

Prog.Lógica - AARE - Avaliação Parcial da Aula 4

28 Pontos

Regras:

- Apresentar exemplos de entrada/saída. Quando fizer sentido, entrada que gere **true** e outras que gerem **false**
 - Cópia (Internet/outros) **gera nota zero na avaliação**
1. (7 pts) Você deve implementar as regras necessárias para resolver o problema Jumping Frog disponível no site Uri Online Judge. A “entrada” do problema será composta duas variáveis: **N**, o tamanho do salto e **L** uma lista com os tamanhos dos canos (no site existe uma variável *a* mais, que não será necessária). O Prolog produzirá true/false...não precisa imprimir na tela.
 2. (7 pts) O problema da **Soma Par** consiste em determinar se uma lista possui 2 elementos distintos cuja soma seja igual a um determinado valor. Por exemplo SomaPar([1, 3, 4, 1], 5) é verdade pois 4+1=5 e SomaPar([1, 3, 4, 1], 6) é falso. Desenvolva as regras para determinar se uma determinada lista é “SomaPar” para um determinado valor.
 3. (7 pts) O problema da **Soma de Sub conjuntos** (ou Sub Set Sum - SSS) consiste em determinar se uma lista possui um subconjunto cuja soma seja igual a um determinado valor. Por exemplo SSS([1, 3, 4, 1], 6) é verdade pois 1+4+1=6 e SSS([1, 3, 4, 5], 2) é falso. Desenvolva as regras para determinar se uma determinada lista é “SSS” para um determinado valor.
 4. (7 pts) O problema de intercalar 2 listas ordenadas consiste em “juntar as listas” e produzir/verificar uma nova lista ordenada. Por exemplo *intercala*([1, 2, 5, 7], [3, 6], [1, 2, 3, 5, 6, 7]). Desenvolva as regras Prolog para intercalar 2 listas. (Atenção: O prolog possui predicados para concatenar listas e ordenar listas...**não** use estes predicados. Você deve implementar uma regra que iterativamente execute este processo)
 5. (Bonus 10pts) Criar as regras para o Problema da Ponte e Zumbis. A regra deve ser “genérica” e buscar a solução de acordo com o problema não somente “testar” uma solução apresentada