

Projeto de Implementação de um Compilador para a Linguagem **T++**

Análise Sintática (Trabalho – 2ª parte)

Prof. Rogério Aparecido Gonçalves¹

¹*Universidade Tecnológica Federal do Paraná (UTFPR)*

Departamento de Computação (DACOM)

rogerioag@utfpr.edu.br

10 de abril de 2019

Resumo

Este documento apresenta a especificação da 2ª parte do trabalho de implementação da disciplina. O objetivo nessa etapa é projetar e implementar a fase de *Análise Sintática* do compilador para a linguagem **T++**.

Sumário

1	Análise Sintática	2
1.1	Instruções Gerais	2
1.2	Implementação	2
1.3	Linguagens de programação para a implementação	3
1.4	Descrição da Gramática da Linguagem	3
1.5	Testes	3
1.6	Documentação	3
1.7	Avaliação	4
1.8	Entrega e apresentação	5
1.9	Referências	5

1 Análise Sintática

1.1 Instruções Gerais

1. Faça download do arquivo do modelo de estrutura do trabalho e relatório disponível na página da disciplina no moodle. Descompacte e trabalhe nos arquivos e estrutura fornecida, pois será a mesma estrutura que deverá ser entregue ao final do projeto.
2. Siga a estrutura fornecida para desenvolver o trabalho.
3. O relatório deve ter a descrição do trabalho e dos programas, o código fonte dos programas, uma explicação sobre o funcionamento do programa, o processo de tradução com exemplos de instruções dos três formatos e um exemplo de execução do seu programa reproduzindo a saída gerada.
4. Deverão ser entregues:
 - a) O código fonte dos programas.
 - b) Relatório em **pdf** que pode ser feito no formato do LibreOffice ou no Latex.
5. O projeto deve seguir a estrutura de diretórios e arquivos, disponível no formato. A estrutura do projeto é apresentada na Figura 1.

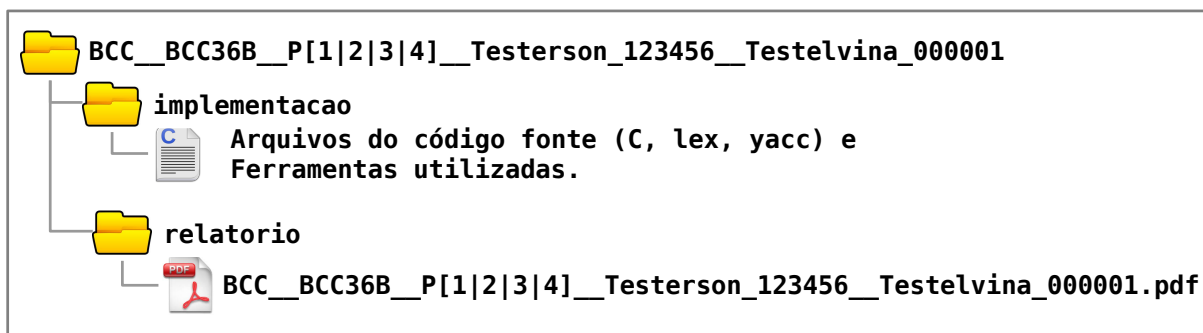


Figura 1: Formato de Entrega

1.2 Implementação

A implementação da análise sintática deverá ser realizando utilizando o mesmo conjunto de ferramentas compatíveis utilizado na análise léxica (apresentados na próxima seção).

A análise sintática deverá ser implementada baseada em uma descrição formal (GLC - Gramática Livre de Contexto) constando na documentação. Os símbolos analisados sintaticamente

poderão ser obtidos por meio de uma função `getToken` obtido da análise léxica ou um método apropriado existente na biblioteca utilizada para o desenvolvimento das duas primeiras análises.

A estrutura da árvore sintática já deverá estar empregada. Para cada derivação da análise sintática, deverá ser observado a necessidade de adicionar um nó na árvore sintática. No final da implementação desta primeira parte do trabalho, uma **árvore sintática abstrata** (AST) deverá estar montada na memória.

1.3 Linguagens de programação para a implementação

Para a implementação do compilador, pode ser utilizado qualquer linguagem de programação¹. É recomendado que seja utilizado uma linguagem que dê suporte à estruturas de dados de alto nível e preferencialmente que exista bibliotecas para a construção da varredura e gramática. Algumas LPs/ferramentas conhecidas são:

- C/C++ - Flex/Bison [3][4]
- Python - PLY (que possui ferramenta Léxica e Sintática) [5]
- Java - JFlex/Jacc [6][7]

Obs.: A implementação de referência será apresentada na linguagem C.

1.4 Descrição da Gramática da Linguagem

Uma especificação da gramática da Linguagem T++ está disponível neste [link](#).

1.5 Testes

Alguns casos de testes estão disponíveis no moodle institucional junto com essa especificação. Serão executados esses testes e outros testes que o professor julgar necessário durante a avaliação desta parte do trabalho.

1.6 Documentação

Durante toda a disciplina o aluno criará uma documentação formal da implementação do compilador para a linguagem. Sendo os relatório com conteúdo acumulativo, isto é, as fases subsequentes irão complementar o conteúdo existente das fases anteriores. A documentação nesta etapa do trabalho deverá conter os seguintes itens:

- A descrição da Gramática no padrão BNF.

¹Que tenha suporte às ferramentas de especificação do sistema de varredura e da gramática e que também tenha suporte ao [LLVM](#).

- Discussão sobre o formato na Análise Sintática realizado pela ferramenta (se é LL(1), LR(1), LALR(1), SLR(1), etc).
- Discussão sobre a implementação e utilização da ferramenta `yacc`.
- Discussão sobre implementação da Árvore Sintática.

Utilize o formato de artigo da SBC [8] para fazer o relatório.

1.7 Avaliação

A avaliação da Análise Sintática será realizada da seguinte forma:

Programa de exemplo `T++` de **entrada** na linha de comando. **Saída** será a validação da análise sintática. Além disto uma **representação da árvore sintática abstrata** (ASA) deverá ser mostrada na saída do programa. A saída da análise sintática por meio de uma ASA é de extrema importância para a continuação da terceira parte do trabalho (Análise Semântica). Deverá ser possível verificar erros `shift-reduce` e `reduce-reduce` (debuggers deverão estar ativos para a realização da avaliação).

É importante que uma classe de análise de erros já esteja preparada. Os erros mais comuns em desenvolvimento de programas estão relacionados com a Análise Sintática.

- Serão avaliados, dentre outros critérios:
 - a) Da implementação:
 - O funcionamento do programa.
 - O capricho e a organização na elaboração do projeto.
 - A corretude da implementação em relação ao que foi pedido no trabalho.
 - A colocação em prática dos conceitos que foram discutidos em sala de aula de forma correta.
 - A qualidade do projeto e da implementação (descrição e elaboração do projeto e o passo a passo da implementação).
 - b) Do relatório:
 - O conteúdo e a forma que foi apresentado, se o formato é o mesmo solicitado.
 - Organização das ideias e do processo de tradução.
 - O capricho na elaboração e na formatação do texto, bem como o conteúdo do texto.
- Não serão avaliados os trabalhos:
 - a) Que cheguem fora do prazo.
 - b) Que não forem feitos nas ferramentas solicitadas.
 - c) Que não estão no formato especificado.
 - d) Que não foram compactados em um só arquivo.
 - e) Que não tiverem identificação (nome e matrícula).
 - f) Que forem cópias de outros trabalhos ou materiais da internet.
 - g) Que não seguirem todas estas instruções.
- Não se esqueça que o trabalho contribui com **1,5** pontos da nota.

1.8 Entrega e apresentação

O trabalho será **individual** e deverá ser entregue até o dia **22/05/2019** no moodle da disciplina em um pacote compactado. A estrutura do projeto com os arquivos do projeto (fonte e relatório) deve ser compactada (zipados) e o arquivo compactado deve ser enviado pelo moodle utilizando a opção de submissão “Trabalho 2a. parte - Análise Sintática”, o nome do arquivo compactado deve seguir o padrão de nomes do formato.

Deverá ser especificado na entrega o mecanismo de execução do *parser* para a realização da correção.

1.9 Referências

- [1] LOUDEN, Kenneth C. Compiladores: princípios e práticas. São Paulo, SP: Thomson, c2004. xiv, 569 p. ISBN 8522104220.
- [2] <http://web.cecs.pdx.edu/~mpj/jacc/>
- [3] <http://flex.sourceforge.net/>
- [4] <http://www.gnu.org/software/bison/>
- [5] <http://www.dabeaz.com/ply/>
- [6] <http://jflex.de/>
- [7] <http://web.cecs.pdx.edu/~mpj/jacc/>
- [8] Formato para publicação de artigos da SBC