Centro Universitário de Belo Horizonte – Uni-BH
Instituto de Engenharia e Tecnologia
Curso: Ciência da Computação
Disciplina: Compiladores
Professor: Gustavo Alves Fernandes

Trabalho Prático II – Análise Sintática

Descrição do trabalho

Nesta etapa, você deverá implementar um analisador sintático descendente (top-down) para a linguagem *Pyscal*, 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 analisar sintático utilizando o algoritmo de Parser Preditivo Recursivo (Procedimentos para cada Não-terminal) ou o algorimto do Parser Preditivo Não-Recursivo (Tabela Preditiva e Pilha).

O analisador sintático deverá reportar possíveis erros ocorridos no programa-fonte. O analisador deverá informar qual o erro encontrado e sua localização no arquivo-fonte. A recuperação de erros a ser adotada deverá ser o Modo Pânico. Contudo, se o número de erros ultrapassar o limite de 5 erros (sintáticos), o programa deverá abortar a análise.

Para implementar o analisador sintático, você deverá modificar a estrutura gramatical da linguagem. 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 tornar a gramática LL(1). Antes disso, você deve tratar operações de gramáticas formais, 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 podem nem ocorrer.

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.

Etapa	**Data de entrega**	**Valor**	**Multa por atraso**
Analisador Léxico e Tabela de símbolos	11/04/2017	10 pontos	2pts/dia
Analisador Sintático	30/05/2017	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 na variável Expressao;
2. Tratar os trechos descritos por expressões regulares da gramática formal;
3. Eliminar a recursividade à esquerdar e fatorar a gramática;
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;
4. Relatório contendo testes realizados com programas (de acordo com a gramática) corretos e errados (no mínimo, 3 certos e 3 errados), e também deverá conter a descrição de cada função/método do Parser.

Para avaliar a correção, o programa deverá exibir os erros sintáticos ocorridos, informando as respectivas linhas e colunas. Além, é claro, de possíveis erros léxicos já tratados pelo Lexer.

Regras:

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 ou Python.

A recuperação de erro deverá ser em Modo Pânico, conforme discutido em sala. Mensagens de erros correspondentes devem ser apresentadas, indicando a linha e coluna da ocorrência do erro.

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(). Você deverá construir esses dois métodos conforme visto em sala para que o modo pânico gere “menos confusão” ao Parser.