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

Exame de Recurso de
GCS – Gramáticas na Compreensão de Software

Data: 29 de Janeiro de 2020
Hora: 09:00

1 Questão teórica (9 valores)

Leia com atenção as 3 questões abaixo e responda a cada uma com clareza e rigor.

a) Fale sobre o conceito de *IDE, Ambiente Integrado de Desenvolvimento* dizendo para que serve, quem o usa, o que dele se espera, qual a relação com o compilador e dê exemplos de instâncias.

b) Discorra sobre o conceito de *Qualidade de uma Linguagem de Programação* referindo-se, entre outras coisas, às características usadas na sua avaliação, explicando em particular o que entende por *expressividade*, *consistência*, *escalabilidade* e *modularidade*.

c) Diga justificando se a afirmação seguinte é verdadeira ou falsa
"*Gramáticas de Atributos ou Visitors são duas abordagens práticas na resolução de problemas em processamento de linguagens; ambas apresentam vantagens e desvantagens dependendo do tipo de problema em questão.*"

2 Questão prática sobre GAs (5 valores)

Considere a seguinte GA, escrita em notação do AnTLR, para descrever um conhecido jogo de cartas.

```
grammar Jogo;

canasta returns [int qt, boolean limpa, int pontos]
        : c1=carta[0] {$canasta.qt=$c1.cnt; $canasta.limpa=(!$c1.joker);}
        (c2=carta[$canasta.qt] {$canasta.qt=$c2.cnt; $canasta.limpa=($canasta.limpa && (!$c2.joker));})*
          '.'
          { if ($canasta.qt>=7) { if ($canasta.limpa) $canasta.pontos=400; else $canasta.pontos=200; }
            else { System.out.println("ENGANO:Nao e uma canasta..."); $canasta.pontos=0; }
            System.out.println("A canasta feita vale: " + $canasta.pontos); }
        ;
carta   [int N]
        returns [int cnt, boolean joker]
        : '(' NUM ',' NAIPE ')' {$carta.cnt = $carta.N+1; $carta.joker = false;}
        | 'Joker' {$carta.cnt = $carta.N+1; $carta.joker = true;}
        ;

NUM     : [0-9ARDV] ;
NAIPE   : 'esp' | 'copas' | 'ouros' | 'paus' ;
WS      :   ( [ \t\r\n] ) -> skip ;
```

e responda então às alíneas seguintes

a) descreva com detalhe toda a linguagem definida pela GA acima.

b) ilustre a explicação anterior com uma frase errada (diga onde está o erro) e ainda uma frase sintática e semanticamente correta.

c) acrescente à GA dada atributos sintetizados ou herdados para dobrar a pontuação final se todas as cartas forem do mesmo naipe.

d) altere a GA dada de modo a permitir que o jogo seja formado por uma ou mais canastas, calculando a pontuação final como sendo a soma do valor dessas canastas.

3 Questão prática sobre Visitors (6 valores)

Considere a seguinte Gramática, escrita em notação do AnTLR, para descrever a divulgação e venda de bilhetes de eventos. Cada evento tem uma designação, uma data e uma sala de espetáculos associados, e pode apresentar ou não reservas já efetuadas. As salas de espetáculos para além de uma designação, apresentam também os seus lugares disponíveis, divididos por categorias, acrescido do preço de cada lugar nessa categoria. Cada reserva apresenta o nome do indivíduo que a efetuou, assim como a informação do lugar reservado (por exemplo A 10, significa, cadeira 10 na categoria A).

```
grammar EventBooking;
start   : event+ ;
event   : 'Name' ':' STR ',' date room bookings? ;
date    : INT'/'INT'/'INT '-' INT':'INT;
room    : 'Room' ':' STR seats;
seats   : '--PRICING--' seat (',' seat)*;
seat    : 'Category' ID '-' INT '@' INT;
bookings: '--BOOKINGS--' booking (',' booking)*;
booking : STR '-' ID INT;

INT: [0-9]+;
STR : '"'~('"')*'"';
ID : [A-Z];
WS: [ \n\t] -> skip;
```

O seguinte exemplo é uma frase simples desta gramática:

```
Name: "Ah, Os Dias Felizes", 2020/03/25 - 16:00
Room: "Teatro Sá da Bandeira"
--PRICING--
Category A - 50 @ 25, Category B - 100 @ 20, Category C - 200 @ 15
--BOOKINGS--
"Manuel José" - A 20, "Manuel José" - A 21, "Maria Duarte" - C 200

Name: "Romeu e Julieta", 2020/10/10 - 21:30
Room: "Teatro Dona Maria II"
--PRICING--
Category A - 100 @ 45, Category B - 100 @ 35,
Category C - 200 @ 25, Category D - 250 @ 15
```

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

```
public class EventBookingBaseVisitor<T> extends AbstractParseTreeVisitor<T> implements EventBookingVisitor
@Override public T visitStart(EventBookingParser.StartContext ctx) { return visitChildren(ctx); }
@Override public T visitEvent(EventBookingParser.EventContext ctx) { return visitChildren(ctx); }
@Override public T visitDate(EventBookingParser.DateContext ctx) { return visitChildren(ctx); }
@Override public T visitRoom(EventBookingParser.RoomContext ctx) { return visitChildren(ctx); }
@Override public T visitSeats(EventBookingParser.SeatsContext ctx) { return visitChildren(ctx); }
@Override public T visitSeat(EventBookingParser.SeatContext ctx) { return visitChildren(ctx); }
@Override public T visitBookings(EventBookingParser.BookingsContext ctx) { return visitChildren(ctx); }
@Override public T visitBooking(EventBookingParser.BookingContext ctx) { return visitChildren(ctx); }
}
```

responda às alíneas seguintes:

a) crie um visitor que calcule a receita de todos os eventos descritos.

b) crie um visitor que calcula a percentagem de atendimento para cada um dos eventos.