

TEMA V. Exemples de L.P.A.N.

2721-Llenguatges de
Programació

Ramon Mas

PROLOG

- Són les inicials de PROgramming in LOGic
- Introduït per Alain Colmerauer a principis dels 70
- Dissenyat pel processament d'informació simbòlica

PROLOG

Programar en PROLOG consisteix en:

- Declarar un conjunt de fets coneguts
- Definir regles que descriguin les relacions entre els objectes
- Fer preguntes sobre els objectes i les seves relacions

PROLOG

- Programar en PROLOG consisteix en dir-li el “què” és veritat i demanar-li que ho demostrï i tregui conclusions
- Als altres llenguatges es diu el què, “com” i quan ha de fer l'ordenador

PROLOG

- Els coneixements es descriuen amb les clàusules de Horn, que es caracteritzen per tenir com a màxim un terme a la part de conclusió:

$$p \text{ :- } q_1, q_2, \dots, q_n$$

Conceptes bàsics: les constants

- S'utilitzen per donar nom als objectes concrets, a les seves relacions i a les seves propietats. Hi ha dos tipus de constants:
 - Els àtoms que són cadenes de lletres, dígit, guions i subratllats començant per minúscula. També podem tenir àtoms tancats entre cometes
 - Numèrics

Conceptes bàsics: els fets

- Serveixen per descriure propietats ja conegudes dels objectes i les seves relacions. Tant els noms dels objectes com de les relacions han de començar sempre en lletra minúscula
- Els fets es descriuen amb un nom de predicat i els objectes afectats com a arguments separats per comes i entre parèntesi
- **Acaben sempre en punt**

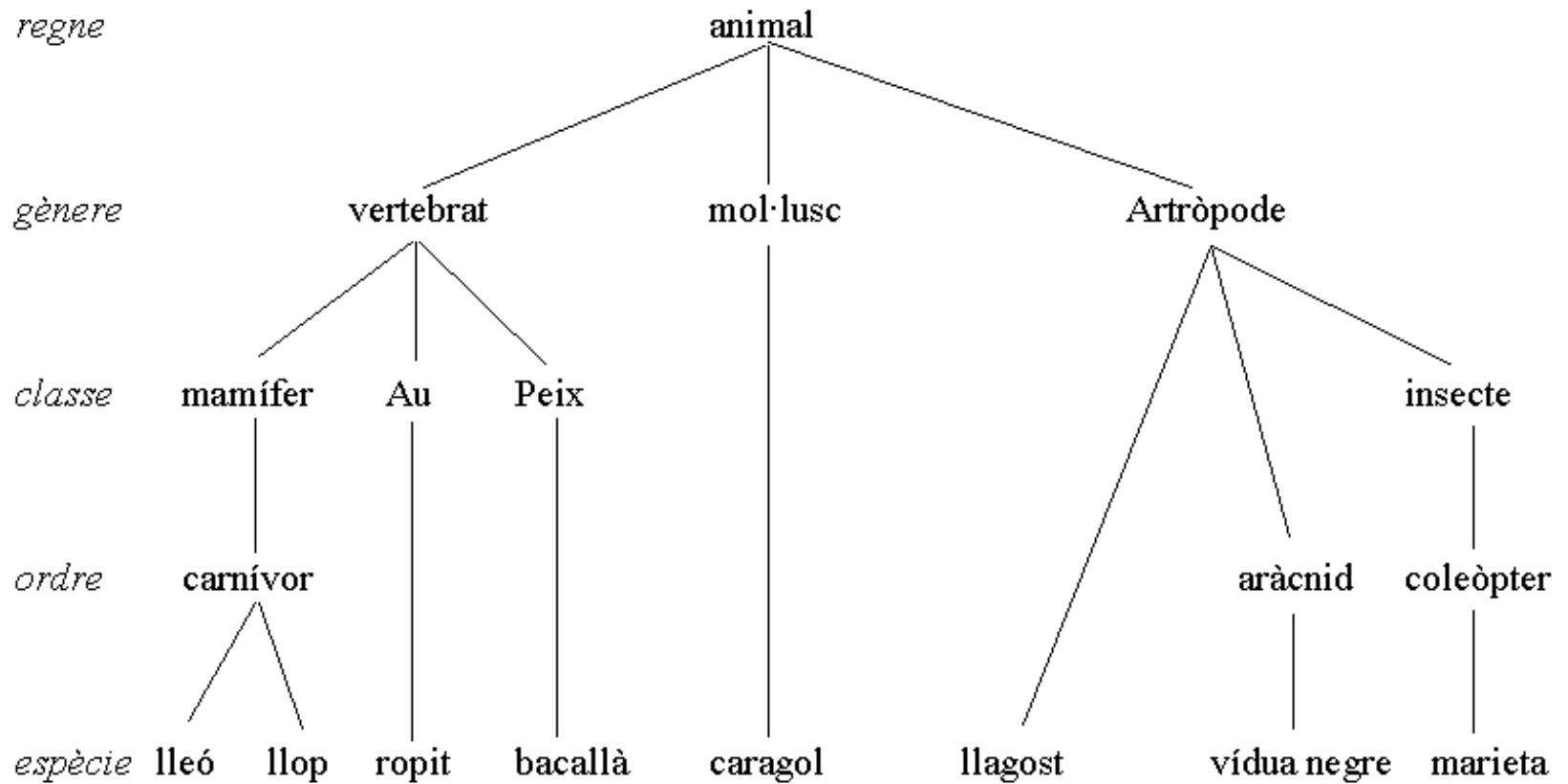
Conceptes bàsics: els fets

`nom_de_predicat(argument1, argument2...).`

ex: `menja(ca, pa).`

descriu la relació “el ca menja pa”

Conceptes bàsics: els fets



Conceptes bàsics: els fets

es-un(marieta, coleopter).

es-un(coleopter, insecte).

es-un(insecte, artropode).

es-un(artropode, animal).

es-un(vidua-negre, aracnid).

es-un(aracnid, artropode).

es-un(llagost, artropode).

es-un(caragol, molusc).

es-un(molusc, animal).

es-un(bacalla, peix).

es-un(peix, vertebrat).

es-un(vertebrat, animal).

es-un(ropit, au).

es-un(au, vertebrat).

es-un(llop, carnivor).

es-un(lleo, carnivor).

es-un(carnivor, mamifer).

es-un(mamifer, vertebrat).

Conceptes bàsics: estructures

- Una estructura és un objecte que es compon d'una col·lecció d'altres objectes anomenats components
- S'especifiquen amb un nom (functor) i separant les components per comes i entre parèntesis

`functor(component1, component2, ...)`

Conceptes bàsics: estructures

- Ens poden servir per guardar la informació de forma ordenada:

```
llibre(titol(el-licenciat-vidriera),  
      autor(nom(miguel),cognom(cervantes)),  
      any(1613)).
```

Conceptes bàsics: les preguntes

- PROLOG considera veritat tot el que té dins la base de coneixements i fals el que no hi és
- Una resposta afirmativa indica que ha demostrat el que demanàvem, una resposta negativa vol dir que amb les dades que té no ho pot demostrar

Conceptes bàsics: les preguntes

?- es-un(coleopter,insecte).

yes

?- es-un(coleopter,bitxo).

no

Conceptes bàsics: les variables

- S'utilitzen per representar objectes que inicialment són incògnites a les nostres preguntes
- Sempre comencen amb majúscula o subratllat “_”
- Si no ens interessa la identitat d'un objecte, podem utilitzar la variable anònima “_”
- Dues variables anònimes dins la mateixa regla, son variables diferents

Conceptes bàsics: les variables

Exemple:

```
?- es-un(coleopter,_).  
yes
```

demanam si coleòpter és alguna cosa, però
no ens interessa el què

Conceptes bàsics: les variables

- Si una variable representa un objecte, llavors es diu que està instanciada. Si no està definit el que representa, llavors es diu que no està instanciada
- **Variables amb el mateix nom dins clàusules distintes són completament diferents**
- L'objectiu principal del PROLOG és instanciar totes les variables amb el que s'anomena procés d'UNIFICACIÓ

El procés d'unificació

- La unificació és un mecanisme amb el que s'intenta que una hipòtesi plantejada tengui una forma “compatible” amb una de les clàusules de la base de coneixements
- Es cerca la base de coneixements (d'amunt avall i d'esquerra a dreta) cercant un predicat unificable amb la pregunta, si es troba llavors s'intenten unificar tots els arguments en ordre i valor

El procés d'unificació

?- es-un(X, artropode).

X=insecte; // el ; indica que es cerqui una

X=aracnid; // altra solució

X=llagost;

no

?- es-un(X, artropode).

X=insecte <return>

yes

Regles d'unificació

- Dues variables no instanciades, unifiquen i es converteixen en la mateixa variable
- Dues variables instanciades, unifiquen únicament si tenen el mateix valor
- Una variable no instanciada i una instanciada unifiquen i la primera agafa el valor de la segona
- Els números i els àtoms unifiquen si són iguals
- Dues estructures unifiquen si tenen el mateix nom, el mateix número d'arguments i tots ells es poden unificar

Conceptes bàsics: conjuncions

La conjunció “i” es representa amb una coma “,” i ens permet combinar més d’un objectiu dins una mateixa pregunta:

?- es-un(lleo,X), es-un(au,Y).

X=carnivor

Y=vertebrat;

no

Conceptes bàsics: conjuncions

Dins una conjunció, dues variables amb el mateix nom són la mateixa variable (estan dins la mateixa clàusula)

```
?- es-un(lleo,X), es-un(llop,X).
```

```
X=carnivor;
```

```
no
```

```
?- es-un(lleo,X), es-un(X,Y).
```

```
X=carnivor
```

```
Y=mamifer;
```

```
no
```

Conceptes bàsics: disjuncions

La conjunció “o” es representa amb un punt i coma “;” i ens permet demanar si es verifica alguna d’entre varies preguntes:

?- es-un(lleo,X); es-un(ropit,X).

X=carnivor

X=au;

no

És equivalent a demanar dues preguntes independents, X no és la mateixa variable

Conceptes bàsics: negacions

La negació ens permet fer preguntes sobre la falsetat d'un fet. S'utilitza el predicat "not"

?- not(es-un(lleo,carnivor)).

no

Conceptes bàsics: regles d'inferència

Les regles s'utilitzen si es vol establir una dependència entre fets. Tenen la forma:

$\text{cap} \text{ :- } \text{cos}.$

on el cap especifica el fet que es vol descriure i el cos són les condicions que s'han de verificar perquè el cap sigui cert

Conceptes bàsics: regles d'inferència

```
fill(pep, pere).
```

```
fill(lluis, pere).
```

```
pare(X,Y):-fill(Y,X).
```

```
germa(X,Y):-pare(Z,X),pare(Z,Y).
```

```
?- germa(pep,lluis).
```

```
yes
```

Operadors

En PROLOG, els operadors aritmètics i relacionals són:

`+, -, *, /, //, mod, =, \=, =<, >=, <, >, is`

què respon PROLOG si en l'exemple anterior demanam:

`?- germa(X,Y).`

com es pot arreglar ?