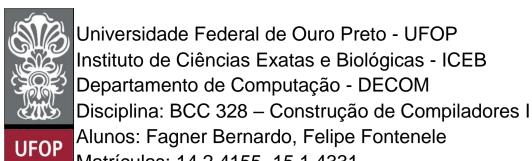
JOSÉ ROMILDO MALAQUIAS UNIVERSIDADE FEDERAL DE OURO PRETO BCC328 – CONSTRUÇÃO DE COMPILADORES I

ESPECIFICAÇÃO DA LINGUAGEM FF CODE

FAGNER DE OLIVEIRA BERNARDO FELIPE FONTENELE DE ÁVILA MAGALHÃES

OURO PRETO 2018





Alunos: Fagner Bernardo, Felipe Fontenele

Matrículas: 14.2.4155, 15.1.4331



A LINGUAGEM FF CODE

A linguagem FF Code é uma pequena linguagem, que foi idealizada para ser simples e intuitiva. Ela possui forte influência de algumas linguagens já consolidadas, como a linguagem C, C++ e da linguagem Torben, desenvolvida para fins didáticos.

Ela suporta o uso de comentários de linha, comentários de bloco, literais booleanos, literais strings, literais inteiros, operadores aritméticos, operadores relacionais, operadores lógicos e atribuições. A mesma ainda é case sensitive, diferenciando letras maiúsculas e minúsculas, a extensão do seu arquivo é .ff e aceita sequência de escapes, tais como, \n, \t, \a, entre outros. Cada especificação suportada possui uma descrição clara e objetiva.

Operadores Aritméticos:

Os operadores aritméticos suportados pela linguagem são: - (subtração), * (multiplicação), / (divisão), % (resto da divisão inteira), + (soma).

Operadores Relacionais

Os operadores relacionais suportados pela linguagem são: != (diferente), > (maior que), >= (maior ou igual que), < (menor que), <= (menor ou igual que), == (igualdade).

Operadores Lógicos

Os operadores lógicos suportados pela linguagem são: ! (negação lógica), & (e lógico), | (ou lógico).

Atribuição

A linguagem FF Code suporta a operação de atribuição pelo símbolo =.

GRAMÁTICA LIVRE DE CONTEXTO

A linguagem FF Code possui uma gramática livre de contexto, que será apresentada a seguir, e que define a sintaxe de todas as construções permissíveis na linguagem. Os subtítulos em negrito indicam a que aspectos da linguagem a gramática livre de contexto está associada.





Disciplina: BCC 328 – Construção de Compiladores I Alunos: Fagner Bernardo, Felipe Fontenele

DE M

Matrículas: 14.2.4155, 15.1.4331

Programa

Program -> Funs

Funs -> Fun

Funs -> Fun Funs

Declaração de Funções

Fun
$$\rightarrow$$
 TypeId (TypeIds) = Exp

Tipo booleano

TypeId -> bool id

Tipo inteiro

TypeId -> int id

Exp -> fbool

Exp -> fint

Literais

Exp -> fbool

Exp -> fint

Exp -> fstring

Variável

 $Exp \rightarrow id$

Atribuição

 $Exp \rightarrow id = Exp$

Operadores aritméticas

 $Exp \rightarrow Exp + Exp$

 $Exp \rightarrow Exp - Exp$

 $Exp \rightarrow Exp * Exp$

 $Exp \rightarrow Exp / Exp$

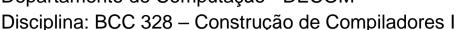
Resto da divisão

Exp -> Exp % Exp

 $Exp \rightarrow -Exp$

Operadores relacionais





Alunos: Fagner Bernardo, Felipe Fontenele

Matrículas: 14.2.4155, 15.1.4331





Exp -> Exp == Exp Exp -> Exp != Exp Exp -> Exp > Exp Exp -> Exp >= Exp Exp -> Exp < Exp Exp -> Exp <= Exp

Operadores lógicos

Exp -> Exp & Exp Exp -> Exp | Exp

Expressão Condicional

Exp -> if Exp do Exp else Exp

Expressão de Repetição

Exp -> while Exp do Exp

Chamada de função

 $Exp \rightarrow id (Exps)$

Expressão de declaração

 $Exp \rightarrow let id = Exp in Exp$

Expressão Sequência

 $Exp \rightarrow (Exps)$

Palavras Reservadas

if, do, else, let, in, while, main, import, ioff, fin, fout.

Precedência Dos Operadores

Operador	->	associatividade
-(unário)	->	NA
*, %	->	esquerda
+, -, (binário)	->	esquerda
==, !=, >, >=, <, <=	->	NA
&	->	esquerda
	->	esquerda





Disciplina: BCC 328 – Construção de Compiladores I

Alunos: Fagner Bernardo, Felipe Fontenele

Matrículas: 14.2.4155, 15.1.4331



= -> direita do, else -> direita

ASPECTOS LÉXICOS

Inicialmente é importante explicitar que, a ocorrência de caracteres brancos (espaços, tabulações horizontais e nova linha) e comentários entre símbolos léxicos são ignorados, tendo utilidade única de separação de símbolos léxicos.

Comentários de linha

A linguagem FF Code tem suporte para comentários de linha. Cada comentário a ser escrito deve ser precedido do símbolo <\$>.

Comentários de bloco

A linguagem FF Code tem suporte para comentários de bloco. Cada bloco de comentário a ser escrito deve possuir um símbolo <\\$> no início do comentário e outro <\$/>no final do comentário.

Literais inteiros

Os literais inteiros suportados pela linguagem FF Code são formados por uma sequência de um ou mais dígitos decimais.

Literais booleanos

A linguagem FF Code suporta tipos booleanos, onde **TRUE** representa uma sentença verdadeira e **FALSE** representa uma sentença falsa.

Literais string

Os literais string são construídos a partir de uma cadeia de caracteres entre o símbolo de aspas duplas.

Identificadores





Disciplina: BCC 328 – Construção de Compiladores I Alunos: Fagner Bernardo, Felipe Fontenele

Motrículae: 14.2.4455, 45.4.4224



Matrículas: 14.2.4155, 15.1.4331

São sequências de letras maiúsculas, minúsculas, dígitos decimais e sublinhados (_), começando com uma letra. Letras maiúsculas e minúsculas são distintas em um identificador.

BIBLIOTECA PADRÃO

A linguagem FF Code possui somente uma biblioteca que por si própria é completa para as tarefas dos programadores. Para isso, no início de cada arquivo de código fonte .ff é necessário fazer o import ioff, fazendo assim a chamada da biblioteca padrão. A entrada e saída de dados se dá pelos streams fin e fout respectivamente.

EXEMPLOS DE CÓDIGO

Abaixo, segue alguns exemplos de códigos nessa linguagem:

```
import ioff

int main() =
(
          fint a, b
          fprint("A soma dos valores e: ", a+b)
)

int multiplicacao(int num1, int num2)
(
          fprint("A multiplicação dos números e: ", num1*num2)
)
```