

Trabalho Prático II – Análise Sintática

Descrição do trabalho

Nesta etapa, você deverá implementar um analisador sintático descendente (top-down) para a linguagem *PortuGolo*, cuja descrição encontra-se no enunciado do Trabalho Prático I.

Seu analisador sintático deverá ser um analisador de uma única passada. Dessa forma, ele deverá interagir com o analisador léxico para obter os tokens do arquivo-fonte. Você deve implementar seu analisador sintático utilizando o algoritmo de Parser Preditivo Recursivo (Tabela Preditiva e Procedimentos para cada Não-terminal) ou o algoritmo de Parser Preditivo Não-Recursivo (Tabela Preditiva e Pilha).

O analisador sintático deverá reportar o primeiro erro ocorrido no programa-fonte e abortar a compilação. O analisador deverá informar qual o erro encontrado e sua localização no arquivo-fonte.

Para implementar o analisador sintático, você deverá modificar a estrutura gramatical da linguagem, ou seja, você deverá adequá-la às restrições de precedência de operadores, eliminar a recursividade à esquerda e fatorar a gramática, ou seja, essa gramática não é LL(1). Portanto, você deverá verificar as regras que infringem as restrições das gramáticas LL(1) e adaptá-las para torná-la uma gramática LL(1). Antes disso, você deve também tratar as regras gramaticais descritas por expressões regulares, ou seja, um trecho com * é o fecho de Kleene (de 0 até várias produções), um trecho sob + é o fecho de Kleene positivo (de 1 até várias produções) e trechos sob o símbolo ? ocorrem 1 vez ou não ocorrem.

Cronograma e Valor

O trabalho vale 30 pontos no total. Ele deverá ser entregue por etapas, sendo a segunda etapa correspondendo ao Analisador Sintático, conforme consta na tabela abaixo.

Etapas	Data de entrega	Valor	Multa por atraso
Analisador Léxico	23/09/2018	10 pontos	2pts/dia
Analisador Sintático	11/11/2018	10 pontos	2pts/dia
Analisador Semântico	A definir	10 pontos	2pts/dia

O que fazer?

1. Tratar a ambiguidade da gramática referente à precedência de operadores para o não-terminal Op;
2. Tratar os trechos descritos por expressões regulares da gramática formal;
3. Eliminar a recursividade à esquerda e fatorar a gramática em: ListaRotina; ListaCmd; Expressao; ListaID; CmdSe.
4. Implementar os algoritmos de Parser Preditivo Recursivo ou Não-Recursivo;

O que entregar?

1. A nova versão da gramática;
2. Apresentar o cálculo do FIRST, FOLLOW e Tabela Preditiva.
3. Programa com todos os arquivos-fonte;

Regras:

O trabalho deverá ser entregue por e-mail para gustavo.fernandes@prof.unibh.br com o assunto TP2_COMP_CC2018_2.

O trabalho poderá ser realizado individualmente ou em dupla.

Não é permitido o uso de ferramentas para geração do analisador sintático.

A implementação deverá ser realizada em uma das linguagens C, C++, Java, Python ou Ruby.

Trabalhos total ou parcialmente iguais receberão avaliação nula.

Ultrapassados cinco (5) dias, após a data definida para entrega, nenhum trabalho será recebido.

Pontuação extra (3 pontos)

É possível tratar o modo pânico na recuperação de erros sintáticos construindo os métodos `synch()` e `skip()`. Se o número de erros ultrapassar o limite de 5 erros (sintáticos), o programa deverá abortar a análise. Você deverá construir esses dois métodos conforme visto em sala para que o modo pânico mantenha o Parser em execução. Lembre-se que a implementação desses métodos varia conforme o Parser adotado.

Em anexo (pasta: *parser_exemplo*) segue um exemplo de uma Gramática com implementação do Parse Top-Down Recursivo e Modo Pânico. **ATENÇÃO:** a gramática do exemplo não tem relação com a gramática do *PortuGolo*.