CHAPTER 2

A —> 0A1
A —> B
B —> #

- The symbol is called ***variable***
- The string contains variables and other symbols is called ***terminal***
- A sequence of substitutions to obtain a string is called ***derivation***
- ***Context-free language:*** any language that can be generated by some context-free grammar
- Definition for a ***Context-free grammar:***
  - It's a 4-tuple (V, Σ, R, S)
    - *V:* a finite set called ***variables***
    - Σ: a finite set, disjoint from V, called ***terminals***
    - *R:* a finite set of ***rules***, with each rule being a variable and a string of variables and terminals
    - *S:* the ***start variable***
- ***The language of the grammar*** is $\{w \in \Sigma^* \mid S \overset{*}{\Rightarrow} w\}$
- If a grammar generates some string ambiguously, we say that the grammar is ***ambiguous*** (aka generates the same string in several different ways, resulting in different parse trees).
- A derivation of string w in a grammar G is called ***leftmost derivation*** if at every step, the leftmost variable is the one getting replaced.
- A string is ***derived ambiguously*** in a context-free grammar G if it has 2 or more different leftmost derivations.
- Some context-free language can be generated only by ambiguous grammar. They're called ***inherently ambiguous***.
- Context-free grammar is in ***Chomsky normal form*** if every rule is of the form:

$$A \rightarrow BC$$

$$A \rightarrow a, \text{ or}$$

$$S \rightarrow \varepsilon$$

a: any terminal
A, B, C: any variables except start variable

- Pushdown automata: similar to nondeterministic finite automata (NFA) with an extra component called stack => allows PDA to recognize non-regular languages.

CHAPTER 1

- A language is called regular language if some finite automata recognizes it.
- Regular operations:
  - $A \cup B = \{x \mid x \in A \text{ or } x \in B\}$
  - $A \circ B = \{xy \mid x \in A \text{ and } y \in B\}$

- $A^* = \{x1x2...xk \mid k \geq 0$ and each $xi \in A\}$ (ε is always a member).
- Star operation: attach any number of strings A together to get a string in the new language

Theorem 1.25: (pg.46)
The class of REG is closed under the union operation.
Theorem 1.26: (pg.47)
The class of REG is closed under the concat operation.

- Nondeterministic: several choices may exist for the next state at any point.
- unary alphabet: an alphabet that contains only one symbol.
- Formal definition of NFA: page 53
- page 59 - 63
- Regular expressions: regular operations to build up expressions describing languages.
- The value of a regular expression is a language.
- Definition 1.52 (pg.64)
- The expression ε: contains a single string named the empty string
- $\varnothing$ represents a language that does not contain any strings.
- circular definition: defining the notion of a regular expression in term of itself.
- inductive definition: define regular expression in terms of smaller regular expression.

Theorem 1.54: (pg.66)
A language is regular if and only if some regular expression describes it.

Theorem 1.70 (pg. 78) ??

CHAPTER 0

- A set is a group of objects represented as a unit.
- the objects in a set are called its elements or members.
  - S = {1, 4}
- For two sets A and B, we say that A is a ***subset*** of B, written $A \subseteq B$, if every member of A also is a member of B
- A is a ***proper subset*** of B, written A B, if A is a subset of B and not equal to B.
- An infinite set contains infinitely many numbers
- Natural Numbers: N = {1,2,3,....}
- Integer: Z= {..., -3,-2,-1,0,1,2,3,...}
- Empty set is set with 0 member $\varnothing$
- A set with one member is sometimes called a ***singleton set***
- A set with two members is called an ***unordered pair***.
- A ***sequence*** of objects is a list of these objects in some order.
  - Ex: (7,21,5)

- Finite sequences are called ***tuples***.
- A 2-tuple is also called an ***ordered pair***
- The power set of A is all the subset of A
- If A and B are two sets, the ***Cartesian product*** or ***cross product*** of A and B, A × B, is the set of all ordered pairs wherein the first element is a member of A and the second element is a member of B.
- A ***function*** is an object that sets up an input–output relationship, also called as ***mapping.***
- The set of possible inputs to the function is called its ***domain***.
- The outputs of a function come from a set called its ***range***
  - f: D-> R
- A function that does use all the elements of the range is said to be ***onto*** the range
- ***infix notation***, with the symbol for the function placed between its two arguments
- ***prefix notation***, with the symbol preceding
- A ***predicate*** or ***property*** is a function whose range is {TRUE, FALSE}
- A property whose domain is a set of k-tuples A × · · · × A is called a ***relation***, a ***k-ary relation***, or a ***k-ary relation on*** A
- Equivalence relations:
  R is ***reflexive*** if for every x, xRx;
  R is ***symmetric*** if for every x and y, xRy implies yRx; and
  R is ***transitive*** if for every x, y, and z, xRy and yRz implies xRz.
- An ***undirected graph***, or simply a ***graph***, is a set of points with lines connecting some of the points.
- The points are called ***nodes/vertices***. The lines are called ***edges***.
- The number of edges at a particular node is the ***degree*** of that node
- ***self-loop***: an edge from a node to itself.
- G is a ***subgraph*** of graph H if the nodes of G are a subset of the nodes of H, and the edges of G are the edges of H on the corresponding nodes.
- 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.
- ***alphabet*** to be any nonempty finite set
- The members of the alphabet are the ***symbols*** of the alphabet
- A ***string over an alphabet*** is a finite sequence of symbols from that alphabet
- The string of length zero is called the ***empty string*** and is written ε
- ***Definitions*** describe the objects and notions that we use
- A ***proof*** is a convincing logical argument that a statement is true.
- A ***theorem*** is a mathematical statement proved true.

Chapter 0.

- Three traditionally central areas of the theory of computation are

  Automata, Computability, and Complexity.

- Complexity theory has pointed cryptographers in the direction of computationally hard problems around which they have designed revolutionary new codes.

- Kurt Godel, Alan Turing, and Alonzo Church are mathematicians, discovered that certain basic problems cannot be solved by computers.

- In Complexity theory, the objective is to classify problems as easy ones and hard ones.
  In Computability theory, the classification of problems is by those that are solvable and those that are not.

- Computability theory introduces several of the concepts used in complexity theory.

- Automata theory deals with the definitions and properties of mathematical models of computation.

- The finite automaton is used in text processing, compilers, and hardware design.

- The context-free grammar, is used in programming languages and artificial intelligence.

* Singleton

Different between

* Set, Subset, Proper subset, Power set

  Assume R be a set $R = \{1, 2, 3\}$

  - Subset $\{1\} \subseteq R$, $\{2\} \subseteq R$, $\{3\} \subseteq R$
  - Proper subset $A = \{1, 2, 3\} \in R$, A is a proper set of R, $A \subseteq R$
  - Power set contain all subset of set

    $P(R) = \{\emptyset, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}\}$.

* Domain is the set of all possible input to a function f(n)

  Range is the output of function f(n)

* Empty Set = $\{\emptyset\}$, Empty String = $\{\}$ or ε
* Function is a object that set up an input and output relationship.

  Relation: A predicate, most typically when the domain is a set of k-tuples.

Formal Definition.

* A sequence of objects is a list of these objects in some order.

  (7, 21, 57) sequence

  The order doesn't matter in a set, but in a sequence it does.

A Sequence can be finite or infinite. Finite Sequence called tuples

infinite sequence called k-tuple.

* A special type of binary relation is called an equivalence relation.

  A binary relation R is an equivalence relation if R satifies 3 conditions:

  reflexive; symmetric and transitive

Chapter 1.

DFA

- A finite automaton is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$, where
  1. Q is a finite set called the States,
  2. Σ is a finite set called the alphabet,
  3. $\delta : Q \times \Sigma \longrightarrow Q$ is the transition function,
  4. $q_0 \in Q$ is the ~~state~~ start state, and
  5. $F \subseteq Q$ is the set of accept states.

NFA

- A nondeterministic finite automaton is
  1. Q is a finite set of states,
  2. Σ is a finite alphabet.
  3. $\delta : Q \times \Sigma_\varepsilon \to \mathcal{P}(Q)$ is the transition function
  4. $q_0 \in Q$ is the start state, and
  5. $F \subseteq Q$ is the set of accept states.

Both DFA and NFA have states, an input alphabet, a transition function, a start state, and a collection of accept state.

However, they differ in the type of ~~transition~~ function.

- In DFA, the transition function takes a state and an input symbol and produces the next state.
- In NFA, the transition function takes a state and an input symbol or the empty string and produces the set of possible next states.

1.4

Language: A set of a strings that some finite automaton recognizes it
Regular language is a language that is ~~defined~~ recognized by finite automaton.
Let A and B be languages. We define the regular operations Union, concatenation and star as follows.

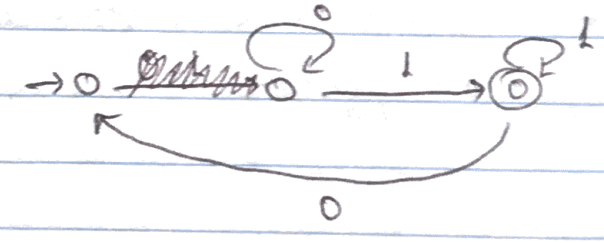
* Union: $A \cup B = \{x \mid x \in A \text{ or } x \in B\}$
* Concatenation: $A \circ B = \{xy \mid x \in A \text{ and } y \in B\}$
* Star: $A^* = \{x_1 x_2 \ldots x_k \mid k \geq 0 \text{ and each } x_i \in A\}$.

Pumping lemma is a technique for proving nonregularity stems from a theorem about regular languages.

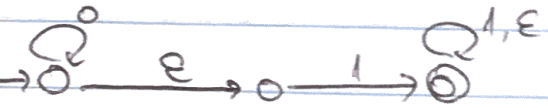
Pumping length is the property states that all strings in the languages can be "pumped" if they are at least as long as a certain special value.

$\Sigma = \{0, 1\}$ w is a strings of the form 0^*1^* $\quad 0^*1^* = \ldots 0 \ldots 1 \ldots 1$

DFA:



NFA:



② CFG: Context-free Grammer is a 4-tuple (V, Σ, R, S)

V is a finite set called varrable

Σ is a finite set of varrable V, called Terminal

R is a finite set of rules.

S is the state varrable.

PDA: Pushdown Automata is a 6-tuple $(Q, \Sigma, T, \delta, q_0, F)$

Q is the set of states.

Σ is the input alphabet.

T is the stack alphabet.

δ is the transition function

q_0 is the start state

F is the set of accept states.

R: Regular Expression ~~if R is~~

$\{a\}$

$\{\varepsilon\}$

$\emptyset$

$(R_1 \cup R_2)$, where R_1 and R_2 are regular expressions.

$(R_1 \circ R_2)$, where ——— u ———

(R_1^*), where R_1 is a regular expressions.

Regular Language is a language that is recognized by finite automata

Regular Expression is a special string for describing a search pattern

Regular Operations are Union, Concatenation and Star

- Union · $A \cup B$
- Concatenation $A \circ B$
- Star A^*

} where A & B be Languages.

④ $M = \{Q, \Sigma, \delta, q_0, F\}$

$Q = \{q_1, q_2, q_3, q_4\}$

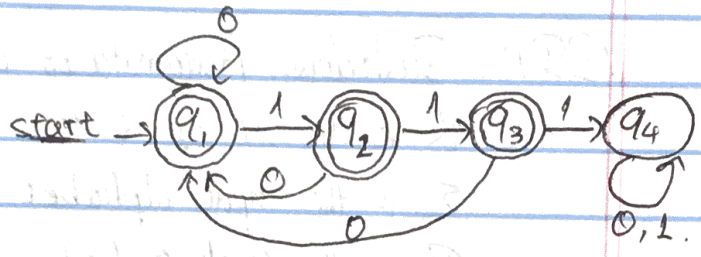
$\Sigma = \{0, 1\}$

$\delta =$

	0	1
q_1	q_1	q_2
q_2	q_1	q_3
q_3	q_1	q_4
q_4	q_4	q_4

$q_0 = q_1$ (start state)

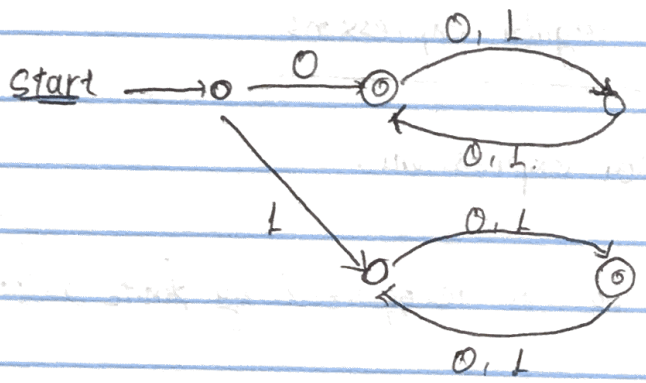
$F = \{q_1, q_2, q_3\}$ (final state)



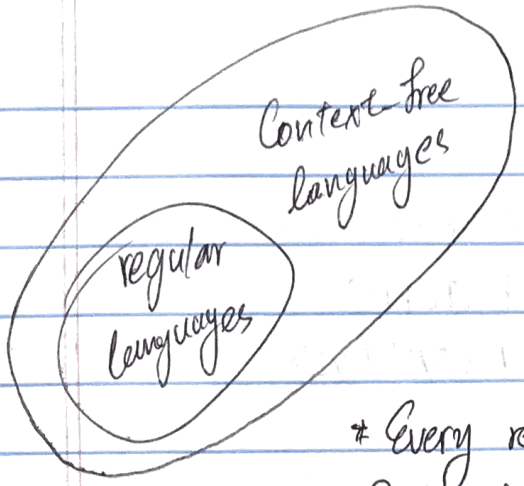
$\{w \mid w$ doesn't contain the substring of $111\}$

⑤ $L = \{w \mid w$ has at most two 1's $\}$.

③



$\{w \mid w$ starts with a 0 and has an odd length or starts with a 1 and has an even length.



Relationship of the regular and context-free languages

* Every regular language is context-free grammer
But not every context-free grammer is regular language.

DPDA: A deterministic pushdown automaton is a 6-tuple $(Q, \Sigma, \Gamma, \delta, q_0, F)$ where Q, Σ, Γ, and F are all finite sets.

- Q is the set of states
- Σ is the input alphabet
- Γ is the stack alphabet
- δ is the transition function
- q_0 is the state state
- F is the set of accept states

The transition function δ must satisfy these condition.

$q \in Q$; $a \in \Sigma$; and $x \in \Gamma$

$\delta(q, a, x)$, $\delta(q, a, \varepsilon)$; $\delta(q, \varepsilon, x)$, and $\delta(q, \varepsilon, \varepsilon)$ is not $\emptyset$.

Grammar G - (V, Σ, R, ⟨EXPR⟩)

V is {⟨EXPR⟩, ⟨TERM⟩, ⟨FACTOR⟩} and Σ is {a, +, x, (,)}.

The rules are ⟨EXPR⟩ → ⟨EXPR⟩ + ⟨TERM⟩ | ⟨TERM⟩

⟨TERM⟩ → ⟨TERM⟩ x ⟨FACTOR⟩ | ⟨FACTOR⟩

⟨FACTOR⟩ → (⟨EXPR⟩) | a