

Enunciado Laboratório 03 de Compiladores.  
Entrega 14/06/2023

**Você deve:**

- **Fazer o trabalho individualmente;**
- **Enviar um arquivo .zip contendo todos os arquivos necessários para rodar o analisador semântico em cima da fatia definida da sua gramática;**
- **Realizar os envios SOMENTE pelo Classroom, até 14/06/2024.**

Adapte o programa anexo para que o programa final faça a análise semântica de uma função na sua linguagem. Seu programa final deve receber um arquivo de programa escrito na sua linguagem sorteada, gerar um objeto em memória que descreve a função (tal como pode ser visto no exemplo em src/src-gram5/Funcao.hpp). Este objeto deverá ser tratado pela função `calcula_ultimo_valor`. Esta função deve simular uma execução da função e retornar o último valor atribuído no último comando. Note que o exemplo do programa anexo é mais simples do que o necessário para sua tarefa, não tendo tipagem (ele considera todo objeto do tipo inteiro).

Para simplificar esta fase, seu programa deve se restringir a um subconjunto da sua linguagem. Este subconjunto deve compreender as seguintes estruturas:

- Declaração de função, com parâmetros e corpo de função
- Declaração de variáveis, com o tipos restritos **a pelo menos**:
  - **um tipo inteiro**
  - **um tipo ponto flutuante**
  - **um tipo booleano ou seu substituto na linguagem.**
- Bloco de comandos, com possível declaração de novas variáveis, caso sua linguagem permita.
- Comando de atribuição
- Expressões com operadores aritméticos, incluindo `+` `-` `*` `/` `%` , variáveis, números inteiros, números ponto flutuante e booleanos.
- Expressões com operadores relacionais, incluindo `<` e `==` (menor e igual), podendo incluir outros
- Expressões com operadores lógicos, incluindo `&&`, `||` e `!` (e, ou e não).
- IF-THEN-ELSE
- WHILE

Seu analisador deve apresentar mensagem de erros se os tipos dos operandos de algum operador for incompatível. Por exemplo, deve haver um erro se for feito `!` ponto flutuante). Também deve haver erro para `(booleano < inteiro)` e para `(inteiro || ponto flutuante)`, caso sua linguagem o faça.

No final, você deve testar o seu analisador passando exemplos de listas de parâmetros para sua função e imprimindo o último valor atribuído. Se o valor a ser impresso for booleano, seu programa deve imprimir `"true"` ou `"false"`. Se o valor for ponto flutuante, ele deve imprimir o número com 2 casas decimais. Se o número for inteiro, ele deve ser impresso da forma usual (decimal).

Note que foi feito um exemplo de conversão para objeto do tipo Funcao a partir de uma linguagem restrita a inteiros descrita no arquivo gramatica.conf. Você deve fazer a sua conversão a partir da declaração de uma função na sua linguagem, com as restrições acima.

**OBS.** Note que, no modelo anexo, eu fiz um parser LR2. A adaptação para ele pode ser entendido lendo-se o arquivo tabela\_lri.conf que tem transições especiais que são LR2, com sintaxe EspecLR2, seguido de :, seguido de múltiplas transições no formato símbolo sublinhado ação, separados por |.