

Primeiro Trabalho de Compiladores

Prof. José de Oliveira Guimarães — DComp — Campus de Sorocaba da UFSCar

13 de setembro de 2021

Faça um compilador para a linguagem Helang¹ dada pela gramática

```
Program ::= VarList { Stat }
VarList ::= { "var" Int Ident ";" }
Stat ::= AssignStat | IfStat | ForStat | PrintStat |
        PrintlnStat | WhileStat
AssignStat ::= Ident "=" Expr ";"
IfStat ::= "if" Expr StatList [
        "else" StatList ]
ForStat ::= "for" Id "in" Expr ".." Expr StatList
PrintStat ::= "print" Expr ";"
PrintlnStat ::= "println" Expr ";"
StatList ::= "{" { Stat } "}"
WhileStat ::= "while" Expr StatList

Expr ::= AndExpr [ "||" AndExpr ]
AndExpr ::= RelExpr [ "&&" RelExpr ]
RelExpr ::= AddExpr [ RelOp AddExpr ]
AddExpr ::= MultExpr { AddOp MultExpr }
MultExpr ::= SimpleExpr { MultOp SimpleExpr }
SimpleExpr ::= Number | '(' Expr ')' | "!" SimpleExpr
                | AddOp SimpleExpr | Ident
RelOp ::= '<' | '<=' | '>' | '>=' | '==' | '!='
AddOp ::= '+' | '-'
MultOp ::= '*' | '/' | '%'
Number ::= ['+' | '-'] Digit { Digit }
```

Caracteres entre aspas são terminais, cadeias entre { e } podem ser repetidas zero ou mais vezes, cadeias entre [e] são opcionais. Ident é composto por letras e números, iniciando por letra. Digit é um dígito e Number é válido em Helang se é válido em Java. A semântica dos operadores relacionais e booleanos é igual aos dos operadores da linguagem C: 0 é considerado

¹He, de Hélio, mais *language*.

falso e diferente de 0, verdadeiro. Variáveis devem ser declaradas antes de serem usadas. Uma variável não pode ser declarada duas vezes. A variável do for não pode ter sido declarada antes e o seu escopo é apenas nas instruções embutidas neste comando. A segunda expressão deve ser maior ou igual à primeira (senão, dê um erro de execução e termine o programa).

Um exemplo de programa seria:

```
var Int s;  
var Int i;  
s = 0;  
for j in 1..100 {  
    s = s + j  
}  
println s;
```

O compilador deve ter as seguintes características:

- (a) deve ser feito em Java e ter uma classe Main;
- (b) o programa deve ser convertido em uma ASA. A classe principal da ASA, Program, deve ter dois métodos:

```
public int run() { ... }  
public void genC() { ... }
```

O método run deve interpretar o programa e genC deve gerar código em C na saída padrão. print e println imprimem na saída padrão.

Para entregar o trabalho no AVA, envie um arquivo zip com os nomes dos dois integrantes do grupo na ordem alfabética, separados por -, e sem acentos. Assim:

Albert Einstein - Isaac Newton.zip

O conteúdo deste zip deve ser algo como

```
ast\  
main\  
lexer\  

```

Utilize a codificação dos arquivos do Windows (Pode-se chamar o programa passando-se um arquivo como parâmetro, como está sendo feito nos compiladores exemplos. Se for usada a opção -run antes do nome do arquivo, este deve ser interpretado. Se for usada a opção -gen, deve-se gerar o código em C. Exemplos:

```
java main.Program -gen "codigoEm.he"  
java main.Program -run "codigoEm.he"
```