

Perfil de PLC - Processamento de Linguagens e de Conhecimento (MiEI-MEI 2019/20)

1º Teste de GCS – Gramáticas na Compreensão de Software

Data: 06 de Janeiro de 2020
Hora: 09:00

1 Questão teórica (9 valores)

Justifique com clareza e rigor as afirmações seguintes:

- a) Uma Gramática de Atributos permite definir formalmente a sintaxe e a semântica (estática e dinâmica) das linguagens de programação.
- b) A *Geração automática de programas*, como se faz no caso concreto dos compiladores (processadores de linguagens), é possível porque o problema a resolver se pode especificar formalmente e é uma forma excelente de reutilização de programas.
- c) A análise estática de código apresenta um conjunto de técnicas e ferramentas bastante úteis na Compreensão de Programas.

2 Questão prática sobre GAs (5 valores)

Considere a seguinte GA, escrita em notação do ANTLR, para descrever os resultados de um inquérito com N perguntas e M respondentes, sendo cada resposta dada na escala 1 a 3 (0 significa que não respondeu).

```
grammar Inquerito;

question returns [int N, int M]
    : cabec {$question.N=0; $question.M=$cabec.M;}
      ( questao[$cabec.M] {$question.N += 1;} )+ '.'
      { if ($question.N > 20) System.out.println("ERRO:..." + $question.N); }
    ;

cabec returns [int M]
    : 'Local' TEXT 'Data' DATA 'Respondentes' NUM
      { $cabec.M = $NUM.int; }
    ;

questao [int M]
    : 'Questao' ID '(' r1=resp[0] ( ',' r2=resp[$r1.sum] {$r1.sum=$r2.sum;} )* ')'
      { System.out.println("Media das respostas a Questao "+ID.text+" = "+$r1.sum/$questao.M); }
    ;

resp [float sumIN]
returns [float sum]
    : NUM
      { if ($NUM.int > 3) {System.out.println("ERRO:..." + $NUM.int); $resp.sum=-1;}
        else
          {$resp.sum = $resp.sumIN + $NUM.int;} }
    ;

ID      : [a-zA-Z][a-zA-Z0-9]* ;
TEXT    : '\'' ~('\'' )* '\'' ;
DATA    : [0-9][0-9][0-9][0-9]'-'[0-9][0-9]'-'[0-9][0-9] ;
NUM      : ('0'..'9')+ ;
WS      : ( [\t\r\n] ) -> skip ;
```

e responda então às alíneas seguintes

- descreva com detalhe a linguagem definida pela GA acima, referindo-se também à parte léxica dessa linguagem;
- ilustre a explicação anterior com uma frase sintática e semanticamente correta.
- acrescente à GA dada atributos sintetizados ou herdados para: (i) garantir que as respostas a cada questão são exatamente M; e (ii) a média das respostas a cada questão não é calculada se houver qualquer erro.

3 Questão prática sobre Visitors (6 valores)

Considere a seguinte Gramática, escrita em notação do ANTLR, para descrever a construção de um desenho a partir de formas geométricas simples (retângulo, triângulo e círculo) sobre uma tela (virtual) – *canvas* – com medidas da sua largura e altura em *pixel* (*px*). Os retângulos e os triângulos são especificados pela sua largura e altura também em *px* e pelo seu centro sob a forma (x,y) , que refere um píxel específico. O círculo é especificado pelo seu raio e o seu centro.

```
grammar Drawing;

drawing : canvas figures
        ;
canvas  : 'canvas' ':' INT 'x' INT
        ;
figures : '--drawing--' figure ';' (figure ';')*
        ;
figure  : rectangle | triangle | circle
        ;
rectangle : 'rec' INT 'x' INT '@' center
          ;
triangle : 'tri' INT 'x' INT '@' center
          ;
circle   : 'cir' INT '@' center
          ;
center   : '(' INT ',' INT ')'
          ;
INT      : [0-9]+ ;
WS       : [ \r\n\t] -> skip ;
```

O seguinte exemplo é uma frase simples desta gramática:

```
canvas: 1024 x 1024
--drawing--
rec 10 x 10 @ (20,21); cir 10 @ (21,21); tri 5 x 10 @ (15,50);
```

Sabendo que o ANTLR gera o seguinte visitor base para a gramática:

```
public class DrawingBaseVisitor<T> extends AbstractParseTreeVisitor<T> implements DrawingVisitor<T> {
    @Override public T visitDrawing(DrawingParser.DrawingContext ctx) { return visitChildren(ctx); }
    @Override public T visitCanvas(DrawingParser.CanvasContext ctx) { return visitChildren(ctx); }
    @Override public T visitFigures(DrawingParser.FiguresContext ctx) { return visitChildren(ctx); }
    @Override public T visitFigure(DrawingParser.FigureContext ctx) { return visitChildren(ctx); }
    @Override public T visitRectangle(DrawingParser.RectangleContext ctx) { return visitChildren(ctx); }
    @Override public T visitTriangle(DrawingParser.TriangleContext ctx) { return visitChildren(ctx); }
    @Override public T visitCircle(DrawingParser.CircleContext ctx) { return visitChildren(ctx); }
    @Override public T visitCenter(DrawingParser.PointContext ctx) { return visitChildren(ctx); }
}
```

responda às alíneas seguintes:

- crie um visitor que verifique que todo o desenho se encontra completamente contido nos limites da tela.
- crie um visitor que calcula o centro gravitacional (CG) do desenho, isto é, o centro definido pelos centros do conjunto de todas as formas geométricas. CG é um centro fictício (X,Y) , onde X a média das abcissas de todos os centros e Y é a média das ordenadas de todos os centros. Para o exemplo acima, o resultado deveria ser: $X = (20 + 21 + 15)/3 = 18.6$ e $Y = (21 + 21 + 50)/3 = 30.6$, ou seja, o centro $(18.6, 30.6)$.