Basic Concepts & Finite Automata

Applications and Limitations of FA

- Applications
  - Text editor for RE
  - Lexical Analysis
- Limitations
  - Well formed parenthesis
  - Palindrome check

Finite State Machine (FSM) with Output

- Moore Machine
  - DFA and NFA are machines that specify YES or NO by indicating Final and Non-Final state respectively
  - Next state is defined by Current State and Current Input Symbol
  - If the length of the input string is n, then length of the output string is n + 1
  - Formal Definition
    - M = (Q , Σ , Δ , δ , λ, q0)
    - Where Q is a finite set of internal states
    - Σ is a finite set of symbols called the input alphabet
    - Δ is Output Alphabet
    - δ : Q X Σ → Q is a Transition Function (State Function)
    - q0 is an initial state q0 ε Q
    - λ : Q → Δ is a output Function
  - Examples (9)
- Mealy Machine
  - The output symbol is dependent on the Present State and Present Input Symbol
  - If the length of the input string is n, then length of the output string is n
  - Formal Definition
    - M = (Q , Σ , Δ , δ , λ, q0)
    - Where Q is a finite set of internal states
    - Σ is a finite set of symbols called the input alphabet
    - Δ is Output Alphabet
    - δ : Q X Σ → Q is a Transition Function (State Function)
    - q0 is an initial state q0 ε Q
    - λ : Q X Σ → Δ is a output Function
  - Examples (8)
- How to differentiate
  - Output
    - Associated with States — Moore
    - Associated with Transitions — Mealy
  - Length of Output
    - n + 1 — Moore
    - n — Mealy
  - Output Function
    - λ : Q → Δ — Moore
    - λ : Q X Σ → Δ — Mealy

Minimization of DFA

- Distinguishable States
  - If for some input string w, δ (p, w) gives an accepting state and δ (q, w) gives a non accepting state or vice versa
  - Then p and q are distinguishable or non – equivalent state
- Indistinguishable States
  - If for some input string w, δ (p, w) and δ (q, w) both produces either accepting state or non accepting state
  - Then p and q are indistinguishable or equivalent state
- Process
  - For an example of how to draw all pairs refer to MinOfDFA.png
  - Step 1: Draw a table for all pair of states (P, Q)
  - Step 2: Mark all pairs where P belongs to set of Final States and Q does not belong to set of Final states
  - Step 3: If there are any unmarked pairs (P, Q) such that [δ (P, x) , δ (Q, x)] is marked then mark [P, Q] where x is an input symbol
  - Step 4: Repeat step 3 until No more marking can be made
  - Step 5: Combine all the unmarked pairs and make them single state in minimized DFA

Conversion from NFA to DFA

- If Epsilon closure is present first eliminate that
- Step 1: The start state of NFA is q0. It will act as initial state of DFA i.e. qo'.
  - Hence q0 will be part of Q'.
  - Find all the transitions from the start state
- Step 2: For each state { q0, q1, q2, … , qi } in Q' the transition for each input symbol Σ can be obtained as
  - I. δ' ({ q0, q1, q2, … , qi } , a) = δ (q0 , a) U δ (q1 , a) U δ (q2 , a) … δ (qi , a) = { q1, q2, … qk } (May be some state)
  - II. Add the state { q1, q2, … qk } to DFA if it is not already in Q'
  - III. Find the transitions for every input symbol from Σ for new state { q1, q2, … qk }.
    - If we get some new state { q1, q2, … qn } which is not in Q' , then add it to Q'
    - New states can be composed of multiple states of the NFA
  - IV. If no new state is generating then stop the process after finding all the transitions.
- Step 3
  - For the state { q1, q2, … , qn } ∈ Q' of DFA if any one state qi is a final state of NFA then { q1, q2, … qn } becomes final state
  - States that cannot be reached from the initial state may be discarded

Nondeterministic (NFA)

- Next state can not be determined uniquely for a given transition with current state and current input
- The Finite Automata is called Non – Deterministic Finite Automata (NFA) when there exists many paths for a specific input from current state to next state.
- Difference in Transition Function
  - Transition Function accepts two parameters one is current state and other is input symbol
  - It is described as δ : Q X Σ → 2^Q
  - Here the power set of Q (2^Q) has been taken because in case of NFA, from a state, transition can occur to any combination of Q states
- Examples (12)
- Epsilon Transition
  - Please refer to first 3 examples to clear your concept in case you haven't already
  - The ε - transition in NFA are given in order to move from one state to another without having any symbol from input set Σ
  - Examples (3)
- Epsilon Closure
  - The epsilon closure (ε - closure) is set of all states which are reachable from p on transitions such that
  - ε - closure (p) = p where p ∈ Q
  - If there exist ε - closure (p) = { q } and δ (q , ε) = r then ε - closure (p) = { q , r }
- Eliminating Epsilon Closure
  - Step 1: Find ε - closure of all states for given NFA
  - Step 2: Find δ' Transition function for NFA without ε
  - Step 3: Construction of Transition Table (δ')
  - Step 4: Construction of Transition Diagram
  - Example
    - δ' (q0 , 0) = ε - closure (δ (ε - closure (q0) , 0))
  - Hint: Will have more final states

Basic Definitions

- Alphabet — Non empty set Σ of objects called symbols
- String — A finite set of sequence from the alphabet
- Σ* — If Σ is an alphabet then we use Σ* to denote the set of strings obtained by concatenating zero or more symbols from Σ
- Language — Language is collection of appropriate strings. It is generally subset of Σ*
- Sentence — A string in a language L will be called as a sentence of L

Operations on Languages

- Union — L1 U L2
- Intersection — L1 ∩ L2
- Difference — L1 - L2
- Concatenation — L1 . L2
  - L1 . L2 = {x . y | x ∈ L1 and y ∈ L2}
  - Special case: L^n, L concatenated with itself n times
- Complement — L (bar) = Σ* - L
- Reverse — L^R: It is the set of all string in reverse format
- Star Closure — L*: L^0 U L^1 U L^2 U L^3 U....
- Positive Closure — L ^ +: L^1 U L^2 U L^3 U....
- Examples
  - Σ = {a, b}
  - L1
    - {a.b^n | n>=0}
    - {a, ab, ab^2, ab^3....}
  - L2
    - {a^n b^n | n>= 0}
    - {ab, a^2 b^2, a^3 b^3....}
  - L3
    - {a^m b^n | n>= 0}
    - {b, ab, a^2 b, a b^2, a^3 b, a^2 b^2.....}
- Operations
  - L1 U L2 — {a, ab, a b^2, a^2 b^2....}
  - L1 ∩ L2 — {ab}
  - L1 - L2 — {a, ab^2, ab^3....}
  - L1 . L2 — {aab, aaab, abab, abaabb,...}
  - L1 ^R — {a, ba, b^2 a, b^3 a....}
  - L2 ^R — {ba, b^2 a ^2, b^3 a ^3....}

Finite Automata

- They are used to represent Finite State Machines
- Representation of FSM
  - Input Tape — Linear tape having number of symbols. Each cell has one symbol
  - Tape Reader — It reads symbols from left to right one at a time
  - Finite Control — It decides the next state on receiving a particular input
- What is Finite Automata?
  - They are models used for computers that have extremely limited memory
  - A finite automata is a collection of five tuples (Q, Σ, delta (δ) , q0 , F)
    - Q is a finite set of states
    - Σ is the input alphabet
    - δ: Q × Σ→Q is a Transition Function or Mapping Function
    - q0 is the initial state, q0 ∈ Q
    - F is a set of finite states
- Why do we use Finite Automata?
  - We use it to model the behavior as present in various electromechanical devices that use a limited amount of memory
- How to use Finite Automata?
  - State Diagram
  - Transition Table
  - Refer to FA1.png
  - If A is a set of all strings that a machine accepts, we say A is the language of machine M and write L(M) = A
  - A machine can accept multiple strings but understands only one language

Deterministic Finite Automata (DFA)

- Each transition in this automata is uniquely determined on current state and current input
- The Finite Automata is called Deterministic Finite Automata (DFA) if there is only one path for specific input from the current state to the next state.
- Examples (15)