



FIPP–Faculdade de Informática de Presidente
Prudente
BCC - BACHARELADO EM CIÊNCIA DA
COMPUTAÇÃO

Fernando Caique Soares de Souza

Compilador fnd

Extended Backus-Naur-Form

t_start = t_begin, t_cmd_var, t_cmd, t_end

t_begin = "begin"

t_end = "end"

t_ponto_virgula = ";;"

t_cmd_var = (t_tipo_var, t_definir_var, t_ponto_virgula) [t_cmd_var]

t_tipo_var = (t_int | t_double | t_exp)

t_int = "int"

t_double = "double"

t_exp = "exp"

t_definir_var = t_id_var [t_atribuidor, t_valor] [t_virgula, t_definir_var]

t_virgula = ";;"

t_id_var = t_letra, {t_algarismo | t_letra}*

t_cmd = (t_atribuicao, t_ponto_virgula) |
(t_cmd_if | t_cmd_while | t_cmd_for) [t_cmd]

t_cmd_if = t_if, t_abre_parenteses, t_bool, t_fecha_parenteses, t_abre_chaves,
t_cmd, t_fecha_chaves [t_cmd_else]

t_abre_parenteses = "("

t_fecha_parenteses = ")"

t_abre_chaves = "{"

t_fecha_chaves = "}"

t_if = "if"

```
t_bool = (t_not, tbool) |  
    (t_abre_parenteses, t_bool, t_fecha_parenteses) |  
    (  
        (t_valor | operacao_aritmetica | t_id_var | t_bool),  
        t_operador_relacional,  
        (t_valor | operacao_aritmetica | t_id_var | t_bool)  
    )  
    [t_and_or, t_bool]
```

```
t_operador_relacional = ( t_menor | t_maior | t_igual | t_menor_igual | t_maior_igual |  
t_diferente)
```

```
t_menor = "<"
```

```
t_maior = ">"
```

```
t_igual = "=="
```

```
t_menor_igual = "<="
```

```
t_maior_igual = ">="
```

```
t_diferente = "!="
```

```
t_and_or = ( t_and | t_or)
```

```
t_and = "&&"
```

```
t_or = "||"
```

```
t_not = "!"
```

```
t_cmd_else = t_else, (t_cmd_if | t_abre_chaves, t_cmd, t_fecha_chaves)
```

```
t_else = "else"
```

```
t_cmd_while = t_while, t_abre_parenteses, t_bool, t_fecha_parenteses,  
t_abre_chave, t_cmd, t_fecha_chave
```

```
t_while = "while"
```

t_cmd_for = t_for, t_abre_parentese, [atribuicao], t_ponto_virgula, t_bool,
t_ponto_virgula, [t_atribuicao], t_fecha_parenteses, t_abre_chave, t_cmd,
t_fecha_chave

t_do = "do"

t_letra = (t_letra_minuscula | t_letra_maiuscula)

t_letra_minuscula = "a"-"z"

t_letra_maiuscula = "A"-"Z"

t_algarismo = "0"-"9"

t_valor = t_valor_int | t_valor_double | t_valor_exp

t_valor_int = {t_algarismo}+

t_valor_double = t_valor_int, [t_ponto, t_valor_int]

t_ponto = "."

t_valor_exp = t_valor_double, (t_mais|t_menos), t_e, t_valor_int

t_e = "e"

t_operador_aritmetico = (t_mais | t_menos | t_mult | t_div)

t_mais = "+"

t_menos = "-"

t_mult = "**"

t_div = "/"

t_atribuidor = "="

t_atribuicao = t_id_var, t_atribuidor, (t_id_var | t_valor | t_operacao_aritmetica)

$t_operacao_aritmetica = ((t_abre_parenteses, t_operacao_aritmetica, t_fecha_parenteses) |$
 $((t_valor | t_id_var | t_operacao_aritmetica), t_operador_aritmetico, (t_valor | t_id_var | t_operacao_aritmetica))) [operador_aritmetico, (t_valor, id_var | t_operacao_aritmetica)]$

First e Follow

Índice	Expressão	First	Follow
1	t_start	first(t_begin)	{'\$'}
2	t_begin	{'begin'}	first(t_cmd_var)
3	t_end	{'end'}	{'\$'}
4	t_cmd_var	first(t_tipo_var)	first(t_cmd)
5	t_tipo_var	{int, double, exp}	first(t_id_var)
6	t_definir_var	first(t_id_var)	first(t_virgula) U first(t_cmd)
7	t_virgula	{','}	first(t_definir_var)
8	t_id_var	first(t_letra)	first(t_atribuidor) U first(t_virgula) U {';'} U first(t_and_or) U first(t_operador_relacional) U first(t_operador_aritmetico) U {'}'}
9	t_cmd	{'do', 'while', 'if'} U first(t_id_var)	first(t_end) U {'}'
10	t_cmd_if	{'if'}	follow(t_cmd)
11	t_abre_parenteses	{'('}	{'!', '('} U first(t_valor) U

			first(id_var) U
12	t_fecha_parenteses	{'})}	{';', '{' } U first(t_operador_aritmetico) U first(t_operador_relacional)
13	t_abre_chave	{'{'}	first(t_cmd)
14	t_fecha_chave	{'}'}	follow(t_cmd) U first(t_else) U first(t_end)
15	t_bool	first(t_not) U first(t_valor) U first(t_id_var) U {'('}	{'})' } U first(t_and_or) U first(t_operador_relacional) U first(operador_aritmetico)
16	t_operador_relacional	{ '<' , '>' , '==' , '<=' , '>=' , '!=' }	first(t_valor) U first(t_id_var)U first(t_bool)
17	t_menor	{ '<' }	first(t_id_var) U first(t_valor) U {'('}
18	t_maior	{ '>' }	follow(t_menor)
19	t_igual	{ '==' }	follow(t_menor)
20	t_menor_igual	{ '<=' }	follow(t_menor)
21	t_maior_igual	{ '>=' }	follow(t_menor)
22	t_diferente	{ '!=' }	follow(t_menor)
23	t_and_or	{ '&&' , ' ' }	first(t_valor) U first(t_id_var) U first(t_bool)
24	t_and	{ '&&' }	follow(t_and_or)
25	t_or	{ ' ' }	follow(t_and_or)
26	t_not	{ '!' }	first(t_bool)
27	t_cmd_else	{ 'else' }	follow(t_cmd_if)
28	t_else	{ 'else' }	{ 'if' , '{' }

29	t_cmd_while	{'while'}	follow(t_cmd)
30	t_while	{'while'}	{'('}
31	t_cmd_for	{'for'}	{';'}
32	t_for	{'for'}	first(t_cmd)
33	t_ponto_virgula	{';'}	first(t_cmd_var) U first(t_cmd)
34	t_letra	first(t_letra_minuscul a) U first(t_letra_maiuscul a)	first(t_letra) first(t_algarismo) U follow(t_id_var)
35	t_letra_minuscula	{a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z}	follow(t_letra)
36	t_letra_maiuscula	{A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z}	follow(t_letra)
37	t_algarismo	{0, 1, 2, 3, 4, 5, 6, 7, 8, 9}	follow(t_letra) U follow(t_valor_int) U {'.'}
38	t_valor	first(t_algarismo) U first(t_id_var)	first(t_and_or) U first(t_operador_relacion al) U follow(t_bool) U first(virgula) U follow(t_definir_var) U follow(atribuicao) U first(operador_aritmetico) U follow(operacao_aritmeti ca)
39	t_valor_int	first(t_algarismo)	follow(t_valor)
40	t_valor_double	first(t_algarismo)	follow(t_valor)
41	t_valor_exp	first(t_algarismo)	follow(t_valor)
42	t_e	{'e'}	first(t_valor_int)
43	t_ponto	{'.'}	first(t_valor_int)

44	t_operador_aritmetico	{+, -, *, /}	first(t_valor) U first(t_id_var) U {'}' , '('}
48	t_mais	{'+'}	follow(t_operador_aritmetico)
49	t_menos	{'-'}	follow(t_mais)
50	t_mult	{'*'}	follow(t_mais)
51	t_div	{'/'}	follow(t_mais)
52	t_atribuidor	{'='}	first(t_id_var) U first(t_valor) U {'('}
53	t_atribuicao	first(t_id_var)	first(t_ponto_virgula)
54	t_operacao_aritmetica	first(t_abre_parenteses) U first(t_valor) U first(t_id_var)	first(t_fecha_parenteses) U follow(t_valor)
55	t_comentario	{'//', '/*'}	follow(t_comentario_linha) U follow(t_comentario_bloco)
56	t_comentario_linha	{'//'}	{'\n'}
57	t_comentario_bloco	{'/*'}	qualquer coisa