TIKRIT UNIVERSITY
COLLEGE OF COMPUTER SCIENCE AND MATHEMATICS
DEPARTMENT OF COMPUTER SCIENCE

SUBJECT OF COMPILER1

DATE OF ISSUE: 2024 - 2025

CLASS: 3TH STAGE

SEMESTER 1

LECTURE NO. : 3

PREPARED BY

Lecturer:
Mohanad Dawood Al-Roomi

&

Assistant Lecturer:
Luay Ibrahim Klalif

Lexical Errors

Type of errors for Lexical analysis

Lexical errors include misspellings of identifiers, keywords, or operators... Etc.

Ex:

```
fi ( a == f(x))
```

Problem of lexical analyzer cannot tell:

- **Whether fi is a misspelling of the keyword if .**
- **Or fi is an undeclared function identifier .**

Terms Tokens, Patterns, and Lexemes

Source program

Ex.:

```
if ( x > y )
   x = 10 ;
while …
```

Lexical analyzer phase

1

Lexemes
if
(
x
>
y
)
x
=
10
;
while

(1) Lexeme:

A lexeme: is one character or a sequence of characters in the source program that denotes the meaning a basic lexical unit of the source program and is identified (breaks up) by the lexical analyzer.

Tokens, Patterns, and Lexemes

Source program

Ex.:
if (x > y)
x = 10 ;
while …

Lexical analyzer phase

2

Token table	
Token-Name	Token-Attribute
if	Kw
(	Pun
x	ID
>	Op
y	ID
)	Pun
x	ID
=	Op
10	Num
;	Pun
while	K.w

3

Symbol table		
Sq.	Token-name	type
1	x	
2	y	

4

Token stream:
<if,k.w> <(,pun> <Id,1>
<>,Op_GT> <Id,2>
<),pun> <Id,1>
<=,Op_Ass.> <10,Num>
<;,pun> < while,k.w>

(2) Token:

A token: is a pair consisting of a token name and an optional attribute value (token-name; attribute-value). The first component token-name is an abstract symbol that is used during syntax analysis, and the second component points to a kind of lexical unit, e.g., a particular keyword, or a sequence of input characters denoting an identifier.

Tokens, Patterns, and Lexemes

Source program

```
Ex.:
if ( x > y )
   x = 10 ;
while …
```

(3) Patterns:

Patterns: Each token has a pattern that describes which sequences of characters can form the lexemes corresponding to that token. The set of words, or strings of characters, that match a given pattern is called a language.

Pattern: is a description of a grammar of language that the tokens may take.

<Statement> → if (<Expression>) <Statement>
| if (<Expression>) <Statement> else <Statement>
| ℇ

<Expression> → <Term> relop <Term> | <Term>

<Term> → id | number

<relop> → < | > | <= | >= | == | !=

<Id> → L (L | D)*

<L> → A | B |… | Z | a | b | …| z | _

<D> → 0 |…| 9

Number → digits OptionalFraction OptionalExponent

<Ds> → DD*

OptionalFraction → . Ds | ε

OptionalExponent → (E (+ | - | ε) Ds) | ε

ملاحظة: القاعدة النحوية لتعريف if سوف يولد مشكلة التداخل من اليسار لذلك يجب ان تكتب بالصيغة التالية:

stmt → if (*expr*) *stmt* [else *stmt*]? | ℰ

Specification of Tokens

Alphabet, Strings and Languages

1. **Alphabet:** is any finite set of symbols.

- Binary alphabet ⟶ {0,1}
- Digit alphabet ⟶ {0,1,2,3,4,5,6,7,8,9}
- Letter alphabet ⟶ {a,b,c,…,z, A,B,C,…,Z}
- ASCII alphabet: a set of digital codes (0 , 1) is used in many software systems.
- Unicode alphabet include 100000 characters.

ASCII: American Standard Code for Information Interchange, a set of digital codes (0 , 1) representing letters, numerals, and other symbols, widely used as a standard format in the transfer of text between computers.

Unicode: an international encoding standard for use with different languages and scripts, by which each letter, digit, or symbol is assigned a unique numeric value.

Specification of Tokens

Alphabet, Strings and Languages

2. String ("Sentence" and "Word"): A string over an alphabet is a finite sequence of symbols drawn from the alphabet.

- The length of a string s, usually written |s|, is the number of occurrences of symbols in s.

010111011100 is string from the binary alphabet. The string of length 12.

01 is string from the binary alphabet. The string of length 2.

0 is string from the binary alphabet. The string of length 1.

banana is string from the Letter alphabet. The string of length 6.

εthe empty string , the string of length zero.

Specification of Tokens

Alphabet, Strings and Languages

3. Language: is any countable set of strings over some fixed alphabet.

Strings	Language
010111011100 01 0	Ex1: This is machine Language.

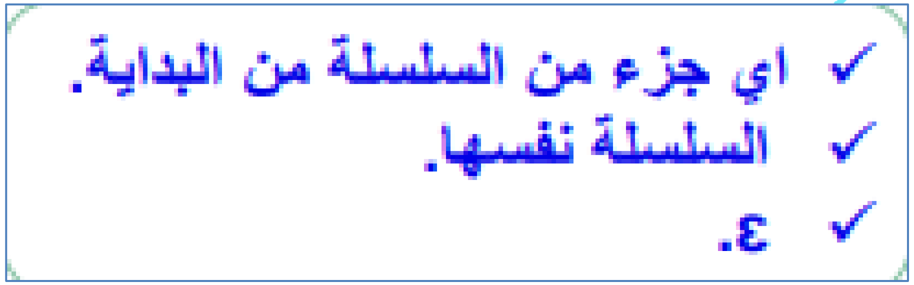
{
if (x==y)
x = x + 1;
for (int i =0 ; i<10; i++)
y = y + I ;
}
	Ex2: This is source program of C++ Language.

<{,Pun> <if,k.w> <(, Pun > <Id,1 pointer to symbol-table entry for x> <==,Op> <Id,2 pointer to symbol-table entry for y) <(, Pun > … < , >	Ex3: This is Language of token stream by C++.
The empty set, or {Ɛ}, the set containing only the empty string.	Abstract languages like Ø .
The set of all grammatically correct English sentences.	English Language.

Terms for Parts of Strings

1. A prefix of string s is any string obtained by removing zero or more symbols from the end of s.

   Ex: ban, banana, and ε are prefixes of banana.



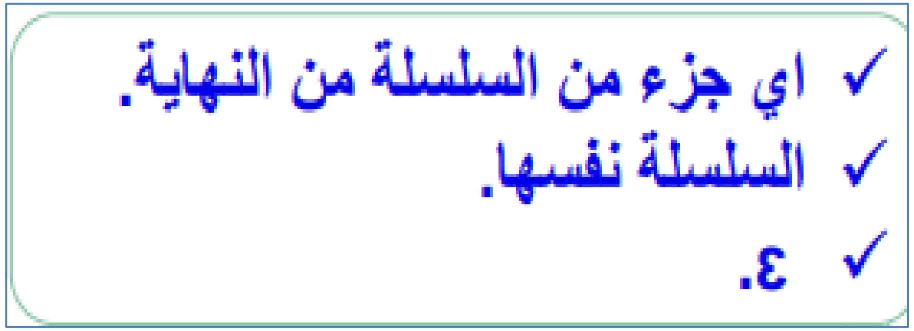
✓ اي جزء من السلسلة من البداية.

✓ السلسلة نفسها.

✓ ε.

2. A suffix of string s is any string obtained by removing zero or more symbols from the beginning of s.

   Ex: nana, banana, and ε are suffixes of banana.



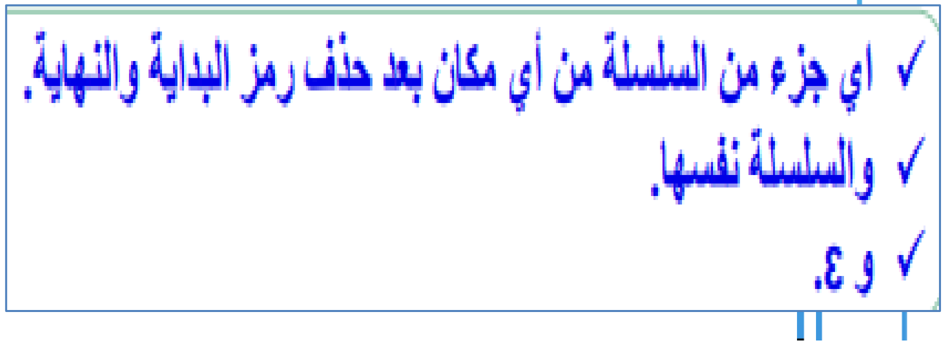
✓ اي جزء من السلسلة من النهاية.

✓ السلسلة نفسها.

✓ ε.

3. A substring of s is obtained by deleting any prefix and any suffix from s.

   Ex: banana, anan, nan, and ε are substrings of banana.



✓ اي جزء من السلسلة من أي مكان بعد حذف رمز البداية والنهاية

✓ والسلسلة نفسها.

✓ و ε.

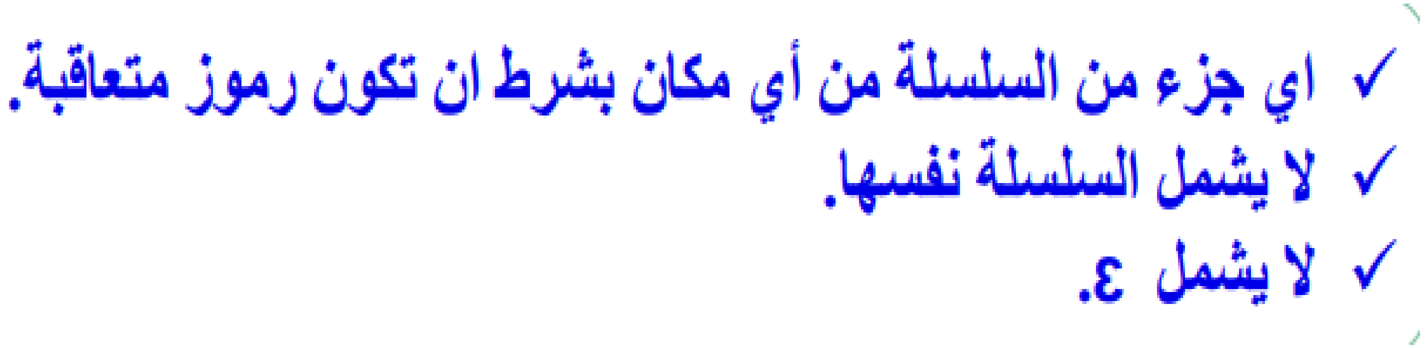
Terms for Parts of Strings

4. The proper prefixes, suffixes, and substrings of a string s are those, prefixes, suffixes, and substrings, respectively, of s that are not ε or not equal to s itself.

Ex: ba, ban, and banan are proper prefixes of banana.

Ex: na, nana and anana are proper suffixes of banana.

Ex: anan, nan are proper substrings of banana.



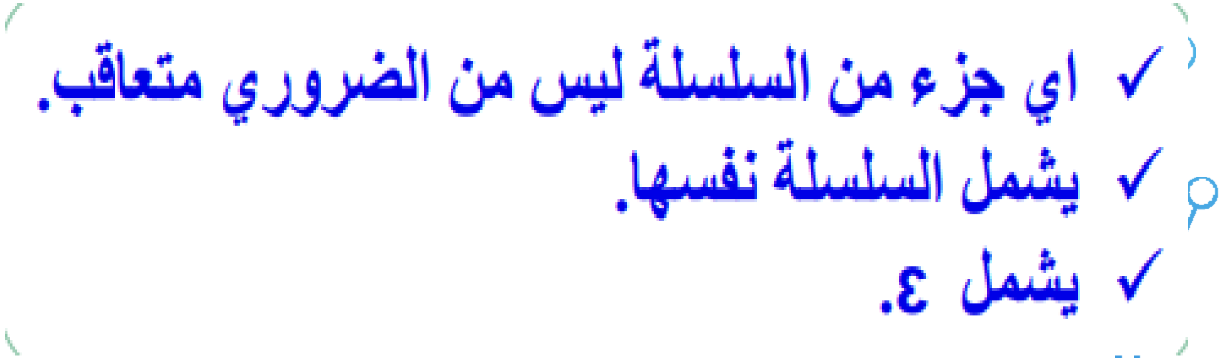
✓ اي جزء من السلسلة من أي مكان بشرط ان تكون رموز متعاقبة.

✓ لا يشمل السلسلة نفسها.

✓ لا يشمل ε.

5. A subsequence of s is any string formed by deleting zero or more not necessary consecutive positions of s.

Ex: baan is a subsequence of banana.



✓ اي جزء من السلسلة ليس من الضروري متعاقب.

✓ يشمل السلسلة نفسها.

✓ يشمل ε.

Specification of Tokens

Concatenation: التتابع

- If x and y are strings,
- Then the concatenation of x and y, denoted xy,
- is the string formed by appending y to x.
- Ex: if x = dog and y = house, then xy = doghouse.

- The empty string is concatenation.
- Any string s, εs
  = sε = s.

Specification of Tokens

Exponentiation : التسلسل

Exponentiation of strings is a product of concatenation.

Since:

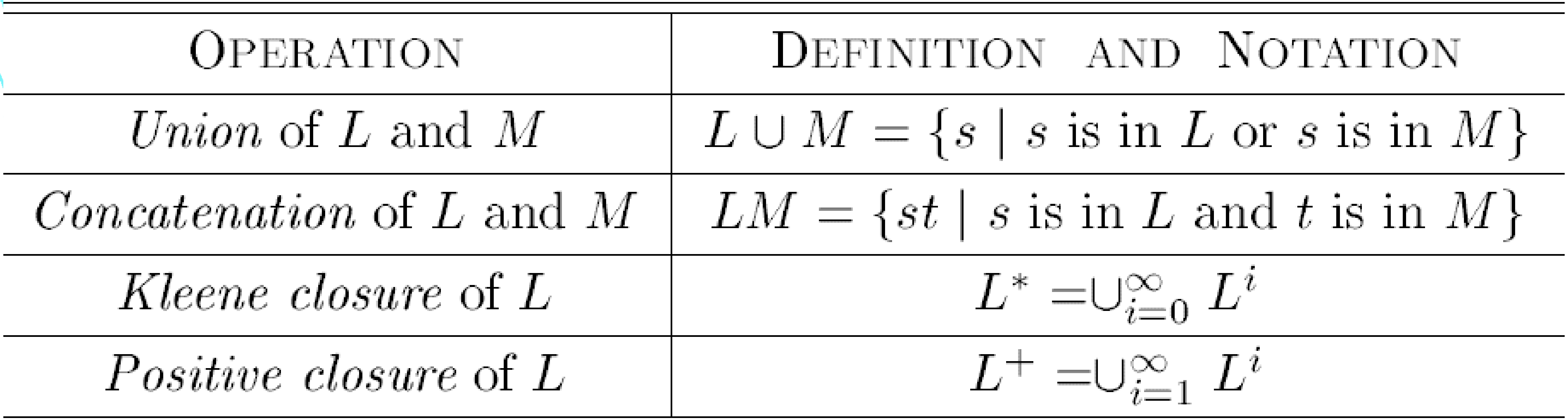
$s^0 = \varepsilon$

$\left.\begin{array}{l} \varepsilon s = s\varepsilon = s \end{array}\right\}$ define: s^0 to be ε

$\left.\begin{array}{l} s^1 = s \\ s^2 = ss \\ s^3 = sss \end{array}\right\}$ $i > 0$, define s^i to be $s^{i-1}s$.

الأس في السلاسل
هي
انتاج من تسلسل

Operations on Languages



OPERATION	DEFINITION AND NOTATION
Union of L and M	$L \cup M = \{s \mid s$ is in L or s is in $M\}$
Concatenation of L and M	$LM = \{st \mid s$ is in L and t is in $M\}$
Kleene closure of L	$L^* = \cup_{i=0}^{\infty} L^i$
Positive closure of L	$L^+ = \cup_{i=1}^{\infty} L^i$

Figure 3.6: Definitions of operations on languages

Operations on Languages

1. L U D is the set of letters and digits. 62 strings of length one.

$$\left.\begin{array}{l} \text{A | B | …| Z | a | b | …| z} \quad = 52 \\ \\ \text{0 | 1| …| 9} \quad = 10 \end{array}\right\} \quad 62$$

2. LD is the set of 520 strings of length two.

A0	B0	…	Z0	a0	b0	…	z0
A1	B1	…	Z1	a1	b1	…	z1
A2	B2	…	Z2	a2	b2	…	z2
…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…
A9	B9	…	Z9	a9	b9	…	z9
Total 520							

Operations on Languages

3. L^4 is the set of all 4 -letter strings.

4. L^* is the set of all strings of letters, including ε, the empty string.

5. $L (L \cup D)^*$ is the set of all strings of letters and digits beginning with a letter.

6. D^+ is the set of all strings of one or more digits.

7. ? Zero or one instance.

THANK YOU