Lexical Analysis.

1

Dr. D. M. Akbar Hussain
Department of Electronic Systems

Lexical Analysis.

Basic Idea:
Read the source code and generate tokens, it is similar what humans will do to read in; just taking on the input and breaking it down in pieces.

Each token is a sequence of characters representing some information.

2

Dr. D. M. Akbar Hussain
Department of Electronic Systems

Lexical Analysis.

Possible Types of Token:

- Keywords: if, while, ...
- Special Symbols: +, *, <, >= .....
- Identifiers: User defined strings of characters .....

Scanning is a special case of pattern matching:

Method used are *Regular Expressions* and *Finite Automata*.

3

Dr. D. M. Akbar Hussain
Department of Electronic Systems

Lexical Analysis.

It is simple but time consuming process:

- Must be efficient
- Responsibility is to recognize tokens nothing else:
  - int foo = 123;
  - foo int 123 =;

What is the out come ?

Is this its job ? (Syntax analysis.)

4

Dr. D. M. Akbar Hussain
Department of Electronic Systems

Lexical Analysis.

Lexical analysis may also perform one or more of the following:

- Deleting comments.
- Inserting line numbers.
- Evaluating constants. (this is controversial, whether it should be done here or not)

5

Dr. D. M. Akbar Hussain
Department of Electronic Systems

Scanning Generator

```
#include
.....
#include

main()
{
char in;
        in = getch ( );
        if ( isalpha (in) )
                in = getch ( );
        else
                error ();
        while ( isalpha (in)|| isdigit (in) )
                in = getch ( );
}
```

For an identifier

6

Dr. D. M. Akbar Hussain
Department of Electronic Systems

Regular Expressions (RE)

- RE represent pattern of strings composed of characters.
- It is a very convenient method to represent identifiers and constant.

A language is a set of strings.
String is finite sequence of symbols.
Symbols themselves are taken from a finite alphabet.

7

Dr. D. M. Akbar Hussain
Department of Electronic Systems

Regular Expressions (RE)

RE r is defined by a set of strings with which it matches: L(r)

Which means r contain characters (could be alphabet + special symbols e.g. meta-characters).

Some Basic RE:

- L(x) = {x} Expression matches x.
- L(ε) = {ε} Empty string with no characters.
- L(Ø) = { } Ø matches no string (empty set).

8

Dr. D. M. Akbar Hussain
Department of Electronic Systems

RE Operations

1. Choice (alternation)
2. Concatenation
3. Repetition

Choice: Given two RE M and N, it is written with the alternation operator | (vertical bar) as M|N.

A string is in the language M|N if it is in the language M or in the language N.

It is a union, e.g., for a language a|b it contains strings a and b.

L(a|b) = L(a) U L(b) = {a,b}

9

Dr. D. M. Akbar Hussain
Department of Electronic Systems

RE Operations

Concatenation: Given two RE M and N, concatenation operator makes a new RE M.N. Which means a string is in the language M.N if it is the concatenation of any two strings a and b such that a is in language M and b is in the language N.

e.g.: (a|b).a defines the language containing aa and ba.

e.g.: S1 = {aa, b}, S2 = {a, bb}

S1.S2 = {aaa, aabb, ba, bbb}

10

Dr. D. M. Akbar Hussain
Department of Electronic Systems

RE Operations

Repetition: Given a RE M its repetition is M* (also called kleene closure).

A string is in the language M* if it is concatenation of zero or more strings all of which are in M.

e.g.: a* matches ε, a, aa, aaa, aaaa, ...

a* = {ε}UaUaaUaaaU.....

$$a^* = \bigcup_{n=0} a^n \qquad a^0 = \varepsilon$$

Basically it is infinite set of union. Remember each element is a finite concatenation of strings.

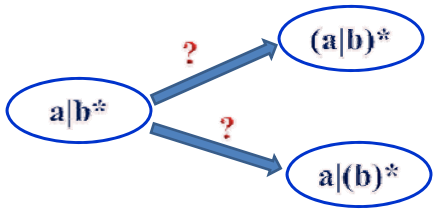
e.g. (a|bb)* = ε, a, bb, aa, bbb, aaa, abb, bbaa, ...

e.g. (0|1)*.0 = 0, 00,10,010, ... (Binary numbers)

11

Dr. D. M. Akbar Hussain
Department of Electronic Systems

RE Operations



(a|b)* → ε, a, b, aa, bb, ab, abb, ...

a|(b)* → ε, a, b, bb, bbb, ...

*** Has the highest precedence then concatenation and alternation**

Some short cuts (abbreviations):

ab|c means (a.b)|c (a|) mean (a|ε)

[abcd] mean (a|b|c|d)

[b - g] mean [bcdefg]

12

Dr. D. M. Akbar Hussain
Department of Electronic Systems

RE Operations Extension

Why it is required:

Example: a* ➡ repetition zero or more, this could be problem for natural numbers, we need to have one instance in which it guarantees that we have a match, disallowing the empty string.

Binary number example: (0 | 1)* ➡ ε, 0, 00, 11, ...

Although, we can solve it by: (0 | 1) (0 | 1)*

But this is a simple situation and it is easy, but things may not be that simple in a language.

So the solution is **M^+ ⟶ (M.M*).**

13

Dr. D. M. Akbar Hussain
Department of Electronic Systems

Extension RE Operations

Another Situation Demands: "any character"

Common requirement is to have a match of any character in the alphabet. Which means we have to literally write each character with alternation.

The extension is to use meta character "." period, which gives the option not to put every character alternative.

Example: .*b.* True for all strings containing at least one b.

14

Dr. D. M. Akbar Hussain
Department of Electronic Systems

Range of Characters

Usually we do like this for range of characters:

a | b | c | d | ..... | z

0 | 1 | 2 | 3 | ..... | 9

So the extension is

[a - z]

[0 - 9]

Any Character not in a given set:

It is not possible to get to a situation, where we have RE for all characters except one character. Tilde or Carat can be used.

For example: ~a → a character which is not a

~(a | b | c) → a character which is neither a,b or c.

In Lex: → ^a, ^(a | b | c)

Optional Occurrence: M? → (M | ε)

15

Dr. D. M. Akbar Hussain
Department of Electronic Systems

RE Notations

a	An ordinary character stands for itself
ε	The empty string
M|N	Alternation choosing from M or N
M.N / MN	Concatenation, an M followed by N
M*	Repetition zero or more times
M+	Repetition one or more times
M?	Optional, Zero or one occurrence of M
[a - zA - Z]	Character set alternation
.	Period, any single character except new line
~M /^M	A character which is not M (Tilde or Carat)
"a.+*"	Quotation, a string in quotes stands for itself literally

16

Dr. D. M. Akbar Hussain
Department of Electronic Systems

Finite Automata

- RE are very convenient for specifying token, but some formalism is required to implement it.
- Finite automata is a mathematical way of describing machines.
- In particular for the process of recognizing input tokens FA is the proper formalism.
- FA are the best construct to implement scanners.

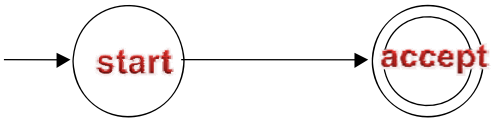
Formal Definition:

- FA is a finite automaton with finite set of states.
- Edges lead from one state to other states.
- Each edge is labeled with a symbol.
- One is unique start state and some are final states.

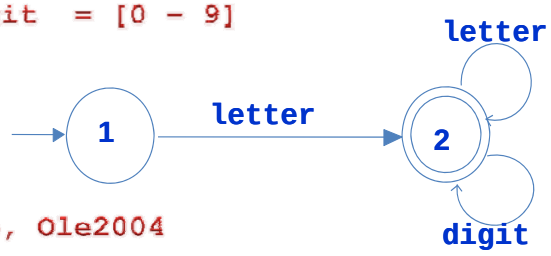
17

Dr. D. M. Akbar Hussain
Department of Electronic Systems

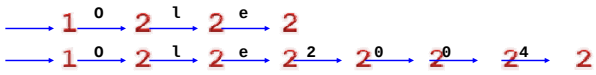
Finite Automata




```
letter = [a - zA - Z]
digit  = [0 - 9]
```



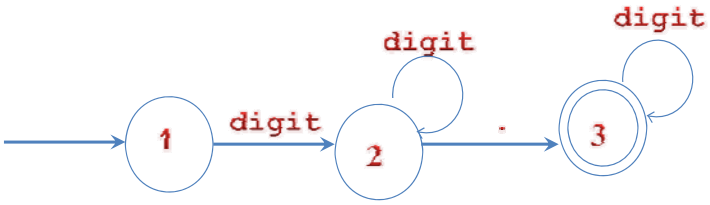
Ole, Ole2004



18

Dr. D. M. Akbar Hussain
Department of Electronic Systems

Example Finite Automata for Real



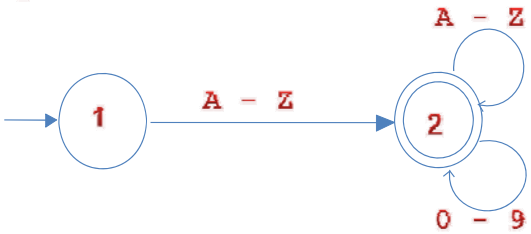
19

Dr. D. M. Akbar Hussain
Department of Electronic Systems

Deterministic Finite Automata (DFA)

No two edges leaving from same state are labeled with the same symbol.

Fox example for an ID:



20

Dr. D. M. Akbar Hussain
Department of Electronic Systems

Deterministic Finite Automata (DFA)

(K, δ, Σ, S, A)

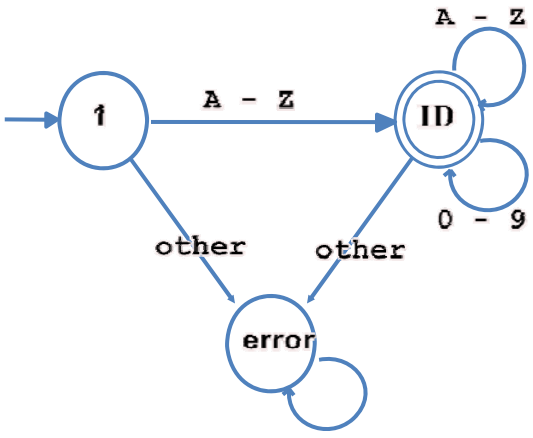
K "Set of states"
δ "Transition function T:(K x Σ) = δ
Σ "Set of Alpha-bets"
S "Unique start state" $S \in K$ **S is an element of K**
A "Accepting state" $A \subset K$ **A proper subset of K**

21

Dr. D. M. Akbar Hussain
Department of Electronic Systems

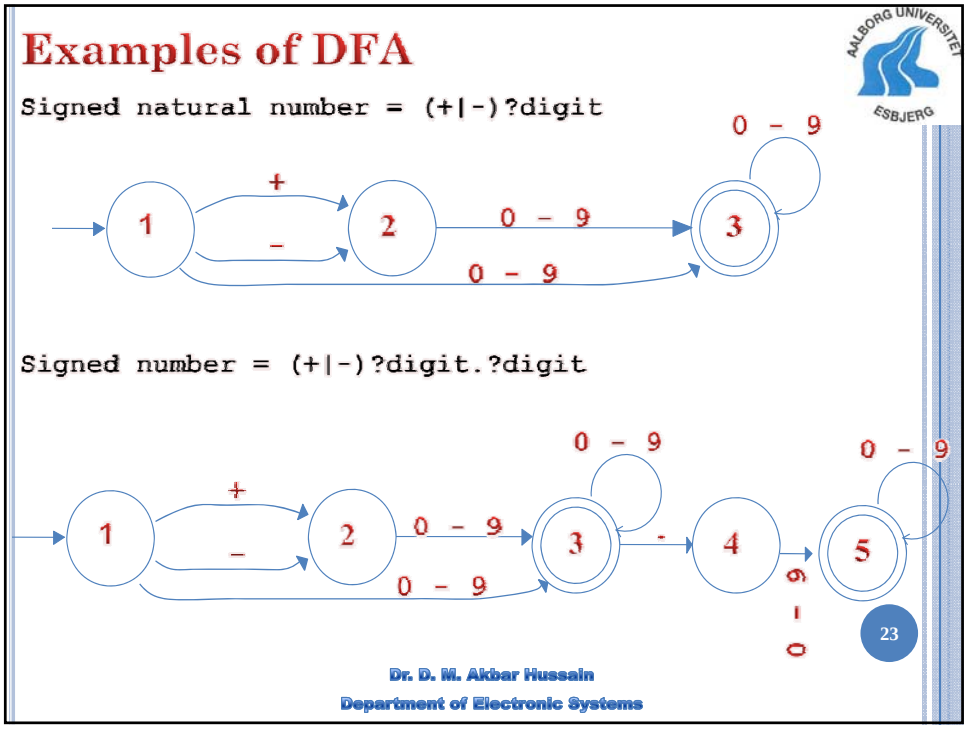
Deterministic Finite Automata (DFA)

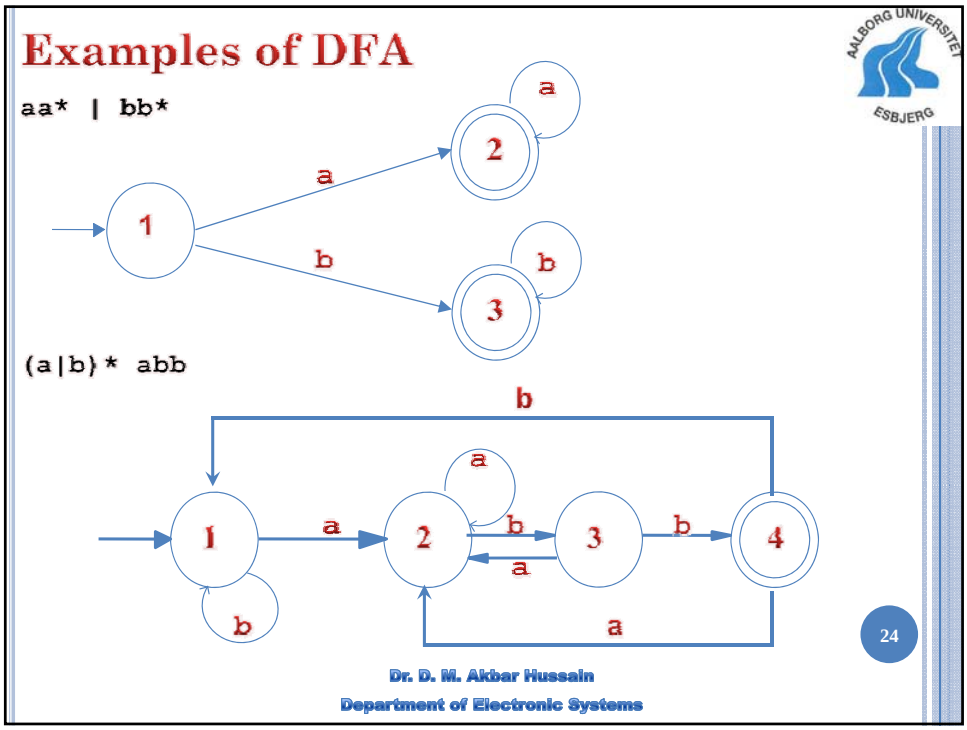
Last example has no such provision for example if there is a character other than alphabet.



22

Dr. D. M. Akbar Hussain
Department of Electronic Systems


Examples of DFA
Signed natural number = (+|-)?digit
0 - 9
+
-
1
2
0 - 9
3
0 - 9
Signed number = (+|-)?digit.?digit
0 - 9
0 - 9
+
-
1
2
0 - 9
3
.
4
5
0 - 9
23
Dr. D. M. Akbar Hussain
Department of Electronic Systems


Examples of DFA
aa* | bb*
a
2
a
1
b
b
3
(a|b)* abb
b
a
1
a
2
b
3
b
4
a
b
a
24
Dr. D. M. Akbar Hussain
Department of Electronic Systems

DFA Conventions

- Error transitions are not explicitly shown.
- Input symbols that result in the same transition are grouped together (this set can even be given a name).
- Still not displayed: stopping conditions and actions.
- **Principle of Longest Sub-string (or Maximal Munch):**
  - The DFA matches the longest possible input string before stopping.

25

Dr. D. M. Akbar Hussain
Department of Electronic Systems

Non Deterministic Finite Automata (NFA)

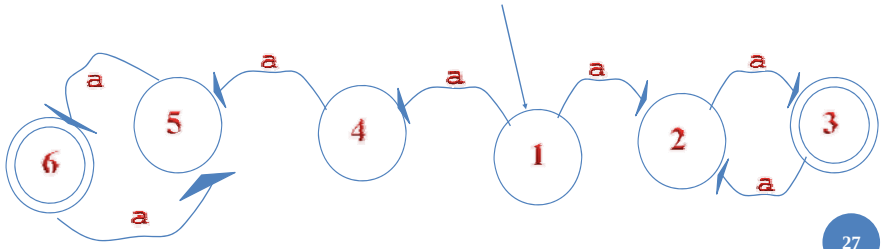
- DFA actually does not represent every thing, it **provides us an outline** of its operations.
- Mathematically, DFA must have a transition for every state and character.
- What action is to be taken when an error occurs.
- What action is to be taken when an accepting condition reached.

26

Dr. D. M. Akbar Hussain
Department of Electronic Systems

Non Deterministic Finite Automata (NFA)

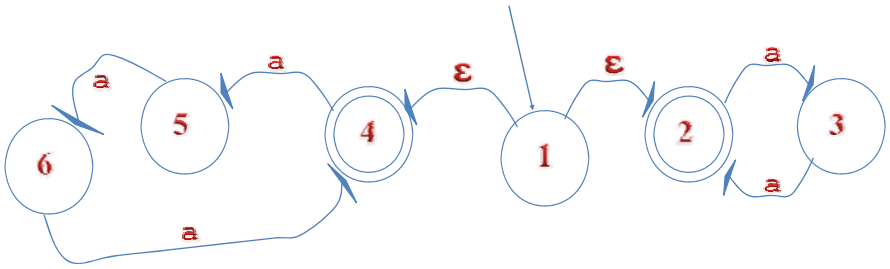
- Basically NFA is an automaton which has choice of edges, labeled with the same symbols to follow out of a state.
- Or it can have special edge labeled with "ε" that can be followed without eating any symbol from the input (without consulting the input).



27

Dr. D. M. Akbar Hussain
Department of Electronic Systems

NFA Example

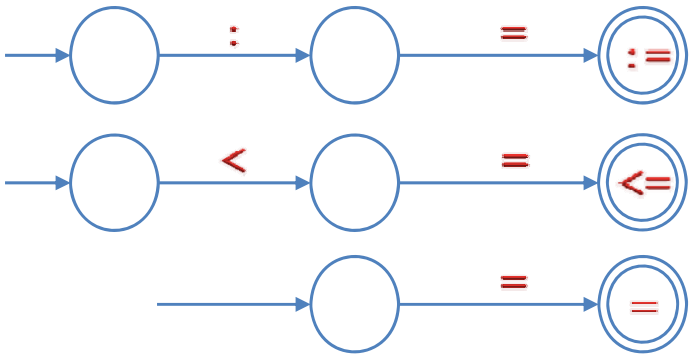


28

Dr. D. M. Akbar Hussain
Department of Electronic Systems

How to NFA

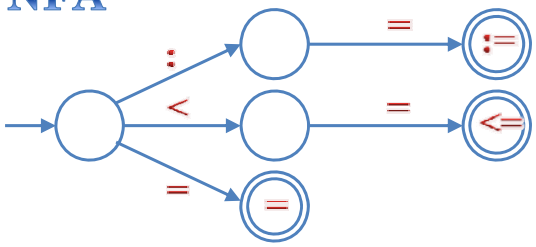
Suppose we have three tokens ":=", "<=", "=":



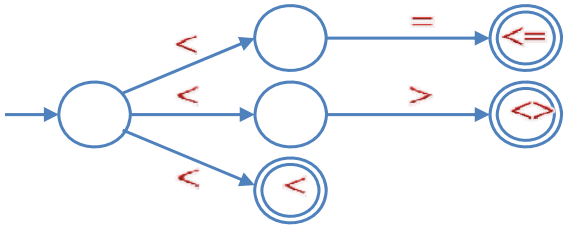
29

Dr. D. M. Akbar Hussain
Department of Electronic Systems

How to NFA



Suppose we have tokens "<=", "<>", "<" can we do this:

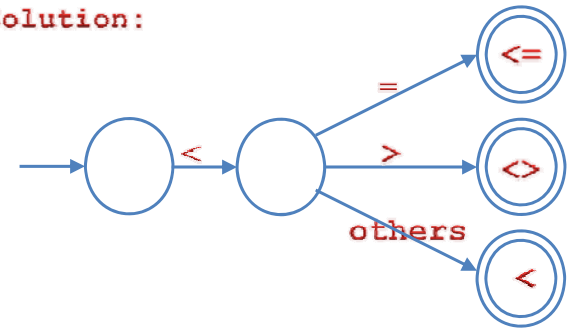


30

Dr. D. M. Akbar Hussain
Department of Electronic Systems

How to NFA

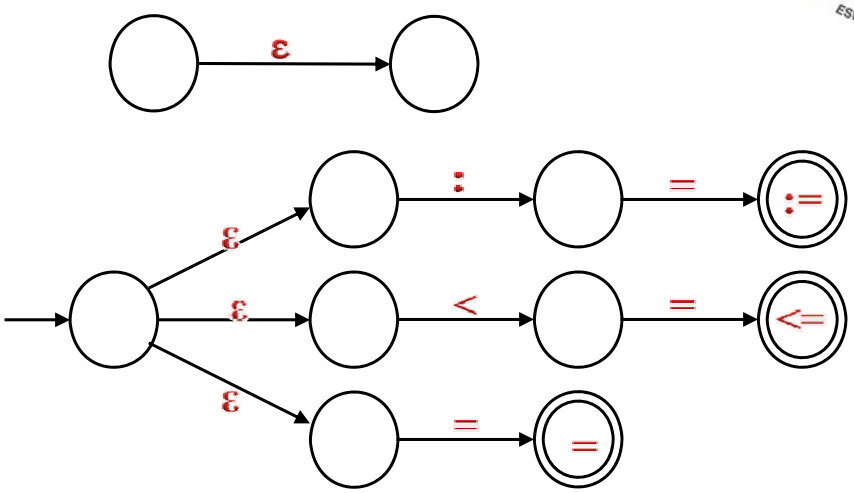
Solution:



31

Dr. D. M. Akbar Hussain
Department of Electronic Systems

How to NFA

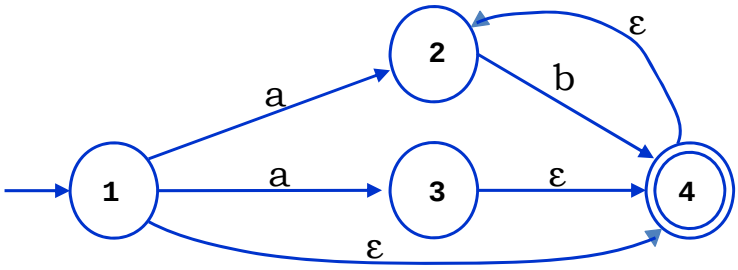


32

Dr. D. M. Akbar Hussain
Department of Electronic Systems

NFA [ab+|ab*|b*]

The definition of NFA is similar but we need to expand the alphabet set to include ε.



"abb" which can be accepted by any of the Following sequences:

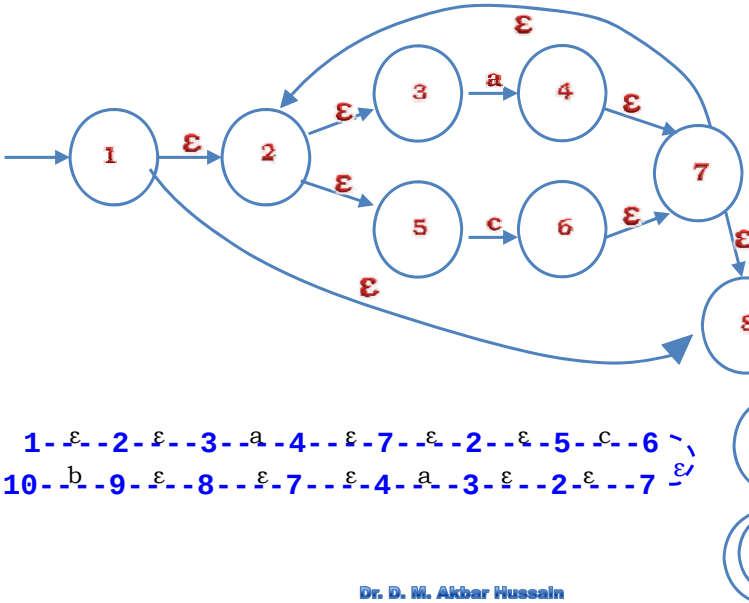
1 -a- 2 -b- 4 -ε- 2 -b- 4

1 -a- 3 -ε- 4 -ε- 2 -b- 4 -ε- 2 -b- 4

33

Dr. D. M. Akbar Hussain
Department of Electronic Systems

NFA "acab" RE = (a|c)*b



1 -ε- 2 -ε- 3 -a- 4 -ε- 7 -ε- 2 -ε- 5 -c- 6

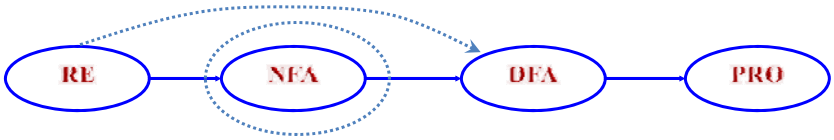
10 -b- 9 -ε- 8 -ε- 7 -ε- 4 -a- 3 -ε- 2 -ε- 7

34

Dr. D. M. Akbar Hussain
Department of Electronic Systems

RE to DFAs

- RE and DFAs are equivalent.
- However RE are preferred over DFAs because of their compactness in token description.
- Typically, scanning starts with RE and through DFAs finally end with scanner program.
  - *Generally*, this translation construct an intermediate construction using NFA which finally construct the DFA.
  - There is a possibility of direct construction from RE to DFA but generally not preferred because of complexity.

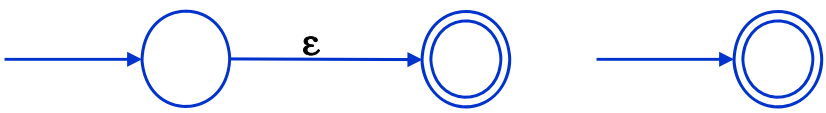


35

Dr. D. M. Akbar Hussain
Department of Electronic Systems

Thomson Construction

- It construct a machine by using the ε transition to glue together the machines for each piece of RE, which corresponds to the whole expression.

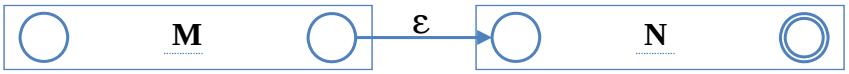


36

Dr. D. M. Akbar Hussain
Department of Electronic Systems

Thomson Construction

- Concatenation:



RE = MN so Thomson construction follows:

Dr. D. M. Akbar Hussain
Department of Electronic Systems

Thomson Construction

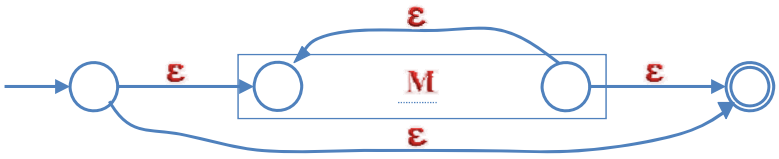
- Alternation/Choice:
  RE = M|N so Thomson construction follows:

38

Dr. D. M. Akbar Hussain
Department of Electronic Systems

Thomson Construction

- Repetition:
  RE = M* so Thomson construction follows:



39

Dr. D. M. Akbar Hussain
Department of Electronic Systems

Thomson Construction

- Thomson construction is not unique, there are other possibilities e.g., for MN:

However, it is only possible if the accepting state has no transitions to other states.

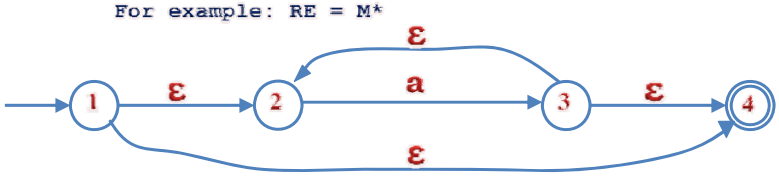
40

Dr. D. M. Akbar Hussain
Department of Electronic Systems

Subset Construction (NFA to DFA)

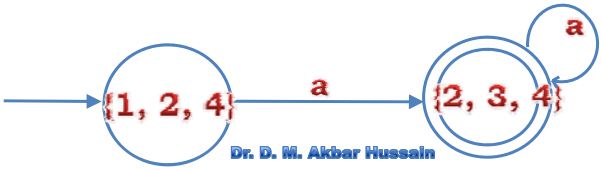
- Obviously, first thing to remove is ε transitions and second the multiple transitions from a state on a single character input.

For example: RE = M*



"ε closure" of a single state S is a set of states reachable by a series of zero or more ε transitions and denoted by $\overline{S}$.

$\overline{1}$ = {1, 2, 4}, $\overline{2}$ = {2}, $\overline{3}$ = {2, 3, 4}, $\overline{4}$ = {4}



41

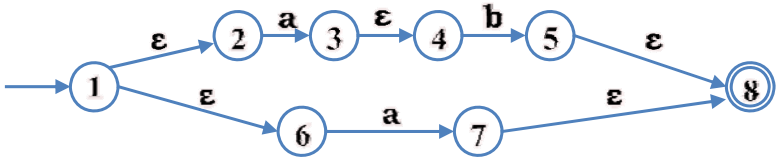
Dr. D. M. Akbar Hussain
Department of Electronic Systems

Subset Construction Example

RE = ab|a*

$\overline{1}$ = {2, 6, 1}, $\overline{2}$ = {2}, $\overline{3}$ = {3, 4}, $\overline{4}$ = {4}, $\overline{5}$ = {8, 5}, $\overline{6}$ = {6}

$\overline{7}$ = {8, 7}, $\overline{8}$ = {8}

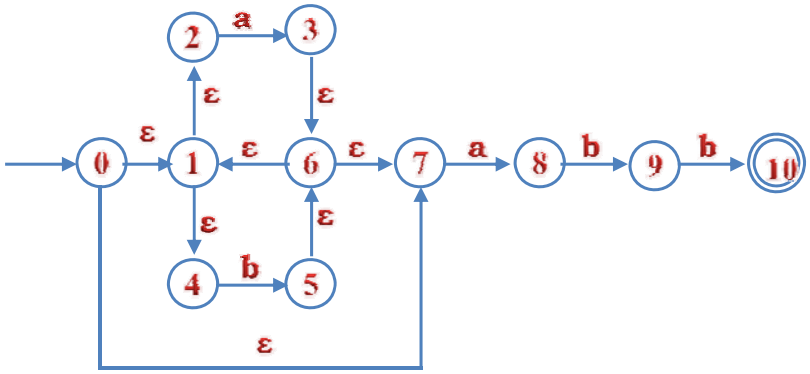


42

Dr. D. M. Akbar Hussain
Department of Electronic Systems

Another Example (First NFA)

RE = (a|b)* abb

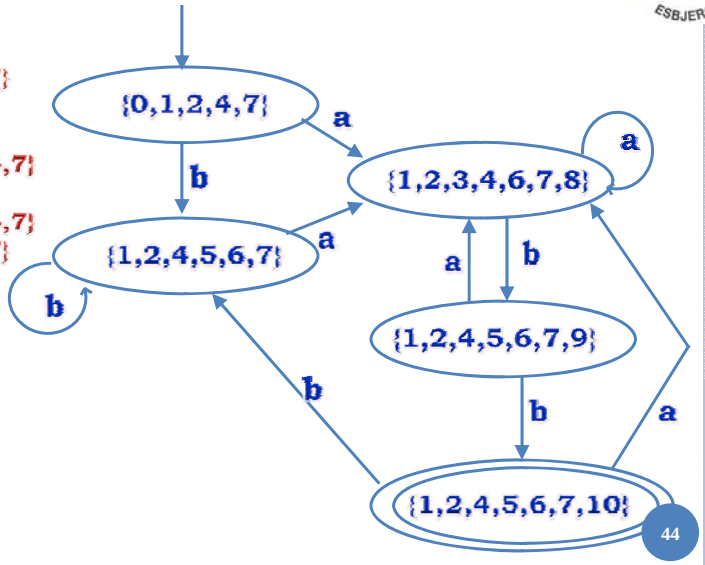


43

Dr. D. M. Akbar Hussain
Department of Electronic Systems

Subset Construction

0 = {0,1,2,4,7}
1 = {1,2,4}
2 = {2}
3 = {3,6,1,2,4,7}
4 = {4}
5 = {5,6,1,2,4,7}
6 = {6,1,2,4,7}
7 = {7}
8 = {8}
9 = {9}
10 = {10}



44

Dr. D. M. Akbar Hussain
Department of Electronic Systems

Minimization of States in a DFA

- We have seen that derivation from RE to DFA result in more states (complex).
- Building scanner with more states is not going to be appreciated.
- But good news is that FA theory states that for a given DFA there is an equivalent DFA containing minimum states. This minimum state DFA is unique.

45

Dr. D. M. Akbar Hussain
Department of Electronic Systems

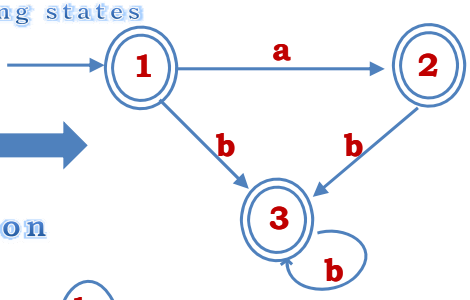
Minimization of States in a DFA

Divide the given set of states into:

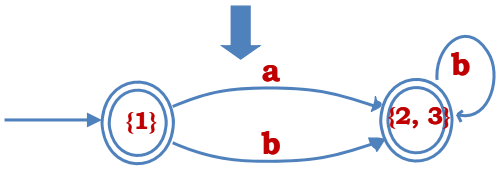
1. Accepting states
2. Not accepting states

(a|ε)b*

Actual DFA

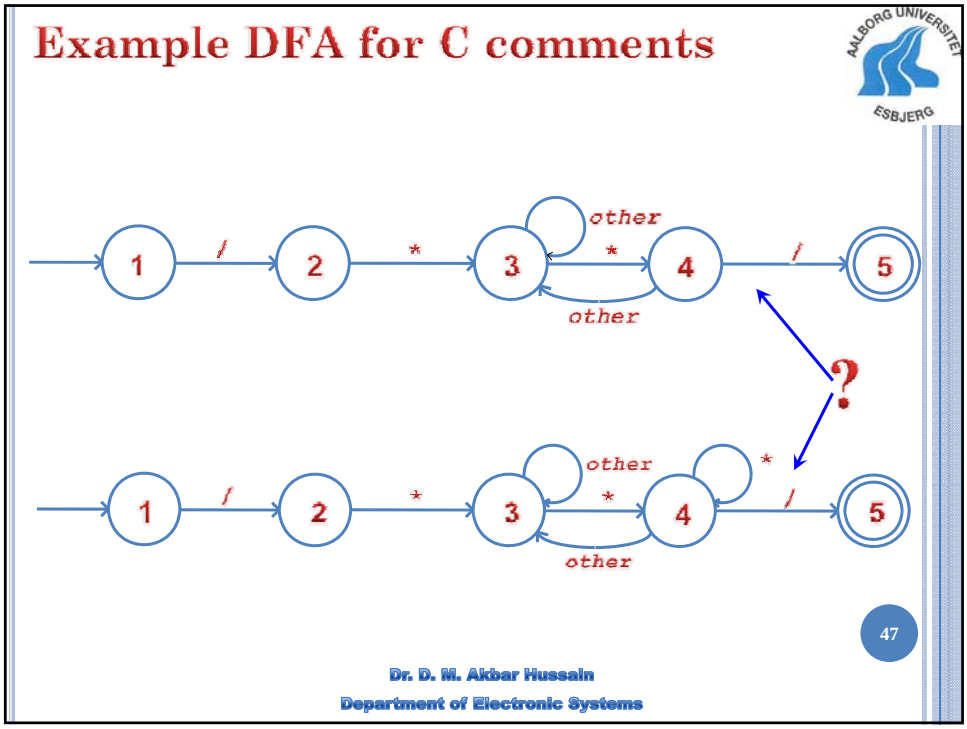


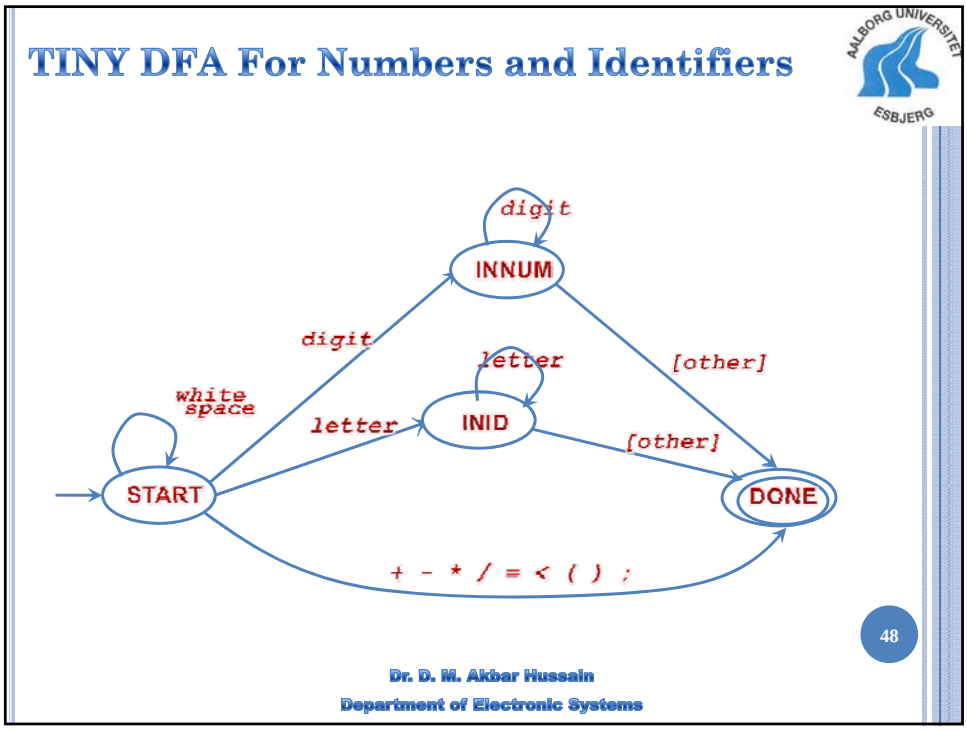
Minimized Version

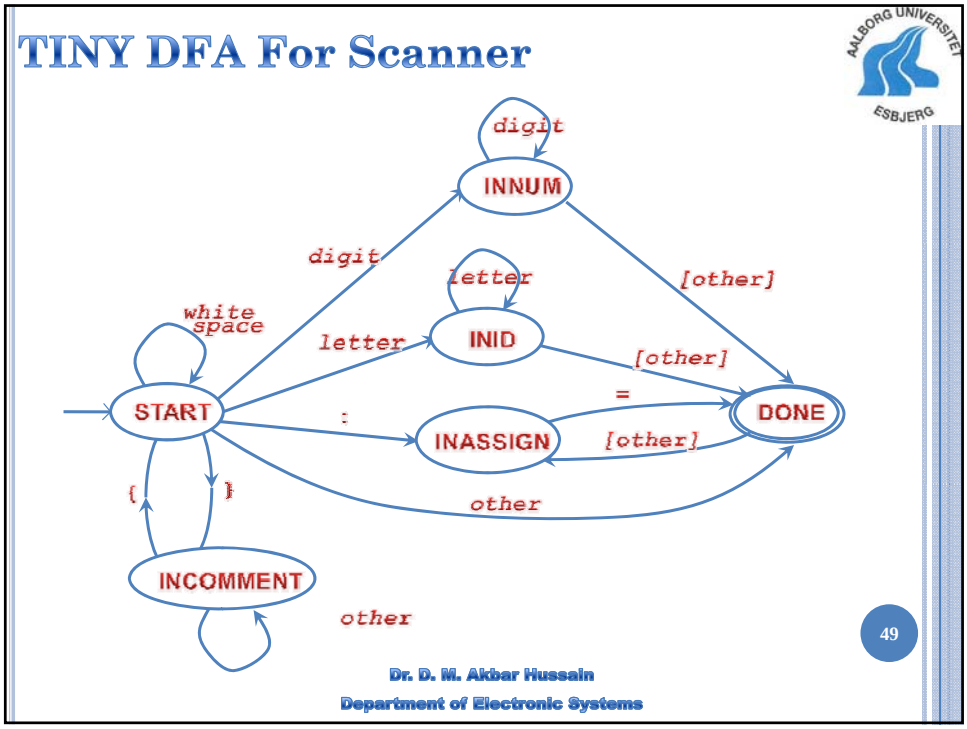


46

Dr. D. M. Akbar Hussain
Department of Electronic Systems


Example DFA for C comments
AALBORG UNIVERSITET ESBJERG
1
/
2
*
3
other
*
4
/
5
other
?
1
/
2
*
3
other
*
4
*
/
5
other
47
Dr. D. M. Akbar Hussain
Department of Electronic Systems


TINY DFA For Numbers and Identifiers
AALBORG UNIVERSITET ESBJERG
digit
INNUM
digit
letter
[other]
white space
letter
INID
[other]
START
DONE
+ - * / = < ( ) ;
48
Dr. D. M. Akbar Hussain
Department of Electronic Systems


TINY DFA For Scanner
AALBORG UNIVERSITET ESBJERG
digit
INNUM
digit
letter
[other]
white space
letter
INID
[other]
=
START
:
INASSIGN
[other]
DONE
{
}
other
INCOMMENT
other
49
Dr. D. M. Akbar Hussain
Department of Electronic Systems