Maxima

Daniel Miranda

7 de Julho de 2017

Conjuntos

If

Block

Números Aleatórios

Atividade:Pôquer

Substituições

For

Plan

- Conjuntos
- If
- Block
- Números Aleatórios
- Atividade:Pôquer
- Substituições
- For

Conjuntos

Se um membro é listado mais de uma vez, a simplificação elimina o elemento redundante.

(%i1**)**

```
{d, b, d};
```

(%o1**)** $\{b, d\}$

Conjuntos

- União union
- Intersecção intersection
- cardinalidade cardinality
- produto cartesiano `cartesian_product`
- Conjunto das partes powerset

setify

Constrói um conjunto de elementos de uma lista

```
 a:[1,2,3,4];
powerset(setify(a),3);
```

Plan

Conjuntos

If

Block

Números Aleatórios

Atividade:Pôquer

Substituições

For

If

A instrução if é usada para a execução condicional. A sintaxe é:

```
if condição then expr_1  else expr_2
```

If

```
g(x):=(if x>0 then sin(x) else x^2);
wxplot2d([g(x)], [x,-5,5])$
```

Plan

Conjuntos

If

Block

Números Aleatórios

Atividade:Pôquer

Substituições

For

Block

Blocos são **instruções compostas**, mas também permitem que o usuário utilize variáveis "fictícias"para valores que são locais para o bloco.

```
block ([v_1, ..., v_m], expr_1, ..., expr_n)
```

O primeiro [] no bloco, pode conter uma lista de variáveis e atribuições de variáveis, tais como [a: 3, b, c: []], o que faria com que as três variáveis a, b, e c não se refiram à sua valores globais, mas sim tenham esses valores especiais enquanto o código é executado dentro do bloco, ou funções dentro chamados de dentro do bloco.
Os valores das variáveis fora do bloco são preservados.

```
a:2;
block([a],a:3);
a;
```

```
a:2;
block([a],a:2,a:a+1,a^2);
a;
```

Plan

Conjuntos

If

Block

Números Aleatórios

Atividade:Pôquer

Substituições

For

random

random(x)
Retorna um número pseudo-aleatório. Se x é um número inteiro, randomx) devolve um número inteiro de 0 a x-1, inclusive. Se x é um número de ponto flutuante, random(x) retorna um número não negativo em ponto flutuante menor que x. random reclama com um erro se x não é nem um inteiro nem um float, ou se x não for positivo.

random

```
( %i1)
  random(7)
(%o1)     5
( %i2)
  random(5.5)
(%o2)     4.007866868699341
```

Escolha Aleatória

```
random_element(l):= part(l, 1+random(length(l))
    );
```

Ou:

```
random_element(l):= l[1+random(length(l))];
```

(%i3)

```
random_element([a, b, c]);
```

(%o3) c

Atividade: Dados

Sorteie um dado 100 vezes e faça o histograma.

```
load (descriptive)
s1 : makelist(random(6)+1,i,1,400);
histogram (
          s1,
          nclasses      = 6,
          title         = "saida do dado",
          xlabel        = "",
          ylabel        = "frequencia absoluta",
          fill_color    = grey,
            fill_density = 0.6)
```

Plan

Conjuntos

If

Block

Números Aleatórios

Atividade:Pôquer

Substituições

For

Atividade: Poquer

Gerando as Cartas do Baralho:

```
valores:[2,3,4,5,6,7,8,9,10,J,Q,K,A];
naipe:[ouro,copas,espada,paus];
cartas:create_list([i,j],i,valores,j,naipe);
length(cartas);
```

Sorteando

Faça um programa que sorteie mãos de cartas com e sem substituição.

Sorteando mãos 1

```
random_element(l):= part(l, 1+random(length(l))
    );
mao(x):=[random_element(cartas),random_element(
    cartas),random_element(cartas),
  random_element(cartas),random_element(cartas)
      ];
     cemjogos:makelist(mao(x),i,1,100));
```

Sorteando mãos 2

Removendo as mãos sem cartas repetidas

```
maoss(x):=block( [],
      carta1:random_element(cartas),
      cartaslocal:delete(carta1,cartas),
      carta2:random_element(cartaslocal),
      cartaslocal:delete(carta2,cartaslocal),
      carta3:random_element(cartaslocal),
       cartaslocal:delete(carta3,cartaslocal),
      carta4:random_element(cartaslocal),
       cartaslocal:delete(carta4,cartaslocal),
      carta5:random_element(cartaslocal),
      [carta1,carta2,carta3,carta4,carta5]
      );
         cemjogosss:makelist(maoss(x),i,1,100));
```

Qual a probabilidade de uma mão ter o valete de espada?

```
makelist(member([J,espada],i),i,miljogos);
delete(false,makelist(member([J,espada],i),i,
    miljogos));
length(%)/
length(miljogos),numer;
```

Plan

Conjuntos

If

Block

Números Aleatórios

Atividade:Pôquer

Substituições

For

subst

Funções que desempenham substituição, com crescente nível de sofisticação:
subst (..) sintática, símbolos e sub-expressões completas

```
expr:a+b*c+b*c*d;
subst(b*c=k,expr);
```

(%o0) k+b*c*d+a

letsimp

let Define uma regra de substituição para letsimp;
letsimp Aplica repetidamente a substituição definida por let até que nenhuma alteração seja feita na expressão.

```
expr : a + b*c + b*c*d
let (b*c, k);
letsimp (expr);
```

(%o0) b*c –> k d*k+k+a

Qual a probabilidade de uma mão ter uma carta de ouros?

```
evento:create_list([i,copas],i,valores);
contacopas(k):=apply("+",subst([false=0,true
    =1],create_list(member(j,miljogos[k]),j,
    evento)));
temcopas(k):= if contacopas(k)>0 then 1 else 0;
create_list(temcopas(k),k,1,length(miljogos));
```

Plan

Conjuntos

If

Block

Números Aleatórios

Atividade:Pôquer

Substituições

For

for

A declaração for é usada para executar iteração. Devido à sua grande generalidade a instrução do será descrita em duas partes. Em primeiro lugar a forma usual será dada que é análoga à utilizada em várias outras linguagens de programação; em seguida, as outras funcionalidades serão mencionadas.

for

Há três variantes desta forma que diferem apenas nas suas condições de terminação. Eles são:

```
for variable: initial_value step increment
  thru limit do body

for variable: initial_value step increment
  while condition do body

for variable: initial_value step increment
  unless condition do body
```

```
for a:-3 thru 26 step 7 do display(a)
```

(%o0)

a = - 3

a = 4

a = 11

a = 18

a = 25

```
s: 0
for i: 1 while i <= 10 do s: s+i;
s;
```

(%o0) done 55

Qual a probabilidade de um flush Flush ?(mão com cinco cartas de mesmo naipe).