Universidade Federal do Ceará
Campus Crateús

Ciência da Computação
Compiladores

Expressões Regulares da Linguagem C

Francisco David Nascimento Sousa 412772
Lucas Chaves Evangelista 427671

1 de Setembro de 2020

1 Expressões regulares da linguagem C

Descrição das expressões regulares:

Palavra:

```
W = [A-Za-z]
```

Número:

```
D = [0-9]
```

Identificador:

```
id = <W>(<W> + <D>)*
```

Tipo de dados:

```
DT = int | float | char | void
```

Char literal:

```
CL = '<W>'
```

String literal:

```
SL = "(<W> + <D>)*"
```

Inteiro literal:

```
IL => (<D>)+
```

Flutuante literal:

```
FL => (<D>)+.(<D>)+
```

Operadores booleana:

```
boolean_operator = == | != | ! | >= | > | <= | < | && | ||
```

Operadores aritméticos:

```
arithmetic_operator = + | - | * | | / | % | ++ | -- | += |
    *= | /=
```

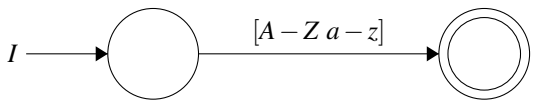
Delimitadores:

```
delimiter = , | ;
```

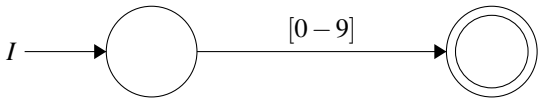
2 Autômatos C

Autômatos das expressões acima:

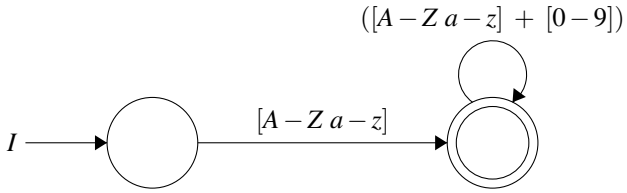
Letra:



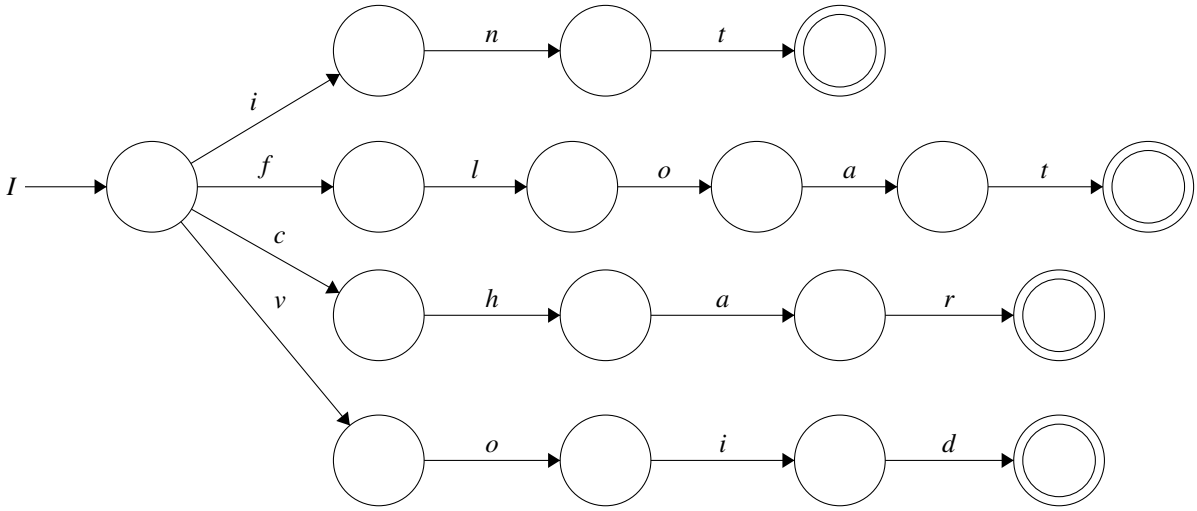
Número:



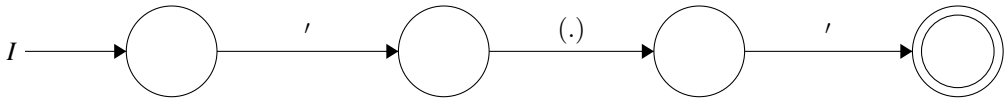
Identificador:



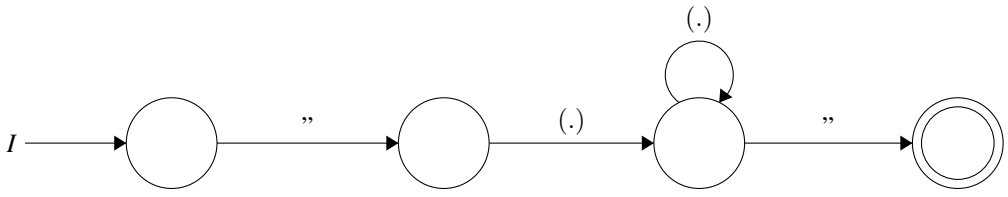
Tipo de dados:



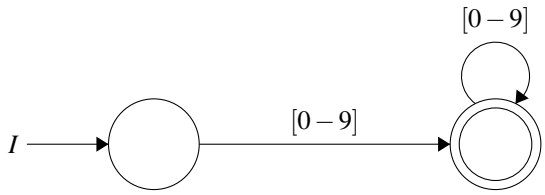
Caractere:



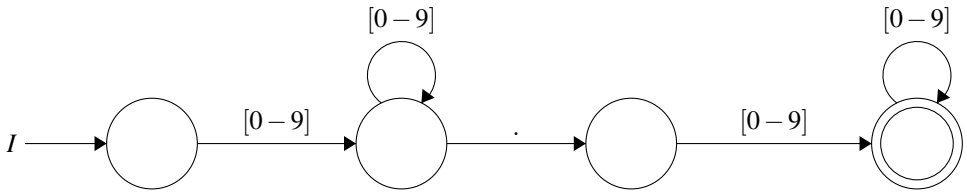
String:



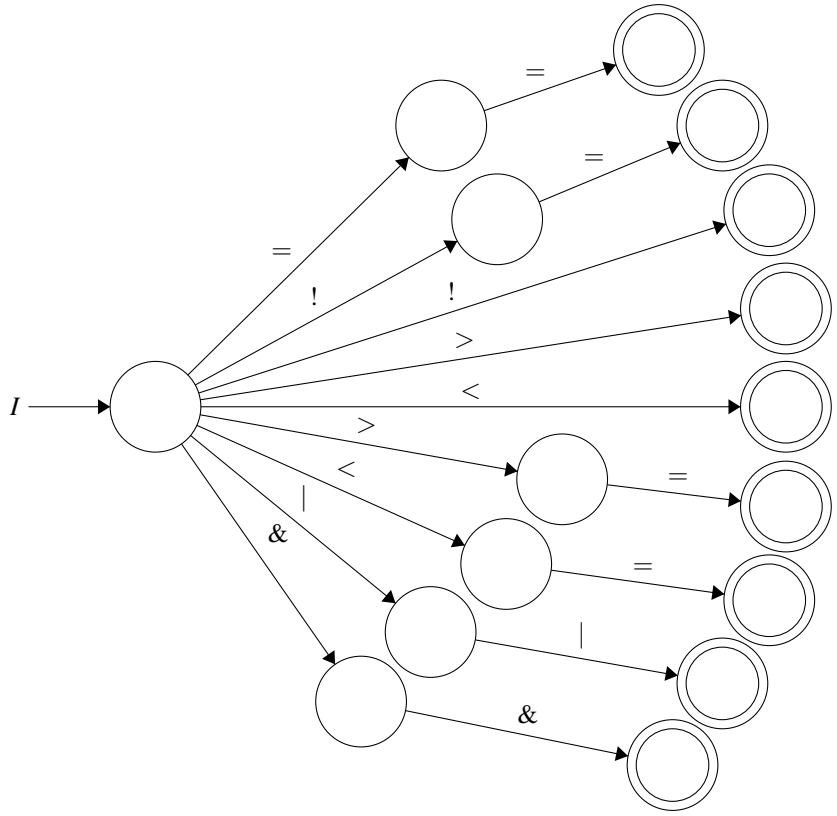
Inteiro literal:



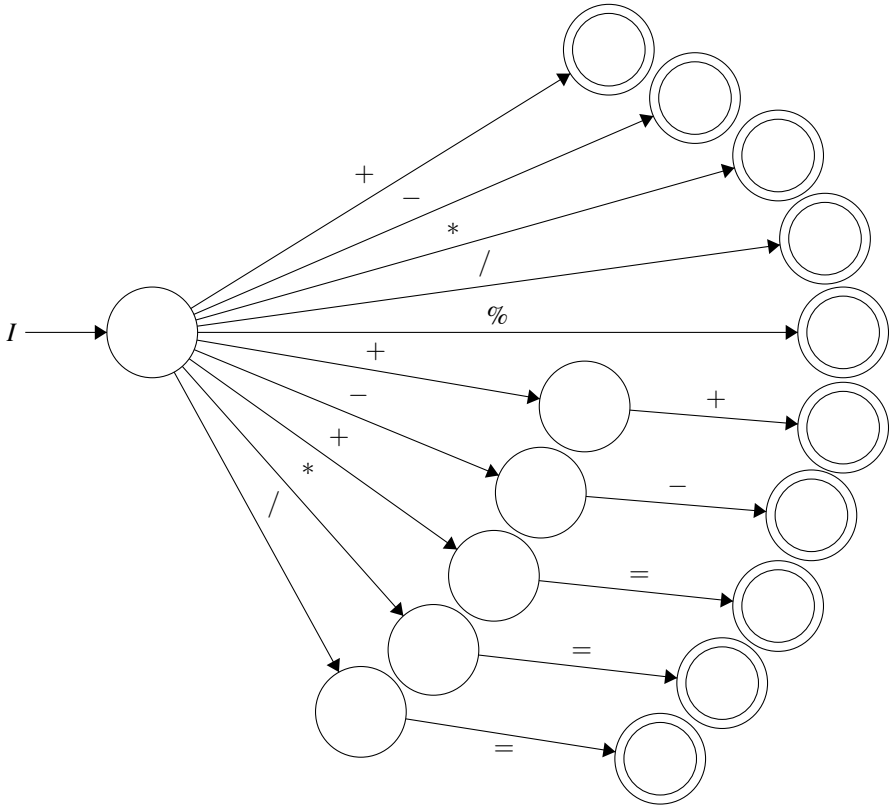
Ponto flutuante literal:



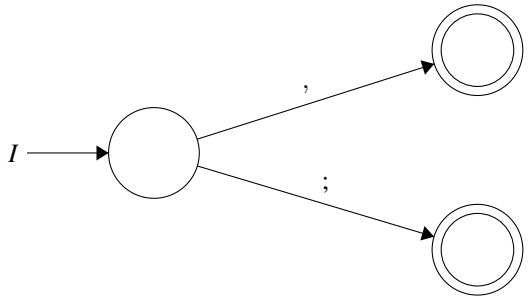
Operadores booleanos:



Operadores aritméticos:



Delimitadores:



3 Definições da gramática da linguagem C

P ⇒ void main () <B> (1)

E ⇒ <id> = <E> (2)
⇒ !<E> <E_1> (3)
⇒ -<E> <E_1> (4)
⇒ +<E> <E_1> (5)
⇒ (<E>) <E_1> (6)
⇒ <IL> <E_1> (8)
⇒ <CL> <E_1> (9)
⇒ <FL> <E_1> (10)
⇒ <SL> <E_1> (11)

E_1 ⇒ || <E> (12)
⇒ == <E> (13)
⇒ ! = <E> (14)
⇒ ≤ <E> (15)
⇒ < <E> (16)
⇒ ≥ <E> (17)
⇒ > <E> (18)
⇒ && <E> (19)
⇒ + <E> (20)
⇒ − <E> (21)
⇒ ∗ <E> (22)
⇒ / <E> (23)
⇒ % <E> (24)
⇒ ; <E> (25)

IF_ST ⇒ if (<E>) <STMT> else <STMT> (26)

B ⇒ { <STMT_LIST> } (27)

VAR ⇒ <DT> <id>; (28)

STMT ⇒ <E> (29)
⇒ <IF_ST> (30)
⇒ <B> (31)
⇒ <VAR> (32)

STMT_LIST ⇒ <STMT> <STMT_LIST> (33)
⇒ ε (34)

Descrição dos não terminais

Não terminais	Descrição
P	Programa principal
E	Expressão da linguagem
E_1	Auxiliar das Expressões
IF_ST	Declaração de comando if
B	Declaração de bloco de código
VAR	Declaração de variável
STMT	Declaração
STMT_LIST	Lista de declarações

Conjunto First

Não terminal	First
P	$\{void\}$
E	$\{id, !, -, +, (, IL, CL, FL, SL\}$
E_1	$\{\|\|, ==, !=, <=, <, >=, >, \&\&, +, -, *, /, \%, ;\}$
IF_ST	$\{if\}$
B	$\{\{\}$
VAR	$\{DT\}$
STMT	$\{id, !, -, +, (, IL, CL, FL, SL, if, \{, DT\}$
STMT_LIST	$\{id, !, -, +, (, IL, CL, FL, SL, if, \{, DT\}$

Conjunto Follow

Não terminal	Follow
P	$\{\$\}$
E	$\{\|\|, ==, !=, <=, <, >=, >, \&\&, +, -, *, /, \%,)\}$
E_1	$\{\|\|, ==, !=, <=, <, >=, >, \&\&, +, -, *, /, \%,)\}$
IF_ST	$\{\}$
B	$\{\}$
VAR	$\{\}$
STMT	$\{\}$
STMT_LIST	$\{)\}$

Tabela preditiva

	void	DT	(	)	{	}	if	!	else	id	IL	CL	FL	SL
P	1													
E			6					3		2	8	9	10	11
E_1														
IF_ST							26							
B					27									
VAR		28												
STMT		32	29		31		30	29		29	29	29	29	29
STMT_LIST		33	33		33	34	33	33		33	33	33	33	33

	$!=$	;	$\leq$	$<$	$\geq$	$>$	$+$	$-$	$*$	$/$	&&	$==$	$\|\|$	%
P														
E							5	4						
E_1	14	25	15	16	17	18	20	21	22	23	19	13	12	24
IF_ST														
B														
VAR														
STMT							29	29						
STMT_LIST							33	33						

4 Diagramas de transição

Abaixo está os diagramas para as gramáticas.

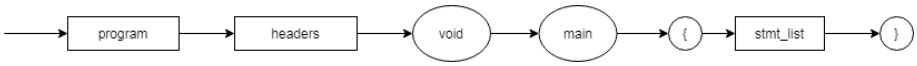


Figura 1: Programa principal

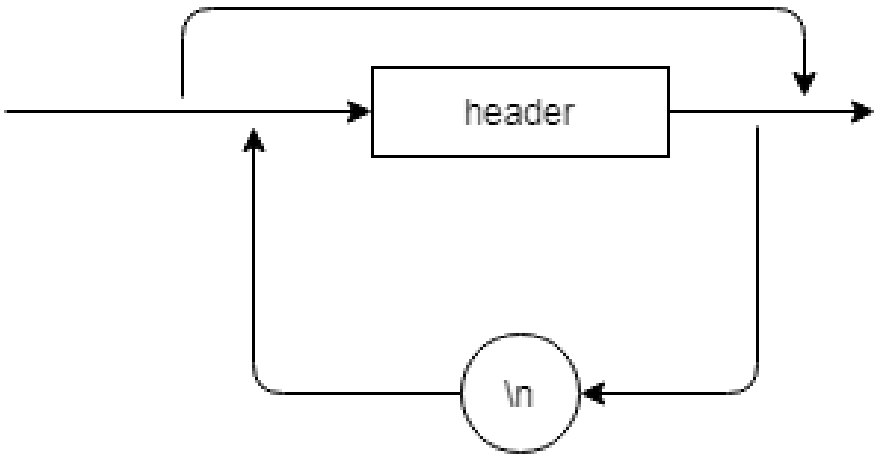


Figura 2: Cabeçalhos

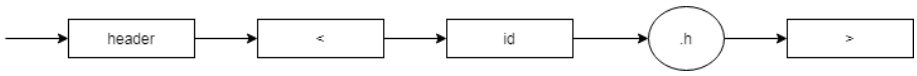


Figura 3: Cabeçalho

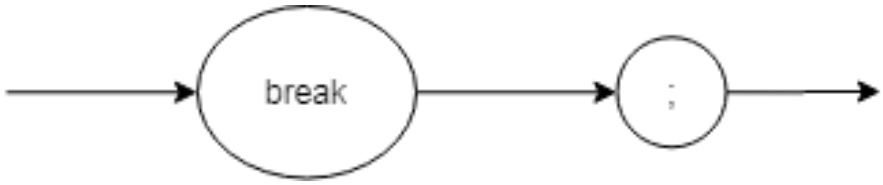


Figura 4: Declaração break

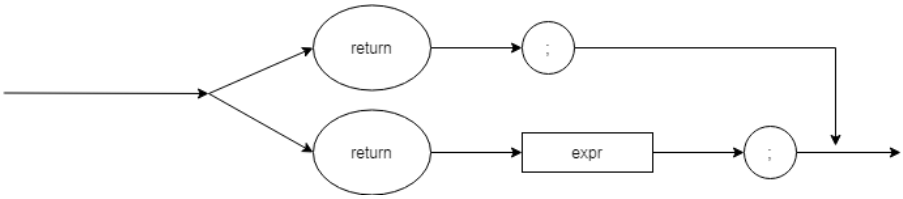


Figura 5: Declaração return

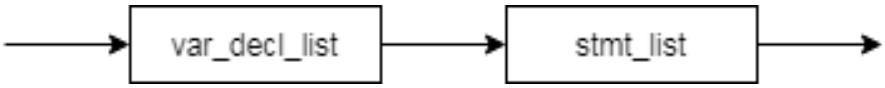


Figura 6: Declaração de bloco de comandos

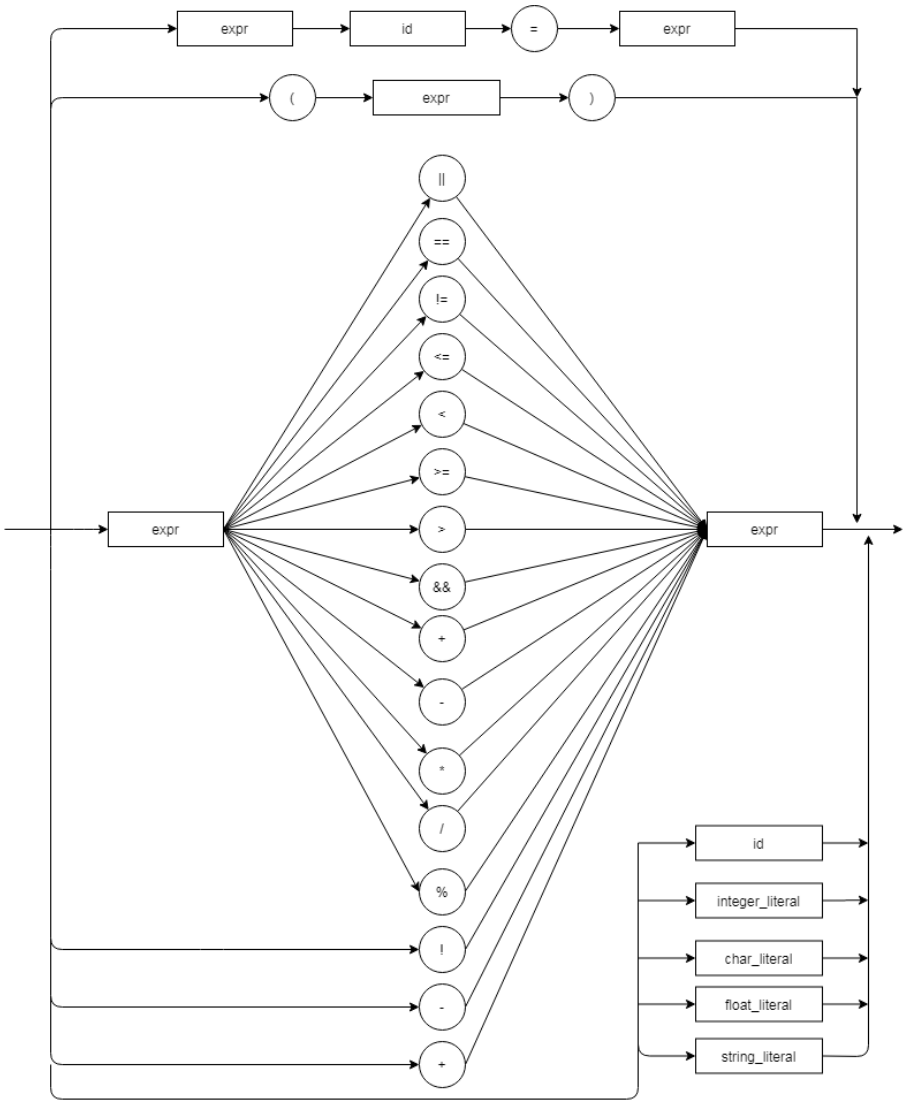


Figura 7: Expressão

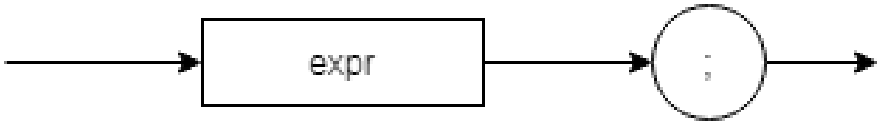


Figura 8: Declaração de expressão

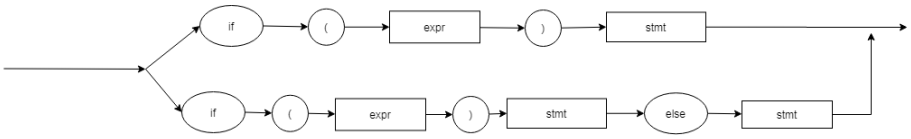


Figura 9: Declaração de condicional

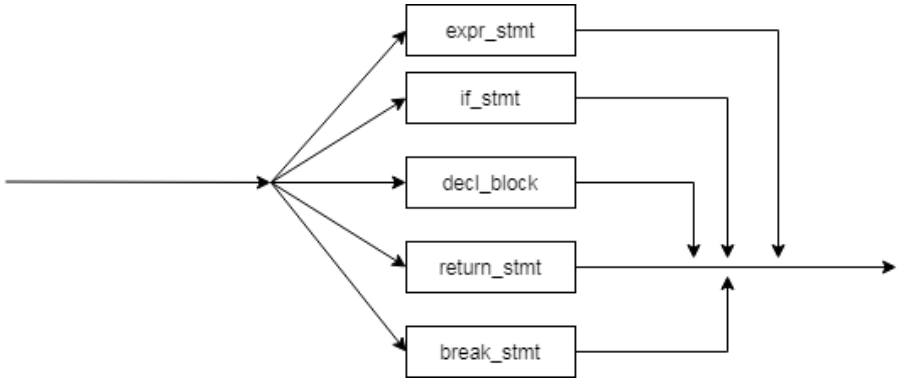


Figura 10: Declaração

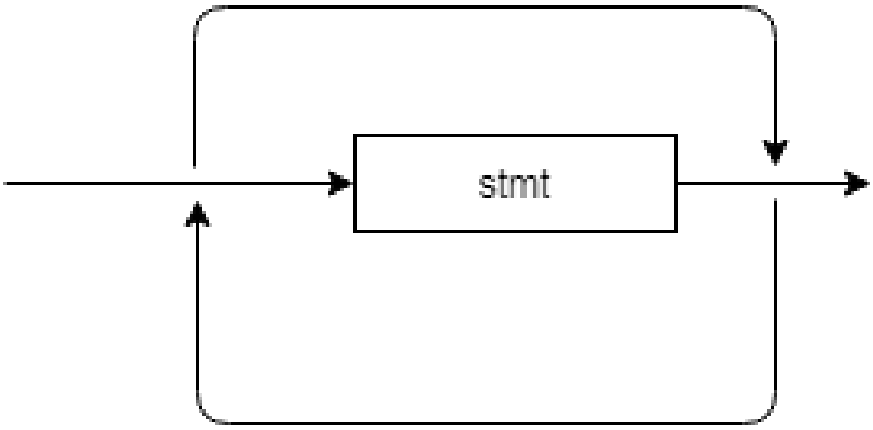


Figura 11: List de declaração

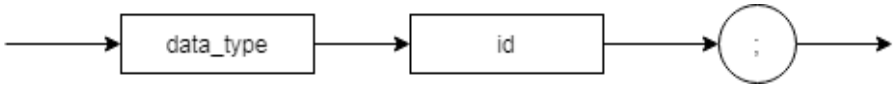


Figura 12: Declaração de variável

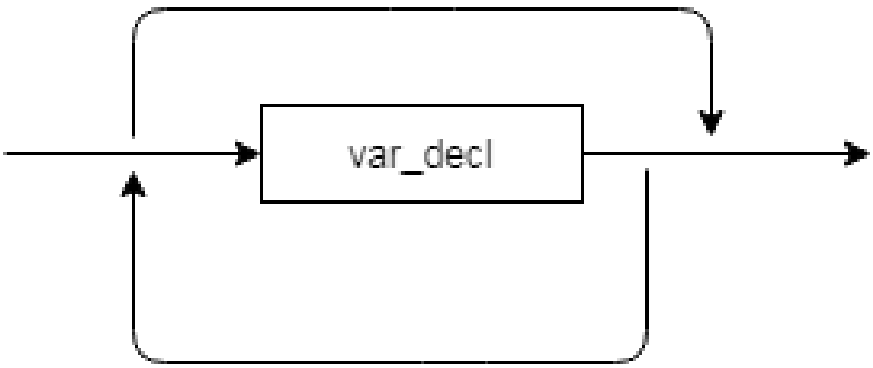


Figura 13: List de declaração de variáveis