Exemples d'unificació amb "="

?- X=lleo, X=Y.

X=lleo

Y=lleo

yes

?- llop = lleo.

no

?- llop = llop.

yes

?- 4+2 = 6.

no

?- 6 is 4+2.

yes

?-f(X,a)=f(a,X) ?

?-agrada(juana,X)=
agrada(X,pere) ?

?-f(X,Y)=f(P,P) ?

Exemple

```
poblacio(palma, 300000).
poblacio(manacor, 50000).
poblacio(santa-maria, 4000).
superficie(palma, 10).
superficie(manacor, 6).
superficie(santa-maria, 2).
densitat(X,Y):- poblacio(X, P), superficie(X,S), Y is P/S.
```

Exemple

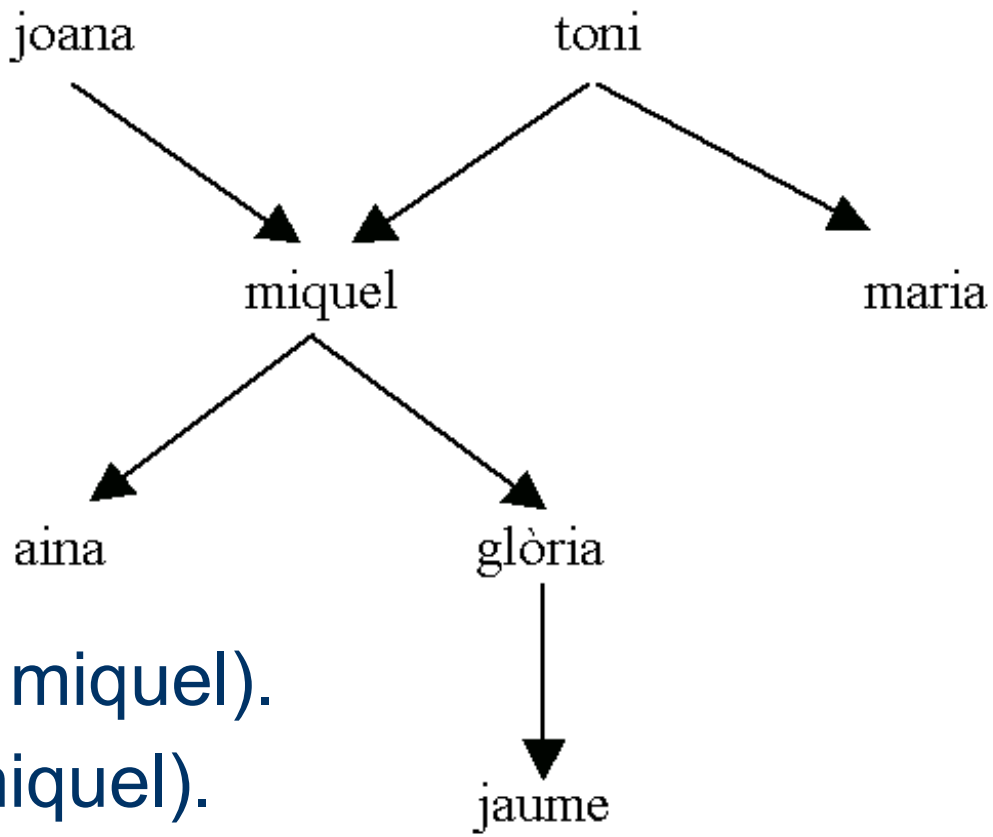
```
horoscop(aries,21,3,21,4).
horoscop(toro,21,4,21,5).
horoscop(geminis,21,5,21,6).
horoscop(cancer,21,6,21,7).
horoscop(leo,21,7,21,8).
horoscop(virgo,21,8,21,9).
horoscop(libra,21,9,21,10).
horoscop(escorpi,21,10,21,11).
horoscop(sagitari,21,11,21,12).
horoscop(capricorni,21,12,21,1).
horoscop(acuari,21,1,21,2).
horoscop(peixos,21,2,21,3).
```

Exemple

```
signe(Dia, Mes, Signe):-
  horoscop(Signe,D1,M1,D2,M2),
  Mes=M1, Dia>=D1.

signe(Dia, Mes, Signe):-
     horoscop(Signe,D1,M1, D2, M2),
     Mes=M2, Dia=<D2.
```

Recursivitat




```
ascendent(joana, miquel).
ascendent(toni, miquel).
ascendent(toni, maria).
ascendent(miquel, aina).
ascendent(miquel, glòria).
ascendent(glòria, jaume).
```

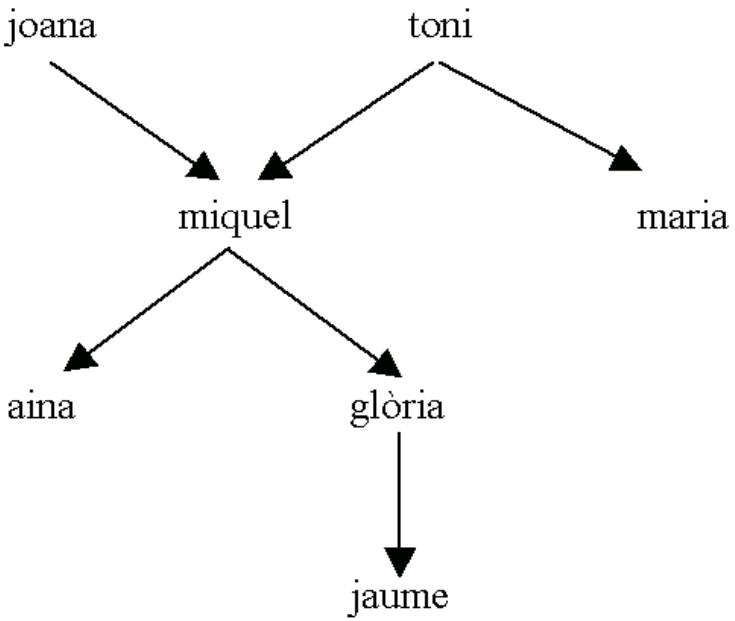
Antecessor

Volem saber si una persona és antecessor (a qualsevol nivell) d'una altra

R1. antecessor(X,Z):-ascendent(X,Z).

R2. antecessor(X,Z):-ascendent(X,Y),
antecessor(Y,Z).

?- Antecessor(toni,g



R1. X=toni, Z=gloria

ascendent(toni,gloria).

no

R2. ascendent(toni,Y), antecessor(Y,gloria).

ascendent(toni,Y)

Y=miquel

antecessor(miquel,gloria).

R1. ascendent(miquel,gloria).

yes

Backtracking o reavaluació

- És juntament amb la unificació, un dels mecanismes fonamentals de PROLOG
- Consisteix en, una vegada una hipòtesi no s'ha pogut demostrar, intentar demostrar-la seguint un altre camí

Backtracking o reavaluació

- PROLOG intenta verificar o satisfer un objectiu cercant a la base de coneixements des del principi
- Si troba un fet o un cap d'una regla que es pugui unificar, fa una marca en aquest punt i instància totes les variables necessàries
- Si és una regla el que ha trobat, llavors també intentarà satisfer els nous objectius (subobjectius)
- Si no troba cap fet o cap de regla unificable, l'objectiu falla i el procés de *backtracking* comença intentant resatisfer l'objectiu inicial

Backtracking o reavaluació

R2. antecessor(X,Z):- ascendent(X,Y),
antecessor(Y,Z).

R1. antecessor(X,Z):- ascendent(X,Z).

?- antecessor(toni, gloria).

R2: X=toni, Z=gloria

ascendent(toni, Y), antecessor(Y, gloria).

ascendent(toni,Y)

Y=miquel

antecessor(miquel, gloria)

antecessor(miquel,gloria)

R2: X=miquel, Z=gloria

ascendent(miquel, Y), antecessor(Y, gloria).

ascendent(miquel,Y)

Y=aina

antecessor(aina, gloria)

R2: X=aina, Z=gloria

ascendent(aina, Y), antecessor(Y, gloria)

ascendent(aina,Y)

no

R1: X=aina, Z=gloria

ascendent(aina, gloria)

no

antecessor(miquel,gloria)

Y=gloria
antecessor(gloria, gloria)
R2: X=gloria, Z=gloria
ascendent(gloria, Y), antecessor(Y, gloria)
ascendent(gloria,Y)
Y=jaume
antecessor(jaume,gloria)
R2:ascendent(jaume,Y), ..
no
R1:acendent(jaume,gloria)
no
R1: X=gloria, Z=gloria
no
no

antecessor(miquel,gloria)

R1: X=miquel, Z=gloria

ascendent(miquel, gloria)

si

Canviant l'ordre dins el cos

R1. antecessor(X,Z):-ascendent(X,Z).
R2. antecessor(X,Z):-antecessor(X,Y), ascendent(Y,Z).

antecessor(toni, gloria).
R1: X=toni, Z=gloria.
ascendent(toni, gloria).
no
R2: antecessor(toni, Y), ascendent(Y, gloria).
antecessor(toni,Y)
R1: ascendent(toni, Y)
Y=miquel
ascendent(miquel, gloria)
si
si

Canviant l'ordre de R1 i R2

R2. antecessor(X,Z):- antecessor(X,Y), ascendent(Y,Z).
R1. antecessor(X,Z):- ascendent(X,Z).

antecessor(toni, gloria).
R2: antecessor(toni, Y), ascendent(Y, gloria).
antecessor(toni,Y)
R2: antecessor(toni, Y'), ascendent(Y', Y).
antecessor(toni,Y')
R2: antecessor(toni, Y''), ascendent(Y'', Y').
antecessor(toni,Y'')

...

Llistes

- Una llista en PROLOG és una seqüència ordenada d'elements tancats entre claudàtors i separats per comes, de qualsevol longitud. Els elements poden ser termes o altres llistes
- Exemples:

  [a,e,i,o,u]

  [mamifer(llop), aracnid(vidua-negre), au(ropit)]

  [[]]

Llistes

- La creació i la descomposició de llistes es fa utilitzant el procés d'unificació i l'operador "|" que separa una llista amb el seu cap i la seva coa

  ?- [X|Y] = [a,e,i,o,u].

  X=a

  Y=[e,i,o,u]

  ?-L=[e,i,o,u],X=[a|L]

  X=[a,e,i,o,u]

Unificació de llistes

[a,b,c] = [c,b,a] → distint ordre

[X] = [a,b,c] → distinta longitud

[X|L] = [a,b,c] → X=a, L=[b,c]

[X,Y,Z|L] =[a,e,i,o,u] → X=a,Y=e,Z=i,L=[o,u]

[X|Y] = [ ] → no !

[X|Y] = [[a,[b,c]],d] → X=[a,[b,c]],Y=[d]

[X|Y] = [a] → X=a,Y=[]

Exercici: Predicats amb llistes

- Pertany un element a una llista
- Afegir dues llistes
- Trobar el darrer element d'una llista
- Invertir una llista
- Esborrar un element d'una llista
- Permutar els elements d'una llista