

Início Segunda, 8 Janeiro 2018, 17:03

Estado Teste enviado

Data de submissão: Segunda, 8 Janeiro 2018, 19:02

Tempo gasto 1 hora 58 minutos

Nota **9,30** de um máximo de 20,00 (**47%**)

Pergunta 1

Respondida

Pontuou 2,500 de 2,500

Explique o que faz o seguinte programa em Prolog. O predicado ***between(+Inf,+Sup,-X)*** unifica **X** com valores no intervalo fechado **[Inf,Sup]**.

```
:- use_module(library(lists)).
:- use_module(library(between)).

prog1(N,M,L1,L2) :-
    length(L1,N),
    N1 is N-1, length(L2,N1),
    findall(E,between(1,M,E),LE),
    fill(L1,LE,LE_),
    fill(L2,LE_,_),
    check(L1,L2).

fill([],LEf,LEf).
fill([X|Xs],LE,LEf) :-
    select(X,LE,LE_),
    fill(Xs,LE_,LEf).

check([],[]).
check([A,B|R],[X|Xs]) :-
    A+B == X,
    check([B|R],Xs).
```

O programa em Prolog gera duas listas, L1 e L2, em que L1 tem o tamanho N e L2 tem o tamanho N - 1.

Cada elemento de cada uma das listas tem que estar obrigatoriamente dentro do intervalo [1;M]. O preenchimento destas listas é feito pelo predicado fill.

A soma de cada par de elementos da lista L1 é igual ao elemento da lista L2, de forma a explicar melhor, veremos o seguinte exemplo:

Sendo $N = 4$, $M = 10$.

Uma possível solução seria: $L1 = [1, 2, 4, 5]$; $L2 = [3, 6, 9]$

O predicado check garante que: $1+2 = 3$; $2+4 = 6$; $4+5 = 9$

Comentário:

Pergunta 2

Correto

Pontuou 2,500 de 2,500

Identifique a dimensão do espaço de procura do problema anterior:

Selecione uma opção de resposta:

- ☐ a. $2 * M * N$
- ☐ b. $M^{L1} + M^{L2}$
- ☐ c. $N * M$
- ☐ d. $(2 * N - 1)^M$
- ☐ e. [Não Responder]
- ☒ f. M^{2N-1} ✓

A sua resposta está correta.

Pergunta 3

Respondida

Pontuou 2,300 de 3,000

O Severino verificou que o programa anterior obtinha soluções simétricas, pelo que pensou em reduzir o espaço de procura através da eliminação de simetrias. Resolveu implementar uma versão equivalente usando programação em lógica com restrições (PLR):

```
:- use_module(library(clpfd)).

prog2(N,M,L1,L2) :-
    length(L1,N),
    N1 is N-1, length(L2,N1),
    domain(L1,1,M),
    domain(L2,1,M),
    check(L1,L2),
    labeling([],L1).
```

Esta resolução, incompleta, vai obter soluções que não fazem parte do problema original. Complete-a, corrija-a e concretize a intenção do Severino de eliminar simetrias.

```
prog2(N,M,L1,L2) :-
    length(L1,N),
    N1 is N-1, length(L2,N1),
    domain(L1,1,M),
    domain(L2,1,M),
    all_distinct(L1),
    check(L1,L2),
    labeling([],L1).

check([],[]).
check([A,B | R],[X | Xs]) :-
    A+B #= X,
    check([B | R],Xs).
```

Comentário:

Pergunta 4

Respondida

Pontuou 2,000 de 6,000

Uma escola de ginástica acrobática pretende ter um programa que obtenha, de forma automática, emparelhamentos de alunos para as suas aulas.

Dadas as alturas dos homens e das mulheres presentes na aula em igual número, pretendem-se emparelhamentos em que a diferença de alturas entre o homem e a mulher seja inferior a um delta. O homem nunca poderá ser mais baixo do que a mulher.

Construa um programa em PLR que permita obter estes emparelhamentos. Soluções simétricas devem ser evitadas. O predicado

gym_pairs(+MenHeights,+WomenHeights,+Delta,-Pairs) recebe as alturas dos homens e das mulheres (listas com o mesmo tamanho) e a diferença máxima de alturas; devolve em **Pairs** os emparelhamentos de pessoas, identificadas pelo seu índice, que cumpram as restrições.

Exemplos:

```
| ?- gym_pairs([95,78,67,84],[65,83,75,80],10,Pairs).  
no
```

```
| ?- gym_pairs([95,78,67,84],[65,83,75,86],10,Pairs).  
Pairs = [1-4,2-3,3-1,4-2] ? ;  
no
```

```
| ?- gym_pairs([95,78,67,84],[65,77,75,86],10,Pairs).  
Pairs = [1-4,2-2,3-1,4-3] ? ;  
Pairs = [1-4,2-3,3-1,4-2] ? ;  
no
```

gym_pairs(MenHeights, WomenHeights, Delta, Pairs) :-

length(MenHeights, Length),
length(WomenHeights, Length),
length(MenIndex, Length),
length(WomenIndex, Length),

domain(MenIndex, 1, Length),
domain(WomenIndex, 1, Length),

getIndexs(MenHeights, MenIndex),
getIndexs(WomenHeights, WomenIndex),
checkPairs(MenHeights, WomenHeights, Delta),

keys_and_values(Pairs, MenIndex, WomenIndex),

Comentário:

Restrições mal colocadas e incompletas.

Pairs não pode ser lista de pares (labeling).

Não evita simetrias nem garante unicidade.

Pergunta 5

Respondida

Pontuou 0,000 de 6,000

Uma escola de patinagem artística pretende ter um programa que obtenha, de forma automática, emparelhamentos de alunos para as suas aulas.

Dadas as alturas dos homens e das mulheres presentes na aula, pretendem-se emparelhamentos em que a diferença de alturas entre o homem e a mulher seja inferior a um delta. O homem nunca poderá ser mais baixo do que a mulher.

Por vezes não é possível emparelhar todas as pessoas presentes numa aula. Contudo, é útil saber que pares é possível formar, ficando as pessoas não emparelhadas a assistir à aula. Pode até acontecer que o número de homens e de mulheres na aula sejam diferentes, o que inviabiliza a constituição de pares para todas as pessoas.

Construa um programa em PLR que permita obter o maior número possível de emparelhamentos. O predicado

optimal_skating_pairs(+MenHeights,+WomenHeights,+Delta,-Pairs) recebe as alturas dos homens e das mulheres (listas não necessariamente com o mesmo tamanho) e a diferença máxima de alturas; devolve em **Pairs** o maior número possível de emparelhamentos de pessoas, identificadas pelo seu índice, que cumpram as restrições.

Exemplos:

```
| ?- optimal_skating_pairs([95,78,67,84],[65,83,75,80],10,Pairs).  
Pairs = [2-3,3-1,4-2]
```

```
| ?- optimal_skating_pairs([95,78,67,84,65,90,77],[65,83,75,80],10,Pairs  
)  
Pairs = [4-4,5-1,6-2,7-3]
```

```
| ?- optimal_skating_pairs([65,83,75,80],[95,78,67,84,65,90,77],10,Pairs  
)  
Pairs = [1-5,2-2,3-3,4-7]
```

```
| ?- optimal_skating_pairs([95,78,67,84,65,90,77],[55,83,75,80],10,Pairs  
)  
Pairs = [4-4,6-2,7-3]
```

```
| ?- optimal_skating_pairs([55,83,75,80],[95,78,67,84,65,90,77],10,Pairs  
)  
Pairs = [2-2,3-3,4-7]
```

```
optimal_skating_pairs(MenHeights, WomenHeights, Delta, Pairs) :-  
    length(MenHeights, Length1),  
    length(WomenHeights, Length2),  
    length(MenIndex, Length1),  
    length(WomenIndex, Length2),  
  
    domain(MenIndex, 1, Length1),  
    domain(WomenIndex, 1, Length2),  
  
    getIndexs(MenHeights, MenIndex),  
    getIndexs(WomenHeights, WomenIndex),  
    checkPairs(MenHeights, WomenHeights, Delta),  
  
    keys_and_values(Pairs, MenIndex, WomenIndex),
```

Comentário:

getIndexs sem sentido.

checkPairs sem ligação aos pares

maximize de variável instanciada