

UNIVERSIDADE FEDERAL DE OURO PRETO
DEPARTAMENTO DE COMPUTAÇÃO
CONSTRUÇÃO DE COMPILADORES I

Trabalho Prático 3
Interpretador da linguagem *Torben*

Ouro Preto, 2018
Aluna: Marina de Souza Mendes (15.1.5978)
Professor: Dr. José Romildo Malaquias

1. A linguagem *Torben*

A linguagem *Torben* é uma linguagem de programação que foi desenvolvida por Torben Ægidius Mogensen para fins didáticos. Como qualquer linguagem de programação, ela possui suas próprias regras e sintaxe. Neste trabalho, baseado na linguagem *Torben*, usamos os seguintes operadores e tipos:

- comentários de linha (começam com o caracter #)
- comentários de bloco (bloco delimitado pelas sequências {# e #})
- literais booleanos (LITBOOL)
- literais numéricos (LITDOUBLE e LITINT)
- literais strings (LITSTRING)
- literais void (LITVOID)
- o tipo string (STRING)
- tipos numéricos (DOUBLE e INT)
- tipo booleano (BOOL)
- tipo void (VOID)
- operadores aritméticos: +(adição, PLUS), - (subtração, MINUS), * (multiplicação, TIMES), / (divisão, DIV), % (resto da divisão, MOD) e ^ (potência, POWER)
- operadores relacionais: ~= (diferente, NE), > (maior que, GT), >= (maior ou igual a, GE), < (menor que, LT), <= (menor ou igual a, LE), == (igual a, EQ)
- operadores lógicos: && (e lógico, AND), e || (ou lógico, OR).
- atribuição (=, ASSIGN)
- expressão de atribuição (LET, IN)
- expressão de repetição (WHILE, DO)
- expressão condicional (IF, THEN, ELSE)

- delimitadores e separadores (LPAREN '(', RPAREN ')', COMMA ',')

Os arquivos *parse.cup* e *lexer.jflex* foram adaptados para que fosse possível as análises dos novos tipos. Além disso, foi necessária a adaptação de algumas classes para a criação de funções para a linguagem.

Para melhor visualização:

Símbolos e expressões	Significado
LITBOOL	Literal booleano
LITINT	Literal inteiro
LITDOUBLE	Literal double
LITSTRING	Literal string
LITVOID	Literal void
ID	Variável
ASSIGN	Atribuição
PLUS, MINUS, TIMES, DIV, MOD, POWER	Operadores lógicos
EQ, NE, LT, LE, GT, GE, AND, OR	Operadores relacionais
LPAREN, RPAREN, COMMA	Símbolos auxiliares: (,) e ,
VOID, BOOL, INT, DOUBLE, STRING	Tipos da linguagem
IF, THEN, ELSE WHILE, DO	Expressões condicionais e de repetição
LET, IN	Declaração e escopo de variável

A gramática livre de contexto estendida:

Production rule	Internal representation
$Program \rightarrow Funs$	Programa
$Funs \rightarrow Fun$	Sequência de definições de funções
$Funs \rightarrow Fun Funs$	
$Fun \rightarrow TypeId (TypeIds) = Exp$	Definição de função
$TypeId \rightarrow \text{bool } id$	Tipo e identificador
$TypeId \rightarrow \text{int } id$	
$TypeId \rightarrow \text{string } id$	
$TypeId \rightarrow \text{double } id$	
$TypeId \rightarrow \text{void } id$	
$TypeIds \rightarrow TypeId$	Sequência de tipos e identificadores
$TypeIds \rightarrow TypeId , TypeIds$	
$Exp \rightarrow \text{litdouble}$	Expressões
$Exp \rightarrow \text{litint}$	
$Exp \rightarrow \text{litstring}$	
$Exp \rightarrow \text{litbool}$	
$Exp \rightarrow \text{litvoid}$	
$Exp \rightarrow id$	
$Exp \rightarrow Exp + Exp$	
$Exp \rightarrow Exp - Exp$	
$Exp \rightarrow Exp * Exp$	
$Exp \rightarrow Exp / Exp$	
$Exp \rightarrow Exp \wedge Exp$	
$Exp \rightarrow Exp < Exp$	
$Exp \rightarrow Exp > Exp$	

Production rule	Internal representation
$Exp \rightarrow Exp \leq Exp$	
$Exp \rightarrow Exp \geq Exp$	
$Exp \rightarrow Exp \sim Exp$	
$Exp \rightarrow Exp == Exp$	
$Exp \rightarrow Exp \&\& Exp$	
$Exp \rightarrow Exp Exp$	
$Exp \rightarrow id(Exps)$	
$Exp \rightarrow \text{if } Exp \text{ then } Exp \text{ else } Exp$	Condicional
$Exp \rightarrow \text{let } id = Exp \text{ in } Exp$	
$Exp \rightarrow \text{while } Exp \text{ do } Exp$	Laço de repetição
$Exps \rightarrow Exp$	Sequência de expressões
$Exps \rightarrow Exp , Exps$	

Funções:

Função	Parâmetros	Retorno	Descrição
print	string	void	Imprime um texto
printint	int	void	Imprime um inteiro
printbool	bool	void	Imprime um bool
printdouble	double	void	Imprime um double
round	double	int	Arredonda o double para o inteiro mais próximo
ceil	double	int	Arredonda para cima
floor	double	int	Arredonda para baixo
cast2double	int	double	Converte de int para double
size	string	int	Retorna o tamanho da string
char	int, string	string	Retorna o caracter na posição dada da string
substr	int, int, string	string	Retorna uma substring entre duas posições (inicial e final) da string original
concat	string, string	string	Concatena duas strings
sumdouble	double, double	double	Soma dois números do tipo double

sumint	int, int	int	Soma dois inteiros
return	int	int	Retorna o inteiro