A is the language of machine $M \Rightarrow L(M) = A$

  $\Rightarrow$ M recognizes A or M accepts A if $A = \{w \mid M \text{ accepts } w\}$ (w: member of Q)

★ A machine may accept several strings but always recognizes only one language ★ ‖ If the machine accepts no strings, it still recognizes 1 language, namely, the empty language $\emptyset$.

Example: $A = \{w \mid w$ contains at least one 1 and an even number of 0s follow the last 1$\}$

$L(M_1) = A$

or $L(M_2) = \{w \mid w$ ends in a 1$\}$

✳ Empty string ε $\Rightarrow$ Because the start state is an accept state, M accepts the empty string ε

✳ A language is called a regular language if some finite automaton recognizes it

✳ Regular operations: Union $\cup$, Concatenation $A \circ B = \{x \mid x \in A \text{ and } x \in B\} \| = \{xy \mid x \in A \text{ and } y \in B\}$, Star $A^* = \{x_1 x_2 \ldots x_k \mid k \geq 0 \text{ and each } x_i \in A\}$

Unary operation not Binary operation

✳ The class of regular languages is closed under the union operation (Thm 1.25)

concatenation (Thm 1.26)

2/ Nondeterminism

DFA

+ has exactly 1 exiting transition arrow for each symbol in the alphabet
+ labels on the transition arrows are symbols from the alphabet

NFA

+ a state may have zero, one, or many exiting arrows for each alphabet symbol
+ has arrows labeled with members of the alphabet or ε

$N_1(Q, \Sigma, \delta, q_0, F)$

$\delta: Q \times \Sigma_\varepsilon \to \mathcal{P}(Q)$

* Two machines are equivalent if they recognize the same language

-> Every nondeterministic finite automation has an equivalent deterministic finite automaton

* A language is regular if and only if some nondeterministic finite automation recognizes it
* The class of regular languages is closed under the union/star/concatenation operation
* A language is regular if and only if some regular expression describes it