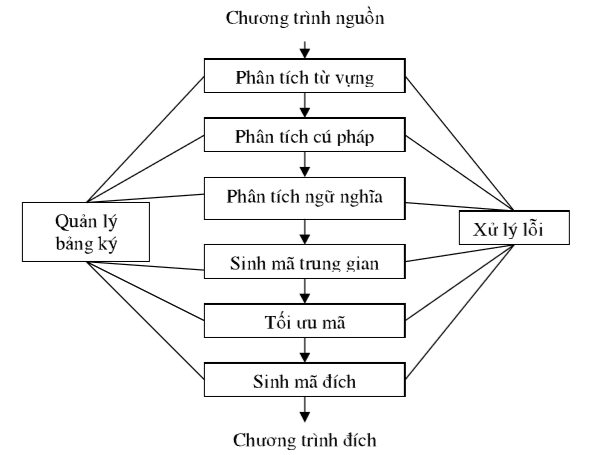
1. Chương trình dịch:

Chương trình dịch là một chương trình máy tính làm công việc dịch một chuỗi các câu lệnh được viết bằng một ngôn ngữ lập trình (gọi là ngôn ngữ nguồn hay mã nguồn), thành một chương trình tương đương nhưng ở dưới dạng một ngôn ngữ máy tính mới (gọi là ngôn ngữ đích) và thường là ngôn ngữ ở cấp thấp hơn, như ngôn ngữ máy. Chương trình mới được dịch này gọi mã đối tượng.



Hình 1: Mô hình của một trình biên dịch

Quy trình thông thường của một trình biên dịch được trình bày như sau:



Hinh 2: Các giai đoạn thực thi của trình biên dịch

Flex là chương trình thực hiện chức năng phân tích từ vựng của một tình biên dịch.

3. Biểu thức chính quy

Biểu thức chính quy là một chuỗi miêu tả một bộ các chuỗi khác, theo những quy tắc cú pháp nhất định. Một biểu thức chính quy định nghĩa một khuôn mẫu tìm kiếm chuỗi.

Flex sử dụng biểu thức chính quy để làm khuôn mẫu định nghĩa xâu đầu vào. Bất cứ khi nào xâu đầu vào thỏa mãn một khuôn mẫu bất kỳ, chương trình sẽ thực hiện các hành động tương ứng đã được khai báo sẵn.

Khi chương trình flex thực thi, nó sẽ quét các xâu đầu vào và tìm kiếm khuân mẫu phù hợp nhất với xâu đó. Nếu tìm được nhiều hơn một khuân mẫu phù hợp với xâu đầu vào, chương trình sẽ chọn thực thi hành động của luật xuất hiện sớm hơn.

Khi một xâu đã được xác định phù hợp với một khuân mẫu nào đó, xâu đó sẽ được lưu trong biến yytext và độ dài của nó được lưu trong biến yyleng.

Không có luật nào được tìm thấy, chương trình sẽ tự động quét các từ tiếp theo.

Một số biểu thức chính quy:

|  |  |
| --- | --- |
| Biểu thức chính quy | Mô tả |
| ‘.’ | Khớp với bất cứ ký tự nào |
| [xyz] | Khớp với ký tự a hoặc b hoặc c |
| [abj-oZ] | Ký tự a hoặc b hoặc 1 ký tự trong khoảng từ j dến o hoặc ký tự Z |
| [^A-Z] | Không phải là một ký tự trong khoảng từ A đến Z |
| r\* | Không hoặc nhiều ký tự r cạnh nhau |
| r+ | Một hoặc nhiều ký tự r cạnh nhau |
| r? | Không hoặc một ký tự r |
| r{2,5} | 2 đến 5 ký tự r cạnh nhau |
| r{2,} | Nhiều hơn 2 ký tự r |
| ‘x’ | Ký tự x |
| ‘r$’ | Ký tự r ở cuối dòng |

4. Ngôn ngữ PL0

Là ngôn ngữ lập trình đơn giản, phục vụ trong giảng dạy, có cấu trúc tựa Pascal, chứa đặc trưng của một ngôn ngữ lập trình bậc cao.

Từ vựng của PL/0:

* Chữ cái: a-z,A-Z
* Chữ số 0-9
* Dấu đơn + - \* / **%** ( ) **[** **]** > < = , ; .
* Dấu kép := >= <= <>
* Từ khóa begin, end, if, then, while, do, call, odd, to const, var, procedure, program, else, for
* Số nguyên có tối đa 9 chữ số
* Định danh độ dài tối đa 10 ký tự

Từ tố của PL/0:

* Số nguyên NUMBER
* Định danh IDENT
  + Nếu từ vựng trùng với khóa, từ vựng sẽ mang ý nghĩa từ tố trùng với tên từ khóa
* Toán tử
  + + (PLUS) - (MINUS)
  + \* (TIMES) / (SLASH) %(PERCENT)
  + = (EQU) <> (NEQ)
  + < (LSS) <= (LEQ)
  + > (GRT) >= (GEQ)
  + ( (LPARENT) ) (RPARENT)
  + [ (LBRACK) ] (RBRACK)
  + . (PERIOD)
  + , (COMMA)
  + ; (SEMICOLON)
  + := (ASSIGN)
* Các trường hợp khác: từ tố NONE

5. Demo chương trình PL/0

Chương trình phân tích cú pháp ngôn ngữ PL/0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | %{ | |  |  | |  | #include<string.h> | |  | %} | |  | DIGIT [0-9] | |  | ID [a-zA-Z][A-Za-z0-9]\* | |  |  | |  |  | |  | %% | |  |  | |  |  | |  |  | |  |  | |  | {DIGIT}+ { | |  | int value = 0; | |  | int i= 0; | |  | int L = strlen(yytext); | |  |  | |  | if(L>9){ | |  | printf("\nSo qua lon %s",yytext); | |  | } | |  | else{ | |  | value = value\*10 + yytext[i]-48; | |  | i=i+1; | |  | while(i<L){ | |  | value = value\*10 + yytext[i]-48; | |  | i = i + 1; | |  | } | |  |  | |  |  | |  | if(i==L) { | |  | printf("\nNUMBER: %s", yytext); | |  | } | |  | } | |  |  | |  | } | |  |  | |  | BEGIN|CALL|CONST|DO|ELSE|END|FOR|IF|ODD|PROCEDURE|PROGRAM|THEN|TO|VAR|WHILE { | |  | printf("\n%s", yytext ); | |  | } | |  |  | |  |  | |  | {ID} { | |  | char s[11]; | |  | int i=0; | |  | if(strlen(yytext) >=10){ | |  | while(i<10){ | |  | s[i] = yytext[i]; | |  | //printf("\nIDENT: %c", yytext[i] ); | |  | i+=1; | |  | } | |  | s[i]='\0'; | |  | printf("\nIDENT: %s", s ); | |  | }else | |  | printf("\nIDENT: %s", yytext ); | |  |  | |  | } | |  | "+" printf( "\nPLUS"); | |  | "-" printf( "\nMINUS"); | |  | "\*" printf( "\nTIMES"); | |  | "/" printf( "\nSLASH"); | |  | "<>" printf( "\nNEQ"); | |  | "<=" printf( "\nLEQ"); | |  | "<" printf( "\nLSS"); | |  | ">=" printf( "\nGEQ"); | |  | ">" printf( "\nGRT"); | |  | ":=" printf( "\nASSIGN"); | |  | "%" printf( "\nPERCENT"); | |  | "=" printf( "\nEQU"); | |  | "(" printf( "\nLPARENT"); | |  | ")" printf( "\nRPARENT"); | |  | "[" printf( "\nLBRACK"); | |  | "]" printf( "\nRBRACK"); | |  | "." printf( "\nPERIOD"); | |  | "," printf( "\nCOMMA"); | |  | ";" printf( "\nSEMICOLON"); | |  | '.' ; | |  | %% | |  |  | |  | int main( int argc, char \*\*argv ) | |  | { | |  | ++argv, --argc; /\* skip over program name \*/ | |  | if ( argc > 0 ) | |  | yyin = fopen( argv[0], "r" ); | |  | else | |  | yyin = stdin; | |  |  | |  | yylex(); | |  | } | |

Với đầu vào là một file

|  |
| --- |
| Program Example2;  Var asdfsdfvdfvfdfdbdfvcvdsxcdsdsdxdscvf,b,c;  Procedure gcd(a; b; var d);  Begin  While a <> b do  if a > b then a := a - b  else b := b - a;  d := b;  End;  Begin  Call Readln(a); Call Readln(b);  Call gcd(a,b,c);  Call Writeln(c);  End |

Sẽ cho ta kết quả 