



MATA56 - Paradigmas de Linguagens de Programação

Prova 1

Aluno:

1. Você tem 110 minutos para fazer esta prova.
2. Você pode consultar qualquer material que estiver com você, mas não pode acessar a Internet ou consultar qualquer fonte externa, inclusive o seu colega ☺.
3. Você pode usar computador, mas não pode usar nenhum mecanismo de comunicação de dados do computador.
4. A codificação Cn, usada no início dos enunciados, indica quantas cláusulas eu usei para solucionar uma dada questão. Por exemplo: C2 indica que eu resolvi a questão utilizando apenas duas cláusulas.
5. Você só pode usar cláusulas pré-definidas quando estas forem explicitamente mencionadas nos enunciados das questões, qualquer outra cláusula auxiliar deve ser codificada por você.
6. Assine a prova e as folhas de resposta logo no início do exame (i.e., AGORA mesmo).
7. Coloque a caneta de lado assim que o professor anunciar o final da prova.

Questão 1 (2 pontos) - C2: Escreva em PROLOG as cláusulas `liste(+N, ?L)` que lista N números em ordem decrescente de N para 1. Por exemplo:

```
?- liste(5, L).  
   L = [5,4,3,2,1].
```

Questão 2 (2 pontos) – C2: Escreva em PROLOG as cláusulas para testar se dois conjuntos C1 e C2 representados por listas são iguais (a ordem dos elementos não importa). As cláusulas devem retornar corretamente metas do tipo: `iguais(?C1, ?C2)`. Por exemplo:

```
?- iguais([1,2,3], [2,1,3]).  
true.
```

Você pode assumir a existência das cláusulas `member(+elem, +list)` e `delete(+list, +elem, -list)` do PROLOG.

Questão 3 (2 pontos) – C6: Escreva em PROLOG as cláusulas `mescleLL(+LL, ?L)`, que recebe uma lista de listas ordenadas (não vazia) e as mescla em uma única lista ordenada. Por exemplo:

```
?- mescleLL([[4,7],[1,3,6],[2,5,8]], L).  
X = [1,2,3,4,5,6,7,8].
```

Dica: escreva a rotina `mescle(+Lista1, +Lista2, ?ListaR)` como função auxiliar.

Questão 4 (2 pontos) – C3 (C4 é mais fácil): Escreva em PROLOG a cláusula `decodifica(+LL, ?L)` que produz uma lista L a partir da decodificação de LL. LL é uma lista de listas, onde cada lista Li de LL contém um par `[N, A]` que diz quantas vezes o átomo A vai se repetir em L. Por exemplo:

```
?- decodifica([[3,a],[2,b],[2,a],[1,d]], L).  
L = [a, a, a, b, b, a, a, d].
```

Você pode assumir a existência da cláusula `append/3` do PROLOG.

Questão 5 (2 pontos) – C4: Define-se uma lista genérica como uma lista em que cada elemento pode ser um átomo ou uma lista genérica. Escreva em PROLOG a cláusula `profundidade(+LG, ?N)`, que recebe uma lista genérica LG e retorna sua profundidade. Por exemplo:

```
?- profundidade([a,b,c,[d],e,f], N).  
N = 2.  
?- profundidade([a,b,[d]], [[e,[f]]], f, P).  
P = 4.
```

Assuma a existência das cláusulas `is_list(L)` e `atom(A)` e da expressão `max(N1, N2)` do Prolog.