Prova Final de Compiladores (IF688) – 2017.1

Prof. Marcelo d’Amorim

Aluno:

1. [2,0 pontos] Identifique quais das tarefas abaixo se refere à análise léxica (L), sintática (S), às duas (LS), ou deixe em branco se não for realizada por nenhuma das duas fases.

( ) Verifica se uma variável foi declarada antes de ser utilizada

( ) Faz parte do front-end de um compilador

( ) Verifica se o número de parâmetros passados em uma chamada a um método está correto

( ) Verifica se o tipo dos parâmetros passados em uma chamada a um método está correto

( ) Verifica se o “else” aparece sempre depois de um “if”

( ) Interage com a tabela de símbolos (lê ou escreve)

( ) É especificada utilizando gramáticas livres de contexto (BNF / EBNF)

( ) Lê o arquivo de entrada e gera os tokens

( ) Cria a abstract syntax tree na memória

( ) Recebe os tokens como entrada

1. [2 pontos] Escreva um parser recursivo descendente para a gramática abaixo:

A ::= Id "(" B ")"

B ::= Id | Id "," B

onde Id ::= [a-ZA-Z]+

Obs. caracteres entre aspas indicam símbolos terminais

1. [2 pontos] Explique o que são dependências de dados e dependências de controle.
2. [2 pontos] Ilustre, com exemplos, três otmizações de código vistas em sala de aula.
3. [2 pontos] Explique vantanges e desvantages de ambientes de execução de um programa usarem coletores de lixo.