Resolução do exame primeira chamada

1. Na formula clausal fica:

1) $\mathsf{r(X,Y)} \ \vee \ \neg\mathsf{r(X,Z)} \ \vee \ \neg\mathsf{r(Z,Y)}$ (transitividade)

2) $\neg\mathsf{r(X,Y)} \ \vee \ \neg\mathsf{r(Y,X)}$ (assimetria)

Q) $\neg\exists\mathsf{X}\ \mathsf{r(X,X)}$ (irreflexividade)

Queremos provar que $1) \cup 2) \models Q)$

Para usar resolução tentamos provar que $1) \cup 2) \ \cup \ \neg Q) \vdash \Box$

Negando Q) fica

Q) $\exists\mathsf{X}\ \mathsf{r(X,X)}$

Skolemizando fica

Q) $\mathsf{r}(\alpha,\alpha)$ onde α é a constante de skolem.

Resolvendo temos:

2) com Q) obtém-se 3) $\neg\mathsf{r}(\alpha,\alpha)$

3) com Q) dá $\Box$

2)

```
% inserir ordenadamente elementos numa arvore binaria
insere(X,nil,tree(X,nil,nil)).
insere(X,tree(Y,L,R),tree(Y,TA,R)) :-
         X < Y,
         insere(X,L,TA).
insere(X,tree(Y,L,R),tree(Y,L,TA)) :-
         insere(X,R,TA).

%query principal
treeconv(Tree,Bin) :- faz([Tree],nil,Bin).

% processa arvore nao binaria
faz([],Tree,Tree).
faz([t(X,Lista)|More],Tree,Nova) :-
         insere(X,Tree,NT),
         faz(Lista,NT,NT2),
         faz(More,NT2,Nova).
```

2.1) Uma simples arvore de prova que deixo ao vosso criterio. O importante e' explicar bem as resolucoes com os respectivos unificadores.

3) i)

```
%assumimos que a regra de computacao e' a que escolhe
%o "mais a esquerda". Mas contamos o numero de vezes
% que Body e' []. Notar que Body fica no inicio da lista

solve(T-T,0).
solve([G|Goals]-T,N) :-
      clausula(G,Body-Goals,Temp),
      solve(Body-T,N1),
      N is Temp + N1.


%clausulas agora sao do tipo
% (notar numero para indicar unitarias)
clausula(p(X),[q(X)|T]-T,0).
clausula(q(a),T-T,1).
```

ii)

```
solve([]).
solve([X is L]) :- % unico goal e nao calculavel
      verifica(L),
      write(X is L,' nao e calculavel').
solve([X is L,G|Goals]) :- % escolher outro goal
      verifica(L),
      solve([G,X is L|Goals]).
solve([X is L|Goals]) :- % calcular a expressao
      X is L,
      solve(Goals).
solve([G|Goals]) :-
      clausula(G,Body),
      append(Body,Goal,Newgoals),
      solve(Newgoals).

% transforma expressao aritmetica
% numa lista de parametros
verifica(L):-
      L=.. Lista,
      conta(Lista).

% identifica variaveis nao instanciadas.
conta([L|Ls]):- var(L).
conta([X|Ls]):- compound(X),
                X =.. Lista,
                conta(Lista),
                conta(Ls).
conta([L|Ls]):- conta(Ls).
```