Universidade Federal de Uberlândia

Bacharelado em Ciência da Computação

GBC071 Construção de Compiladores

Prof. Luiz Gustavo Almeida Martins

Lucas de Paula Martins – 12221BCC001

Cauê Grassi Ribeiro da Silva - 12221BCC019

Entrega Final

Uberlândia, Minas Gerais

12 de fevereiro de 2026

Universidade Federal de Uberlândia

Bacharelado em Ciência da Computação

GBC071 Construção de Compiladores

1. Projeto da Linguagem

1.1. Gramática Livre de Contexto (GLC)

G = (V, T, P, S)

Definições:

- G = gramática livre de contexto (GLC) da linguagem
- T = conjunto de símbolos terminais (Tokens)
- V = conjunto de símbolos não terminais (Variáveis Sintáticas)
- P = conjunto de produções (Regras de Transição)
- S = símbolo inicial da gramática (Símbolo Sentencial)

Conjunto de Terminais (T)

T = {"void", "int", "char", "float", "main", "if", "then", "elsif", "else", "while", "do", "for", "+", "-", "*", "/", "**", "==", "!=", "<", ">", "<=", ">=",":=","(", ")", "[", "]", ",", ";", "id", "const_int" | "const_float" | "char_const"}

Connto de Variáveis (V)

V = {<programa>, <bloco>, <lista_comandos>, <comando>, <decl>, <lista_decl>, <tipo>, <lista_ids>, <atribuicao>, <if>, <lista_elsif>, <elsif>, <else>, <entidade>, <while>, <do>, <for>, <expressao>, <termo>, <fator>, <condicao>, <operando_cond>, <operador_relacional>, <constante>}

Símbolo Inicial S

S = {<programa>}

Conjunto de Produções (P)

(1) Estrutura do Programa

<programa> ::= <tipo> "main" "(" ")" <bloco>
<bloco> ::= "[" < lista_declaracao> <lista_comandos> "]"
<lista_comandos> ::= <comando> <lista_comandos> | ε
<comando> ::= <if>
| <while>
| <do>
| <for>
| <atribuicao>

(2) Declarações

Universidade Federal de Uberlândia

Bacharelado em Ciência da Computação

GBC071 Construção de Compiladores

<lista_declaracao> ::= <declaracao> <lista_declaracao> | ε
<declaracao> ::= <tipo> <lista_ids> “;”
<tipo> ::= "void" | "int" | "char" | "float"
<lista_ids> ::= "id" | "id" “,” <lista_ids>

(3) Comandos

<atribuicao> ::= <id> “:=” <expressao> “;”
<if> ::= "if" "(" <condicao> ")" "then" <entidade> <lista_elsif>
<lista_elsif> ::= <elsif> <lista_elsif> | <else> | ε
<elsif> ::= "elsif" "(" <condicao> ")" "then" <entidade>
<else> ::= "else" <entidade>
<entidade> ::= <bloco> | <comando>
<while> ::= "while" "(" <condicao> ")" "do" <entidade>
<do> ::= "do" <entidade> "while" "(" <condicao> ")" ";"
<for> ::= "for" "(" "id" ";" <num> ";" <num> ";" <expressao> ")" <entidade>

(4) Expressões Aritméticas

<expressao> ::= <expressao> “+” <expressao>
| <expressao> “-” <expressao>
| <expressao> “*” <expressao>
| <expressao> “/” <expressao>
| <expressao> “**” <expressao>
| "id" | "const_int" | "const_float" | "char_const"

(5) Condições

<condicao> ::= <operando_cond> <operador_relacional> <operando_cond>
<operando_cond> ::= "id" | "const_int" | "const_float" | "char_const"
<operador_relacional> ::= "==" | "!=" | "<" | ">" | "<=" | ">="

1.2. Identificação dos Tokens da Linguagem

(1) Palavras-chave

Token	Lexema	Atributo	Descrição
T_VOID	void	—	Tipo void
T_INT	int	—	Tipo inteiro
T_CHAR	char	—	Tipo caractere
T_FLOAT	float	—	Tipo float
T_MAIN	main	—	Função principal
T_IF	if	—	Comando condicional
T_THEN	then	—	Parte do if
T_ELSIF	elsif	—	Else condicional
T_ELSE	else	—	Else simples
T_WHILE	while	—	Laço while

Universidade Federal de Uberlândia

Bacharelado em Ciência da Computação

GBC071 Construção de Compiladores

T_DO	do	—	Início do bloco do-while

(2) Operadores Aritméticos

Token	Lexema	Atributo
T_SOMA	+	—
T_SUB	-	—
T_MULT	*	—
T_DIV	/	—
T_POT	**	—

(3) Operadores Relacionais

Token	Lexema	Atributo
T_IGUAL	==	—
T_DIF	!=	—
T_MENOR	<	—
T_MAIOR	>	—
T_MENOR_IG	<=	—
T_MAIOR_IG	>=	—

(4) Símbolos e Separadores

Token	Lexema	Atributo
T_ABREPAR	(	—
T_FECHAPAR	)	—
T_ABREBLOCO	[	—
T_FECHABLOCO	]	—
T_VIRG	,	—
T_PV	;	—
T_ATRIB	:=	—

(5) Constantes e Identificadores

Token	Exemplo	Atributo
T_ID	idade	string (lexema)
T_NUM_INT	123	valor inteiro
T_NUM_FLOAT	3.14, 0.1E-2	valor float
T_CHAR_CONST	'A'	caractere

Universidade Federal de Uberlândia

Bacharelado em Ciência da Computação

GBC071 Construção de Compiladores

(6) Comentários e Tokens Especiais

Token	Padrão	Atributo
T_COMENT	{% ... %}	ignorado
T_WS	espaços, tabs, quebras	ignorado
T_EOF	fim do arquivo	—
T_ERROR	qualquer sequência inválida	string (lexema)

1.3. Expressões Regulares de Cada Token

Palavras-chave

"void" → T_VOID

"int" → T_INT

"char" → T_CHAR

"float" → T_FLOAT

"main"→ T_MAIN

"if" → T_IF

"then" → T_THEN

"elsif" → T_ELSIF

"else" → T_ELSE

"while"→ T_WHILE

"do” → T_DO

"for” → T_FOR

Operadores Aritméticos

"+" → T_SOMA

"-" → T_SUB

"*" → T_MULT

"/" → T_DIV

"**" → T_POT

Operadores Relacionais

"==" → T_IGUAL

Universidade Federal de Uberlândia

Bacharelado em Ciência da Computação

GBC071 Construção de Compiladores

"!=" → T_DIF

"<=" → T_MENOR_IG

">=" → T_MAIOR_IG

"<" → T_MENOR

">" → T_MAIOR

Símbolos / Separadores

"(" → T_ABREPAR

")" → T_FECHAPAR

"[" → T_ABREBLOCO

"]" → T_FECHABLOCO

"," → T_VIRG

";" → T_PV

":=" → T_ATRIB

Identificadores e Constantes

[A-Za-z_] [A-Za-z0-9_]* → T_ID

[0-9]+ → T_NUM_INT

[0-9]+\.[0-9]+(E [+-]?[0-9]+)? → T_NUM_FLOAT

'[A-Za-z]' → T_CHAR_CONST

Comentários

\{%(.|\n)*?%\} → T_COMENT

Espaços e Separadores (ignorados)

[\t\r\n]+ → T_WS

Tokens especiais

EOF → T_EOF

. → T_ERROR