

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN CUỐI KÌ MÔN NHẬP MÔN NGÔN NGỮ LẬP
TRÌNH**

Lexical analyzer and syntax analysis for PASC language

Người hướng dẫn: **GV HÀ LÊ HOÀI TRUNG**

Người thực hiện: **NGUYỄN TRUNG TÍNH - 51603330**

NGUYỄN TUẤN HUY - 51603144

Lớp : 16050302

Khoá : 20

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2019

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN CUỐI KÌ MÔN NHẬP MÔN NGÔN NGỮ LẬP
TRÌNH**

Lexical analyzer and syntax analysis for PASC language

Người hướng dẫn: **GV HÀ LÊ HOÀI TRUNG**

Người thực hiện: **NGUYỄN TRUNG TÍNH - 51603330**

NGUYỄN TUẤN HUY - 51603144

Lớp : 16050302

Khoá : 20

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2019

LỜI CẢM ƠN

Đồ án này được hoàn thành dưới sự hỗ trợ tận tình của thầy Hà Lê Hoài Trung. Qua việc làm đề tài này chúng em đã học hỏi được những kinh nghiệm thầy truyền đạt lại và những kiến thức mới trong quá trình làm. Chúng em gửi lời cảm ơn sâu sắc tới thầy. Một lần nữa chúng em xin cảm ơn!

ĐỒ ÁN ĐƯỢC HOÀN THÀNH TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

Tôi xin cam đoan đây là sản phẩm đồ án của riêng chúng tôi và được sự hướng dẫn của thầy Hà Lê Hoài Trung. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình. Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

TP. Hồ Chí Minh, ngày tháng năm

Tác giả

(ký tên và ghi rõ họ tên)

Nguyễn Trung Tính

Nguyễn Tuấn Huy

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

Phần xác nhận của GV hướng dẫn

Tp. Hồ Chí Minh, ngày tháng năm
(kí và ghi họ tên)

Phần đánh giá của GV chấm bài

Tp. Hồ Chí Minh, ngày tháng năm
(kí và ghi họ tên)

TÓM TẮT

Trong đề án, chúng tôi sẽ trình bày văn phạm phi ngữ cảnh và các sơ đồ trạng thái của những token. Có nghĩa là chúng tôi sẽ viết một công cụ phân tích ngữ nghĩa của một đoạn chương trình PASC, đưa ra những loại token đã được phát hiện cùng với đó là công cụ để sinh ra cây cú pháp, kiểm tra cú pháp của cây (Abstract Syntax Tree hay gọi tắt là AST). Và văn phạm phi ngữ cảnh sẽ giúp chúng ta hiểu hơn quá trình phân tích mà công cụ thực hiện mà mô hình hóa thành một cây cú pháp để biết được cấu trúc như thế nào.

MỤC LỤC

LỜI CẢM ƠN	i
PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN	iii
TÓM TẮT	iv
MỤC LỤC	1
DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ.....	2
CHƯƠNG 1 – VĂN PHẠM PHI NGỮ CẢNH VÀ LƯỢC ĐỒ TRẠNG THÁI.....	3
1.1 Văn phạm phi ngữ cảnh	3
1.2 Lược đồ trạng thái.....	5
CHƯƠNG 2 – DEMO	9
2.1 Phase 1.....	9
2.1.1 TestCase-1:	9
2.1.2 TestCase-2:	9
2.1.3 TestCase-3:	9
2.1.4 TestCase-4:	10
2.1.5 TestCase-5:	10
2.1.6 TestCase-6:	11
2.2 Phase 2.....	11
TÀI LIỆU THAM KHẢO	13

DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ

DANH MỤC HÌNH

Hình 1 Comparasion	5
Hình 2 Identifier	6
Hình 3 Number	6
Hình 4 String	7
Hình 5 Whitespace	7
Hình 6 Symbol	8
Hình 7: testcase 1 và output	9
Hình 8: testcase 2 và output	9
Hình 9: testcase 3 và output	9
Hình 10: testcase 4 và output	10
Hình 11: testcase 5 và output	10
Hình 12: testcase 6 và output	11
Hình 13: Code demo phase_2	11
Hình 14: Parse_tree	12

CHƯƠNG 1 – VĂN PHẠM PHI NGỮ CẢNH VÀ LƯỢC ĐỒ TRẠNG THÁI

Sau đây để hiểu rõ hơn về cấu trúc để phân tích ngữ nghĩa và cú pháp của chương trình được viết bằng ngôn ngữ PASC, chúng tôi sẽ trình bày 2 phần:

1.1 Văn phạm phi ngữ cảnh

Ký hiệu bắt đầu cho chương trình là *program*.

program: PROGRAMnumber expr SEMInumber declare block

//Expresstion:

expr → expr operator expr
 | expr assign
 | paren
 | constvalue
 | expr COMMANumber expr
 | declare
 | ifStatement
 | loopStatement
 | identifier
 | constnumber
 | string
 | forStatement
 | repeat

//Block

block → BEGINnumber expr + ENDnumber;

//Loop-Statment

loopStatement → WHILEnumber condition expr + (while)
 | LOOPnumber expr + ENDLOOPnumber (loop) ;

//If-Statement

ifStatement → IFnumber condition THENnumber expr + (ELSEnumber (block|expr+))
 ENDIFnumber

 | IFnumber condition THENnumber expr + ENDIFnumber;

//Condition

condition → expr compare=(EQnumber|LEnumber|GEnumber|NEnumber) expr
 | condition ANDnumber condition
 | condition ORnumber condition;

```

//Operator
operator → (PLUSnumber|MINUSnumber|TIMESnumber|DIVnumber);

//Assign
assign → COLEQnumber expr SEMInumber
        | expr COLEQnumber constnumber;

//Const-number
constnumber → ICONTSnumber;

//const-value
constvalue → CONSTnumber expr SEMInumber;

//Paren
paren → LPAREN expr RPAREN;

//Identifier
identifier → IDnumber;

//declare-value
declare → VARnumber expr SEMInumber;

//String
string → '(' expr '*';

//for
forStatement → FORnumber COLEQnumber expr TOnumber constnumber 'do' expr+
              | FORnumber COLEQnumber expr DOWNTOnumber constnumber 'do' expr+;

//repeat
repeat → REPEATnumber expr+ UNTILnumber condition SEMInumber;

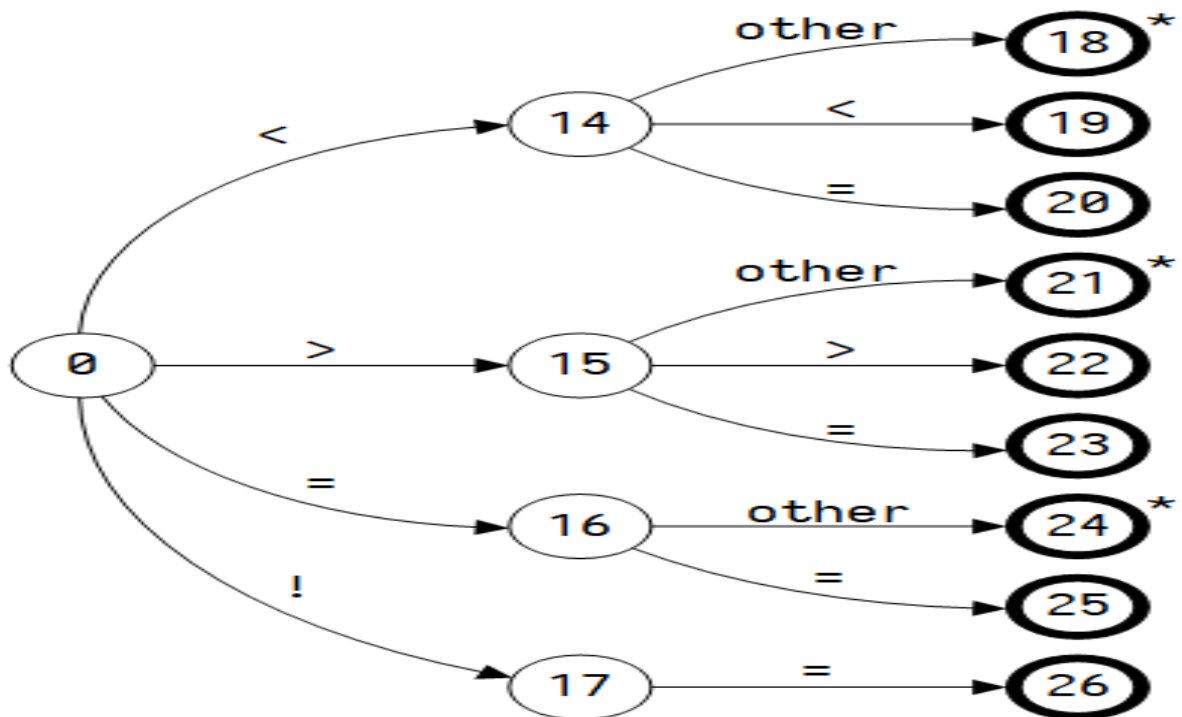
```

Định danh một số từ dùng trong văn phạm phi ngữ cảnh:

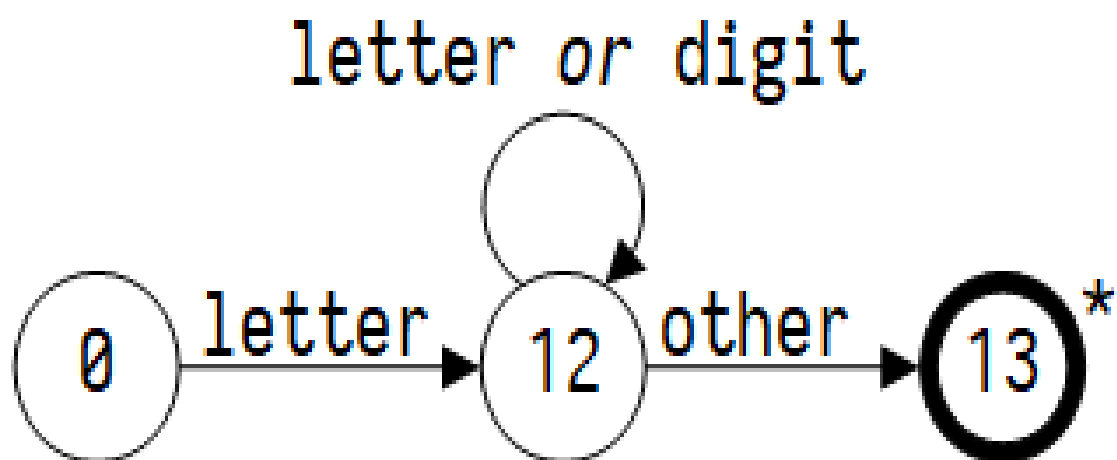
<i>// Whitespace</i>		DOWNTOnumber	→ 'downto'
WS → [\t\n\r]+ -> channel(HIDDEN)		ELSEnumber	→ 'else'
		ELSIFnumber	→ 'elsif'
<i>// Keywords</i>		ENDnumber	→ 'end'
VARnumber	→ 'var'	ENDIFnumber	→ 'endif'
ANDnumber	→ 'and'	ENDLOOPnumber	→ 'endloop'
ARRAYnumber	→ 'array'	ENDRECnumber	→ 'endrec'
BEGINnumber	→ 'begin'	EXITnumber	→ 'exit'
CONSTnumber	→ 'const'	FORnumber	→ 'for'
DIVnumber	→ 'div'	FORWARDnumber	→ 'forward'

FUNCTIONnumber	→ 'function'	// Operators & signs	
IFnumber	→ 'if'	PLUSnumber	→ '+'
ISnumber	→ 'is'	MINUSnumber	→ '-'
LOOPnumber	→ 'loop'	TIMESnumber	→ '*'
NOTnumber	→ 'not'	COLEQnumber	→ ':='
OFnumber	→ 'of'	LPARENnumber	→ '('
ORnumber	→ 'or'	RPARENnumber	→ ')'
PROCEDUREnumber	→ 'procedure'	DOTnumber	→ '.'
PROGRAMnumber	→ 'program'	SEMInumber	→ ';'
RECORDnumber	→ 'record'	COLONnumber	→ ':'
REPEATnumber	→ 'repeat'	COMMAnumber	→ ','
RETURNnumber	→ 'return'	LTnumber	→ '<'
THENnumber	→ 'then'	GTnumber	→ '>'
TOnumber	→ 'to'	EQnumber	→ '=='
TYPEnumber	→ 'type'	LEnumber	→ '<='
UNTILnumber	→ 'until'	GEnumber	→ '>='
WHILEnumber	→ 'while'	NEnumber	→ '<>'
// Literals		// Identifiers	
ICONTSnumber	→ [0-9]+	IDnumber	→ [A-Za-z0-9]+

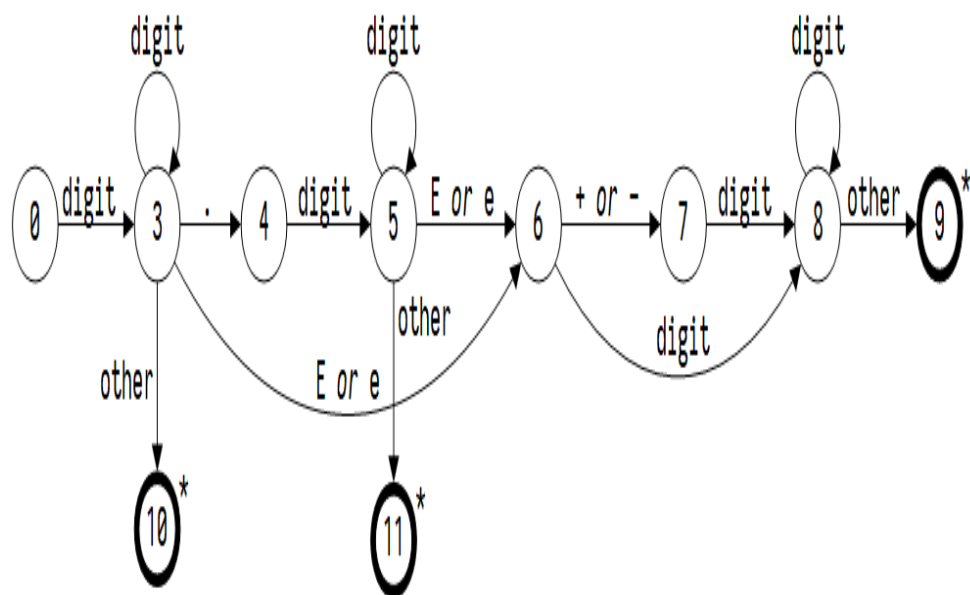
1.2 Lược đồ trạng thái



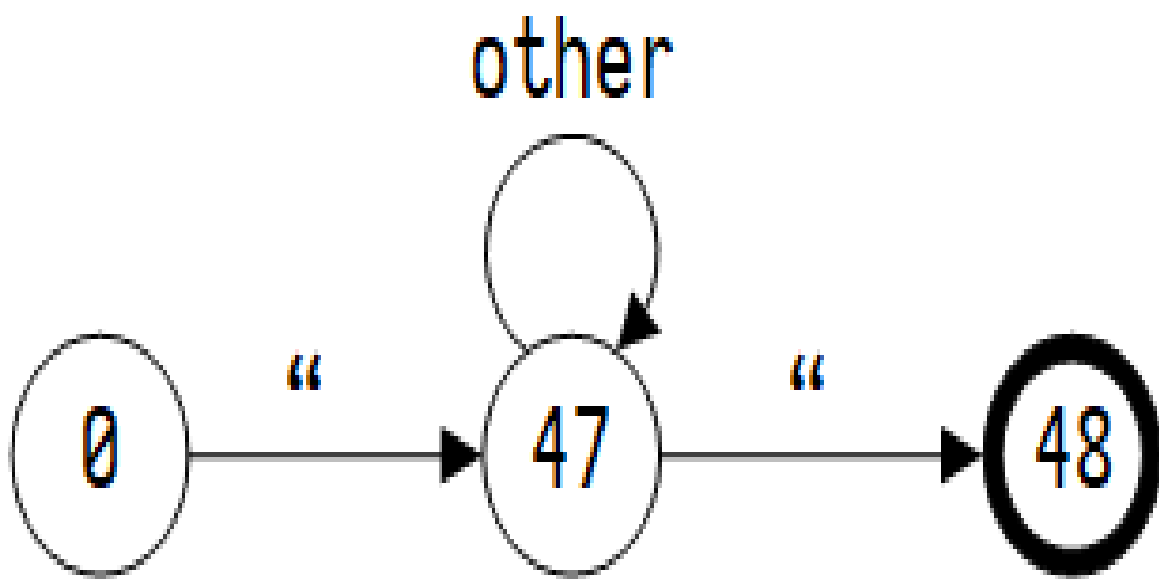
Hình 1: Comparison



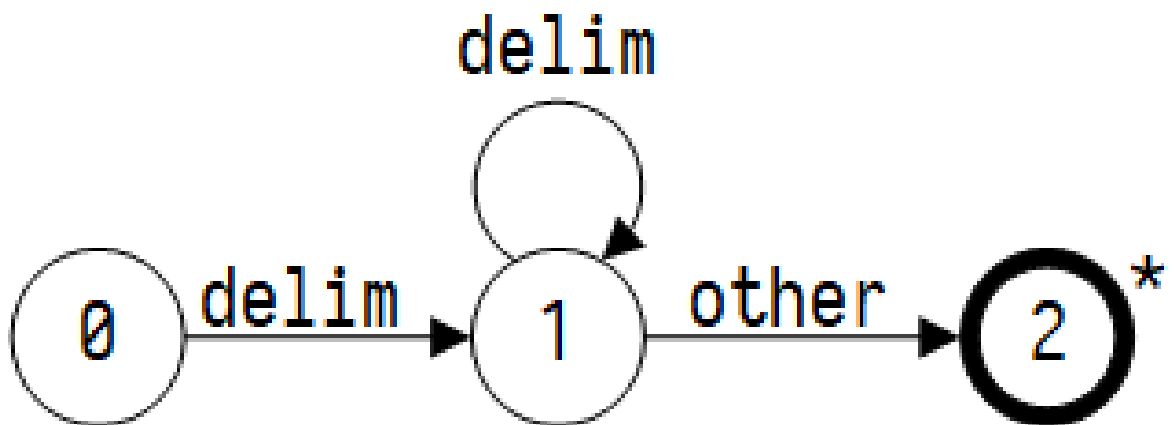
Hình 2: Identifier



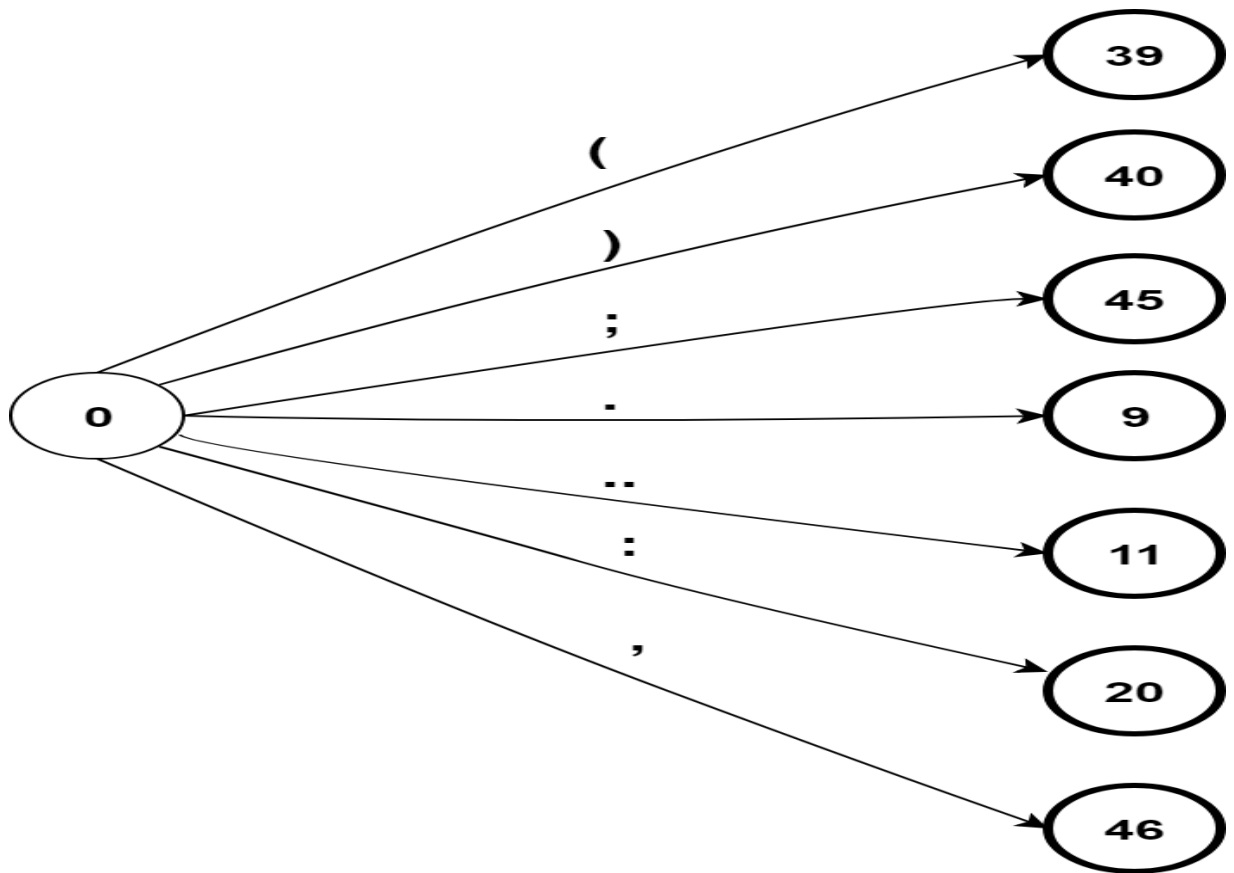
Hình 3: Number



Hình 4: String



Hình 5: Whitespace

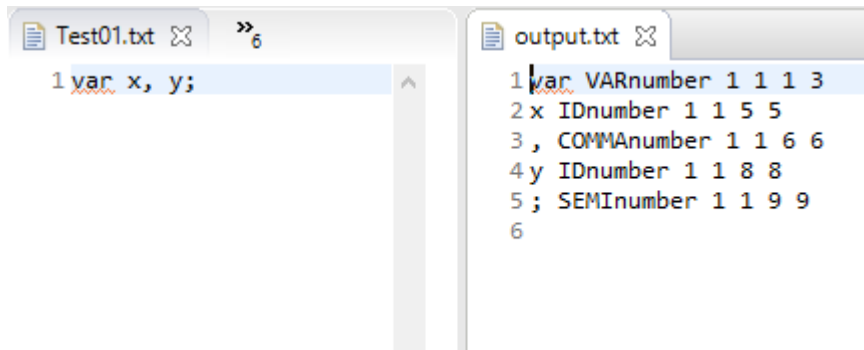


Hình 6: Symbol

CHƯƠNG 2 – DEMO

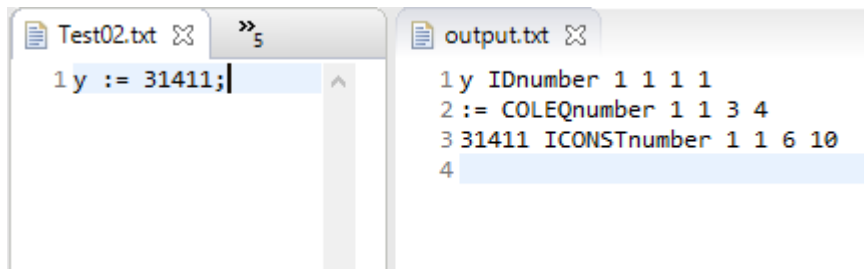
2.1 Phase 1

2.1.1 TestCase-1:



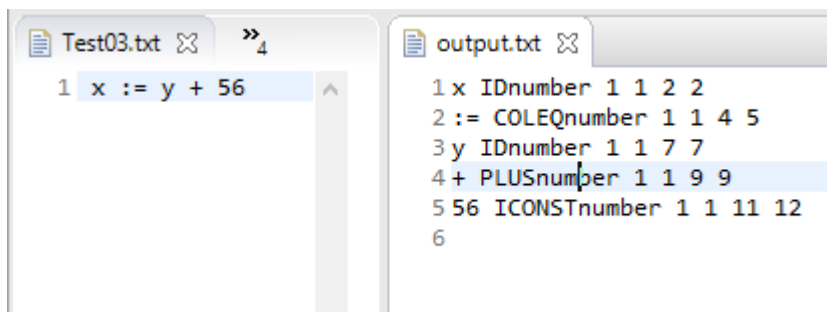
Hình 7: testcase 1 và output

2.1.2 TestCase-2:



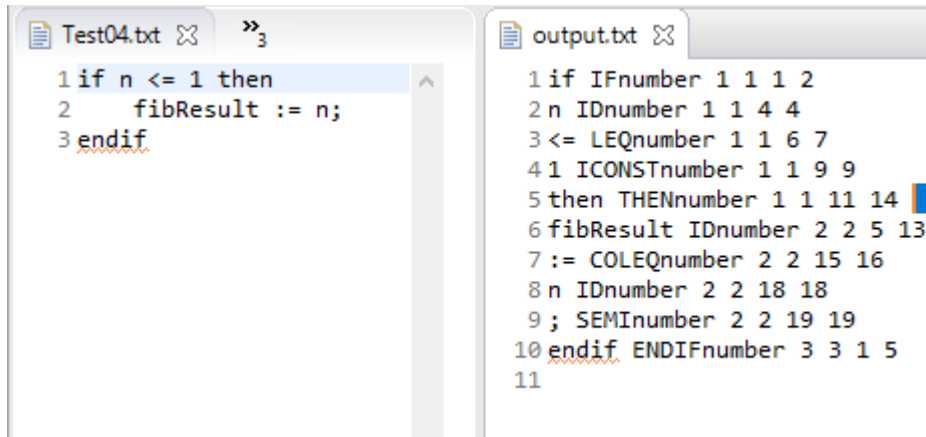
Hình 8: testcase 2 và output

2.1.3 TestCase-3:



Hình 9: testcase 3 và output

2.1.4 TestCase-4:



The screenshot shows a code editor with two tabs: 'Test04.txt' and 'output.txt'. The 'Test04.txt' tab contains the following code:

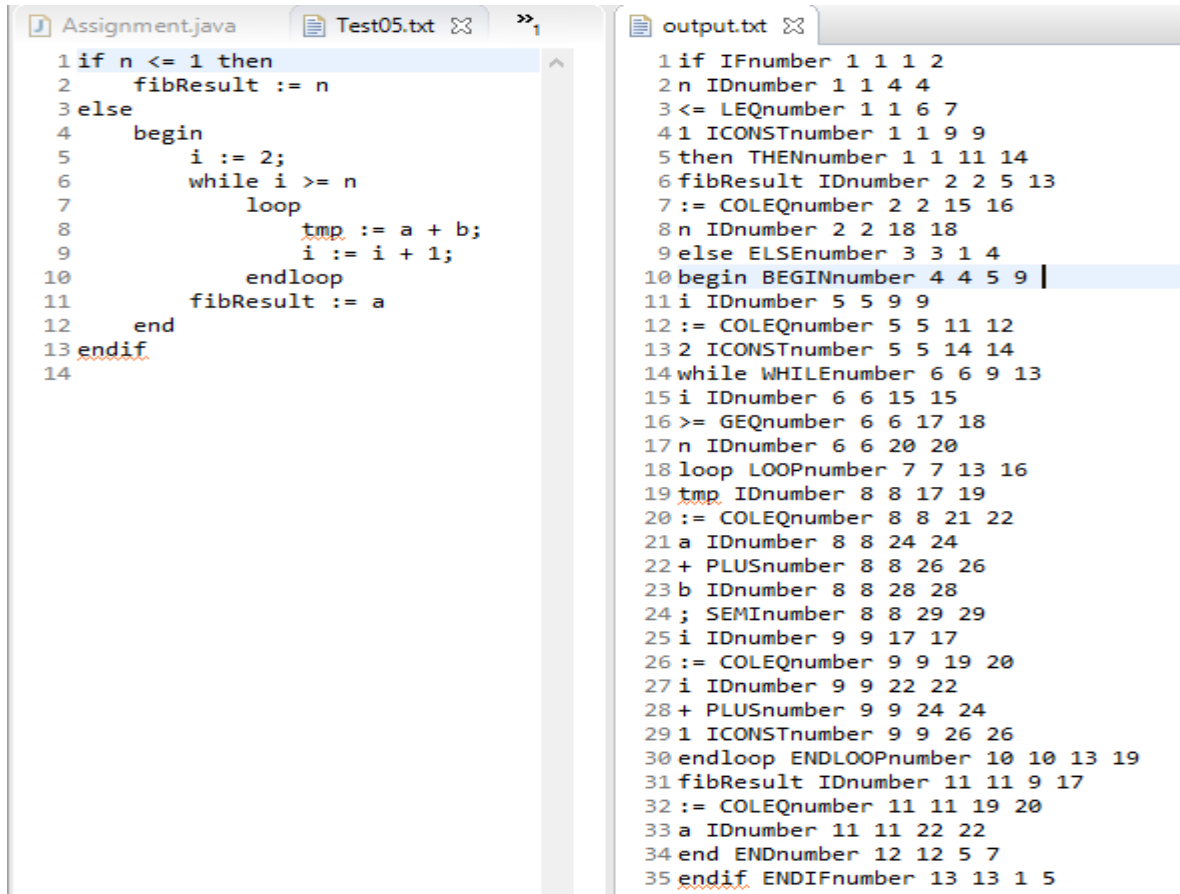
```
1 if n <= 1 then
2   fibResult := n;
3 endif
```

The 'output.txt' tab contains the following output:

```
1 if IFnumber 1 1 1 2
2 n IDnumber 1 1 4 4
3 <= LEQnumber 1 1 6 7
4 1 ICONSTnumber 1 1 9 9
5 then THENnumber 1 1 11 14
6 fibResult IDnumber 2 2 5 13
7 := COLEQnumber 2 2 15 16
8 n IDnumber 2 2 18 18
9 ; SEMInumber 2 2 19 19
10 endif ENDIFnumber 3 3 1 5
11
```

Hình 10: testcase 4 và output

2.1.5 TestCase-5:



The screenshot shows a code editor with three tabs: 'Assignment.java', 'Test05.txt', and 'output.txt'. The 'Test05.txt' tab contains the following code:

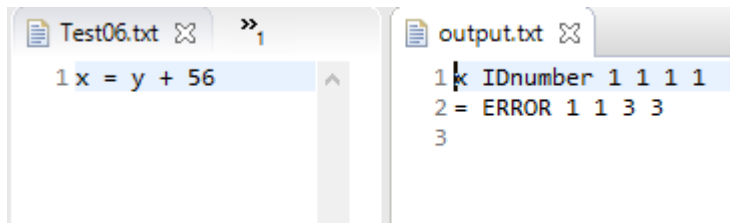
```
1 if n <= 1 then
2   fibResult := n
3 else
4   begin
5     i := 2;
6     while i >= n
7       loop
8         tmp := a + b;
9         i := i + 1;
10      endloop
11      fibResult := a
12    end
13 endif
14
```

The 'output.txt' tab contains the following output:

```
1 if IFnumber 1 1 1 2
2 n IDnumber 1 1 4 4
3 <= LEQnumber 1 1 6 7
4 1 ICONSTnumber 1 1 9 9
5 then THENnumber 1 1 11 14
6 fibResult IDnumber 2 2 5 13
7 := COLEQnumber 2 2 15 16
8 n IDnumber 2 2 18 18
9 else ELSEnumber 3 3 1 4
10 begin BEGINnumber 4 4 5 9
11 i IDnumber 5 5 9 9
12 := COLEQnumber 5 5 11 12
13 2 ICONSTnumber 5 5 14 14
14 while WHILEnumber 6 6 9 13
15 i IDnumber 6 6 15 15
16 >= GEQnumber 6 6 17 18
17 n IDnumber 6 6 20 20
18 loop LOOPnumber 7 7 13 16
19 tmp IDnumber 8 8 17 19
20 := COLEQnumber 8 8 21 22
21 a IDnumber 8 8 24 24
22 + PLUSnumber 8 8 26 26
23 b IDnumber 8 8 28 28
24 ; SEMInumber 8 8 29 29
25 i IDnumber 9 9 17 17
26 := COLEQnumber 9 9 19 20
27 i IDnumber 9 9 22 22
28 + PLUSnumber 9 9 24 24
29 1 ICONSTnumber 9 9 26 26
30 endloop ENDLOOPnumber 10 10 13 19
31 fibResult IDnumber 11 11 9 17
32 := COLEQnumber 11 11 19 20
33 a IDnumber 11 11 22 22
34 end ENDnumber 12 12 5 7
35 endif ENDIFnumber 13 13 1 5
```

Hình 11: testcase 5 và output

2.1.6 TestCase-6:



Hình 12: testcase 6 và output

2.2 Phase 2

```

program Test;
var x,y;
begin
  if n <= 1 then
    fibResult := n;
  else
    begin
      i := 2;
      while i >= n
        loop
          tmp := a + b;
          i := i + 1;
        endloop
      fibResult := a;
    end
  endif
end

```

Hình 13: Code demo phase_2

Hình 14: Parse_tree

TÀI LIỆU THAM KHẢO

1. **<https://dzone.com/articles/building-a-compiler-for-your-own-language-from-the-1>**
2. **<https://tomassetti.me/parse-tree-abstract-syntax-tree/>**