1EE Compiladores – 2017.1, Prof. Marcelo d’Amorim

Aluno:

1. [2 pontos] Responda V ou F para as afirmações abaixo:

( ) Um “cross compiler” é um compilador que produz código para uma máquina distinta daquela que ele executa.

( ) O processo de “bootstrapping” é caracterizado pela compilação de uma linguagem utilizando um compilador escrito na própria linguagem.

( ) Nem todos os tokens consumidos por um parser durante o processo de compilação são representados em uma árvore abstrata.

( ) Análise léxica produz árvores sintáticas.

( ) Checagem de tipos pode ser feita estaticamente ou dinamicamente.

( ) A verificação que o número e tipos de parâmetros que ocorrem em uma chamada de função são consistentes com as declaradas na função correspondente é feita durante a análise sintática.

( ) O compilador não gera erros durante a fase de análise léxica.

( ) Recursão à direita pode causar loop infinito e inviabilizar a construção de um parser recursivo descendente.

( ) Gramáticas de atributos podem ser usadas tanto para expressar regras de tipos como para expressar regras de construção dos nós de uma árvore sintática.

( ) O compilador de Java gera código para um interpretador.

1. [2 pontos] Considere a gramática abaixo:

Program ::= Statements “;” ***Programa***

Statements ::= Statement ***SimpleStmt***

| Statement “;” Statements ***MultStmt***

Statement ::= "print" Identifier ***PrintStmt***

| Identifier "=" Expression ***AssignmentStmt***

Expression ::= number ***Literal***

| Expression "+" Number ***AddExpression***

Id ::= “a” | “b” | “c” ***Id***

Number ::= “0” | “1” | “2” | “3” | “4” | “5” | “6” | “7” | “8” | “9” ***Number***

Cada produção da gramática acima está associada a uma estrutura de dados com nome apresentado no canto direito, em fonte itálica. Por exemplo, um programa é representado por new Programa(stmts), onde stmts corresponde a lista de comandos que aparece ao lado direto da única produção da classe gramatical Program. Construa árvore sintática abstrata (AST) para o programa abaixo usando estas estruturas de dados. Apresente tanto a representação gráfica da AST como a representação programática.

**a = 3 + 4 + 2; print (a);**

1. [2 pontos] A partir da gramática acima, elabore uma gramática de atributos com regras semânticas para produção de ASTs. Dica: a aula prática de análise sintática mostrou como isto é feito.
2. [2 pontos] Modifique a gramática acima, se necessário, para descrever um parser recursivo descendente na linguagem de sua preferência.
3. [2 pontos] Explique vantagens e desvantagens da checagem de tipos dinâmica e da checagem de tipos estática