MUST

Wisdom & Virtue

MIRPUR UNIVERSITY OF SCIENCE AND TECHNOLOGY (MUST), MIRPUR

DEPARTMENT OF SOFTWARE ENGINEERING

Formal Methods in Software Engineering

Lecture [8]: Automata Theory

Engr. Samiullah Khan

(Lecturer)

Topics discussed in Today's Lectures

- Automata Theory

Automata Theory

- Automata are abstract mathematical models of machines that perform computations on an input by moving through a series of states or configurations
- Automata theory is related to formal language, since there is a correspondence b/n certain families of automata & classes of languages generated by grammar formalisms
- A language is accepted by an automaton when it accepts all the strings in the language
- **Pushdown automata** involve a pushdown store: a sequence in which symbols can only be added and removed from one end, with the effect that the first symbols in, are the last ones out.

Deterministic Finite State Machine or Deterministic Finite Automaton (DFA)

- It is a finite state machine where for each pair of state and input symbol there is a deterministic next state.
- A DFA is a 5-tuple, (S, Σ, T, s, A), consisting of:
  - S = A finite set of states
  - Σ = A finite set of Symbols called Alphabet (0s,1s), input the machine will use
  - T = A Transition Quick Facts about: Transition function
    - A mathematical relation such that each element of one set is associated with at least one element of another set function (T: S × Σ → S)
  - s = A start state (s . S)
  - A = A set of accept states (A . S)

Deterministic Finite State Machine or Deterministic Finite Automaton (DFA)

Let M be a DFA such that M = (S, Σ, T, s, A), and X = x1x2 ... xn be a string over the alphabet Σ. M accepts the string X if a sequence of states, r0,r1, ..., rn, exists in S with the following conditions:

1. r0 = s
2. ri+1 = T(ri, xi), for i = 0, ..., n-1
3. rn ∈ A.

As shown in the first condition, that machine starts in the start state s. The second condition says that given each character of string X, the machine will transition from state to state as ruled by the transition function T. The last condition says that the machine accepts if the last input of X causes the machine to be in one of the accepting states. Otherwise, it is said to reject the string. The set of strings it accepts form a language, which is the language the DFA recognizes.

Non-Deterministic Finite State Machine or Non-Deterministic Finite Automaton (DNFA)

- Is a finite state machine where for each pair of state and input symbol there may be several possible next states.
- A NFA is a 5-tuple, (S, Σ, T, s, A), consisting of:
  - S = A finite set of states
  - Σ = A finite set called the alphabet, (0,1 etc), input the machine will use
  - T = A Transition Quick Facts about: Transition function
    - A mathematical relation such that each element of one set is associated with at least one element of another set function (T: S × Σ → P(S))
  - s = A start state (s . S)
  - A = A set of accept states (A . S)

where P(S) is the power set of S and ε is the empty Quick Facts about: string A linear sequence of symbols (characters or words or phrases) string.

Non-Deterministic Finite State Machine or Non-Deterministic Finite Automaton (NDFA)

Let M be an NFA such that M = (S, Σ, T, s, A), and X be a string over the alphabet Σ that can be written as x1x2 ... xn . M accepts the string X if a sequence of states,
r0,r1, ..., rn, exists in S with the following conditions:

1. r0 = s
2. ri+1 ∈ T(ri, xi), for i = 0, ..., n-1
3. rn ∈ A.

The machine starts in the start state and reads in a string of symbols from its alphabet. It uses the transition relation T to determine the next state(s) using the current state and the symbol just read or the empty string. If, when it has finished reading, it is in an accepting state, it is said to accept the string, otherwise it is said to reject the string. The set of strings it accepts form a language, which is the language the NFA recognizes.

Difference b/w DFA and NDFA

DFA	NDFA
The transition from a state is to a single particular next state for each input symbol. Hence it is called deterministic.	The transition from a state can be to multiple next states for each input symbol. Hence it is called non-deterministic.
Empty string transitions are not seen in DFA.	NDFA permits empty string transitions.
Backtracking is allowed in DFA	In NDFA, backtracking is not always possible.
Requires more space.	Requires less space.
A string is accepted by a DFA, if it transits to a final state.	A string is accepted by a NDFA, if at least one of all possible transitions ends in a final state.

THANKS