COMPILADORES

Universidade Federal de Alagoas
Instituto de Computação

GimmeDaddy - Análise sintática

Leandro Martins de Freitas Douglas Henrique Maximo da Silva

2017.2

Sumário

1 Errata

Fizemos algumas modificações e correções nos códigos exemplos e também em algumas categorias.

1.1 Categorias

A categoria MAIN foi retirada, porém a restrição de que o código só é aceito com um método main ainda existe.

As categorias OPR_DIF, OPR_MEN_IG, OPR_MEN, OPR_MAI_IG, OPR_MAI e OPR_IGUAL foram substituídas por uma única categoria chamada OPR.

As categorias SE e SENAO foram substituídas por IF e ELSE, respectivamente.

As categorias ENTRADA, SAIDA e SAIDA_LN foram substituídas por READ, PRINT e PRINTLN, respectivamente.

Adicionamos uma categoria chamada IN, que faz parte reformulação (explicada na próxima seção) da estrutura de repetição **for**.

1.2 Nova construção for

Na versão anterior da linguagem, a construção da estrutura de repetição **for** dava-se da seguinte maneira:

```
1 for (<variavelControle> = <valorInicial>; <condicaoLogica>){
2    <corpoDoBloco>
3 }
```

Agora fazemos assim:

```
1 for <variavelControle> in ([<valorInicial>,]<valorFinal>){
2    <corpoDoBloco>
3 }
```

Onde [<valorInicial>,] indica uma expressão opcional. Caso seja inserida, a repetição vai de <valorInical> a <valorFinal> - 1. Caso a expressão seja omitida, a repetição começa em 0 e vai até <valorFinal> - 1. Em ambos os casos o incremento da variável de controle é de uma unidade.

1.3 Correção no programa exemplo: ShellSort

Modificação no for e na declaração das variáveis - não é permitido declarações de variáveis na mesma linha, o que ocorria no código anterior.

```
1  void shellSort(int vet[], int size) {
2      int i;
3      int j;
4      int value;
5      int gap;
6      gap = 1;
7
8      while(gap < size) {
9          gap = 3*gap+1;
10     }
11
12     while(gap > 1) {
13         gap = gap/3;
14         for i in (gap, size) {
15             value = vet[i];
16             j = i - gap;
17             while (j >= 0 & value < vet[j]) {
18                 vet [j + gap] = vet[j];
19                 j = j - gap;
20             }
21             vet [j + gap] = value;
22         }
23     }
24 }
25
26 void main() {
27   int size;
28   int i;
29
30   read("%i", size);
31
32   int vet[size];
33
```

```
34      for i in (size) {
35        read("%i", vet[i]);
36      }
37
38      for i in (size - 1) {
39        print("%i, ", vet[i]);
40      }
41      print("%i\n", vet[size-1]);
42
43      shellSort(vet, size);
44
45      for i in (size - 1) {
46        print("%i, ", vet[i]);
47      }
48      print("%i\n", vet[size-1]);
49    }
```

2 Gramática

Escolhemos implementar um analisador descendente preditivo tabular.

2.1 Gramática não tratada

```
Programa = Cod
Cod = Funcao CodAux
CodAux = Funcao CodAux
CodAux = epsilon

Funcao = Tipo 'id' '[' ']' '(' LParam ')' '{' Bloco '}'
Funcao = Tipo 'id' '(' LParam ')' '{' Bloco '}'
Funcao = 'void' 'id' '(' LParam ')' '{' Bloco '}'

Tipo = 'boolean'
Tipo = 'int'
Tipo = 'float'
Tipo = 'char'

LParam = Tipo 'id' LParAux
LParam = epsilon
LParAux = '[' ']'
LParAux = '[' ']' ',' Tipo 'id' LParAux
LParAux = ',' Tipo 'id' LParAux
LParAux = epsilon

Bloco = Declaracao Bloco
Bloco = Print Bloco
Bloco = Read Bloco
Bloco = For Bloco
Bloco = While Bloco
Bloco = If Bloco
Retorno = 'return' ExpConcat ';'
Bloco = 'id' '[' ExpArit ']' '=' ExpConcat ';' Bloco
Bloco = 'id' '=' ExpConcat ';' Bloco
Bloco = 'id' '(' LArgs ')' ';' Bloco
```

```
Bloco = epsilon

Declaracao = Tipo 'id' ';'
Declaracao = Tipo 'id' '[' ExpConcat ']' ';'

For = 'for' 'id' 'in' '(' Range ')' '{' Bloco '}'
Range = ExpArit ',' ExpArit
Range = ExpArit

While = 'while' '(' ExpBool ')' '{' Bloco '}'

If = 'if' '(' ExpBool ')' '{' Bloco '}' 'else' '{' Bloco '}'
If = 'if' '(' ExpBool ')' '{' Bloco '}'

ExpConcat = ExpConcat '++' ExpBool
ExpConcat = ExpBool

LArgs = ExpConcat LArgsAux
LArgs = epsilon
LArgsAux = ',' ExpConcat LArgsAux
LArgsAux = epsilon

Print = 'print' '(' 'cte_cad_ch' PrintAux ')' ';'
PrintAux = ',' LArgs
PrintAux = epsilon

Read = 'read' '(' 'cte_cad_ch' LId ')' ';'

LId = ',' 'id' '[' ExpConcat ']' LId
LId = ',' 'id' LId
LId = epsilon

ExpBool = ExpBool '|' TermBool
ExpBool = TermBool

TermBool = TermBool '&' ExpRel
TermBool = ExpRel
```

```
ExpRel = '!' ExpRel
ExpRel = ExpRel 'opr' ExpArit
ExpRel = ExpArit

ExpArit = ExpArit 'opa_ad' TermArit
ExpArit = TermArit

TermArit = TermArit 'opa_mult' FatArit
TermArit = FatArit

FatArit = 'id' '(' LArgs ')'
FatArit = 'id' '[' ExpConcat ']'
FatArit = 'id'
FatArit = 'opa_nega' FatAritRes
FatArit = '(' ExpBool ')'
FatArit = 'cte_int'
FatArit = 'cte_float'
FatArit = 'cte_char'
FatArit = 'cte_cad_ch'
FatArit = 'cte_bool'
```

2.2 Gramática tratada

Gramática fatorada e sem recursão à esquerda.

```
d0: Programa = Cod
d1: Cod = Funcao CodAux
d2: CodAux = Funcao CodAux
d3: CodAux = epsilon

d4: Funcao = Tipo 'id' LParV '(' LParam ')' '{' Bloco '}'
d5: Funcao = 'void' 'id' '(' LParam ')' '{' Bloco '}'

d6: Tipo = 'boolean'
d7: Tipo = 'int'
d8: Tipo = 'float'
d9: Tipo = 'char'
```

```
d10: LParam = Tipo 'id' LParAux
d11: LParam = epsilon
d12: LParAux = LParV LParAuxRes
d13: LParAuxRes = ',' Tipo 'id' LParAux
d14: LParAuxRes = epsilon
d15: LParV = '[' ']'
d16: LParV = epsilon

d17: Bloco = Declaracao Bloco
d18: Bloco = Print Bloco
d19: Bloco = Read Bloco
d20: Bloco = For Bloco
d21: Bloco = While Bloco
d22: Bloco = If Bloco
d23: Bloco = Retorno Bloco
d24: Bloco = 'id' BlocoAux
d25: Bloco = epsilon

d26: Declaracao = Tipo 'id' DeclFim
d27: DeclFim = ';'
d28: DeclFim = '[' ExpConcat ']' ';'

d29: For = 'for' 'id' 'in' '(' Range ')' '{' Bloco '}'
d30: Range = ExpArit RangeAux
d31: RangeAux = ',' ExpArit
d32: RangeAux = epsilon

d33: While = 'while' '(' ExpBool ')' '{' Bloco '}'

d34: If = 'if' '(' ExpBool ')' '{' Bloco '}' Else
d35: Else = 'else' '{' Bloco '}'
d36: Else = epsilon

d37: Retorno = 'return' ExpConcat ';'

d38: BlocoAux = '[' ExpArit ']' Atribuicao Bloco
```

```
d39: BlocoAux = Atribuicao Bloco
d40: BlocoAux = CompletaFc ';' Bloco
d41: CompletaFc = '(' LArgs ')'

d42: Atribuicao = '=' ExpConcat ';'

d43: ExpConcat = ExpBool TermConcat
d44: TermConcat = '++' ExpBool TermConcat
d45: TermConcat = epsilon

d46: LArgs = ExpConcat LArgsAux
d47: LArgs = epsilon
d48: LArgsAux = ',' ExpConcat LArgsAux
d49: LArgsAux = epsilon

d50: Print = 'print' '(' 'cte_cad_ch' PrintAux ')' ';'
d51: PrintAux = ',' LArgs
d52: PrintAux = epsilon

d53: Read = 'read' '(' 'cte_cad_ch' LId ')' ';'
d54: LId = ',' 'id' LIdAux
d55: LIdAux = '[' ExpConcat ']' LId
d56: LIdAux = LId
d57: LId = epsilon

d58: ExpBool = TermBool FatBool
d59: FatBool = '|' TermBool FatBool
d60: FatBool = epsilon
d61: TermBool = ExpRel TermBoRes
d62: TermBoRes = '&' ExpRel TermBoRes
d63: TermBoRes = epsilon

d64: ExpRel = '!' ExpRel
d65: ExpRel = ExpArit TermRel
d66: TermRel = 'opr' ExpArit TermRel
d67: TermRel = epsilon
```

```
d68: ExpArit = TermArit FatArit
d69: FatArit = 'opa_ad' TermArit FatArit
d70: FatArit = epsilon

d71: TermArit = FatAritRes TermAriRes
d72: TermAriRes = 'opa_mult' FatAritRes TermAriRes
d73: TermAriRes = epsilon

d74: FatAritRes = 'id' CFuncAux
d75: CFuncAux = CompletaFc
d76: CFuncAux = '[' ExpConcat ']'
d77: CFuncAux = epsilon
d78: FatAritRes = 'opa_nega' FatAritRes
d79: FatAritRes = '(' ExpBool ')'
d80: FatAritRes = 'cte_int'
d81: FatAritRes = 'cte_float'
d82: FatAritRes = 'cte_char'
d83: FatAritRes = 'cte_cad_ch'
d84: FatAritRes = 'cte_bool'
```

3 Tabela de análise

Pela extensão da tabela, tivemos que dividí-la.

	'id'	'('	')'	'{'	'}'	'void'	'boolean'	'int'	'float'	'char'	','	'['
Programa						d0	d0	d0	d0	d0		
Cod						d1	d1	d1	d1	d1		
CodAux						d2	d2	d2	d2	d2		
Funcao						d5	d4	d4	d4	d4		
Tipo							d6	d7	d8	d9		
LParam			epsilon				d10	d10	d10	d10		
LParAux			d12								d12	d12
LParAuxRes			epsilon								d13	
LParV		epsilon	epsilon								epsilon	d15
Bloco	d24				epsilon		d17	d17	d17	d17		
Declaracao							d26	d26	d26	d26		
DeclFim												d28
For												
Range	d30	d30										
RangeAux			epsilon								d31	
While												
If												
Else	epsilon				epsilon		epsilon	epsilon	epsilon	epsilon		
Retorno												
BlocoAux		d40										d38
CompletaFc		d41										
Atribuicao												
ExpConcat	d43	d43										
TermConcat			epsilon								epsilon	

	$	';'	'for'	'in'	'while'	'if'	'else'	'return'	'='	'++'	'print'	'cte_cad_ch'
Programa												
Cod												
CodAux												
Funcao												
Tipo												
LParam												
LParAux												
LParAuxRes												
LParV												
Bloco			d20		d21	d22		d23			d18	
Declaracao												
DeclFim		d27										
For			d29									
Range												d30
RangeAux												
While					d33							
If						d34						
Else			epsilon		epsilon	epsilon	d35	epsilon			epsilon	
Retorno								d37				
BlocoAux									d39			
CompletaFc												
Atribuicao									d42			
ExpConcat												d43
TermConcat	epsilon	epsilon								d44		

	'read'	'|'	'&'	'!'	'opr'	'opa_ad'	'opa_mult'	'opa_nega'	'cte_int'	'cte_float'	'cte_char'	'cte_bool'	EOF
Programa													
Cod													
CodAux													epsilon
Funcao													
Tipo													
LParam													
LParAux													
LParAuxRes													
LParV													
Bloco	d19												
Declaracao													
DeclFim													
For													
Range								d30	d30	d30	d30	d30	
RangeAux													
While													
If													
Else	epsilon												
Retorno													
BlocoAux													
CompletaFc													
Atribuicao													
ExpConcat				d43				d43	d43	d43	d43	d43	
TermConcat													

	'id'	'('	')'	'{'	'}'	'void'	'boolean'	'int'	'float'	'char'	','	'['
LArgs	d46	d46	epsilon									
LArgsAux			epsilon								d48	
Print												
PrintAux			epsilon								d51	
Read												
LId			epsilon								d54	
LIdAux			d56								d56	d55
ExpBool	d58	d58										
FatBool			epsilon								epsilon	
TermBool	d61	d61										
TermBoRes			epsilon								epsilon	
ExpRel	d65	d65										
TermRel			epsilon								epsilon	
ExpArit	d68	d68										
FatArit			epsilon								epsilon	
TermArit	d71	d71										
TermAriRes			epsilon								epsilon	
FatAritRes	d74	d79										
CFuncAux		d75	epsilon								epsilon	d76

	']'	';'	'for'	'in'	'while'	'if'	'else'	'return'	'='	'++'	'print'	'cte_cad_ch'
LArgs												d46
LArgsAux												
Print											d50	
PrintAux												
Read												
LId												
LIdAux												
ExpBool												d58
FatBool	epsilon	epsilon								epsilon		
TermBool												d61
TermBoRes	epsilon	epsilon								epsilon		
ExpRel												d65
TermRel	epsilon	epsilon								epsilon		
ExpArit												d68
FatArit	epsilon	epsilon								epsilon		
TermArit												d71
TermAriRes	epsilon	epsilon								epsilon		
FatAritRes												d83
CFuncAux	epsilon	epsilon								epsilon		

	'read'	'|'	'&'	'!'	'opr'	'opa_ad'	'opa_mult'	'opa_nega'	'cte_int'	'cte_float'	'cte_char'	'cte_bool'	EOF
LArgs				d46				d46	d46	d46	d46	d46	
LArgsAux													
Print													
PrintAux													
Read	d53												
Lld													
LldAux													
ExpBool				d58				d58	d58	d58	d58	d58	
FatBool		d59											
TermBool				d61				d61	d61	d61	d61	d61	
TermBoRes		epsilon	d62										
ExpRel				d64				d65	d65	d65	d65	d65	
TermRel		epsilon	epsilon		d66								
ExpArit								d68	d68	d68	d68	d68	
FatArit		epsilon	epsilon		epsilon	d69							
TermArit								d71	d71	d71	d71	d71	
TermAriRes		epsilon	epsilon		epsilon	epsilon	d72						
FatAritRes								d78	d80	d81	d83	d84	
CFuncAux		epsilon	epsilon		epsilon	epsilon	epsilon						

4 Saída dos programas exemplo

4.1 Hello World

```
1  void main() {
       Programa = Cod
       Cod = Funcao CodAux
       Funcao = 'tipo_void' 'id' 'ab_parente' LParam 'fe_parente'
           'ab_chave' Bloco 'fe_chave'
           [001, 001] (0021, TIPO_VOID) {void}
           [001, 006] (0000,       ID) {main}
           [001, 010] (0003, AB_PARENTE) {(}
       LParam = epsilon
           [001, 011] (0004, FE_PARENTE) {)}
           [001, 013] (0005, AB_CHAVE) {{}
2  print("Hello World!");
       Bloco = Print Bloco
       Print = 'print' 'ab_parente' 'cte_cad_ch' PrintAux 'fe_parente'
           'pont_virg'
           [002, 007] (0035,    PRINT) {print}
           [002, 012] (0003, AB_PARENTE) {(}
           [002, 013] (0009, CTE_CAD_CH) {Hello World!}
       PrintAux = epsilon
           [002, 027] (0004, FE_PARENTE) {)}
           [002, 028] (0001, PONT_VIRG) {;}
3  }
       Bloco = epsilon
           [003, 001] (0006, FE_CHAVE) {}}

Sintaxe correta! Análise encerrada!
```

4.2 Fibonacci

```
1  void fib(int limite, int seq[]) {
       Programa = Cod
       Cod = Funcao CodAux
       Funcao = 'tipo_void' 'id' 'ab_parente' LParam 'fe_parente'
           'ab_chave' Bloco 'fe_chave'
```

```
            [001, 001] (0021, TIPO_VOID) {void}
            [001, 006] (0000,       ID) {fib}
            [001, 009] (0003, AB_PARENTE) {(}
        LParam = Tipo 'id' LParAux
        Tipo = 'tipo_int'
            [001, 010] (0017, TIPO_INT) {int}
            [001, 014] (0000,       ID) {limite}
        LParAux = LParV LParAuxRes
        LParV = epsilon
        LParAuxRes = 'virgula' Tipo 'id' LParAux
            [001, 020] (0002,  VIRGULA) {,}
        Tipo = 'tipo_int'
            [001, 022] (0017, TIPO_INT) {int}
            [001, 026] (0000,       ID) {seq}
        LParAux = LParV LParAuxRes
        LParV = 'ab_colchet' 'fe_colchet'
            [001, 029] (0015, AB_COLCHET) {[}
            [001, 030] (0016, FE_COLCHET) {]}
        LParAuxRes = epsilon
            [001, 031] (0004, FE_PARENTE) {)}
            [001, 033] (0005, AB_CHAVE) {{}
2   int aux;
        Bloco = Declaracao Bloco
        Declaracao = Tipo 'id' DeclFim
        Tipo = 'tipo_int'
            [002, 003] (0017, TIPO_INT) {int}
            [002, 007] (0000,       ID) {aux}
        DeclFim = 'pont_virg'
            [002, 010] (0001, PONT_VIRG) {;}
3   aux = 0;
        Bloco = 'id' BlocoAux
            [003, 003] (0000,       ID) {aux}
        BlocoAux = Atribuicao Bloco
        Atribuicao = 'atribuicao' ExpConcat 'pont_virg'
            [003, 007] (0008, ATRIBUICAO) {=}
        ExpConcat = ExpBool TermConcat
        ExpBool = TermBool FatBool
```

```
        TermBool = ExpRel TermBoRes
        ExpRel = ExpArit TermRel
        ExpArit = TermArit FatArit
        TermArit = FatAritRes TermAriRes
        FatAritRes = 'cte_int'
            [003, 009] (0012,  CTE_INT) {0}
        TermAriRes = epsilon
        FatArit = epsilon
        TermRel = epsilon
        TermBoRes = epsilon
        FatBool = epsilon
        TermConcat = epsilon
            [003, 010] (0001, PONT_VIRG) {;}
5   while(aux < limite) {
        Bloco = While Bloco
        While = 'while' 'ab_parente' ExpBool 'fe_parente' 'ab_chave'
            Bloco 'fe_chave'
            [005, 003] (0033,    WHILE) {while}
            [005, 008] (0003, AB_PARENTE) {(}
        ExpBool = TermBool FatBool
        TermBool = ExpRel TermBoRes
        ExpRel = ExpArit TermRel
        ExpArit = TermArit FatArit
        TermArit = FatAritRes TermAriRes
        FatAritRes = 'id' CFuncAux
            [005, 009] (0000,       ID) {aux}
        CFuncAux = epsilon
        TermAriRes = epsilon
        FatArit = epsilon
        TermRel = 'opr' ExpArit TermRel
            [005, 013] (0007,      OPR) {<}
        ExpArit = TermArit FatArit
        TermArit = FatAritRes TermAriRes
        FatAritRes = 'id' CFuncAux
            [005, 015] (0000,       ID) {limite}
        CFuncAux = epsilon
        TermAriRes = epsilon
```

```
        FatArit = epsilon
        TermRel = epsilon
        TermBoRes = epsilon
        FatBool = epsilon
            [005, 021] (0004, FE_PARENTE) {)}
            [005, 023] (0005, AB_CHAVE) {{}
6   if(aux == 0 | aux == 1) {
        Bloco = If Bloco
        If = 'if' 'ab_parente' ExpBool 'fe_parente' 'ab_chave' Bloco
            'fe_chave' Else
            [006, 005] (0029,       IF) {if}
            [006, 007] (0003, AB_PARENTE) {(}
        ExpBool = TermBool FatBool
        TermBool = ExpRel TermBoRes
        ExpRel = ExpArit TermRel
        ExpArit = TermArit FatArit
        TermArit = FatAritRes TermAriRes
        FatAritRes = 'id' CFuncAux
            [006, 008] (0000,       ID) {aux}
        CFuncAux = epsilon
        TermAriRes = epsilon
        FatArit = epsilon
        TermRel = 'opr' ExpArit TermRel
            [006, 012] (0007,      OPR) {==}
        ExpArit = TermArit FatArit
        TermArit = FatAritRes TermAriRes
        FatAritRes = 'cte_int'
            [006, 015] (0012,  CTE_INT) {0}
        TermAriRes = epsilon
        FatArit = 'opa_ad' TermArit FatArit
            [006, 017] (0023,   OPA_AD) {|}
        TermArit = FatAritRes TermAriRes
        FatAritRes = 'id' CFuncAux
            [006, 019] (0000,       ID) {aux}
        CFuncAux = epsilon
        TermAriRes = epsilon
        FatArit = epsilon
```

```
        TermRel = 'opr' ExpArit TermRel
            [006, 023] (0007,      OPR) {==}
        ExpArit = TermArit FatArit
        TermArit = FatAritRes TermAriRes
        FatAritRes = 'cte_int'
            [006, 026] (0012,  CTE_INT) {1}
        TermAriRes = epsilon
        FatArit = epsilon
        TermRel = epsilon
        TermBoRes = epsilon
        FatBool = epsilon
            [006, 027] (0004, FE_PARENTE) {)}
            [006, 029] (0005, AB_CHAVE) {{}
7   seq[aux] = aux;
        Bloco = 'id' BlocoAux
            [007, 007] (0000,       ID) {seq}
        BlocoAux = 'ab_colchet' ExpArit 'fe_colchet' Atribuicao Bloco
            [007, 010] (0015, AB_COLCHET) {[}
        ExpArit = TermArit FatArit
        TermArit = FatAritRes TermAriRes
        FatAritRes = 'id' CFuncAux
            [007, 011] (0000,       ID) {aux}
        CFuncAux = epsilon
        TermAriRes = epsilon
        FatArit = epsilon
            [007, 014] (0016, FE_COLCHET) {]}
        Atribuicao = 'atribuicao' ExpConcat 'pont_virg'
            [007, 016] (0008, ATRIBUICAO) {=}
        ExpConcat = ExpBool TermConcat
        ExpBool = TermBool FatBool
        TermBool = ExpRel TermBoRes
        ExpRel = ExpArit TermRel
        ExpArit = TermArit FatArit
        TermArit = FatAritRes TermAriRes
        FatAritRes = 'id' CFuncAux
            [007, 018] (0000,       ID) {aux}
        CFuncAux = epsilon
```

```
        TermAriRes = epsilon
        FatArit = epsilon
        TermRel = epsilon
        TermBoRes = epsilon
        FatBool = epsilon
        TermConcat = epsilon
            [007, 021] (0001, PONT_VIRG) {;}
8   } else {
        Bloco = epsilon
            [008, 005] (0006, FE_CHAVE) {}}
        Else = 'else' 'ab_chave' Bloco 'fe_chave'
            [008, 007] (0030,     ELSE) {else}
            [008, 012] (0005, AB_CHAVE) {{}
9   seq[aux] = seq[aux - 1] + seq[aux - 2];
        Bloco = 'id' BlocoAux
            [009, 007] (0000,       ID) {seq}
        BlocoAux = 'ab_colchet' ExpArit 'fe_colchet' Atribuicao Bloco
            [009, 010] (0015, AB_COLCHET) {[}
        ExpArit = TermArit FatArit
        TermArit = FatAritRes TermAriRes
        FatAritRes = 'id' CFuncAux
            [009, 011] (0000,       ID) {aux}
        CFuncAux = epsilon
        TermAriRes = epsilon
        FatArit = epsilon
            [009, 014] (0016, FE_COLCHET) {]}
        Atribuicao = 'atribuicao' ExpConcat 'pont_virg'
            [009, 016] (0008, ATRIBUICAO) {=}
        ExpConcat = ExpBool TermConcat
        ExpBool = TermBool FatBool
        TermBool = ExpRel TermBoRes
        ExpRel = ExpArit TermRel
        ExpArit = TermArit FatArit
        TermArit = FatAritRes TermAriRes
        FatAritRes = 'id' CFuncAux
            [009, 018] (0000,       ID) {seq}
        CFuncAux = 'ab_colchet' ExpConcat 'fe_colchet'
```

```
    [009, 021] (0015, AB_COLCHET) {[}
ExpConcat = ExpBool TermConcat
ExpBool = TermBool FatBool
TermBool = ExpRel TermBoRes
ExpRel = ExpArit TermRel
ExpArit = TermArit FatArit
TermArit = FatAritRes TermAriRes
FatAritRes = 'id' CFuncAux
    [009, 022] (0000,       ID) {aux}
CFuncAux = epsilon
TermAriRes = epsilon
FatArit = 'opa_ad' TermArit FatArit
    [009, 026] (0023,   OPA_AD) {-}
TermArit = FatAritRes TermAriRes
FatAritRes = 'cte_int'
    [009, 028] (0012,  CTE_INT) {1}
TermAriRes = epsilon
FatArit = epsilon
TermRel = epsilon
TermBoRes = epsilon
FatBool = epsilon
TermConcat = epsilon
    [009, 029] (0016, FE_COLCHET) {]}
TermAriRes = epsilon
FatArit = 'opa_ad' TermArit FatArit
    [009, 031] (0023,   OPA_AD) {+}
TermArit = FatAritRes TermAriRes
FatAritRes = 'id' CFuncAux
    [009, 033] (0000,       ID) {seq}
CFuncAux = 'ab_colchet' ExpConcat 'fe_colchet'
    [009, 036] (0015, AB_COLCHET) {[}
ExpConcat = ExpBool TermConcat
ExpBool = TermBool FatBool
TermBool = ExpRel TermBoRes
ExpRel = ExpArit TermRel
ExpArit = TermArit FatArit
TermArit = FatAritRes TermAriRes
```

```
        FatAritRes = 'id' CFuncAux
            [009, 037] (0000,       ID) {aux}
        CFuncAux = epsilon
        TermAriRes = epsilon
        FatArit = 'opa_ad' TermArit FatArit
            [009, 041] (0023,   OPA_AD) {-}
        TermArit = FatAritRes TermAriRes
        FatAritRes = 'cte_int'
            [009, 043] (0012,  CTE_INT) {2}
        TermAriRes = epsilon
        FatArit = epsilon
        TermRel = epsilon
        TermBoRes = epsilon
        FatBool = epsilon
        TermConcat = epsilon
            [009, 044] (0016, FE_COLCHET) {]}
        TermAriRes = epsilon
        FatArit = epsilon
        TermRel = epsilon
        TermBoRes = epsilon
        FatBool = epsilon
        TermConcat = epsilon
            [009, 045] (0001, PONT_VIRG) {;}
10  }
        Bloco = epsilon
            [010, 005] (0006, FE_CHAVE) {}}
11  aux = aux + 1;
        Bloco = 'id' BlocoAux
            [011, 005] (0000,       ID) {aux}
        BlocoAux = Atribuicao Bloco
        Atribuicao = 'atribuicao' ExpConcat 'pont_virg'
            [011, 009] (0008, ATRIBUICAO) {=}
        ExpConcat = ExpBool TermConcat
        ExpBool = TermBool FatBool
        TermBool = ExpRel TermBoRes
        ExpRel = ExpArit TermRel
        ExpArit = TermArit FatArit
```

```
        TermArit = FatAritRes TermAriRes
        FatAritRes = 'id' CFuncAux
            [011, 011] (0000,       ID) {aux}
        CFuncAux = epsilon
        TermAriRes = epsilon
        FatArit = 'opa_ad' TermArit FatArit
            [011, 015] (0023,   OPA_AD) {+}
        TermArit = FatAritRes TermAriRes
        FatAritRes = 'cte_int'
            [011, 017] (0012,  CTE_INT) {1}
        TermAriRes = epsilon
        FatArit = epsilon
        TermRel = epsilon
        TermBoRes = epsilon
        FatBool = epsilon
        TermConcat = epsilon
            [011, 018] (0001, PONT_VIRG) {;}
12  }
        Bloco = epsilon
            [012, 003] (0006, FE_CHAVE) {}}
13  }
        Bloco = epsilon
            [013, 001] (0006, FE_CHAVE) {}}
15  void main() {
        CodAux = Funcao CodAux
        Funcao = 'tipo_void' 'id' 'ab_parente' LParam 'fe_parente'
            'ab_chave' Bloco 'fe_chave'
            [015, 001] (0021, TIPO_VOID) {void}
            [015, 006] (0000,       ID) {main}
            [015, 010] (0003, AB_PARENTE) {(}
        LParam = epsilon
            [015, 011] (0004, FE_PARENTE) {)}
            [015, 013] (0005, AB_CHAVE) {{}
16  int limite;
        Bloco = Declaracao Bloco
        Declaracao = Tipo 'id' DeclFim
        Tipo = 'tipo_int'
```

```
            [016, 003] (0017, TIPO_INT) {int}
            [016, 007] (0000,       ID) {limite}
        DeclFim = 'pont_virg'
            [016, 013] (0001, PONT_VIRG) {;}
17  int aux;
        Bloco = Declaracao Bloco
        Declaracao = Tipo 'id' DeclFim
        Tipo = 'tipo_int'
            [017, 003] (0017, TIPO_INT) {int}
            [017, 007] (0000,       ID) {aux}
        DeclFim = 'pont_virg'
            [017, 010] (0001, PONT_VIRG) {;}
18  aux = 0;
        Bloco = 'id' BlocoAux
            [018, 003] (0000,       ID) {aux}
        BlocoAux = Atribuicao Bloco
        Atribuicao = 'atribuicao' ExpConcat 'pont_virg'
            [018, 007] (0008, ATRIBUICAO) {=}
        ExpConcat = ExpBool TermConcat
        ExpBool = TermBool FatBool
        TermBool = ExpRel TermBoRes
        ExpRel = ExpArit TermRel
        ExpArit = TermArit FatArit
        TermArit = FatAritRes TermAriRes
        FatAritRes = 'cte_int'
            [018, 009] (0012,  CTE_INT) {0}
        TermAriRes = epsilon
        FatArit = epsilon
        TermRel = epsilon
        TermBoRes = epsilon
        FatBool = epsilon
        TermConcat = epsilon
            [018, 010] (0001, PONT_VIRG) {;}
20  read("%i", limite);
        Bloco = Read Bloco
        Read = 'read' 'ab_parente' 'cte_cad_ch' LId 'fe_parente'
            'pont_virg'
```

```
            [020, 003] (0034,     READ) {read}
            [020, 007] (0003, AB_PARENTE) {(}
            [020, 008] (0009, CTE_CAD_CH) {%i}
        LId = 'virgula' 'id' LIdAux
            [020, 012] (0002,  VIRGULA) {,}
            [020, 014] (0000,       ID) {limite}
        LIdAux = LId
        LId = epsilon
            [020, 020] (0004, FE_PARENTE) {)}
            [020, 021] (0001, PONT_VIRG) {;}
21  int sequencia[limite + 1];
        Bloco = Declaracao Bloco
        Declaracao = Tipo 'id' DeclFim
        Tipo = 'tipo_int'
            [021, 003] (0017, TIPO_INT) {int}
            [021, 007] (0000,       ID) {sequencia}
        DeclFim = 'ab_colchet' ExpConcat 'fe_colchet' 'pont_virg'
            [021, 016] (0015, AB_COLCHET) {[}
        ExpConcat = ExpBool TermConcat
        ExpBool = TermBool FatBool
        TermBool = ExpRel TermBoRes
        ExpRel = ExpArit TermRel
        ExpArit = TermArit FatArit
        TermArit = FatAritRes TermAriRes
        FatAritRes = 'id' CFuncAux
            [021, 017] (0000,       ID) {limite}
        CFuncAux = epsilon
        TermAriRes = epsilon
        FatArit = 'opa_ad' TermArit FatArit
            [021, 024] (0023,   OPA_AD) {+}
        TermArit = FatAritRes TermAriRes
        FatAritRes = 'cte_int'
            [021, 026] (0012,  CTE_INT) {1}
        TermAriRes = epsilon
        FatArit = epsilon
        TermRel = epsilon
        TermBoRes = epsilon
```

```
        FatBool = epsilon
        TermConcat = epsilon
            [021, 027] (0016, FE_COLCHET) {]}
            [021, 028] (0001, PONT_VIRG) {;}
23  fib(limite, sequencia);
        Bloco = 'id' BlocoAux
            [023, 003] (0000,       ID) {fib}
        BlocoAux = CompletaFc 'pont_virg' Bloco
        CompletaFc = 'ab_parente' LArgs 'fe_parente'
            [023, 006] (0003, AB_PARENTE) {(}
        LArgs = ExpConcat LArgsAux
        ExpConcat = ExpBool TermConcat
        ExpBool = TermBool FatBool
        TermBool = ExpRel TermBoRes
        ExpRel = ExpArit TermRel
        ExpArit = TermArit FatArit
        TermArit = FatAritRes TermAriRes
        FatAritRes = 'id' CFuncAux
            [023, 007] (0000,       ID) {limite}
        CFuncAux = epsilon
        TermAriRes = epsilon
        FatArit = epsilon
        TermRel = epsilon
        TermBoRes = epsilon
        FatBool = epsilon
        TermConcat = epsilon
        LArgsAux = 'virgula' ExpConcat LArgsAux
            [023, 013] (0002,  VIRGULA) {,}
        ExpConcat = ExpBool TermConcat
        ExpBool = TermBool FatBool
        TermBool = ExpRel TermBoRes
        ExpRel = ExpArit TermRel
        ExpArit = TermArit FatArit
        TermArit = FatAritRes TermAriRes
        FatAritRes = 'id' CFuncAux
            [023, 015] (0000,       ID) {sequencia}
        CFuncAux = epsilon
```

```
        TermAriRes = epsilon
        FatArit = epsilon
        TermRel = epsilon
        TermBoRes = epsilon
        FatBool = epsilon
        TermConcat = epsilon
        LArgsAux = epsilon
            [023, 024] (0004, FE_PARENTE) {)}
            [023, 025] (0001, PONT_VIRG) {;}
25  while(aux < limite - 1) {
        Bloco = While Bloco
        While = 'while' 'ab_parente' ExpBool 'fe_parente' 'ab_chave'
            Bloco 'fe_chave'
            [025, 003] (0033,    WHILE) {while}
            [025, 008] (0003, AB_PARENTE) {(}
        ExpBool = TermBool FatBool
        TermBool = ExpRel TermBoRes
        ExpRel = ExpArit TermRel
        ExpArit = TermArit FatArit
        TermArit = FatAritRes TermAriRes
        FatAritRes = 'id' CFuncAux
            [025, 009] (0000,       ID) {aux}
        CFuncAux = epsilon
        TermAriRes = epsilon
        FatArit = epsilon
        TermRel = 'opr' ExpArit TermRel
            [025, 013] (0007,      OPR) {<}
        ExpArit = TermArit FatArit
        TermArit = FatAritRes TermAriRes
        FatAritRes = 'id' CFuncAux
            [025, 015] (0000,       ID) {limite}
        CFuncAux = epsilon
        TermAriRes = epsilon
        FatArit = 'opa_ad' TermArit FatArit
            [025, 022] (0023,   OPA_AD) {-}
        TermArit = FatAritRes TermAriRes
        FatAritRes = 'cte_int'
```

```
            [025, 024] (0012,  CTE_INT) {1}
        TermAriRes = epsilon
        FatArit = epsilon
        TermRel = epsilon
        TermBoRes = epsilon
        FatBool = epsilon
            [025, 025] (0004, FE_PARENTE) {)}
            [025, 027] (0005, AB_CHAVE) {{}
26  print("%i, ", sequencia[aux]);
        Bloco = Print Bloco
        Print = 'print' 'ab_parente' 'cte_cad_ch' PrintAux 'fe_parente'
            'pont_virg'
            [026, 005] (0035,    PRINT) {print}
            [026, 010] (0003, AB_PARENTE) {(}
            [026, 011] (0009, CTE_CAD_CH) {%i, }
        PrintAux = 'virgula' LArgs
            [026, 017] (0002,  VIRGULA) {,}
        LArgs = ExpConcat LArgsAux
        ExpConcat = ExpBool TermConcat
        ExpBool = TermBool FatBool
        TermBool = ExpRel TermBoRes
        ExpRel = ExpArit TermRel
        ExpArit = TermArit FatArit
        TermArit = FatAritRes TermAriRes
        FatAritRes = 'id' CFuncAux
            [026, 019] (0000,       ID) {sequencia}
        CFuncAux = 'ab_colchet' ExpConcat 'fe_colchet'
            [026, 028] (0015, AB_COLCHET) {[}
        ExpConcat = ExpBool TermConcat
        ExpBool = TermBool FatBool
        TermBool = ExpRel TermBoRes
        ExpRel = ExpArit TermRel
        ExpArit = TermArit FatArit
        TermArit = FatAritRes TermAriRes
        FatAritRes = 'id' CFuncAux
            [026, 029] (0000,       ID) {aux}
        CFuncAux = epsilon
```

```
        TermAriRes = epsilon
        FatArit = epsilon
        TermRel = epsilon
        TermBoRes = epsilon
        FatBool = epsilon
        TermConcat = epsilon
            [026, 032] (0016, FE_COLCHET) {]}
        TermAriRes = epsilon
        FatArit = epsilon
        TermRel = epsilon
        TermBoRes = epsilon
        FatBool = epsilon
        TermConcat = epsilon
        LArgsAux = epsilon
            [026, 033] (0004, FE_PARENTE) {)}
            [026, 034] (0001, PONT_VIRG) {;}
27  aux = aux + 1;
        Bloco = 'id' BlocoAux
            [027, 005] (0000,       ID) {aux}
        BlocoAux = Atribuicao Bloco
        Atribuicao = 'atribuicao' ExpConcat 'pont_virg'
            [027, 009] (0008, ATRIBUICAO) {=}
        ExpConcat = ExpBool TermConcat
        ExpBool = TermBool FatBool
        TermBool = ExpRel TermBoRes
        ExpRel = ExpArit TermRel
        ExpArit = TermArit FatArit
        TermArit = FatAritRes TermAriRes
        FatAritRes = 'id' CFuncAux
            [027, 011] (0000,       ID) {aux}
        CFuncAux = epsilon
        TermAriRes = epsilon
        FatArit = 'opa_ad' TermArit FatArit
            [027, 015] (0023,   OPA_AD) {+}
        TermArit = FatAritRes TermAriRes
        FatAritRes = 'cte_int'
            [027, 017] (0012,  CTE_INT) {1}
```

```
        TermAriRes = epsilon
        FatArit = epsilon
        TermRel = epsilon
        TermBoRes = epsilon
        FatBool = epsilon
        TermConcat = epsilon
            [027, 018] (0001, PONT_VIRG) {;}
28  }
        Bloco = epsilon
            [028, 003] (0006, FE_CHAVE) {}}
29  print("%i\n", sequencia[limite - 1]);
        Bloco = Print Bloco
        Print = 'print' 'ab_parente' 'cte_cad_ch' PrintAux 'fe_parente'
            'pont_virg'
            [029, 003] (0035,    PRINT) {print}
            [029, 008] (0003, AB_PARENTE) {(}
            [029, 009] (0009, CTE_CAD_CH) {%i\n}
        PrintAux = 'virgula' LArgs
            [029, 015] (0002,  VIRGULA) {,}
        LArgs = ExpConcat LArgsAux
        ExpConcat = ExpBool TermConcat
        ExpBool = TermBool FatBool
        TermBool = ExpRel TermBoRes
        ExpRel = ExpArit TermRel
        ExpArit = TermArit FatArit
        TermArit = FatAritRes TermAriRes
        FatAritRes = 'id' CFuncAux
            [029, 017] (0000,       ID) {sequencia}
        CFuncAux = 'ab_colchet' ExpConcat 'fe_colchet'
            [029, 026] (0015, AB_COLCHET) {[}
        ExpConcat = ExpBool TermConcat
        ExpBool = TermBool FatBool
        TermBool = ExpRel TermBoRes
        ExpRel = ExpArit TermRel
        ExpArit = TermArit FatArit
        TermArit = FatAritRes TermAriRes
        FatAritRes = 'id' CFuncAux
```

```
            [029, 027] (0000,       ID) {limite}
        CFuncAux = epsilon
        TermAriRes = epsilon
        FatArit = 'opa_ad' TermArit FatArit
            [029, 034] (0023,   OPA_AD) {-}
        TermArit = FatAritRes TermAriRes
        FatAritRes = 'cte_int'
            [029, 036] (0012,  CTE_INT) {1}
        TermAriRes = epsilon
        FatArit = epsilon
        TermRel = epsilon
        TermBoRes = epsilon
        FatBool = epsilon
        TermConcat = epsilon
            [029, 037] (0016, FE_COLCHET) {]}
        TermAriRes = epsilon
        FatArit = epsilon
        TermRel = epsilon
        TermBoRes = epsilon
        FatBool = epsilon
        TermConcat = epsilon
        LArgsAux = epsilon
            [029, 038] (0004, FE_PARENTE) {)}
            [029, 039] (0001, PONT_VIRG) {;}
30  }
        Bloco = epsilon
            [030, 001] (0006, FE_CHAVE) {}}

Sintaxe correta! Análise encerrada!
```

4.3 Shell Sort

```
1  void shellSort(int vet[], int size) {
       Programa = Cod
       Cod = Funcao CodAux
       Funcao = 'tipo_void' 'id' 'ab_parente' LParam 'fe_parente'
           'ab_chave' Bloco 'fe_chave'
           [001, 002] (0021, TIPO_VOID) {void}
```

```
            [001, 007] (0000,       ID) {shellSort}
            [001, 016] (0003, AB_PARENTE) {(}
        LParam = Tipo 'id' LParAux
        Tipo = 'tipo_int'
            [001, 017] (0017, TIPO_INT) {int}
            [001, 021] (0000,       ID) {vet}
        LParAux = LParV LParAuxRes
        LParV = 'ab_colchet' 'fe_colchet'
            [001, 024] (0015, AB_COLCHET) {[}
            [001, 025] (0016, FE_COLCHET) {]}
        LParAuxRes = 'virgula' Tipo 'id' LParAux
            [001, 026] (0002,  VIRGULA) {,}
        Tipo = 'tipo_int'
            [001, 028] (0017, TIPO_INT) {int}
            [001, 032] (0000,       ID) {size}
        LParAux = LParV LParAuxRes
        LParV = epsilon
        LParAuxRes = epsilon
            [001, 036] (0004, FE_PARENTE) {)}
            [001, 038] (0005, AB_CHAVE) {{}
2   int i;
        Bloco = Declaracao Bloco
        Declaracao = Tipo 'id' DeclFim
        Tipo = 'tipo_int'
            [002, 007] (0017, TIPO_INT) {int}
            [002, 011] (0000,       ID) {i}
        DeclFim = 'pont_virg'
            [002, 012] (0001, PONT_VIRG) {;}
3   int j;
        Bloco = Declaracao Bloco
        Declaracao = Tipo 'id' DeclFim
        Tipo = 'tipo_int'
            [003, 007] (0017, TIPO_INT) {int}
            [003, 011] (0000,       ID) {j}
        DeclFim = 'pont_virg'
            [003, 012] (0001, PONT_VIRG) {;}
4   int value;
```

```
        Bloco = Declaracao Bloco
        Declaracao = Tipo 'id' DeclFim
        Tipo = 'tipo_int'
            [004, 007] (0017, TIPO_INT) {int}
            [004, 011] (0000,       ID) {value}
        DeclFim = 'pont_virg'
            [004, 016] (0001, PONT_VIRG) {;}
5   int gap;
        Bloco = Declaracao Bloco
        Declaracao = Tipo 'id' DeclFim
        Tipo = 'tipo_int'
            [005, 007] (0017, TIPO_INT) {int}
            [005, 011] (0000,       ID) {gap}
        DeclFim = 'pont_virg'
            [005, 014] (0001, PONT_VIRG) {;}
6   gap = 1;
        Bloco = 'id' BlocoAux
            [006, 007] (0000,       ID) {gap}
        BlocoAux = Atribuicao Bloco
        Atribuicao = 'atribuicao' ExpConcat 'pont_virg'
            [006, 011] (0008, ATRIBUICAO) {=}
        ExpConcat = ExpBool TermConcat
        ExpBool = TermBool FatBool
        TermBool = ExpRel TermBoRes
        ExpRel = ExpArit TermRel
        ExpArit = TermArit FatArit
        TermArit = FatAritRes TermAriRes
        FatAritRes = 'cte_int'
            [006, 013] (0012,  CTE_INT) {1}
        TermAriRes = epsilon
        FatArit = epsilon
        TermRel = epsilon
        TermBoRes = epsilon
        FatBool = epsilon
        TermConcat = epsilon
            [006, 014] (0001, PONT_VIRG) {;}
8   while(gap < size) {
```

```
        Bloco = While Bloco
        While = 'while' 'ab_parente' ExpBool 'fe_parente' 'ab_chave'
            Bloco 'fe_chave'
            [008, 007] (0033,    WHILE) {while}
            [008, 012] (0003, AB_PARENTE) {(}
        ExpBool = TermBool FatBool
        TermBool = ExpRel TermBoRes
        ExpRel = ExpArit TermRel
        ExpArit = TermArit FatArit
        TermArit = FatAritRes TermAriRes
        FatAritRes = 'id' CFuncAux
            [008, 013] (0000,      ID) {gap}
        CFuncAux = epsilon
        TermAriRes = epsilon
        FatArit = epsilon
        TermRel = 'opr' ExpArit TermRel
            [008, 017] (0007,     OPR) {<}
        ExpArit = TermArit FatArit
        TermArit = FatAritRes TermAriRes
        FatAritRes = 'id' CFuncAux
            [008, 019] (0000,      ID) {size}
        CFuncAux = epsilon
        TermAriRes = epsilon
        FatArit = epsilon
        TermRel = epsilon
        TermBoRes = epsilon
        FatBool = epsilon
            [008, 023] (0004, FE_PARENTE) {)}
            [008, 025] (0005, AB_CHAVE) {{}
9   gap = 3*gap+1;
        Bloco = 'id' BlocoAux
            [009, 011] (0000,      ID) {gap}
        BlocoAux = Atribuicao Bloco
        Atribuicao = 'atribuicao' ExpConcat 'pont_virg'
            [009, 015] (0008, ATRIBUICAO) {=}
        ExpConcat = ExpBool TermConcat
        ExpBool = TermBool FatBool
```

```
        TermBool = ExpRel TermBoRes
        ExpRel = ExpArit TermRel
        ExpArit = TermArit FatArit
        TermArit = FatAritRes TermAriRes
        FatAritRes = 'cte_int'
            [009, 017] (0012,  CTE_INT) {3}
        TermAriRes = 'opa_mult' FatAritRes TermAriRes
            [009, 018] (0024, OPA_MULT) {*}
        FatAritRes = 'id' CFuncAux
            [009, 019] (0000,       ID) {gap}
        CFuncAux = epsilon
        TermAriRes = epsilon
        FatArit = 'opa_ad' TermArit FatArit
            [009, 022] (0023,   OPA_AD) {+}
        TermArit = FatAritRes TermAriRes
        FatAritRes = 'cte_int'
            [009, 023] (0012,  CTE_INT) {1}
        TermAriRes = epsilon
        FatArit = epsilon
        TermRel = epsilon
        TermBoRes = epsilon
        FatBool = epsilon
        TermConcat = epsilon
            [009, 024] (0001, PONT_VIRG) {;}
10  }
        Bloco = epsilon
            [010, 007] (0006, FE_CHAVE) {}}
12  while(gap > 1) {
        Bloco = While Bloco
        While = 'while' 'ab_parente' ExpBool 'fe_parente' 'ab_chave'
            Bloco 'fe_chave'
            [012, 007] (0033,    WHILE) {while}
            [012, 012] (0003, AB_PARENTE) {(}
        ExpBool = TermBool FatBool
        TermBool = ExpRel TermBoRes
        ExpRel = ExpArit TermRel
        ExpArit = TermArit FatArit
```

```
        TermArit = FatAritRes TermAriRes
        FatAritRes = 'id' CFuncAux
            [012, 013] (0000,       ID) {gap}
        CFuncAux = epsilon
        TermAriRes = epsilon
        FatArit = epsilon
        TermRel = 'opr' ExpArit TermRel
            [012, 017] (0007,      OPR) {>}
        ExpArit = TermArit FatArit
        TermArit = FatAritRes TermAriRes
        FatAritRes = 'cte_int'
            [012, 019] (0012,  CTE_INT) {1}
        TermAriRes = epsilon
        FatArit = epsilon
        TermRel = epsilon
        TermBoRes = epsilon
        FatBool = epsilon
            [012, 020] (0004, FE_PARENTE) {)}
            [012, 022] (0005, AB_CHAVE) {{}
13  gap = gap/3;
        Bloco = 'id' BlocoAux
            [013, 011] (0000,       ID) {gap}
        BlocoAux = Atribuicao Bloco
        Atribuicao = 'atribuicao' ExpConcat 'pont_virg'
            [013, 015] (0008, ATRIBUICAO) {=}
        ExpConcat = ExpBool TermConcat
        ExpBool = TermBool FatBool
        TermBool = ExpRel TermBoRes
        ExpRel = ExpArit TermRel
        ExpArit = TermArit FatArit
        TermArit = FatAritRes TermAriRes
        FatAritRes = 'id' CFuncAux
            [013, 017] (0000,       ID) {gap}
        CFuncAux = epsilon
        TermAriRes = 'opa_mult' FatAritRes TermAriRes
            [013, 020] (0024, OPA_MULT) {/}
        FatAritRes = 'cte_int'
```

```
            [013, 021] (0012,  CTE_INT) {3}
        TermAriRes = epsilon
        FatArit = epsilon
        TermRel = epsilon
        TermBoRes = epsilon
        FatBool = epsilon
        TermConcat = epsilon
            [013, 022] (0001, PONT_VIRG) {;}
14  for i in (gap, size) {
        Bloco = For Bloco
        For = 'for' 'id' 'in' 'ab_parente' Range 'fe_parente'
            'ab_chave' Bloco 'fe_chave'
            [014, 011] (0032,      FOR) {for}
            [014, 015] (0000,       ID) {i}
            [014, 017] (0031,       IN) {in}
            [014, 020] (0003, AB_PARENTE) {(}
        Range = ExpArit RangeAux
        ExpArit = TermArit FatArit
        TermArit = FatAritRes TermAriRes
        FatAritRes = 'id' CFuncAux
            [014, 021] (0000,       ID) {gap}
        CFuncAux = epsilon
        TermAriRes = epsilon
        FatArit = epsilon
        RangeAux = 'virgula' ExpArit
            [014, 024] (0002,  VIRGULA) {,}
        ExpArit = TermArit FatArit
        TermArit = FatAritRes TermAriRes
        FatAritRes = 'id' CFuncAux
            [014, 026] (0000,       ID) {size}
        CFuncAux = epsilon
        TermAriRes = epsilon
        FatArit = epsilon
            [014, 030] (0004, FE_PARENTE) {)}
            [014, 032] (0005, AB_CHAVE) {{}
15  value = vet[i];
        Bloco = 'id' BlocoAux
```

```
    [015, 015] (0000,       ID) {value}
BlocoAux = Atribuicao Bloco
Atribuicao = 'atribuicao' ExpConcat 'pont_virg'
    [015, 021] (0008, ATRIBUICAO) {=}
ExpConcat = ExpBool TermConcat
ExpBool = TermBool FatBool
TermBool = ExpRel TermBoRes
ExpRel = ExpArit TermRel
ExpArit = TermArit FatArit
TermArit = FatAritRes TermAriRes
FatAritRes = 'id' CFuncAux
    [015, 023] (0000,       ID) {vet}
CFuncAux = 'ab_colchet' ExpConcat 'fe_colchet'
    [015, 026] (0015, AB_COLCHET) {[}
ExpConcat = ExpBool TermConcat
ExpBool = TermBool FatBool
TermBool = ExpRel TermBoRes
ExpRel = ExpArit TermRel
ExpArit = TermArit FatArit
TermArit = FatAritRes TermAriRes
FatAritRes = 'id' CFuncAux
    [015, 027] (0000,       ID) {i}
CFuncAux = epsilon
TermAriRes = epsilon
FatArit = epsilon
TermRel = epsilon
TermBoRes = epsilon
FatBool = epsilon
TermConcat = epsilon
    [015, 028] (0016, FE_COLCHET) {]}
TermAriRes = epsilon
FatArit = epsilon
TermRel = epsilon
TermBoRes = epsilon
FatBool = epsilon
TermConcat = epsilon
    [015, 029] (0001, PONT_VIRG) {;}
```

```
16  j = i - gap;
        Bloco = 'id' BlocoAux
            [016, 015] (0000,       ID) {j}
        BlocoAux = Atribuicao Bloco
        Atribuicao = 'atribuicao' ExpConcat 'pont_virg'
            [016, 017] (0008, ATRIBUICAO) {=}
        ExpConcat = ExpBool TermConcat
        ExpBool = TermBool FatBool
        TermBool = ExpRel TermBoRes
        ExpRel = ExpArit TermRel
        ExpArit = TermArit FatArit
        TermArit = FatAritRes TermAriRes
        FatAritRes = 'id' CFuncAux
            [016, 019] (0000,       ID) {i}
        CFuncAux = epsilon
        TermAriRes = epsilon
        FatArit = 'opa_ad' TermArit FatArit
            [016, 021] (0023,   OPA_AD) {-}
        TermArit = FatAritRes TermAriRes
        FatAritRes = 'id' CFuncAux
            [016, 023] (0000,       ID) {gap}
        CFuncAux = epsilon
        TermAriRes = epsilon
        FatArit = epsilon
        TermRel = epsilon
        TermBoRes = epsilon
        FatBool = epsilon
        TermConcat = epsilon
            [016, 026] (0001, PONT_VIRG) {;}
17  while (j >= 0 & value < vet[j]) {
        Bloco = While Bloco
        While = 'while' 'ab_parente' ExpBool 'fe_parente' 'ab_chave'
            Bloco 'fe_chave'
            [017, 015] (0033,    WHILE) {while}
            [017, 021] (0003, AB_PARENTE) {(}
        ExpBool = TermBool FatBool
        TermBool = ExpRel TermBoRes
```

```
ExpRel = ExpArit TermRel
ExpArit = TermArit FatArit
TermArit = FatAritRes TermAriRes
FatAritRes = 'id' CFuncAux
    [017, 022] (0000,       ID) {j}
CFuncAux = epsilon
TermAriRes = epsilon
FatArit = epsilon
TermRel = 'opr' ExpArit TermRel
    [017, 024] (0007,      OPR) {>=}
ExpArit = TermArit FatArit
TermArit = FatAritRes TermAriRes
FatAritRes = 'cte_int'
    [017, 027] (0012,  CTE_INT) {0}
TermAriRes = epsilon
FatArit = epsilon
TermRel = epsilon
TermBoRes = 'opl_e' ExpRel TermBoRes
    [017, 029] (0026,    OPL_E) {&}
ExpRel = ExpArit TermRel
ExpArit = TermArit FatArit
TermArit = FatAritRes TermAriRes
FatAritRes = 'id' CFuncAux
    [017, 031] (0000,       ID) {value}
CFuncAux = epsilon
TermAriRes = epsilon
FatArit = epsilon
TermRel = 'opr' ExpArit TermRel
    [017, 037] (0007,      OPR) {<}
ExpArit = TermArit FatArit
TermArit = FatAritRes TermAriRes
FatAritRes = 'id' CFuncAux
    [017, 039] (0000,       ID) {vet}
CFuncAux = 'ab_colchet' ExpConcat 'fe_colchet'
    [017, 042] (0015, AB_COLCHET) {[}
ExpConcat = ExpBool TermConcat
ExpBool = TermBool FatBool
```

```
        TermBool = ExpRel TermBoRes
        ExpRel = ExpArit TermRel
        ExpArit = TermArit FatArit
        TermArit = FatAritRes TermAriRes
        FatAritRes = 'id' CFuncAux
            [017, 043] (0000,       ID) {j}
        CFuncAux = epsilon
        TermAriRes = epsilon
        FatArit = epsilon
        TermRel = epsilon
        TermBoRes = epsilon
        FatBool = epsilon
        TermConcat = epsilon
            [017, 044] (0016, FE_COLCHET) {]}
        TermAriRes = epsilon
        FatArit = epsilon
        TermRel = epsilon
        TermBoRes = epsilon
        FatBool = epsilon
            [017, 045] (0004, FE_PARENTE) {)}
            [017, 047] (0005, AB_CHAVE) {{}
18  vet [j + gap] = vet[j];
        Bloco = 'id' BlocoAux
            [018, 019] (0000,       ID) {vet}
        BlocoAux = 'ab_colchet' ExpArit 'fe_colchet' Atribuicao Bloco
            [018, 023] (0015, AB_COLCHET) {[}
        ExpArit = TermArit FatArit
        TermArit = FatAritRes TermAriRes
        FatAritRes = 'id' CFuncAux
            [018, 024] (0000,       ID) {j}
        CFuncAux = epsilon
        TermAriRes = epsilon
        FatArit = 'opa_ad' TermArit FatArit
            [018, 026] (0023,   OPA_AD) {+}
        TermArit = FatAritRes TermAriRes
        FatAritRes = 'id' CFuncAux
            [018, 028] (0000,       ID) {gap}
```

```
CFuncAux = epsilon
TermAriRes = epsilon
FatArit = epsilon
    [018, 031] (0016, FE_COLCHET) {]}
Atribuicao = 'atribuicao' ExpConcat 'pont_virg'
    [018, 033] (0008, ATRIBUICAO) {=}
ExpConcat = ExpBool TermConcat
ExpBool = TermBool FatBool
TermBool = ExpRel TermBoRes
ExpRel = ExpArit TermRel
ExpArit = TermArit FatArit
TermArit = FatAritRes TermAriRes
FatAritRes = 'id' CFuncAux
    [018, 035] (0000,       ID) {vet}
CFuncAux = 'ab_colchet' ExpConcat 'fe_colchet'
    [018, 038] (0015, AB_COLCHET) {[}
ExpConcat = ExpBool TermConcat
ExpBool = TermBool FatBool
TermBool = ExpRel TermBoRes
ExpRel = ExpArit TermRel
ExpArit = TermArit FatArit
TermArit = FatAritRes TermAriRes
FatAritRes = 'id' CFuncAux
    [018, 039] (0000,       ID) {j}
CFuncAux = epsilon
TermAriRes = epsilon
FatArit = epsilon
TermRel = epsilon
TermBoRes = epsilon
FatBool = epsilon
TermConcat = epsilon
    [018, 040] (0016, FE_COLCHET) {]}
TermAriRes = epsilon
FatArit = epsilon
TermRel = epsilon
TermBoRes = epsilon
FatBool = epsilon
```

```
        TermConcat = epsilon
            [018, 041] (0001, PONT_VIRG) {;}
19  j = j - gap;
        Bloco = 'id' BlocoAux
            [019, 019] (0000,       ID) {j}
        BlocoAux = Atribuicao Bloco
        Atribuicao = 'atribuicao' ExpConcat 'pont_virg'
            [019, 021] (0008, ATRIBUICAO) {=}
        ExpConcat = ExpBool TermConcat
        ExpBool = TermBool FatBool
        TermBool = ExpRel TermBoRes
        ExpRel = ExpArit TermRel
        ExpArit = TermArit FatArit
        TermArit = FatAritRes TermAriRes
        FatAritRes = 'id' CFuncAux
            [019, 023] (0000,       ID) {j}
        CFuncAux = epsilon
        TermAriRes = epsilon
        FatArit = 'opa_ad' TermArit FatArit
            [019, 025] (0023,   OPA_AD) {-}
        TermArit = FatAritRes TermAriRes
        FatAritRes = 'id' CFuncAux
            [019, 027] (0000,       ID) {gap}
        CFuncAux = epsilon
        TermAriRes = epsilon
        FatArit = epsilon
        TermRel = epsilon
        TermBoRes = epsilon
        FatBool = epsilon
        TermConcat = epsilon
            [019, 030] (0001, PONT_VIRG) {;}
20  }
        Bloco = epsilon
            [020, 015] (0006, FE_CHAVE) {}}
21  vet [j + gap] = value;
        Bloco = 'id' BlocoAux
            [021, 015] (0000,       ID) {vet}
```

```
        BlocoAux = 'ab_colchet' ExpArit 'fe_colchet' Atribuicao Bloco
            [021, 019] (0015, AB_COLCHET) {[}
        ExpArit = TermArit FatArit
        TermArit = FatAritRes TermAriRes
        FatAritRes = 'id' CFuncAux
            [021, 020] (0000,       ID) {j}
        CFuncAux = epsilon
        TermAriRes = epsilon
        FatArit = 'opa_ad' TermArit FatArit
            [021, 022] (0023,   OPA_AD) {+}
        TermArit = FatAritRes TermAriRes
        FatAritRes = 'id' CFuncAux
            [021, 024] (0000,       ID) {gap}
        CFuncAux = epsilon
        TermAriRes = epsilon
        FatArit = epsilon
            [021, 027] (0016, FE_COLCHET) {]}
        Atribuicao = 'atribuicao' ExpConcat 'pont_virg'
            [021, 029] (0008, ATRIBUICAO) {=}
        ExpConcat = ExpBool TermConcat
        ExpBool = TermBool FatBool
        TermBool = ExpRel TermBoRes
        ExpRel = ExpArit TermRel
        ExpArit = TermArit FatArit
        TermArit = FatAritRes TermAriRes
        FatAritRes = 'id' CFuncAux
            [021, 031] (0000,       ID) {value}
        CFuncAux = epsilon
        TermAriRes = epsilon
        FatArit = epsilon
        TermRel = epsilon
        TermBoRes = epsilon
        FatBool = epsilon
        TermConcat = epsilon
            [021, 036] (0001, PONT_VIRG) {;}
22  }
        Bloco = epsilon
```

```
            [022, 011] (0006, FE_CHAVE) {}}
23  }
        Bloco = epsilon
            [023, 007] (0006, FE_CHAVE) {}}
24  }
        Bloco = epsilon
            [024, 003] (0006, FE_CHAVE) {}}
26  void main() {
        CodAux = Funcao CodAux
        Funcao = 'tipo_void' 'id' 'ab_parente' LParam 'fe_parente'
            'ab_chave' Bloco 'fe_chave'
            [026, 003] (0021, TIPO_VOID) {void}
            [026, 008] (0000,       ID) {main}
            [026, 012] (0003, AB_PARENTE) {(}
        LParam = epsilon
            [026, 013] (0004, FE_PARENTE) {)}
            [026, 015] (0005, AB_CHAVE) {{}
27  int size;
        Bloco = Declaracao Bloco
        Declaracao = Tipo 'id' DeclFim
        Tipo = 'tipo_int'
            [027, 005] (0017, TIPO_INT) {int}
            [027, 009] (0000,       ID) {size}
        DeclFim = 'pont_virg'
            [027, 013] (0001, PONT_VIRG) {;}
28  int i;
        Bloco = Declaracao Bloco
        Declaracao = Tipo 'id' DeclFim
        Tipo = 'tipo_int'
            [028, 005] (0017, TIPO_INT) {int}
            [028, 009] (0000,       ID) {i}
        DeclFim = 'pont_virg'
            [028, 010] (0001, PONT_VIRG) {;}
30  read("%i", size);
        Bloco = Read Bloco
        Read = 'read' 'ab_parente' 'cte_cad_ch' LId 'fe_parente'
            'pont_virg'
```

```
            [030, 005] (0034,     READ) {read}
            [030, 009] (0003, AB_PARENTE) {(}
            [030, 010] (0009, CTE_CAD_CH) {%i}
        LId = 'virgula' 'id' LIdAux
            [030, 014] (0002,  VIRGULA) {,}
            [030, 016] (0000,       ID) {size}
        LIdAux = LId
        LId = epsilon
            [030, 020] (0004, FE_PARENTE) {)}
            [030, 021] (0001, PONT_VIRG) {;}
32  int vet[size];
        Bloco = Declaracao Bloco
        Declaracao = Tipo 'id' DeclFim
        Tipo = 'tipo_int'
            [032, 005] (0017, TIPO_INT) {int}
            [032, 009] (0000,       ID) {vet}
        DeclFim = 'ab_colchet' ExpConcat 'fe_colchet' 'pont_virg'
            [032, 012] (0015, AB_COLCHET) {[}
        ExpConcat = ExpBool TermConcat
        ExpBool = TermBool FatBool
        TermBool = ExpRel TermBoRes
        ExpRel = ExpArit TermRel
        ExpArit = TermArit FatArit
        TermArit = FatAritRes TermAriRes
        FatAritRes = 'id' CFuncAux
            [032, 013] (0000,       ID) {size}
        CFuncAux = epsilon
        TermAriRes = epsilon
        FatArit = epsilon
        TermRel = epsilon
        TermBoRes = epsilon
        FatBool = epsilon
        TermConcat = epsilon
            [032, 017] (0016, FE_COLCHET) {]}
            [032, 018] (0001, PONT_VIRG) {;}
34  for i in (size) {
        Bloco = For Bloco
```

```
        For = 'for' 'id' 'in' 'ab_parente' Range 'fe_parente'
            'ab_chave' Bloco 'fe_chave'
            [034, 005] (0032,      FOR) {for}
            [034, 009] (0000,       ID) {i}
            [034, 011] (0031,       IN) {in}
            [034, 014] (0003, AB_PARENTE) {(}
        Range = ExpArit RangeAux
        ExpArit = TermArit FatArit
        TermArit = FatAritRes TermAriRes
        FatAritRes = 'id' CFuncAux
            [034, 015] (0000,       ID) {size}
        CFuncAux = epsilon
        TermAriRes = epsilon
        FatArit = epsilon
        RangeAux = epsilon
            [034, 019] (0004, FE_PARENTE) {)}
            [034, 021] (0005, AB_CHAVE) {{}
35  read("%i", vet[i]);
        Bloco = Read Bloco
        Read = 'read' 'ab_parente' 'cte_cad_ch' LId 'fe_parente'
            'pont_virg'
            [035, 007] (0034,     READ) {read}
            [035, 011] (0003, AB_PARENTE) {(}
            [035, 012] (0009, CTE_CAD_CH) {%i}
        LId = 'virgula' 'id' LIdAux
            [035, 016] (0002,  VIRGULA) {,}
            [035, 018] (0000,       ID) {vet}
        LIdAux = 'ab_colchet' ExpConcat 'fe_colchet' LId
            [035, 021] (0015, AB_COLCHET) {[}
        ExpConcat = ExpBool TermConcat
        ExpBool = TermBool FatBool
        TermBool = ExpRel TermBoRes
        ExpRel = ExpArit TermRel
        ExpArit = TermArit FatArit
        TermArit = FatAritRes TermAriRes
        FatAritRes = 'id' CFuncAux
            [035, 022] (0000,       ID) {i}
```

```
        CFuncAux = epsilon
        TermAriRes = epsilon
        FatArit = epsilon
        TermRel = epsilon
        TermBoRes = epsilon
        FatBool = epsilon
        TermConcat = epsilon
            [035, 023] (0016, FE_COLCHET) {]}
        LId = epsilon
            [035, 024] (0004, FE_PARENTE) {)}
            [035, 025] (0001, PONT_VIRG) {;}
36  }
        Bloco = epsilon
            [036, 005] (0006, FE_CHAVE) {}}
38  for i in (size - 1) {
        Bloco = For Bloco
        For = 'for' 'id' 'in' 'ab_parente' Range 'fe_parente'
            'ab_chave' Bloco 'fe_chave'
            [038, 005] (0032,      FOR) {for}
            [038, 009] (0000,       ID) {i}
            [038, 011] (0031,       IN) {in}
            [038, 014] (0003, AB_PARENTE) {(}
        Range = ExpArit RangeAux
        ExpArit = TermArit FatArit
        TermArit = FatAritRes TermAriRes
        FatAritRes = 'id' CFuncAux
            [038, 015] (0000,       ID) {size}
        CFuncAux = epsilon
        TermAriRes = epsilon
        FatArit = 'opa_ad' TermArit FatArit
            [038, 020] (0023,   OPA_AD) {-}
        TermArit = FatAritRes TermAriRes
        FatAritRes = 'cte_int'
            [038, 022] (0012,  CTE_INT) {1}
        TermAriRes = epsilon
        FatArit = epsilon
        RangeAux = epsilon
```

```
            [038, 023] (0004, FE_PARENTE) {)}
            [038, 025] (0005, AB_CHAVE) {{}
39  print("%i, ", vet[i]);
        Bloco = Print Bloco
        Print = 'print' 'ab_parente' 'cte_cad_ch' PrintAux
            'fe_parente' 'pont_virg'
            [039, 007] (0035,    PRINT) {print}
            [039, 012] (0003, AB_PARENTE) {(}
            [039, 013] (0009, CTE_CAD_CH) {%i, }
        PrintAux = 'virgula' LArgs
            [039, 019] (0002,  VIRGULA) {,}
        LArgs = ExpConcat LArgsAux
        ExpConcat = ExpBool TermConcat
        ExpBool = TermBool FatBool
        TermBool = ExpRel TermBoRes
        ExpRel = ExpArit TermRel
        ExpArit = TermArit FatArit
        TermArit = FatAritRes TermAriRes
        FatAritRes = 'id' CFuncAux
            [039, 021] (0000,       ID) {vet}
        CFuncAux = 'ab_colchet' ExpConcat 'fe_colchet'
            [039, 024] (0015, AB_COLCHET) {[}
        ExpConcat = ExpBool TermConcat
        ExpBool = TermBool FatBool
        TermBool = ExpRel TermBoRes
        ExpRel = ExpArit TermRel
        ExpArit = TermArit FatArit
        TermArit = FatAritRes TermAriRes
        FatAritRes = 'id' CFuncAux
            [039, 025] (0000,       ID) {i}
        CFuncAux = epsilon
        TermAriRes = epsilon
        FatArit = epsilon
        TermRel = epsilon
        TermBoRes = epsilon
        FatBool = epsilon
        TermConcat = epsilon
```

```
            [039, 026] (0016, FE_COLCHET) {]}
        TermAriRes = epsilon
        FatArit = epsilon
        TermRel = epsilon
        TermBoRes = epsilon
        FatBool = epsilon
        TermConcat = epsilon
        LArgsAux = epsilon
            [039, 027] (0004, FE_PARENTE) {)}
            [039, 028] (0001, PONT_VIRG) {;}
40  }
        Bloco = epsilon
            [040, 005] (0006, FE_CHAVE) {}}
41  print("%i\n", vet[size-1]);
        Bloco = Print Bloco
        Print = 'print' 'ab_parente' 'cte_cad_ch' PrintAux
            'fe_parente' 'pont_virg'
            [041, 005] (0035,    PRINT) {print}
            [041, 010] (0003, AB_PARENTE) {(}
            [041, 011] (0009, CTE_CAD_CH) {%i\n}
        PrintAux = 'virgula' LArgs
            [041, 017] (0002,  VIRGULA) {,}
        LArgs = ExpConcat LArgsAux
        ExpConcat = ExpBool TermConcat
        ExpBool = TermBool FatBool
        TermBool = ExpRel TermBoRes
        ExpRel = ExpArit TermRel
        ExpArit = TermArit FatArit
        TermArit = FatAritRes TermAriRes
        FatAritRes = 'id' CFuncAux
            [041, 019] (0000,       ID) {vet}
        CFuncAux = 'ab_colchet' ExpConcat 'fe_colchet'
            [041, 022] (0015, AB_COLCHET) {[}
        ExpConcat = ExpBool TermConcat
        ExpBool = TermBool FatBool
        TermBool = ExpRel TermBoRes
        ExpRel = ExpArit TermRel
```

```
        ExpArit = TermArit FatArit
        TermArit = FatAritRes TermAriRes
        FatAritRes = 'id' CFuncAux
            [041, 023] (0000,       ID) {size}
        CFuncAux = epsilon
        TermAriRes = epsilon
        FatArit = 'opa_ad' TermArit FatArit
            [041, 027] (0023,   OPA_AD) {-}
        TermArit = FatAritRes TermAriRes
        FatAritRes = 'cte_int'
            [041, 028] (0012,  CTE_INT) {1}
        TermAriRes = epsilon
        FatArit = epsilon
        TermRel = epsilon
        TermBoRes = epsilon
        FatBool = epsilon
        TermConcat = epsilon
            [041, 029] (0016, FE_COLCHET) {]}
        TermAriRes = epsilon
        FatArit = epsilon
        TermRel = epsilon
        TermBoRes = epsilon
        FatBool = epsilon
        TermConcat = epsilon
        LArgsAux = epsilon
            [041, 030] (0004, FE_PARENTE) {)}
            [041, 031] (0001, PONT_VIRG) {;}
43  shellSort(vet, size);
        Bloco = 'id' BlocoAux
            [043, 005] (0000,       ID) {shellSort}
        BlocoAux = CompletaFc 'pont_virg' Bloco
        CompletaFc = 'ab_parente' LArgs 'fe_parente'
            [043, 014] (0003, AB_PARENTE) {(}
        LArgs = ExpConcat LArgsAux
        ExpConcat = ExpBool TermConcat
        ExpBool = TermBool FatBool
        TermBool = ExpRel TermBoRes
```

```
            ExpRel = ExpArit TermRel
            ExpArit = TermArit FatArit
            TermArit = FatAritRes TermAriRes
            FatAritRes = 'id' CFuncAux
                [043, 015] (0000,       ID) {vet}
            CFuncAux = epsilon
            TermAriRes = epsilon
            FatArit = epsilon
            TermRel = epsilon
            TermBoRes = epsilon
            FatBool = epsilon
            TermConcat = epsilon
            LArgsAux = 'virgula' ExpConcat LArgsAux
                [043, 018] (0002,  VIRGULA) {,}
            ExpConcat = ExpBool TermConcat
            ExpBool = TermBool FatBool
            TermBool = ExpRel TermBoRes
            ExpRel = ExpArit TermRel
            ExpArit = TermArit FatArit
            TermArit = FatAritRes TermAriRes
            FatAritRes = 'id' CFuncAux
                [043, 020] (0000,       ID) {size}
            CFuncAux = epsilon
            TermAriRes = epsilon
            FatArit = epsilon
            TermRel = epsilon
            TermBoRes = epsilon
            FatBool = epsilon
            TermConcat = epsilon
            LArgsAux = epsilon
                [043, 024] (0004, FE_PARENTE) {)}
                [043, 025] (0001, PONT_VIRG) {;}
45      for i in (size - 1) {
            Bloco = For Bloco
            For = 'for' 'id' 'in' 'ab_parente' Range 'fe_parente'
                'ab_chave' Bloco 'fe_chave'
                [045, 005] (0032,      FOR) {for}
```

```
                [045, 009] (0000,       ID) {i}
                [045, 011] (0031,       IN) {in}
                [045, 014] (0003, AB_PARENTE) {(}
            Range = ExpArit RangeAux
            ExpArit = TermArit FatArit
            TermArit = FatAritRes TermAriRes
            FatAritRes = 'id' CFuncAux
                [045, 015] (0000,       ID) {size}
            CFuncAux = epsilon
            TermAriRes = epsilon
            FatArit = 'opa_ad' TermArit FatArit
                [045, 020] (0023,   OPA_AD) {-}
            TermArit = FatAritRes TermAriRes
            FatAritRes = 'cte_int'
                [045, 022] (0012,  CTE_INT) {1}
            TermAriRes = epsilon
            FatArit = epsilon
            RangeAux = epsilon
                [045, 023] (0004, FE_PARENTE) {)}
                [045, 025] (0005, AB_CHAVE) {{}
46  print("%i, ", vet[i]);
            Bloco = Print Bloco
            Print = 'print' 'ab_parente' 'cte_cad_ch' PrintAux
                'fe_parente' 'pont_virg'
                [046, 007] (0035,    PRINT) {print}
                [046, 012] (0003, AB_PARENTE) {(}
                [046, 013] (0009, CTE_CAD_CH) {%i, }
            PrintAux = 'virgula' LArgs
                [046, 019] (0002,  VIRGULA) {,}
            LArgs = ExpConcat LArgsAux
            ExpConcat = ExpBool TermConcat
            ExpBool = TermBool FatBool
            TermBool = ExpRel TermBoRes
            ExpRel = ExpArit TermRel
            ExpArit = TermArit FatArit
            TermArit = FatAritRes TermAriRes
            FatAritRes = 'id' CFuncAux
```

```
            [046, 021] (0000,       ID) {vet}
        CFuncAux = 'ab_colchet' ExpConcat 'fe_colchet'
            [046, 024] (0015, AB_COLCHET) {[}
        ExpConcat = ExpBool TermConcat
        ExpBool = TermBool FatBool
        TermBool = ExpRel TermBoRes
        ExpRel = ExpArit TermRel
        ExpArit = TermArit FatArit
        TermArit = FatAritRes TermAriRes
        FatAritRes = 'id' CFuncAux
            [046, 025] (0000,       ID) {i}
        CFuncAux = epsilon
        TermAriRes = epsilon
        FatArit = epsilon
        TermRel = epsilon
        TermBoRes = epsilon
        FatBool = epsilon
        TermConcat = epsilon
            [046, 026] (0016, FE_COLCHET) {]}
        TermAriRes = epsilon
        FatArit = epsilon
        TermRel = epsilon
        TermBoRes = epsilon
        FatBool = epsilon
        TermConcat = epsilon
        LArgsAux = epsilon
            [046, 027] (0004, FE_PARENTE) {)}
            [046, 028] (0001, PONT_VIRG) {;}
47  }
        Bloco = epsilon
            [047, 005] (0006, FE_CHAVE) {}}
48  print("%i\n", vet[size-1]);
        Bloco = Print Bloco
        Print = 'print' 'ab_parente' 'cte_cad_ch' PrintAux
            'fe_parente' 'pont_virg'
            [048, 005] (0035,    PRINT) {print}
            [048, 010] (0003, AB_PARENTE) {(}
```

```
    [048, 011] (0009, CTE_CAD_CH) {%i\n}
PrintAux = 'virgula' LArgs
    [048, 017] (0002,  VIRGULA) {,}
LArgs = ExpConcat LArgsAux
ExpConcat = ExpBool TermConcat
ExpBool = TermBool FatBool
TermBool = ExpRel TermBoRes
ExpRel = ExpArit TermRel
ExpArit = TermArit FatArit
TermArit = FatAritRes TermAriRes
FatAritRes = 'id' CFuncAux
    [048, 019] (0000,      ID) {vet}
CFuncAux = 'ab_colchet' ExpConcat 'fe_colchet'
    [048, 022] (0015, AB_COLCHET) {[}
ExpConcat = ExpBool TermConcat
ExpBool = TermBool FatBool
TermBool = ExpRel TermBoRes
ExpRel = ExpArit TermRel
ExpArit = TermArit FatArit
TermArit = FatAritRes TermAriRes
FatAritRes = 'id' CFuncAux
    [048, 023] (0000,      ID) {size}
CFuncAux = epsilon
TermAriRes = epsilon
FatArit = 'opa_ad' TermArit FatArit
    [048, 027] (0023,   OPA_AD) {-}
TermArit = FatAritRes TermAriRes
FatAritRes = 'cte_int'
    [048, 028] (0012,  CTE_INT) {1}
TermAriRes = epsilon
FatArit = epsilon
TermRel = epsilon
TermBoRes = epsilon
FatBool = epsilon
TermConcat = epsilon
    [048, 029] (0016, FE_COLCHET) {]}
TermAriRes = epsilon
```

```
            FatArit = epsilon
            TermRel = epsilon
            TermBoRes = epsilon
            FatBool = epsilon
            TermConcat = epsilon
            LArgsAux = epsilon
                [048, 030] (0004, FE_PARENTE) {)}
                [048, 031] (0001, PONT_VIRG) {;}
 49  }
            Bloco = epsilon
                [049, 003] (0006, FE_CHAVE) {}}

Sintaxe correta! Análise encerrada!
```