

Compiladores – 2019/1

Trabalho Prático T4

30 de maio de 2019

1 Objetivo

O objetivo deste trabalho é a elaboração de um construtor de ASTs (*Abstract Syntax Trees* – árvores de sintaxe abstrata), utilizadas como representação intermediária do código de entrada para o compilador. Este trabalho conclui o desenvolvimento do *front-end* do compilador para a linguagem C-Minus.

2 Descrição do Problema

A implementação desse trabalho envolve a seguinte atividade principal: escrever código que constrói a representação intermediária do programa de entrada como uma AST. É recomendável dividir o desenvolvimento do trabalho nas seguintes etapas:

- Implemente uma estrutura de dados em árvore que seja adequada para uso como uma AST.
- Modifique a gramática do trabalho 3 para incorporar ações semânticas que constroem a AST do programa de entrada.
Obs. 1: Use os arquivos `.dot` e `.pdf` disponibilizados pelo professor como referência para verificar a construção da sua AST.
Obs. 2: A sua AST não precisa ficar idêntica à do professor. Você é livre para construir a sua estrutura como preferir; os exemplos disponibilizados são apenas referências.
- Integre as funcionalidades do código que manipulam as tabelas de símbolos e de literais com a construção da AST. Com isso você terá concluído o desenvolvimento do *front-end* para a linguagem C-Minus.

Exibição de mensagens: As mensagens de erros léxicos, sintáticos e semânticos são as mesmas dos trabalhos anteriores e devem continuar sendo exibidas como antes.

Se o programa estiver correto, o seu compilador deve exibir uma mensagem indicando que o programa foi aceito, como abaixo:

```
$ ./trab4 < program.cm  
PARSE SUCCESSFUL!
```

ATENÇÃO: os arquivos das ASTs disponibilizados pelo professor servem apenas para auxiliar o desenvolvimento do trabalho. As saídas que devem ser exibidas no terminal são AS MESMAS que as do Trabalho 3. O seu *front-end* deve construir a AST do programa de entrada mas NÃO DEVE exibi-la na tela.

3 Implementando e Testando o Trabalho

As convenções léxicas, sintáticas e semânticas da linguagem C-Minus já foram apresentadas nas especificações dos Trabalhos 1, 2 e 3, respectivamente. Você deve integrar os analisadores léxico, sintático e semântico desenvolvidos anteriormente para compor o *front-end* deste trabalho.

Observações importantes:

- O seu *front-end* pode terminar a execução ao encontrar o primeiro erro no programa de entrada.
- Os arquivos de entrada de exemplo são os mesmos do Trabalho 1.
- Valide o seu programa usando os gabaritos disponibilizados na sala da disciplina no AVA.
- Ao submeter o seu trabalho para correção, além dos códigos-fonte envie um arquivo **Makefile** que gera como executável para o seu *front-end* um arquivo de nome **trab4**.

4 Regras para Desenvolvimento e Entrega do Trabalho

- **Data da Entrega:** O trabalho deve ser entregue até às 23:55 h do dia 19/06/2019 (quarta-feira). Não serão aceitos trabalhos após essa data.
- **Grupo:** O trabalho é **individual**.
- **Linguagem de Programação e Ferramentas:** O seu *front-end* deve obrigatoriamente ser desenvolvido na linguagem C.
- **Como entregar:** Pela atividade criada no AVA. Envie um arquivo compactado com todo o seu trabalho.
- **Recomendações:** Modularize o seu código adequadamente. Crie códigos claros e organizados. Utilize um estilo de programação consistente. Comente o seu código extensivamente. Não deixe para começar o trabalho na última hora.

5 Avaliação

- O trabalho vale 2.5 pontos na média parcial do semestre.
- Trabalhos com erros de compilação receberão nota zero. Isso inclui trabalhos com conflitos de *shift-reduce* ou *reduce-reduce*.
- Trabalhos com *memory leak* (vazamento de memória) sofrerão desconto na nota.
- Serão levadas em conta, além da correção da saída do seu programa, a clareza e simplicidade de seu código.
- Caso seja detectado plágio (entre alunos ou da internet), todos os envolvidos receberão nota zero.
- A critério do professor, poderão ser realizadas entrevistas com os alunos, sobre o conteúdo do trabalho entregue. Caso algum aluno seja convocado para uma entrevista, a nota do trabalho será dependente do desempenho na entrevista. (Vide item sobre plágio, acima.)