Compilers  
Lexical Analyzer Generator

28/3/2020

# data structures

## Node

Node is a class used to represent states in both NFA and DFA and it consists of:

* A String to save name of the state.
* An Integer to save value of the priority of this state if it’s final or zero it it’s not final.
* Vector of pointers of edges which contain all edges that go out from this state.

## Edge

Edge is a class used to represent transitions between states in both NFA and DFA and it consists of:

* A Pointer to destination node.
* Two characters: start and end alphabets that are the range of the allowed input transitions.
* A Set of characters that are disallowed to be the input of a transition by this edge.

## NFA

NFA is a class to represent the NFA as a graph and it has:

* Two pointers on start and end nodes of the NFA graph.
* Set of pointer on nodes for all final states in this NFA.

## DFA

DFA is a class to represent the DFA as a graph and it has:

* A Pointer to the start state node.
* A map its key is pointer on the DFA state and its value is another map, this map of key character “the input symbol” and the value is a pointer to a node “2nd DFA state which accessed by that input symbol”.

# Algorithms

## Build NFA from RE

|  |  |
| --- | --- |
|  | Describe THOMPSON’S CONSTRUCTION. |

## Converter from NFA to DFA

|  |  |
| --- | --- |
|  | To convert from NFA to DFA, Subset Construction Algorithm is used. So there is one main function to convert it and two other sub functions.   * Closure: It takes a set of nodes and return a set of all nodes that can be reached by epsilon transition. * Move: it takes a set of nodes and a char symbol and return a set of all nodes that can be reached by this symbol transition. * Convert: It takes a NFA and a set of alphabets used and return a DFA. while (there is an unmarked DFA state in mark) {  mark DFA state = true;  for (each input symbol c in alphabet) {   u = closure(move(T, a));  D\_table[DFA state, c] = u;  if (U is not in DFA states) {  add u as unmarked state to DFA states;  }  }  } So while building D\_table, DFA graph is being built to. |

## DFA Minimizer

|  |  |
| --- | --- |
|  | Describe Minimizer. |

## Maximal Munch

|  |  |
| --- | --- |
|  | Describe finding longest prefix. |

# minimal dfa table

# stream of tokens

# assumptions

# Team members

|  |  |
| --- | --- |
| Name | ID |
| Arsany Atef |  |
| Kirolos Malak Habib |  |
| Michael Said |  |
| Yomna Gamal El-Din Mahmoud | 63 |