

Exercício prático 06

BCC328 - Construção de Compiladores I

Prof. Rodrigo Ribeiro

05-07-2025

Finalização de um interpretador para a linguagem L3

O objetivo desta atividade é finalizar a implementação de um interpretador para a linguagem L3, para a qual adicionamos a verificação de tipos.

A Linguagem L3

A linguagem L3 consiste na extensão de L1 por permitir a definição de variáveis com seus respectivos tipos. A seguir apresentamos a sintaxe, regras do sistema de tipos e alterações necessárias na semântica de L1 para acomodar as novas construções presentes em L3.

Sintaxe da linguagem L3

A sintaxe da linguagem L3 é definida pela seguinte gramática livre de contexto:

$$\begin{aligned} P &\rightarrow SP \mid \lambda \\ S &\rightarrow \textit{let } v : \tau := E; \\ &\quad \mid \textit{read}(E, v); \mid \textit{print}(E); \mid S_1; S_2 \\ T &\mid \textit{Int} \mid \textit{Bool} \mid \textit{String} \\ E &\rightarrow n \mid v \mid s \mid b \mid E + E \mid E - E \mid E * E \\ &\quad \mid E < E \mid E = E \mid E / E \mid E \&\& E \mid ! E \\ &\quad \mid \textit{strcat}(E, E) \mid \textit{strsize}(E) \mid \textit{i2s}(E) \mid \textit{i2b}(E) \\ &\quad \mid \textit{b2s}(E) \mid \textit{b2i}(E) \mid \textit{s2i}(E) \mid \textit{s2b}(E) \end{aligned}$$

A gramática é formada por quatro variáveis: P , T , S e E ; e pelos seguintes tokens (símbolos do alfabeto):

- \textit{let} : inicia a declaração de uma variável.
- v : representam identificadores. O token de identificador segue as regras usuais presentes em linguagens de programação: um identificador começa com uma letra seguida de uma sequência de zero ou mais dígitos ou letras.
- n : representam constantes numéricas. No momento, vamos suportar apenas números inteiros (tanto positivos, quanto negativos).
- s : representam literais de strings. A linguagem L3 utiliza aspas duplas para delimitar literais de string.

- Operadores aritméticos (+,*,⁻ e /) , relacionais (< e =), lógicos (! e &&), operadores sobre strings (strcat e strsize) e operações para conversão entre tipos (i2b, i2s, b2s, b2i, s2i e s2b).

A sintaxe abstrata de L3 é representada pelos seguintes tipos de dados:

```
data L3
  = L3 [S3]

data Ty
  = TString | TInt | TBool

data S3
  = SLet Var Ty E3
  | SAssign Var E3
  | SRead E3 Var
  | SPrint E3
```

Adicionalmente, a sintaxe de expressões é dada pelo seguinte tipo:

```
data E3
  = EValue Value
  -- initially, Nothing, after type checking
  -- we include its type.
  | EVar Var (Maybe Ty)
  -- arithmetic operators
  | EAdd E3 E3
  | EMult E3 E3
  | EMinus E3 E3
  | EDiv E3 E3
  -- Relational operators
  | ELt E3 E3
  | EEq E3 E3
  -- Boolean operators
  | EAnd E3 E3
  | ENot E3
  -- string operators
  | ECat E3 E3
  | ESize E3
  -- type coercion
  | EI2S E3
  | EI2B E3
  | ES2I E3
  | ES2B E3
  | EB2S E3
  | EB2I E3
```

O tipo L3 representa a variável P, S3 denota a variável S e E3 representa a variável E da gramática de L3. Por sua vez, o tipo Ty representa a variável T. A ocorrência de uma variável em uma expressão possui um valor de tipo `Maybe Ty`. Durante a análise sintática, as variáveis possuem o valor `Nothing` para este componente e depois da verificação de tipos, este é substituído por `Just t`, em que `t` é o tipo associado a esta variável no contexto.

Semântica de L3

A semântica de L3 é exatamente a de L1 com novas regras para lidar com novas operações, que é similar a de outros operadores presentes em L1.

Detalhes da entrega

O que deverá ser implementado

Você deverá implementar:

- Analisador léxico para L3.
- Analisador sintático para L3.
- Interpretador para L3.

A seguir, detalharemos a estrutura pré-definida do projeto para L3. A primeira função `lexerOnly` deve realizar a análise léxica sobre o arquivo de entrada e imprimir os tokens encontrados, como feito para a implementação de L1 e L2, em exercícios anteriores.

```
lexerOnly :: FilePath -> IO ()  
lexerOnly file = error "Not implemented!"
```

A segunda função, `parserOnly`, deve realizar a análise léxica e sintática sobre o arquivo de entrada e imprimir a árvore de sintaxe produzida, como feito para a implementação de L1 e L2.

```
parserOnly :: FilePath -> IO ()  
parserOnly file = error "Not implemented!"
```

A terceira função, `typecheckOnly`, deve realizar a análise léxica, sintática e verificação de tipos sobre o arquivo de entrada, imprimindo a árvore com as anotações de tipos em cada ocorrência de variável no programa.

```
typecheckOnly :: FilePath -> IO ()  
typecheckOnly file = error "Not implemented!"
```

Finalmente, a última função, `interpret`, deve realizar a interpretação do programa contido no arquivo fonte fornecido. Para isso, você deverá executar a análise léxica, sintática, verificação de tipos e executar o programa representado pela árvore produzida por todas as etapas de análise de L3.

```
interpret :: FilePath -> IO ()  
interpret file = error "Not implemented!"
```

todas essas funções estão presentes no arquivo `src/L3/L3.hs`, que é o arquivo principal para implementações da linguagem L3. A implementação da árvore sintática para programas L3 está presente no arquivo `Syntax.hs` na pasta `L3.Frontend`.

Adicione novos módulos ou funções a módulos existentes como julgar necessário.

Como será feita a entrega

- As entregas serão feitas utilizando a plataforma Github classroom.