ملفي الماسح والمعرب للعبارات الشرطية

إن في هذا المترجم سيتم تحديد فيما إذا كانت هذه العبارات مقبولة أم لا .

فمثلا العبارة if(a==2) then a+1; else a-1;

ستقتصر مهمة المترجم على تحديد فيما اذا كانت هذه العبارة صحيحة أم لا.

أولاً: ملف وصف الماسح:

```
%{
#include<stdlib.h>
#include<stdio.h>
#include"C:\Lex_Yacc\examples\if-lex\ify.h"
%}

blank [ \n\t]+
alpha [A-Za-z]
digit [0-9]

%%
```

المرحلة الأولى: مرحلة التضمين والتصريح

يتم فيها تضمين مكتبة <stdlib.h> والملف الرأسي (.h) وهو الملف الذي **ينتج عن المعرب ويحتوي تعريفات Token** .

المرحلة الثانية: مرحلة تحديد القوالب

يتم بها التصريح عن كل القوالب التي يستخدمها الماسح لمطابقة رموز سلسلة الدخل وتعريفها لـ Token:

قالب الفراغات blank [\n\t]+

قالب التعرف على الأحرف الأبجدية بأحرفها الكبيرة والصغيرة alpha [A-Za-z]

قالب التعرف على الأرقام digit[0-9]

%%:للفصل بين الأقسام.

```
{blank}
{digit}+    {yylval=atoi(yytext);
        return NUM;}
{alpha}({alpha}|{digit})*
       return ID;
if    return IF;
then    return THEN;
else    return ELSE;
"("    return LPAR;
")"    return RPAR;
";"    return SEMI;
"<"    return LT;
">"    return GT;
"<="    return LE;
">="    return GE;
"=="    return EQ;
"!="    return NE;
"||"    return OR;
"&&"    return  AND ;
"+"             return PLUS ;
"*"             return MULT ;
"-"             return  MINUS ;
"/"             return DIVS ;
"="             return EQUAL ;
"^"         return POWER;
%%
```

المرحلة الثالثة: مرحلة الاستجابة للقوالب

قاعدة القالب الأول تجاهل الفراغ {blank}

قاعدة القالب الثاني لاستجابة للأرقام {digit}+

قاعدة القالب الثالث للاستجابة للمتغيرات ويشترط بها أن تبدأ بحر ف أبجدي

إذا صادف الماسح if سيعيد IF

إذا صادف الماسح then سيعيد THEN

إذا صادف الماسح else سيعيد ELSE

إذا صادف الماسح "(" سيعيد LPAR

إذا صادف الماسح ")" سيعيد RPAR

إذا صادف الماسح ";" سيعيد SEMI

إذا صادف الماسح "<" سيعيد LT

إذا صادف الماسح ">" سيعيد GT

إذا صادف الماسح "<=" سيعيد LE

إذا صادف الماسح ">=" سيعيد GE

إذا صادف الماسح "==" سيعيد EQ

إذا صادف الماسح "!=" سيعيد NE

إذا صادف الماسح "||" سيعيد OR

إذا صادف الماسح "&&" سيعيد AND

إذا صادف الماسح "+" سيعيد plus

إذا صادف الماسح "*" سيعيد MULT

إذا صادف الماسح "-" سيعيد MINUS

إذا صادف الماسح "/" سيعيد DIVS

إذا صادف الماسح "=" سيعيد EQUAL

إذا صادف الماسح "^" سيعيد POWER

ثانياً: ملف وصف المعرب:

```
%{
#include <stdio.h>
#include<conio.h>
#include <stdlib.h>
#include <math.h>
#include"C:\Lex_Yacc\examples\if-lex\ifl.c"
%}

%token ID NUM POWER IF THEN LE
 GE EQ NE OR AND ELSE EQUAL  LT
 GT PLUS MINUS MULT DIVS LPAR
 RPAR SEMI

%right EQUAL
%left AND OR
%left LT GT LE GE EQ NE
%left PLUS MINUS
%left MULT DIVS
%%
```

المرحلة الأولى: مرحلة التضمين

هي المرحلة التي يتم بها تضمين المكتبات المطلوبة

المكتبة stdio من أجل التابع printf()

المكتبة conio من أجل التابع clrscr()

المكتبة stdlib من أجل التابع exit(0)

وتضمين ملف الـ C الذي ينتج من تنفيذ بيئة الماسح .

المرحلة الثانية: مرحلة التصريح عن السلاسل اللفظية Tokenوتحديد الأولويات

إن بهذا التصريح سيتم تحديد المفردات التي يتم التعامل معها وهنا في هذا المعرب هي العمليات المنطقية والحسابية و المتغيرات والأقواس والفاصلة المنقوطة.

المرحلة الثالثة: مرحلة قواعد الاعراب

```
S     : ST {printf("Input accepted.\n")};
ST    : IF LPAR E2 RPAR THEN ST1 SEMI ELSE ST1 SEMI
      | IF LPAR E2 RPAR THEN ST1 SEMI ;
ST1   : ST | E ;
E     : ID EQUAL E
      | E PLUS E
      | E MINUS E
      | MINUS E
      | E MULT E
      | E DIVS E {if($3==0) yyerror("can't devide by zero");}
      | E POWER E
      | E LT E
      | E GT E
      | E LE E
      | E GE E
      | E EQ E
      | E NE E
      | E OR E
      | E AND E
      | ID
      | NUM ;
E2    : E LT E
      | E GT E
      | E LE E
      | E GE E
      | E EQ E
      | E NE E
      | E OR E
      | E AND E
      | ID
      | NUM;
```

الملف S هو جملة ST وبحال صحتها سيطبع عبارة أن الدخل مقبول.

الجملة STتحدد صيغة العبارة المراد اعرابها وهي عبارة شرطية والتي تتألف من مفردات Token ومفردات E2،ST1 .

ST1: هو إما عودة لعبارة شرطية أو تعبير E

التعبير E هو إما اسناد رقم لمتحول أو عملية حسابية لتعبيرين أو عملية منطقية أو رقم أو متحول

التعبير E2 هو حتما عملية منطقية

أو رقم أو متغير لأن E2 سيتم اختبارها بالشرط if

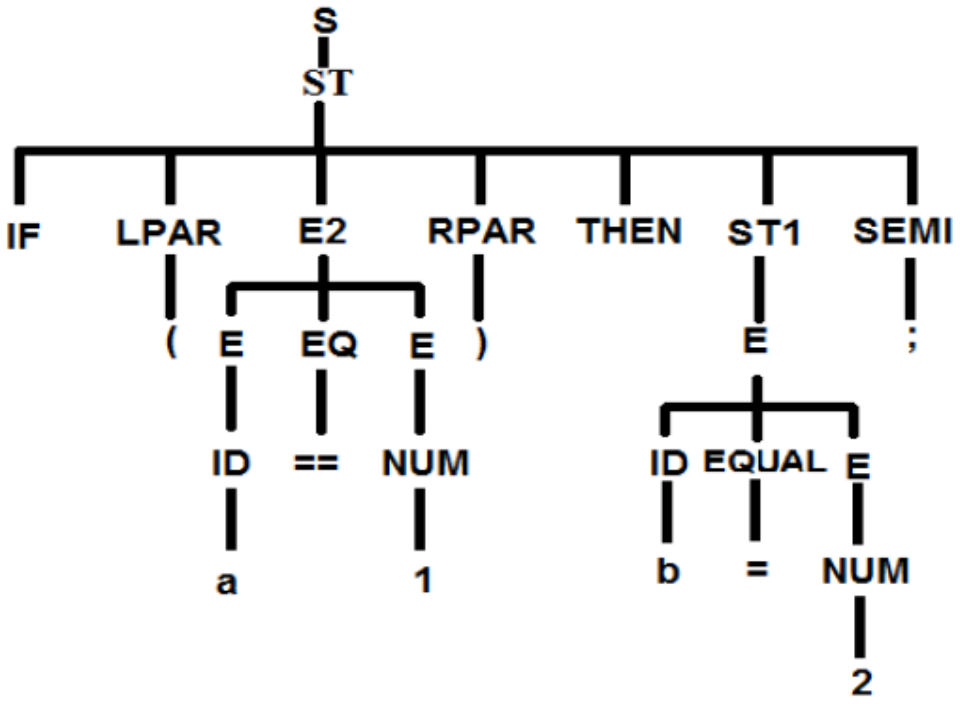
```
%%
int yyerror (char *s)
{
printf("%s\n",s);
}
int yywrap(){
return 1;
}
main()
{
clrscr();
if((yyin=fopen("C:\\Lex_Yacc\\examples\\if-lex\\input.txt","r"))==NULL
{
printf("input.txt not found !\n");
return;
}
  yyparse();
return;
}
```

بناء شجرة الاعراب لقواعد الاعراب للعبارة الشرطية :

نريد بناء شجرة اعراب للعملية التالية: if(a==1) then b=1;

حسب القواعد المحددة

```
S     : ST
ST    : IF LPAR E2 RPAR THEN ST1 SEMI ELSE ST1 SEMI
      | IF LPAR E2 RPAR THEN ST1 SEMI;
ST1   : ST| E;
E     : ID EQUAL E | E PLUS E | E MINUS E | MINUS E | E MULT E | E DIVS E | E LT
E | E GT E | E LE E | E GE E | E EQ E | E NE E | E OR E | E AND E | ID | NUM;
E2    : E LT E | E GT E | E LE E | E GE E | E EQ E | E NE E | E OR E | E AND E | ID |
NUM;
```



من جديد، باستخدام القواعد السابقة سنقوم ببناء شجرة الإعراب للعبارة:

if (a==1) then b=1; else b=2;