Theory of Computation (CS F351)

BITS Pilani
Hyderabad Campus

Prof. R.Gururaj
CS&IS Dept.

Finite Automata

(Chapter-2)

Concepts

innovate achieve lead

1. We look at the models of Computations and devices for accepting and generating languages.

2. A restricted model of a computer is – a *Finite Automaton or a Finite state Machine*

3. *The FA shares one common property with a real computer- Both got a CPU of fixed and finite capacity.*

What a finite automaton does:

- It accepts a string as an input, delivered to it on an input tape.
- It produces no output.
- But it gives an indication of whether the input string is approved or disapproved or accepted or rejected.

We can see it as a language recognition device.

Hence an *Automaton* is a Machine designed to respond to encoded instructions; a robot.

Uses:
Automata are applicable to algorithms and computer programs.

Ex: 1→ The lexical analysis process which identifies program units like operators, identifiers, constants etc., is often based on the simulation of Finite Automaton

Ex : 2→ The problem of finding the occurrences of a string within the other

Operations of a Finite Automaton

Strings are fed into the device through input tape.

1. The tape is divided into squares where we can write a single symbol.
2. The main part of the machine is called as *finite control*.
3. At any given instance of time the *finite control* can be at one of the specified finite no. states.
4. This finite control can sense the symbols written on the tape, with a movable Read head.

Initially the RH is placed at the leftmost square of the tape.

Then the FC is in initial state.

At regular intervals automaton reads one symbol from the input tape then enters a new state depends only on the current state and the symbol read.

That is why this machine is called by the deterministic finite automaton.

After reading an input symbol, the read head moves one square to the right on the tape so that on the next move it will read the symbol in the next square.

This process is repeated again and again-

- read symbol,

- move the head to the right,

- change the state of the FC

At some point , the reading head reaches the end of the input string.

The automaton then indicates the acceptance or rejection of what it has read, by its state at the end.

If the state at the end is one among the final states then it is accepted.

The language accepted by the machine is set of all strings it accepts.

Formal definition

A Deterministic Finite Automaton is a quintuple- $(K, \sum , \delta, s, F)$

K is a finite set of states

$\sum$ is an alphabet

$s \in K$ is the start state

F subset of K is the set of final states

δ *is the transition function from* $K \times \sum \rightarrow K$

Rules of transition for a automaton *M* :

Rules are encoded into the transition function.

Thus the automaton M in state q ∈ *K* *and symbol read from the input tape is*

a ∈ ∑, *the transition function is given as* *δ(q, a)* ∈ *K is the uniquely determined state to which the M passes.*

Since the automaton is not allowed move its head backwards, it can not visit the part of the tape which is already read.

Hence the already read part of the string will not influence the future of the machine.

The language accepted by a automaton *M*, *L(M)*, is the set of all strings accepted by *M*

Ex 1: Let M be the deterministic finite automaton ($K, \sum, \delta, s, F$)

Where $K=\{q_0, q_2\}$

$\sum = \{a, b\}$

$s = \{q_0\}$

$F = \{q_0\}$

And δ is the function tabulated as follows

State (q)	Next symbol(σ)	$\delta(q, \sigma)$
q_0	a	q_0
q_0	b	q_1
q_1	a	q_1
q_1	b	q_0

L(M) is the set of all strings in {a, b}*, that have an even number of *b*s

If the M is given the string *aabba*

The initial *configuration* is $(q_0, aabba)$ *then we have*

1. $(q_0, aabba) \rightarrow (q_0, abba)$
   $\rightarrow (q_0, bba)$
   $\rightarrow (q_1, ba)$
   $\rightarrow (q_0, a)$
   $\rightarrow (q_0, e)$

Therefore $(q_0, aabba)$ *after multiple moves reaches* (q_0, e)

Hence the string aabba is accepted by M

Ex 2: Check if the string *baabbb* is accepted, the previous *M* or not

Deterministic Finite Transducer

Is a device much like DFA, except that its purpose is not to accept strings or languages.

But to transform input string into output strings

It starts in a designated state and moves from state to state depending on the input like DFA.

On each step however it emits a string of zero or more symbols depending on the current state and input symbol.

State diagram is similar to that of a DFA except that the label on an arrow will have *a/w,* meaning if the input symbol is *'a'* follow this arrow and emit *'w'.*

Mealy Machine

Introduced by George H Mealy in 1955.

Is a type FA with out put where the output depends on current state and symbol.

Moore Machine

Introduced by Edward F Moore in 1956.

Is a type FA with out put where the output depends on current state only.

NDFA

1. From a given state there may be more than one transition possible on a symbol.
2. From a given state, on a symbol no transition defined.
3. From a given state we may have a transition on null (e).

If at least one of the above is satisfied the FA is a Non Deterministic Finite Automaton.

Conversion from NDFA to DFA.

- Without Null transitions.
- With null transitions. (We compute E(q) for all qϵK.)

State minimization in a DFA

1. Simplest kind of minimization is eliminating unreachable states.
2. Next, we identify equivalent states.

Two states q_1 and q_2 are equivalent $q_1 \equiv q_2$ if both $\delta(q_1,x)$ and $\delta(q_2,x)$ are final states or non-final states, for all $x \in \sum^*$.

K-equivalence of two states

Two states q_1 and q_2 are k-equivalent ($k \geq 0$) if both $\delta(q_1,x)$ and $\delta(q_2,x)$ are final states or non-final states, for all $x \in \sum^*$ and length $\leq k$.

Hence any two final / non-final states are 0-equivalent.

FA and Regular Expressions

- It is understood that the class of languages accepted by FA remains same even with non-determinism.
- The class of languages accepted by FA are stable.
- Regular Languages are the languages accepted by both DFA and NDFA.
- For every NDFA there exists a DFA.
- Class of languages accepted by both DFA & NDFA are same.

Regular Languages

Regular Languages are closed under :

1. Union
2. Intersection
3. Concatenation
4. Complementation
5. Kleene star

Constructing FA from RE

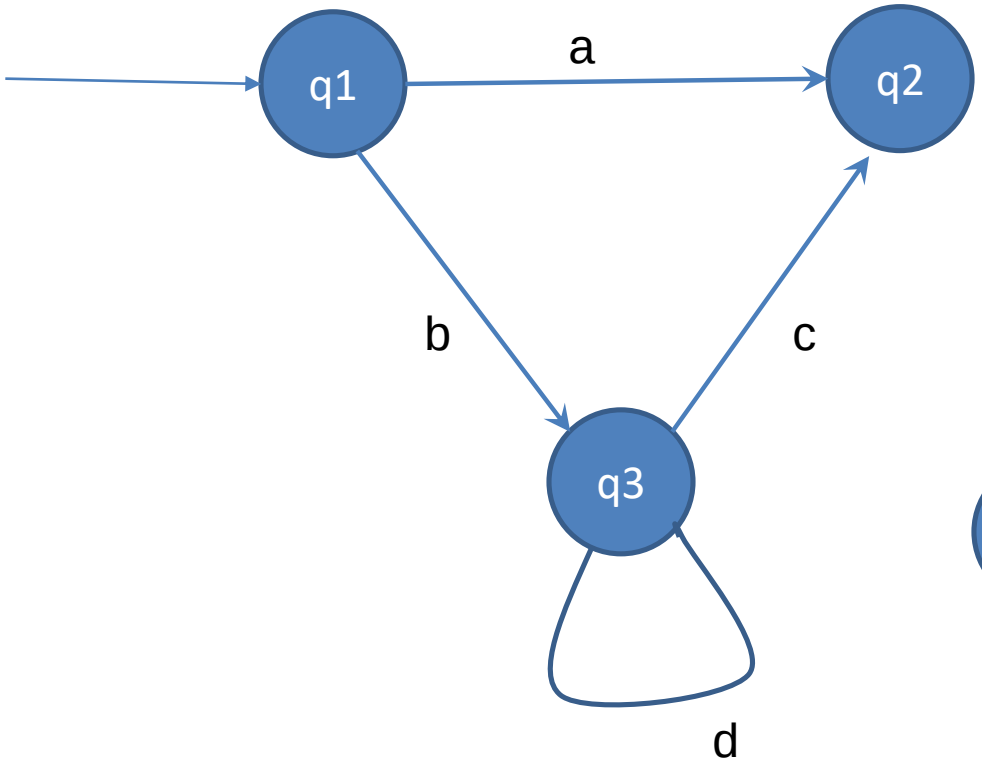
Section 2.3 of Ch.2 of T1 and as per class notes.

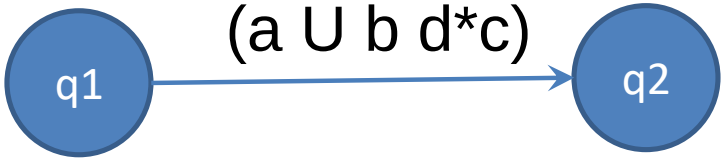
Constructing RE from FA

Section 2.3 of Ch.2 of T1.

For constructing RE for a given NDFA, things will be simplified if we assume that the Non-deterministic Finite Automaton M has following simple properties.

1. It has a single Final state.
2. There is no transition into initial state and no transition leaving the final state.


q1
a
q2
b
c
q3
d


q1
(a U b d*c)
q2

Languages that are not Regular

- It is understood that the Regular Languages can be represented by Res or by DFD or NDFA.
- We scan strings from left to right.
- Regular languages with infinite number of strings are represented by FA with cycles and Res invoking Kleene star. We use simple repetitive structures.
- DFA have memory restrictions.
- We have pumping theorem to prove certain languages are not regular.
- But Pumping theorem cannot be used to prove that a language is regular.
- It is proof by contradiction.

Pumping Lemma for RLs

Pumping Lemma is a necessary condition for a string to be in RL or regular Set.

Pumping means generating or taking out.

This necessary condition is used to prove that certain sets (PLs) are not regular.

If any set fulfills all the conditions of pumping Lemma, it can not be said that it is regular.

But reverse is true, if any set breaks the condition then it is not Regular.

Pumping Theorem for RL

Let M=(Q, ∑, δ, q_0, F) be a finite automaton with *n* number of states.

L be regular set (Language) accepted by M.

Let *w* be a string that belongs to L and $|w| \geq m$.

If $m \geq n$ i.e., the length of the string is greater than the number of states.

Then there exists x,y,z such that $w=xyz$ where $|xy| \leq n$, $|y|>0$, and $xy^q z \in L$ for each $q \geq 0$.

If L is Regular

$\exists N \geq 1$, such that

$\forall$ strings $w \in L$, where $|w| \geq 1$

$\exists x,y,z,$ such that $w=xyz$

and $|xy| \leq N$

and $y \neq 0$

for all $q \geq 0$, $xy^qz \in L$

Summary (Ch.2 of T1)

- Finite Automata
- DFA
- Moore Machine
- Mealy Machine
- NFA
- NFA to DFA Conversions
- State Minimization
- Constructing FA from RE
- RE from FA
- Pumping theorem for Regular languages