# Resolução do exame segunda chamada 2000

1)

Começamos por ter as frases:

$\forall Z\ (\neg Z \in a \vee Z \in b)$

$subset(X, Y) \leftarrow (\forall U\ U \in Y \leftarrow U \in X)$

$subset(X, Y) \rightarrow (\forall U\ U \in Y \leftarrow U \in X)$

Destas frases chegamos as seguintes clausulas:

A) $\neg Z \in a \vee Z \in b$

B1) $\neg subset(X, Y) \vee U \in Y \vee \neg U \in X$

B2) $subset(X, Y) \vee \neg c \in Y$

C) $subset(X, Y) \vee c \in X$

onde $c$ é uma constante de Skolem. O que queremos provar é

Q) $\neg subset(a, b)$

Q) com B2) origina D) $\neg c \in b$

Q) com C) origina E) $c \in a$

D) com A) origna F) $\neg c \in a$

E) com F) dá $\Box$.

2)

```
imagem(nil,nil).
imagem(t(N,E,D),t(N,ND,NE)):-
           imagem(E,NE),
           imagem(D,ND).
```

3)

```
muda([],_,[]).
muda([P|L],A,[NP|NL]):-
      compound(P),
      P =.. [X1,_|Resto],
      NP =.. [X1,A|Resto],
      muda(L,A,NL).
muda([P|L],A,[P|Nova]):-
      muda(L,A,Nova).
```

4)

```
unfolding((A:-B),C,A):-
         B = C.
unfolding(C,(A:-B),A):-
         B = C.
unfolding((A:-B),(C:-D),(A:-D)):-
         B = C.
unfolding((A:-B),(C:-D),(C:-B)):-
         A = D.
```

5)

```
solve([]).
solve(Goals) :-
      escolhe(Goals,G,SemG),
      clausula(G,Body)
      append(Body,SemG,Newgoals),
      solve(Newgoals).

escolhe([G|Gs],NG,L):-
      G =.. X,
      length(X,N),
      escolhe2(Gs,G,N,NG),
      append(A,[NG|B],[G|Gs]),
      append(A,B,L).

escolhe2([],G,_,G).
escolhe2([G|Gs],G2,N,NG):-
      G =.. X, length(X,N1),
      N1 < N,
      escolhe2(Gs,G,N1,NG).
escolhe2([G|Gs],G2,N,NG):-
      escolhe2(Gs,G2,N,NG).
```

6)

```
% na vertical....
vertical([[P|_],[P|_],[P|_],[P|_],_],P).
vertical([_,[P|_],[P|_],[P|_],[P|_]],P).
vertical([[_,P|_],[_,P|_],[_,P|_],[_,P|_],_],P).
vertical([_,[_,P|_],[_,P|_],[_,P|_],[_,P|_]],P).
vertical([[_,_,P|_],[_,_,P|_],[_,_,P|_],[_,_,P|_],_],P).
vertical([_,[_,_,P|_],[_,_,P|_],[_,_,P|_],[_,_,P|_]],P).
vertical([[_,_,_,P|_],[_,_,_,P|_],[_,_,_,P|_],[_,_,_,P|_],_],P).
vertical([_,[_,_,_,P|_],[_,_,_,P|_],[_,_,_,P|_],[_,_,_,P|_]],P).
vertical([[_,_,_,_,P|_],[_,_,_,_,P|_],[_,_,_,_,P|_],[_,_,_,_,P|_],_],P).
vertical([_,[_,_,_,_,P|_],[_,_,_,_,P|_],[_,_,_,_,P|_],[_,_,_,_,P|_]],P).
vertical([
         [_,_,_,_,_,P|_],
         [_,_,_,_,_,P|_],
         [_,_,_,_,_,P|_],
         [_,_,_,_,_,P|_],_],P).
vertical([_,[_,_,_,_,_,P|_],
            [_,_,_,_,_,P|_],
            [_,_,_,_,_,P|_],
            [_,_,_,_,_,P|_]],P).
vertical([[_,_,_,_,_,_,P],
          [_,_,_,_,_,_,P],
          [_,_,_,_,_,_,P],
          [_,_,_,_,_,_,P],_],P).
vertical([_,[_,_,_,_,_,_,P],
            [_,_,_,_,_,_,P],
            [_,_,_,_,_,_,P],
            [_,_,_,_,_,_,P]],P).
```

7)

```
A) length([],0).
B) length([X|Xs],N):- length(Xs,N1), N is N + 1.
```

A) + B) dá C) length([X], 1). com Xs/[] e N1/0.

C) + B) dá D) length([X, Y], 2). com Xs/[X] e N1/1.

B) + B) dá

```
length([X,Y|Xs],N):- length(Xs,N1), N is N1 + 2.
```

Agora bastava construir as árvores.

8)

```
forma_quatro(L):-
      forma(L,[]).

forma(L,L):-
      length(L,4).
forma(L,[]):-
      equipa(Nome,_,_,),
      forma(L,[Nome]).
forma(L,Temp):-
      equipa(Nome,G,1,_),
      not mesmo_grupo(Nome,G,Temp),
      forma(L,[Nome|Temp]).
forma(L,Temp):-
      equipa(Nome,_,2,Pts),
      not mais_pontos(Nome,Pts,Temp),
      forma(L,[Nome|Temp]).

mesmo_grupo(Nome,G,[X|_]):-
      equipa(X,G,_,_),
      X =\= Nome.
mesmo_grupo(Nome,G,[_|Xs]):-
      mesmo_grupo(NOme,G,Xs).

mais_pontos(Nome,P,[X|_]):-
      equipa(X,_,_,Pts),
      Nome =\= X,
      Pts > P.
mais_pontos(Nome,P,[_|Xs]):-
      mais_pontos(Nome,P,Xs).
```