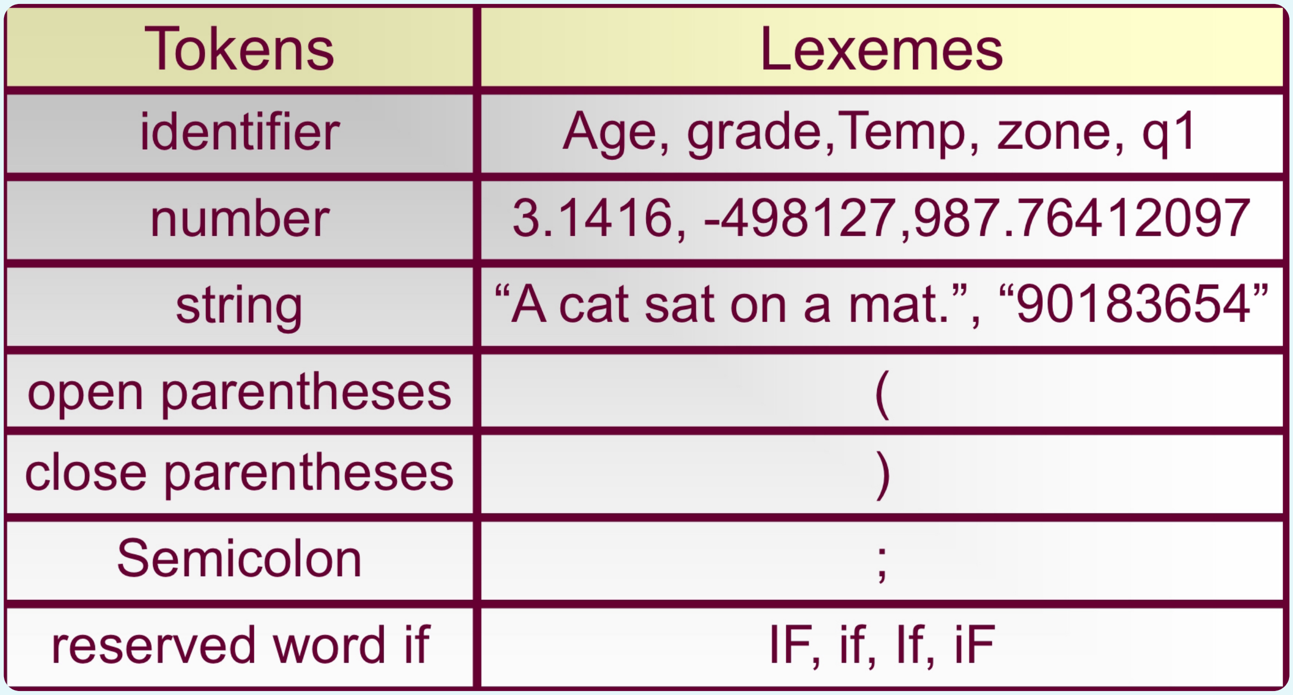
Scanner

Sometime call Lexical Analyzer

Main purpose

Info

Get a stream of characters divedes it into tokens.
Tokens : meaningful unit in source language
Lexemes : string which match pattern of tokens



Tokens	Lexemes
identifier	Age, grade,Temp, zone, q1
number	3.1416, -498127,987.76412097
string	“A cat sat on a mat.”, “90183654”
open parentheses	(
close parentheses	)
Semicolon	;
reserved word if	IF, if, If, iF

Detailed

When a token is found

Info

for sure that it will be passed to next phase how ever as we mentioned in intro scanner will put information into a symbol/literal table as identifier and attribute, and it is nessary to **check is it already in**

Example of attribute
1.Attribute of a variable are name, address, type
2.Attribute of numeric constant is its value

Regular Expressions

Info

As we mentioned Lexemes is string which match **pattern of tokens** so we have to have something to check this condition, so we used regular expression to describe these patterns

Important Regular Expression
λ = ε = e = empty string
ϕ = empty set
r | s = r or s
rs = r followed by s
r* = r 0~∞ ตัว
r^+ = r 1~∞ ตัว
(r) = r? = r 0/1 ตัว
[a-z] = any character from a to z

[A-Za-z] = any alphabet

~(r) = r = any character not `r`

. = any character

General Pattern of Tokens

reservedIF = (l|i)(F|f)

letter = [a-zA-Z]

digit = [0-9]

identifier = letter(letter|digit)*

numeric = (+|-)?$digit^+$(.$digit^+$)?(E(+|-)?$digit^+$)

Comment :

{(~})*}

/* ([^*]*[^/]*)* */

;(~newline)* newline

Disambiguating Rules

Info

IF is reserved word

A reserved word cannot be used as identifier,while keyword can also be identifier.

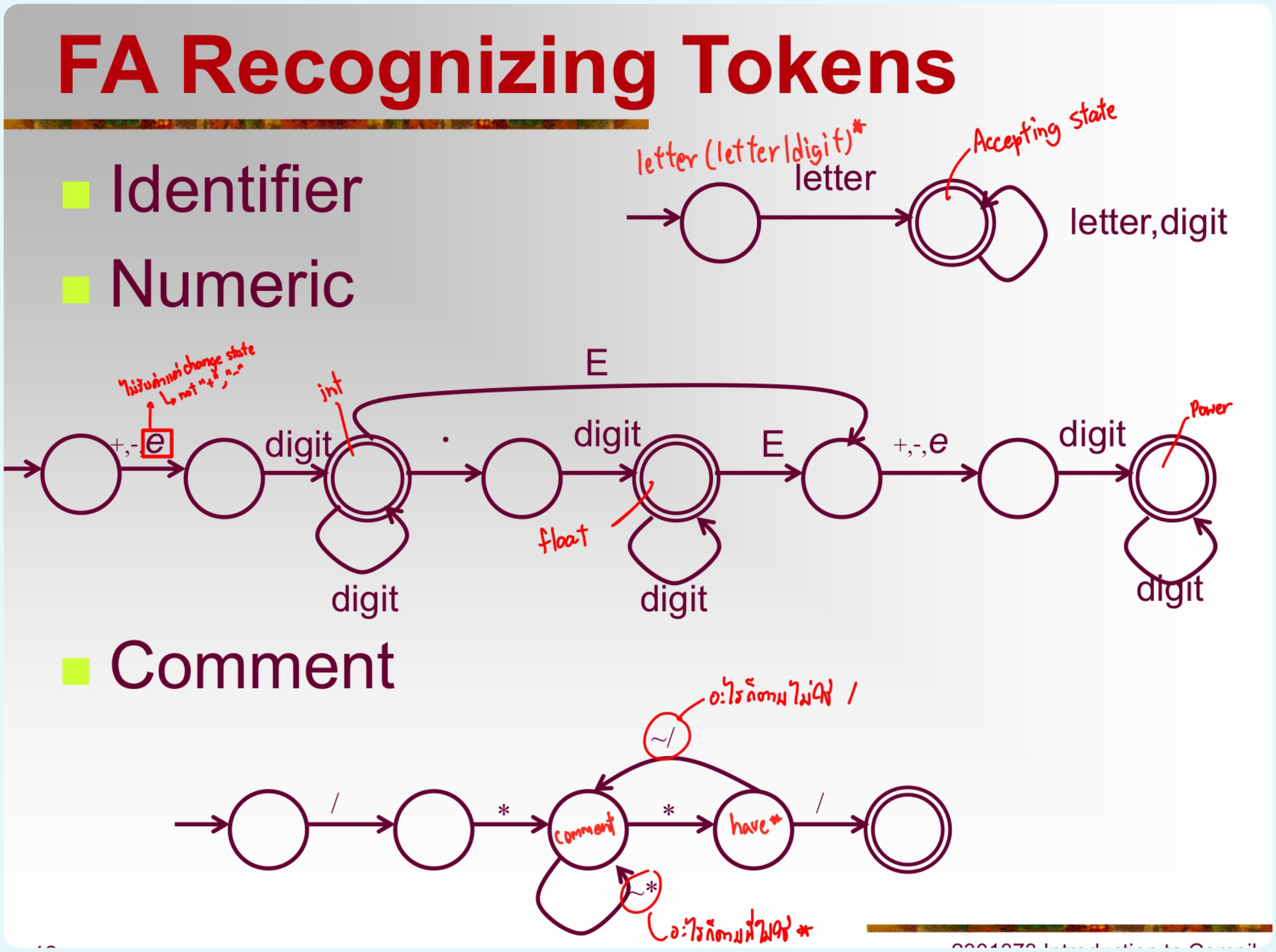
Priciple of longest substring :

When a string can be either a single token or a sequence of tokens, single-token interpretation is preferred.

Interesting thing

Info

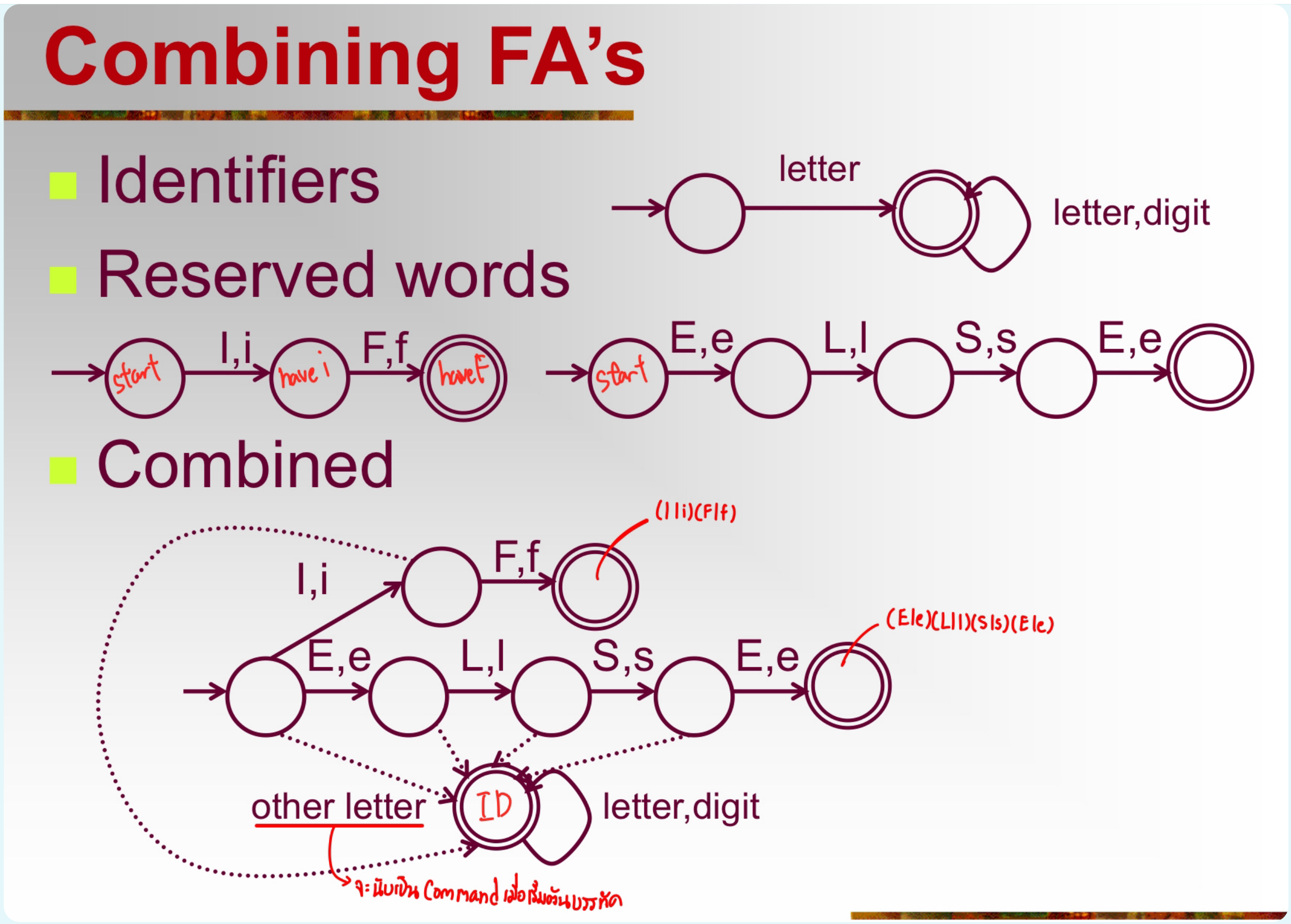
Example NFA of Tokens



FA Recognizing Tokens

- Identifier
- Numeric
- Comment

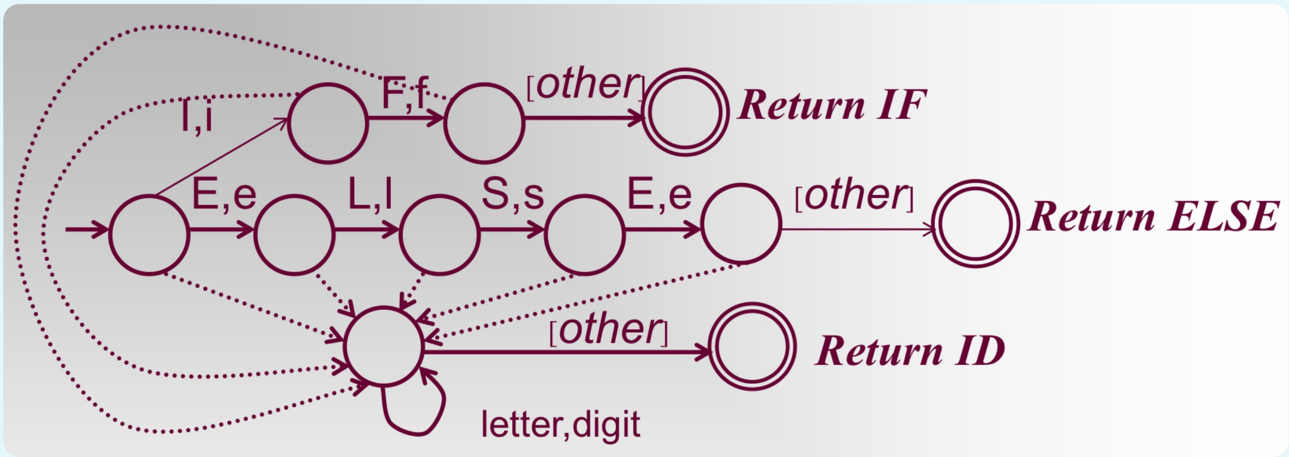
Combining IF with identifier with no lookahead yet



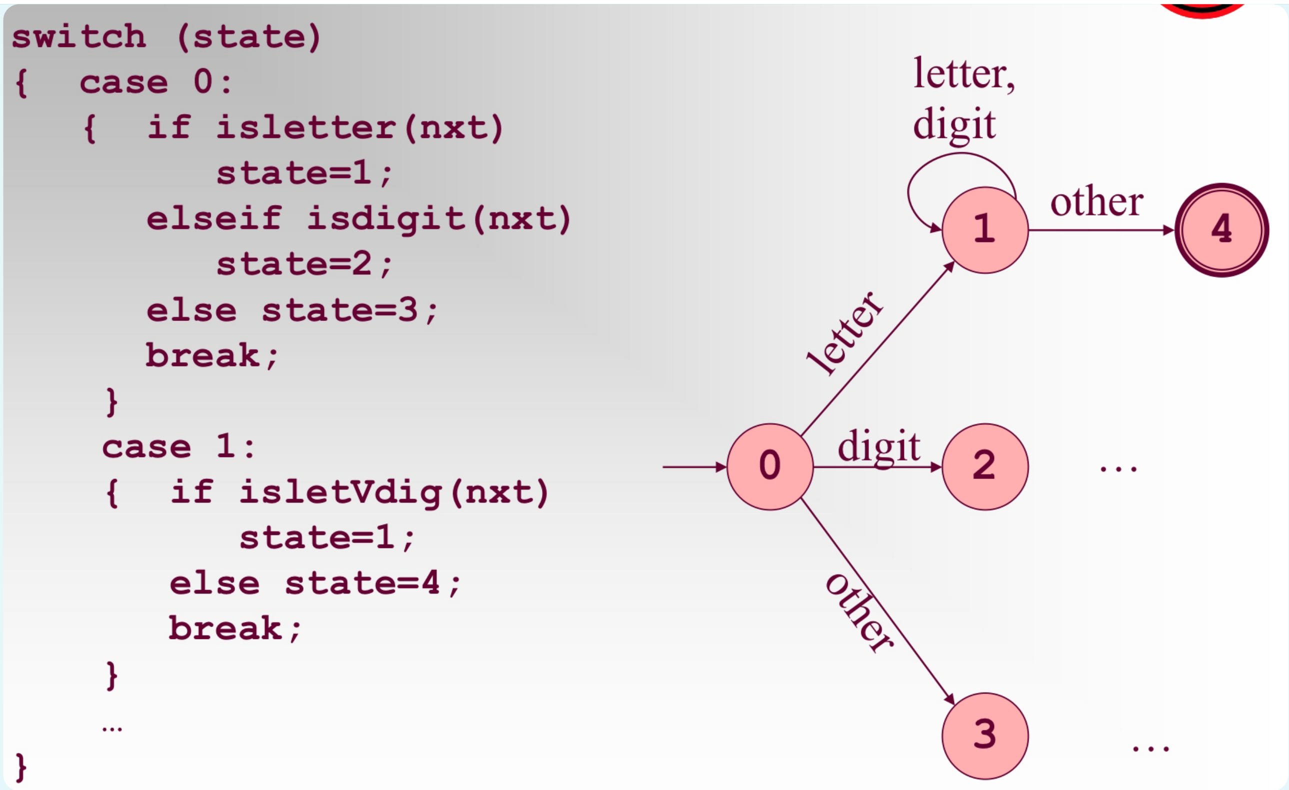
Combining FA's

- Identifiers
- Reserved words
- Combined

Adding lookahead

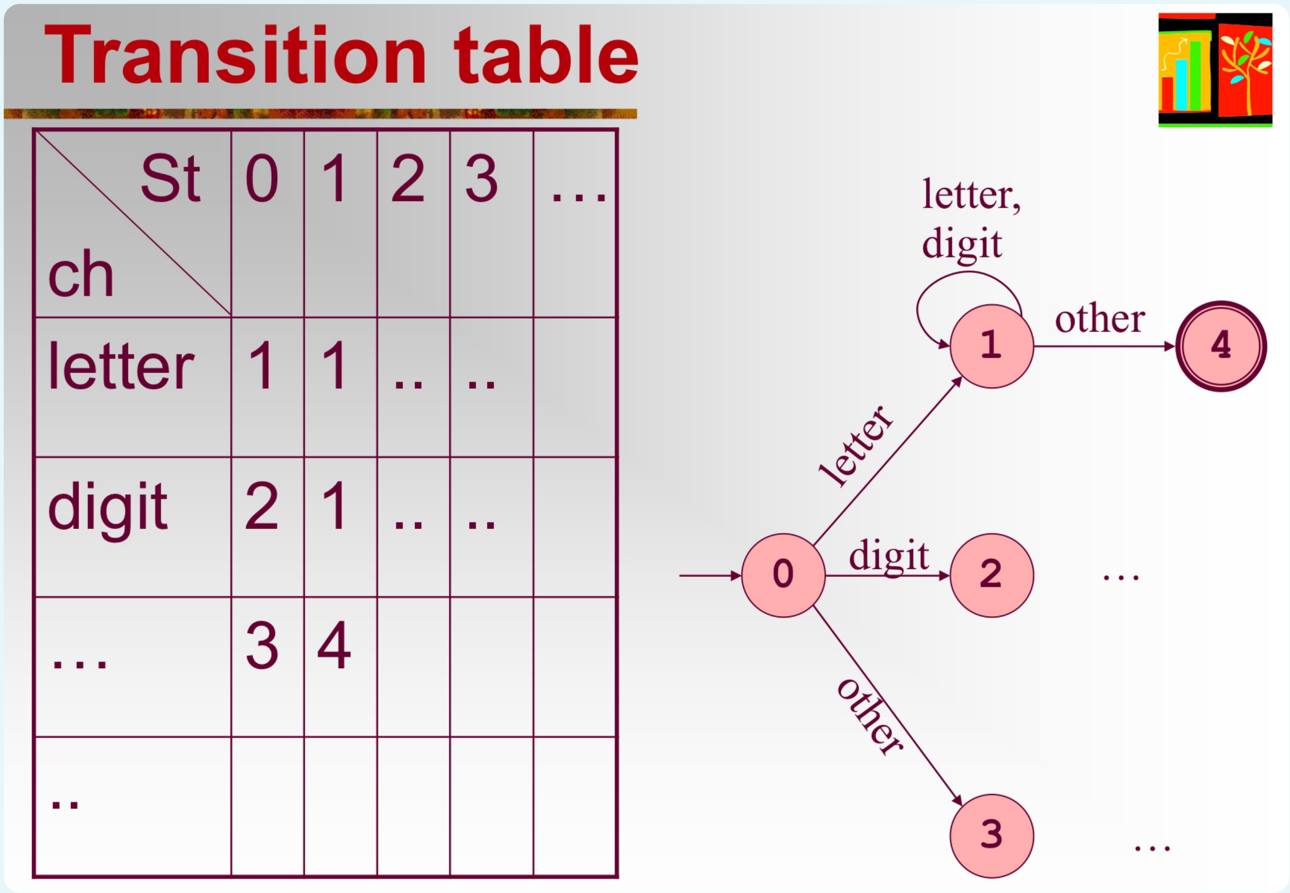


Using switch case to track current state



```
switch (state)
{  case 0:
   {  if isletter(nxt)
         state=1;
      elseif isdigit(nxt)
         state=2;
      else state=3;
      break;
    }
    case 1:
    {  if isletVdig(nxt)
          state=1;
       else state=4;
       break;
    }
    …
}
```

Using transition table help coding



Transition table

St ch	0	1	2	3	…
letter	1	1	..	..	
digit	2	1	..	..	
…	3	4			
..					

Error Handling in Scanner

Info

1.Delete an extraneous character
2.Insert a missing character
3.Replace a wrong character with correct one
4.Transpose 2 adjacent-char to correct position

Buffering

Info

character read into buffer and then scanned by scanner
Scanning done by two pointer(beginP,forwardP)
step 1 fP move char by char find end of token
step 2 check pattern/identify lexeme
step 3 bP and fP move to right char of fwd
Single Buffer :
if not store will be lost cuz overidden
Buffer Pair :
two buffers:
one for reading characters from source
the other for accumulating until complete lexemes
Sentinel : บอกจุดจบlexeme
Sentinel is a special character, often used at the end of the input buffer, to mark its boundaries
Help to ensure that the compiler knows where to

stop reading characters and where to begin forming lexemes.