

Conteúdo

1 Manipulação algébrica	1
2 Resolução de Problemas: Geração e Teste	2

1 Manipulação algébrica

Factorial

Recursivo

```
fac(0,1).
fac(N,F):- N>0 , ! N1 is N-1, fac(N1,F1), F is F1*N.
```

Iterativo

```
faci(N,F):-fac(N,1,F).
fac(0,F,F).
fac(N,F1,F2):- N>0, F is F1*N , N1 is N-1, fac(N1,F,F2).
```

Derivação de Funções de uma Variável

```
deriv(X, X, 1):- !. % variável
deriv(C, X, 0) :- number(C), !. % constante
deriv(F+G, X, DF+DG) :- % soma
deriv(F, X, DF), deriv(G, X, DG).
deriv(F-G, X, DF-DG) :- % subtracção
deriv(F, X, DF), deriv(G, X, DG).
deriv(F*G, X, F*DG + DF*G) :- % produto
deriv(F,X,DF), deriv(G,X,DG).
deriv(F/G, X, (G*DF-F*DG)/(G*G)) :- % divisão
deriv(F,X,DF), deriv(G,X,DG).
```

```
?- deriv(2*x+1, x, D). D = 2*1+0*x+0
```

Simplificações

- `simplifica(Exp1, Exp2)` para efectuar algumas simplificações algébricas, por exemplo:

```
1+2 -> 3      (em geral: operações sobre constantes)
0*X -> 0
X*0 -> 0
1*X -> X
X+X -> 2*X
X*X -> X^2
```

Simplificações

Alguns exemplos:

```
simpl(0*F,0) :- !.  
simpl(F*0,0) :- !.  
simpl(F+0,F1) :- !, simpl(F,F1).  
simpl(0+F,F1) :- !, simpl(F,F1).  
simpl(1*F,F1) :- !, simpl(F,F1).  
simpl(F*1,F1) :- !, simpl(F,F1).  
simpl(F-F,0) :- !.  
simpl(F/F,1) :- F \=0.  
simpl(F^0,1) :- F \= 0.  
simpl(0^C,0) :- C\=0.
```

2 Resolução de Problemas: Geração e Teste

Torres de Hanoi

Predicado `hanoi/1` para resolver o puzzle das Torres de Hanoi.

Objectivo do puzzle é mover todos os discos de uma torre para outra torre, utilizando o menor número possível de movimentos, respeitando as seguintes regras:

- um disco maior não pode ser colocado sobre um disco menor;
- apenas um disco pode ser movido de cada vez;
- um disco deve estar sempre numa das três torres (ou em movimento).

Puzzle lógico

Existem 5 casas de cores diferentes, habitadas por homens de nacionalidades diferentes, cada um dos quais tem um animal de estimação diferente, uma bebida preferida diferente e uma marca de cigarros diferente. E tais que:

- o inglês vive na casa vermelha
- o espanhol tem um cão
- o dono da casa verde bebe café
- o ucraniano bebe chá
- a casa verde fica imediatamente à direita da casa cor de marfim

- o fumador de Winston cria caracóis
- o dono da casa amarela fuma Kools
- o dono da casa do meio bebe leite
- o norueguês mora na primeira casa da esquerda
- o homem que fuma Chesterfields mora na casa ao lado do dono da raposa
- o homem que fuma Kools mora ao lado do dono do cavalo
- o homem que bebe sumo de laranja fuma Lucky Strike
- o japonês fuma Parliaments
- o norueguês vive ao lado da casa azul

Quem é o dono da zebra? Quem prefere água?

Puzzle Lógico

- Soluções: [Casa,Nacionalidade,Animal,Bebida,Cigarro]
- `nac/2, casa/2, etc`
- `direita/3, aolado`
- `pistas/2`
- `questao/3`

Para encontrar a solucao: tentar satisfazer as pistas e a questão

```
solve([]).
solve([C|Cs]):-C, solve(Cs).
```

Puzzle Aritmético

Escrever um predicado `soma/3` para resolver o puzzle

```

DONALD
+GERALD
-----
ROBERT
```

onde cada letra representa uma variável lógica a que se pode atribuir um dígito decimal distinto de forma a que os correspondentes números inteiros (base 10) verifiquem a adição anterior.

Puzzle aritmético

- Representar cada parcela por uma lista tal que a solução seja dada por:

```
?-soma([D,0,N,A,L,D],[G,E,R,A,L,D],[R,0,B,E,R,T]).
```

- soma1/6 atribui um dígito a cada das variáveis e efectua a soma dígito a dígito (com transporte).

```
soma(N1,N2,N):- soma1(N1,N2,N,0,0,[0,1,2,3,4,5,6,7,8,9],_).
```

```
soma1([],[],[],0,0,D,D).
```

```
soma1([D1|N1],[D2|N2],[D|N],C1,C,Dig1,Dig2):-  
    soma1(N1,N2,N,C1,C2,Dig1,Dig0),  
    soma_dig(D1,D2,C2,D,C,Dig0,Dig2).
```

Puzzle aritmético

- Soma dígito a dígito com transporte

```
soma_dig(D1,D2,C1,D,C,Dig1,Dig0):-  
del(D1,Dig1,Dig2),  
del(D2,Dig2,Dig3),  
del(D,Dig3,Dig0),  
S is D1+D2+C1,  
D is S mod 10,  
C is S//10.
```