Page: .....

Compiler

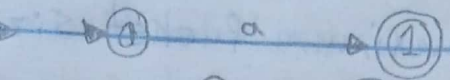
Date: .....

~~*Ch3~~

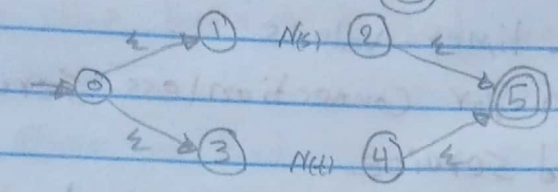
Convert NFA to DFA

RE → NFA use basic

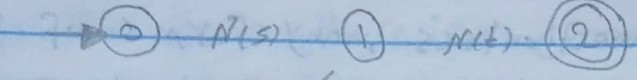
r = a



r = s|t

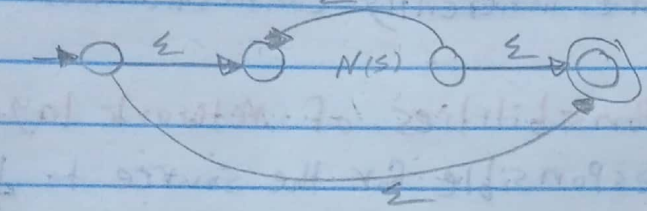


r = st



نهاية أولى بداية ثانية

$r = s^*$



ch1

1- What is the difference between a compiler and an interpreter?

A compiler is a program that can read a program in one language - the source language - and translate it into an equivalent program in another language.

Interpreter directly executes the operations specified in the source program on inputs supplied by the user

2- What are the advantages of (a) a compiler over an interpreter (b) an interpreter over a compiler?

(a) The machine language target program produced by a compiler is usually much faster than an interpreter at mapping inputs to outputs

(b) An interpreter can usually give better error diagnostics than a compiler, because it executes the source program statement by statement

Page: Date:

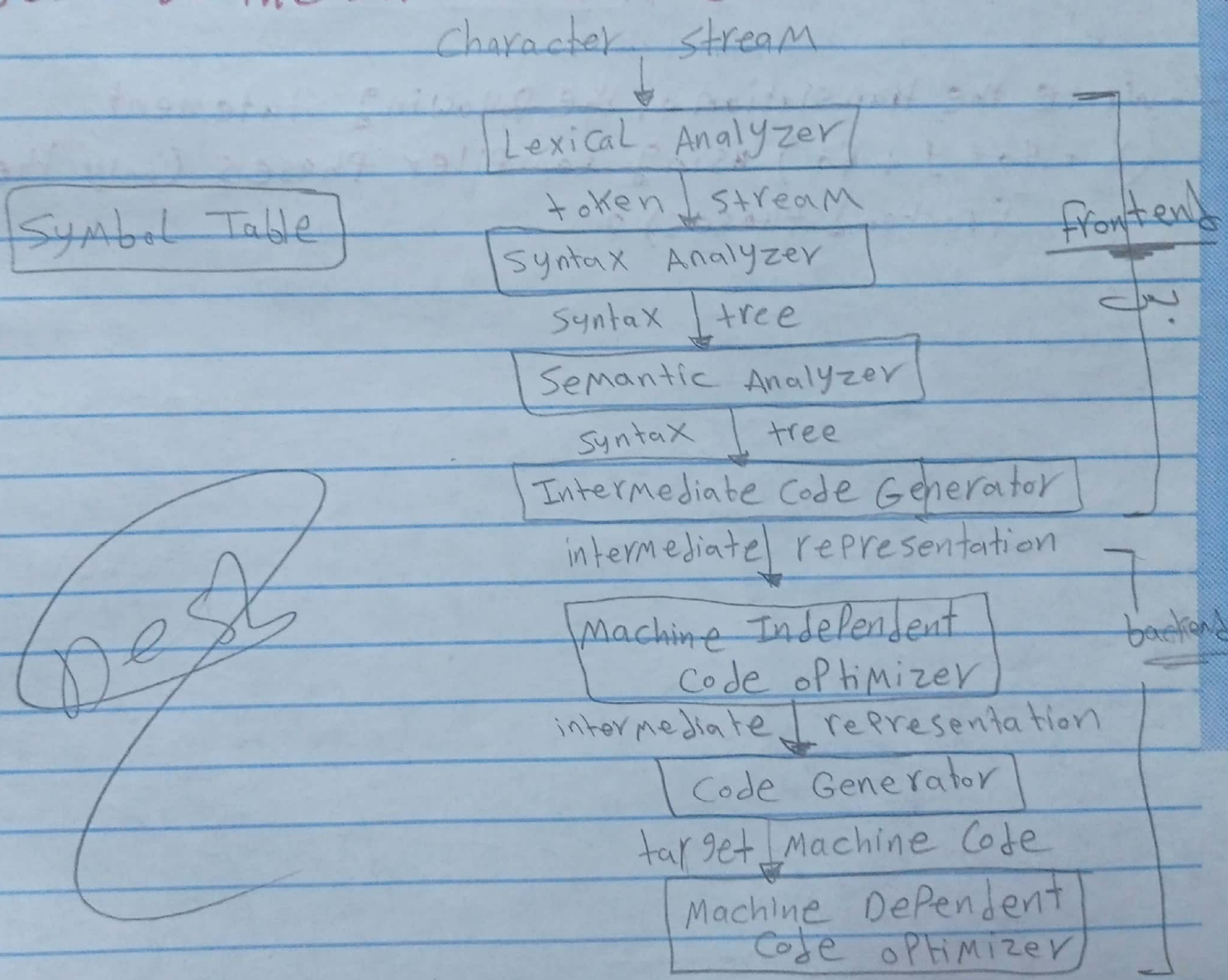
3- What advantages are there to a language processing system in which the compiler produces assembly language rather than machine language?

The compiler may produce an assembly language program as its output, because assembly language is easier to produce as output and is easier to debug

4- Describe some of the tasks that an assembler needs

It translates from the assembler language to machine code. This machine code is relocatable

5- Describe the structure of compiler phases in brief



Page: .... Date: ....

6- Describe the language processing system in brief?

Source program
↓
Preprocessor
↓ modified source program
Compiler
↓ target assembly program
Assembler
↓ relocatable machine code
library files → Linker / Loader
relocatable object files

7- Write the translation of the following statement (X = 40 * y + 50) using compiler phases (only the frontend phases)?

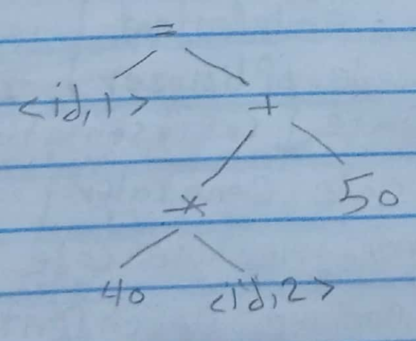
اى تعبير نفس الفكرة
total = 125 * num + 180

X = 40 * y + 50

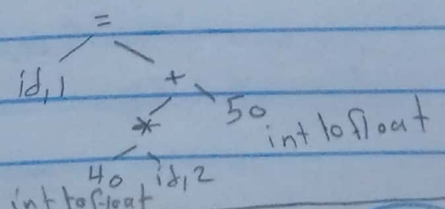
lexical analyzer
↓
<id,1><=><40><*><id,2><+><50>

يمشى فى الأكواد
بتاعت مادة ٣
x, y, z, w
in book

Syntax Analyzer
↓



Semantic analyzer



Intermediate code generator →

t_1 = int to float (50)
t_2 = int to float (40) * id,2
t_3 = $t_2 + t_1$
id,1 = t_3

1- write a simple intermediate Code for the Following java segment?

```
{   int i ; int j ; float [100] a ; float v ; float x ;
    while (true) {
        do i = i+1 ; while ( a[i] < v ) ;
        do j = j-1 ; while ( a[j] > v ) ;
        if( i >= j ) break ;
        x = a[i] ; a[i] = a[j] ; a[j] = x ;
    }
}
```

```
1: i = i+1                 4: j = j-1
2: t1 = a[i]               5: t2 = a[j]
3: if t1 < v goto 1        6: if t2 > v goto 4
          7: if false i >= j goto 9
          8: goto 14
9: x = a[i]                11: a[i] = t3
10: t3 = a[j]              12: a[j] = x
          13: goto 1
          14:
```

2- expr → expr + term | expr - term | term
term → term * factor | term / factor | factor
factor → digit | (expr)
digit → 0|...|9

2.2.4 افهم فكرة وحل

3- S → SS+ | SS* | a

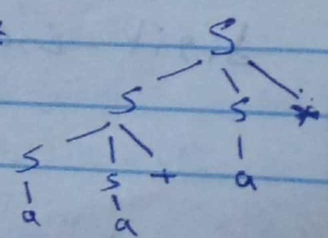
(1) show how the string aa+a* can be generated by this Grammar

(2) Construct a Parse tree for this string

(3) What language does this grammar generate?

(1) S ⇒ SS* ⇒ SS+S* ⇒ aa+a*

(2)



(3) L = { Postfix expression Consisting of digits, plus and multiple signs }

Page: Date:

(4) ① $S \to 0S1 \mid 01$

② $S \to +SS \mid -SS \mid a$ ambiguous

③ $S \to S(S)S \mid \varepsilon$ 2-2-3

④ $S \to aSbS \mid bSaS \mid \varepsilon$

⑤ $S \to a \mid S+S \mid SS \mid S^* \mid (S)$

1- $L = \{ 0^n 1^n \mid n >= 1 \}$

2- $L = \{$ Prefix expression consisting of plus and minus signs $\}$

3- $L = \{$ matched brackets of arbitrary arrangement and nesting includes ε $\}$

4- $L = \{$ String has the same amount of a & b, include ε $\}$

5- $L = \{$ Regular expressions used to describle regular language $\}$

4- look book fg 2-14, 2-15 infix ... to Postfix

5- Construct a syntax-directed translation scheme that translates arithmebic expressions from infix notation into prefix notation in which an operator appears before its operands -xy is the prefix notation for x-y Give annotated parse tree for the inputs 9-5+2 and 9-5*2

Productions Grammar

expr → expr + term
| expr - term
| term
term → term * factor
| term / factor
| factor
factor → digit | (expr)
digit → 0 | ... | 9

translation schemes

expr → expr + term {print("+")}
| expr - term {print("-")}
| term
term → term * factor {print("*")}
| term / factor {print("/")}
| factor
factor → digit {print("digit")}
| (expr)
digit → 0 | ... | 9

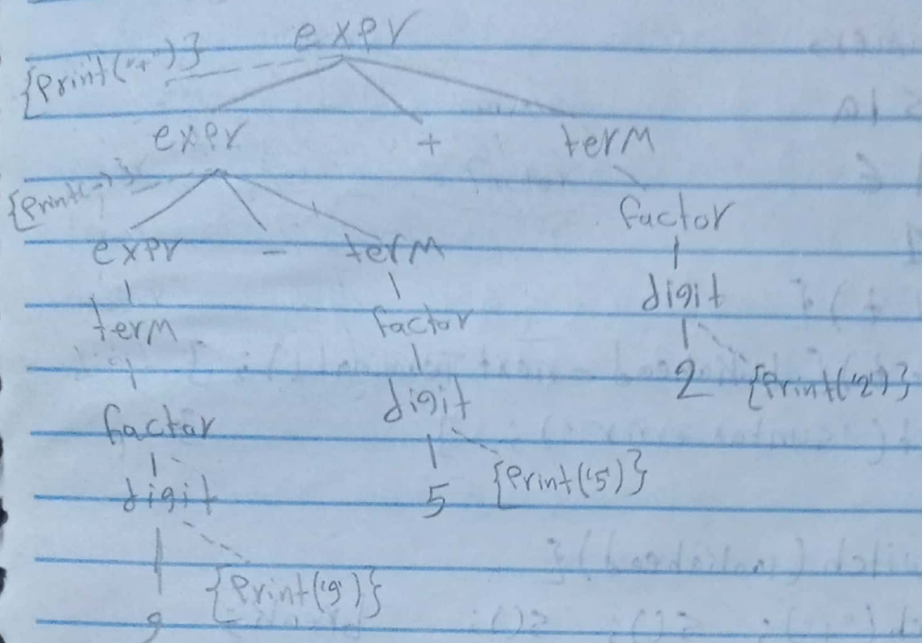
prefix infix Postfix

prefix Postfix infix

* الـ Parse tree بيتكون معي وانا عندي ويدي حسب المطلوب هنا طالب الـ
ادي ترتيب امشي بقواعد ما كما آخذتها في DS

fg 2.14

9 - 5 + 2   9 - 5 * 2



+-952

6 - in book look fg 2.16 و 2.17 و 2.18 خش عليها بس بالاكواد الشكلية

Predictive parsing   وامتحانات فيزياء مايو ♡

```
for ( optexpr ; optexpr ; optexper ) stmt
match(for);   match('(');   optexpr();   match(';');
optexpr();    match(';');   optexper();  match(')');
stmt()
```

الـ terminal بنعمل ماتش كونه terminal من قواعد اللغة
الـ not terminal ,, function

```
void match (terminal t) {
    if (lookahead == t)
        lookahead = next Terminal;
    else  report ("syntax error"); }
void optexer() {
    if (lookahead == expr)  match(expr); }
void stmt() { switch (lookahead) {
  case expr :  match(expr);  match(';');  break;
  case if :  match('if');  match('(');  match(exer);  match(')');
  case for :  match(for) كمل الباقي لآخرها  break;
  case other:  match(other);  break;
  default :  report ("syntax error");  }
}
```

look fg 2.19

كمل لآخرها break stmt() ومتنساش

Page: Date:

7- left Recursion expr → expr + term

A → A α | β

8- Construct recursive-descent parsers, starting with the following grammers

(a) S → + S S | - S S | a

(b) S → S (S) S | ε

(c) S → 0 S 1 | 0 1

```
Void match (Terminal t){
    if(lookahead = t){ lookahead = next Terminal(); }
    else { report('syntax error'); }
}
```

```
(a) Void S(){ switch(lookahead){
    Case "+"   Match("+");  S();  S();   break;
    Case "-"   Match("-");  S();  S();   break;
    Case "a"   Match("a");        break;
    default    report("syntax error"); }  }
(b) Void S(){
      if(lookahead == "(" ){
          match("("); S(); match(")"); S(); } }
(c) Void S(){  switch(lookahead){
    Case "0"   Match("0"); S(); match("1"); break;
    Case "1"   //match(epsilon);  break;
    default    report("syntax error"); }  }
```

9- in book fg 2.23 , 2.24 read 2.5.2   A → λR

الكود يفي للحل مشكلة اللوب 2.5.2   R → αR | βR | ε

الكود دة من جرامر fg 2.23

```
- Void expr(){ term(); rest(); }
- void term(){  if(lookahead is a digit){
      t = lookahead; match(lookahead); print(t); }
      else { report("syntax error");  }  }
```

Abdelkarem

Page: Date:

```
void rest(){ if (lookahead == '+'){
    match('+'); term(); print('+'); rest(); }
  else if (lookahead == '-'){
    match('-'); term(); print('-'); rest(); }
  else { } }
```

Fig 2.25

```
void rest(){ while (true){
    if (lookahead == '+'){
      match('+'); term(); print('+'); continue; }
    else if (lookahead == '-'){
      match('-'); term(); print('-'); continue; }
    break; } }
```

to-write a pseudoCode of a program that takes input from file and remove the new line and multiple spaces?

[1] or skip white space

```
for ( ; ; peek = next input character){
  if (peek is a blank or a tab) do nothing ;
  else if (peek is a newline) line = line + 1 ;
  else break; }
```

Fig 2.29

[2] Grouping digits into integers

```
if (peek holds a digit) { int v=0
  do{ v = v*10 + integer value of digit peek ;
      peek = next input character ;
  } while (peek holds a digit);
  return token(num, v) }
```

Fig 2.30

[3] Recognizing Key words and Identifiers

```
if (peek holds a letter){
  Collect letter or digits into a buffer b;
  s = string formed from the characters in b;
  w = token returned by words.get(s);
  if ( w is not Null)   return w;
  else Enter the key value pair (s, <id, s>) into words;
  return token(id, s);
}
```

Fig 2.31

handle آفضل

الكود الشفهي مط أي عماد في سعاد حسب السؤال المطلوب نظامة كالتالي

space — number — word — comend — relation — number float

```
Public Token Scan(){
//space
   for( ; ; peek = (char) system.in.read()){
      if( peek == ' ' || peek == '\t') continue;
      else if (peek == '\n') line = line + 1;
      else break;                                  }
//number
   if( character.isDigit(peek)){  int v=0
     do{ v = v*10 + character.digit(peek,10);
         peek = (char) system.in.read();
       } while (character.isDigit(peek));
     return new Num(v);   }
//word
   if(character.isLetter(peek)){
     string Buffer b = new stringBuffer();
     do{  b.append(peek);
          peek = (char) system.in.read();
       } while (character.isLetter(peek));
     string s = b.toString();
     word w = (word) words.get(s);
     if( w != null)      return w;
     else     w = new word(Tag.ID, s);
              words.put(s, w)
              return w;              }
// comment   single line comment
   if( peek == '/'){
        peek = (char) stream.read();
        if( peek == '/'){
           for( ; ; peek = (char) stream.read()){
              if (peek == '\n')  break;
          }
        }
   }else  Throw new syntaxException();
```

Page: Date:

~~//comment /*x/~~

```
if( Peek == '/') {
    Peek =(char) stream.read();
    if ( Peek == '*') {
        char PrevPeek = ' ';
        for(;; PrevPeek = Peek, Peek =(char) stream.read()) {
            if ( PrevPeek == '*' && Peek == '/') {
                break; }        }      }
}else    Throw New Syntax Exception();
// Relation sign{<, <=, ==, !=, >=, >}
if( "<=!>" .index of(Peek) > -1) {
    StringBuffer b = new StringBuffer();
    b.append (Peek);
    Peek = (char) stream.read();
    if (Peek == '=') { b.append (Peek); }
    return new Rel (b.toString());          }
//number no type Sensitive float
if( character.isDigit(Peek) || Peek == '.') {
    Boolean isDotExist = false;
    StringBuffer b = new StringBuffer();
    do{ if (Peek == '.') { isDotExist = true; }
        b.append (Peek);
        Peek = (char) stream.read();
    }while (isDotExist == true ? character.isDigit(Peek):
            Character.isDigit(Peek) || Peek == '.');
    return new Num (new float(b.toString())); }
Token t = new Token(Peek);
Peek = ' '; // Peek= blank;
return t;     }
```

أي أمر اساسي في اي كود اعتقد مهمش حسب المطلوب كونه عماد
emad Married 7 soad Code

Mostafa AbdelKarem
01121811782
2022

Desha
mansy
2022

CamScanner المسموحة ضوئيا بـ