

OLPR0001

(26 de março de 2015) – 2a.lista – Sintaxe do MINIZINC

Introdução a Programação por Restrições

Entrega digital, se houver gráficos e figuras, deposite também em separado no  
Dropbox

**Entrega é via Dropbox .. mas terão 10 dias para esta 1a. entrega  
– 3a. feira – 31/março**

Objetivo desta lista é familiarizar-se com a sintaxe do Minizinc, vetores, conjuntos e funções.

## Sumário

1	Vetores	2
2	Conjuntos	3
3	Funções e Predicados	4
4	Considerações Finais:	5

# 1 Vetores

1. Crie um vetor de tamanho 13, com domínio de 1 a 10000. Faça uma (ou mais) restrição tal que a soma de dois elementos adjacentes no vetor forneça o elemento seguinte. Exemplo de uma atribuição inicial válida ao vetor:

1 2 3 5 8 13 . . . .

Lembra algo?

2. Crie um vetor de tamanho 10, com domínio de 1 a 100. Faça uma (ou mais) restrição tal que o vetor final contenha uma sequência crescente, tal que as diferenças entre dois conteúdos subsequentes seja igual a  $k = 2$ . Exemplo de soluções válidas ao vetor resultante:

1 3 5 7 9 11 . . . .

2 4 6 8 10 . . . .

3 5 7 9 11 . . . .

.....

3. Repita o item anterior, exceto que o vetor final contenha uma sequência crescente, com diferenças crescentes entre os números. Exemplo de soluções válidas ao vetor resultante:

1 3 7 12 18 . . . .

2 4 7 11 16 . . . .

.....

## 2 Conjuntos

1. Defina dois conjuntos  $A$  e  $B$  de cardinalidade 10, com domínios inteiros de 1 a 100.
  - (a) Faça uma restrição tal que a união destes dois conjuntos forneça um conjunto final com cardinalidade 15
  - (b) Faça uma restrição tal que a intersecção destes dois conjuntos forneça um conjunto final com cardinalidade 5
2. Dada uma variável  $X$  do tipo conjunto (*set*) com, domínio de inteiros de 1 a 10. Imprima os sub-conjuntos  $X$  com cardinalidade 3. A função cardinalidade e restrição é dada por  $\text{card}(X) = 3$ . Saída esperada:  
 $\{\{1, 2, 3\}, \{1, 2, 4\}, \dots\}$

### 3 Funções e Predicados

Exemplos:

```
function var int: mysquare_F(var int: n) = n*n ;
```

```
predicate mysquare_P(int: n, var int: res) =  
    res = n*n ;
```

a partir disto os mesmos são usados nas restrições do modelo.

Vamos aos exercícios:

1. Dado um vetor bi-dimensional, crie uma função que calcule e retorne a soma de todos elementos desta matriz. Ao fazer esta função, faça uma que imprima os valores da matriz. Teste-a na seção do `output` do Minizinc;
2. Dado dois conjuntos  $A$  e  $B$  de cardinalidade 10, com domínios de 1 a 100, faça uma função que calcule a diferença simétrica entre 2 conjuntos:  $A\Delta B$  (definição: the *symmetric difference* between two sets  $A$  and  $B$ , written  $A\Delta B$ , is the set of all  $x$  such that either  $x \in A$  or  $x \in B$  but not both. In other words,  $A\Delta B := (A \cup B) \setminus (A \cap B)$ )

Ou ainda:  $A\Delta B := (A \setminus B) \cup (B \setminus A)$

3. Considere um domínio  $D_x = \{2, 3, 4\}$ ,  $D_y = \{3, 4, 5\}$  construa funções que faz a interpretação lógica das fórmulas abaixo:

(a)  $\forall x \exists y (x < y)$

(b)  $\exists x \forall y (x < y)$

(c)  $\forall x \exists y (x^2 \neq y)$

Veja, o retorno desta função é um `booleano`. No github do prof há alguns exemplos para idéias.

4. Repita os itens acima usando predicados em Minizinc.

## 4 Considerações Finais:

- ⇒ No fonte a ser entregue, adicione os resultados dentro de comentários.
- ⇒ Faça vários testes. Em geral ninguém faz, mas, é para fazer vários I/O
- ⇒ Assuma e justifique os dados que faltarem (eventualmente pode ocorrer).