(First Order Logic)

Lógica de Primeira Ordem

opera sobre conhecimento preciso

crenças do mundo e regras

(First Order Logic)

Lógica de Primeira Ordem

Predicados

Conectivos

Quantificadores

Constantes

Variáveis

Aridade

$\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$

Quantificador universal: “Para todos”

$\forall X \; \forall Y \; pai(Z,X) \wedge pai(Z,Y) \wedge diff(X,Y) \Rightarrow irmaos(X,Y)$

Quantificador existencial: “Existe pelo menos um”

$\exists X \; \exists Y \; mae(X, Y)$

$mae(ana, joao)$

Cláusulas de Horn

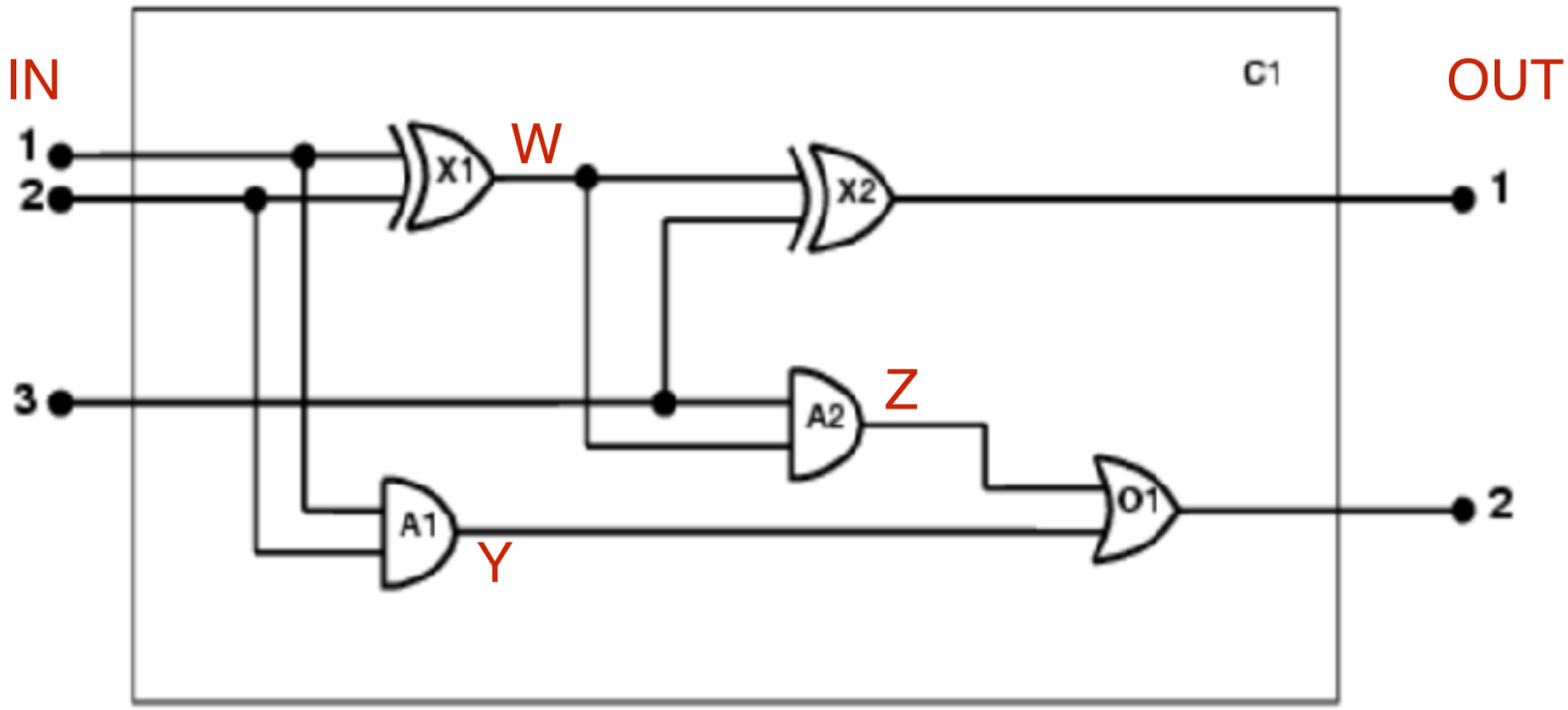
Subconjunto da Lógica de Primeira Ordem

Cláusulas da lógica de primeira ordem possuem custo muito alto

$$C \wedge \ldots \wedge Cn \Rightarrow C \qquad (n \geq 0)$$

Base de fatos

or(0, 0, 0).
or(_, _, 1).
and(1, 1, 1).
and(_, _, 0).
not(1, 0).
not(0, 1).
xor(X, X, 0).
xor(_, _, 1).



Base de regras

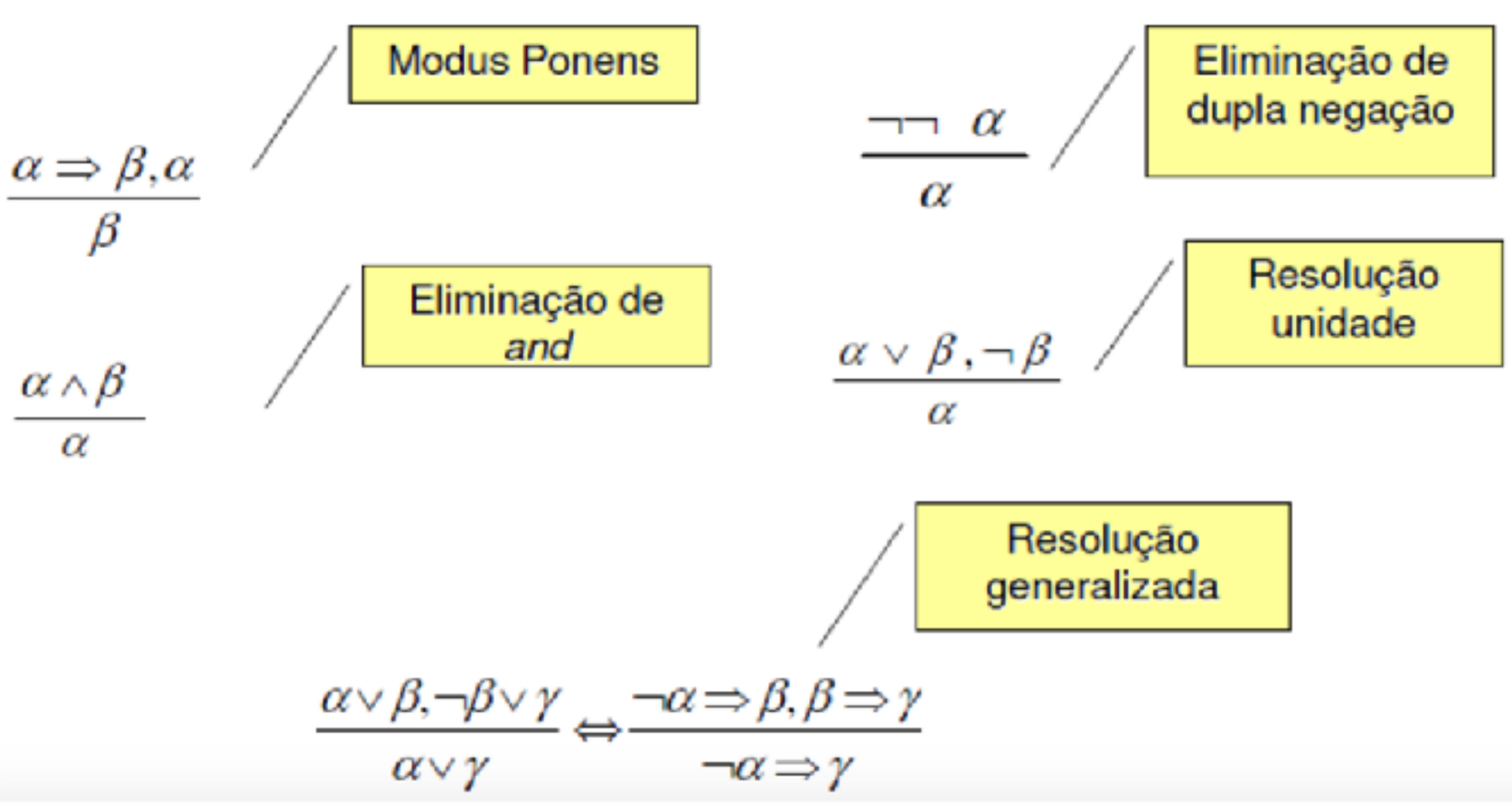
xor(IN1, IN2, W) ∧ and(IN1, IN2, Y) ∧ xor(W, IN3, OUT1)
∧ and(IN3, W, Z) ∧ or(Z, Y, OUT2) ⇒ somador(IN1, IN2, IN3, OUT1, OUT2)

— — — — — — — — — —

somador(0, 1, 1, OUT1, OUT2) ?

Como responder as questões?

Algoritmos de Inferência que fazem uso de Regras de inferência



Algorimtos de inferência

Encadeamento para frente

Encadeamento para trás

Prova por refutação

```
function FOL-FC-ASK(KB, α) returns a substitution or false
  inputs: KB, the knowledge base, a set of first order definite clauses
          α, the query, an atomic sentence
  local variables: new, the new sentences inferred on each iteration

  repeat until new is empty
      new ← { }
      for each rule in KB do
          (p1 ∧ ... ∧ pn ⇒ q) ← STANDARDIZE-VARAIBLES(rule)
          for each θ such that SUBST(θ, p1 ∧ ... ∧ pn) = SUBST(θ, p'1 ∧ ... ∧ p'n)
                      for some p'1 ∧ ... ∧ p'n in KB
              q' ← SUBST(θ, q)
              if q' does not unify with some sentence already in KB or new then
                  add q' to new
                  Φ ← UNIFY(q', α)
                  if Φ is not fail then return Φ
      add new to KB
  return false
```

Figure ?? A conceptually straightforward, but very inefficient forward-chaining algorithm. On each iteration, it adds to *KB* all the atomic sentences that can be inferred in one step from the implication sentences and the atomic sentences already in *KB*. The function STANDARDIZE-VARIABLES replaces all variables in its arguments with new ones that have not been used before.

```
function FOL-BC-ASK(KB, query) returns a generator of substitutions
  return FOL-BC-OR(KB, query, { })

generator FOL-BC-OR(KB, goal, θ) yields a substitution
  for each rule (lhs ⇒ rhs) in FETCH-RULES-FOR-GOAL(KB, goal) do
      (lhs, rhs) ← STANDARDIZE-VARIABLES((lhs, rhs))
      for each θ' in FOL-BC-AND(KB, lhs, UNIFY(rhs, goal, θ)) do
          yield θ'

generator FOL-BC-AND(KB, goals, θ) yields a substitution
  if θ = failure then return
  else if LENGTH(goals) = 0 then yield θ
  else do
      first, rest ← FIRST(goals), REST(goals)
      for each θ' in FOL-BC-OR(KB, SUBST(θ, first), θ) do
          for each θ'' in FOL-BC-AND(KB, rest, θ') do
              yield θ''
```

Figure ?? A simple backward-chaining algorithm for first-order knowledge bases.

(Backward Chaining)

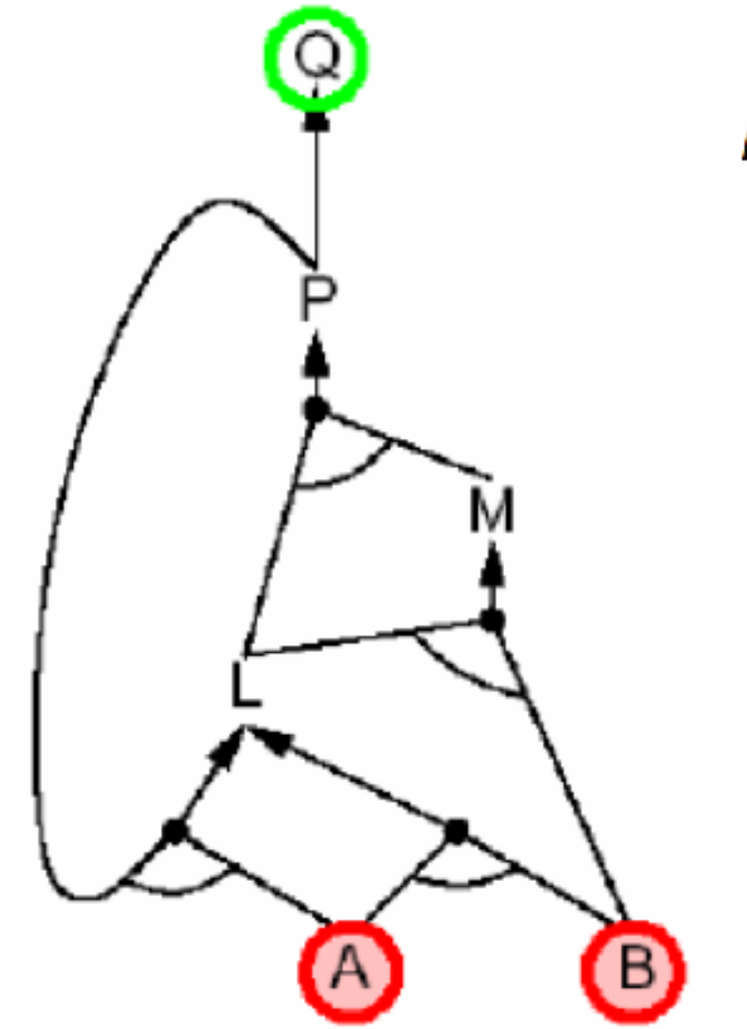
Encadeamento para trás

Raciocínio dirigido à objetivo

Da hipótese aos dados (ex: PROLOG)

Programa responde a perguntas !

Posso provar Q?



meta

$P \Rightarrow Q$

$L \wedge M \Rightarrow P$

$B \wedge L \Rightarrow M$

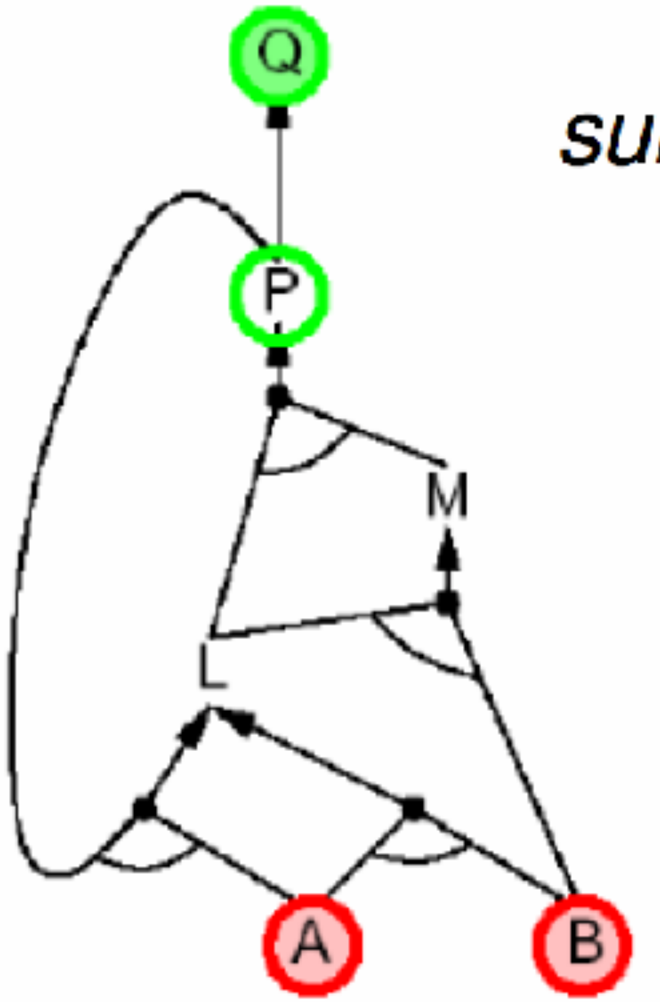
$A \wedge P \Rightarrow L$

$A \wedge B \Rightarrow L$

A

B

sub-meta



$P \Rightarrow Q$

$L \wedge M \Rightarrow P$

$B \wedge L \Rightarrow M$

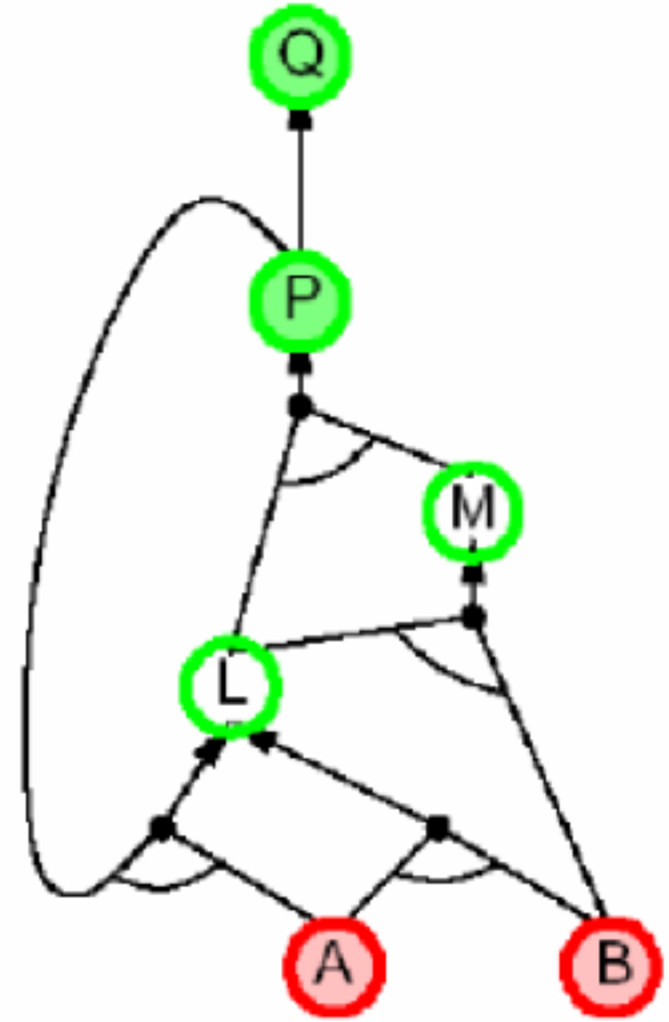
$A \wedge P \Rightarrow L$

$A \wedge B \Rightarrow L$

A

B

sub-meta



$P \Rightarrow Q$

$L \wedge M \Rightarrow P$

$B \wedge L \Rightarrow M$

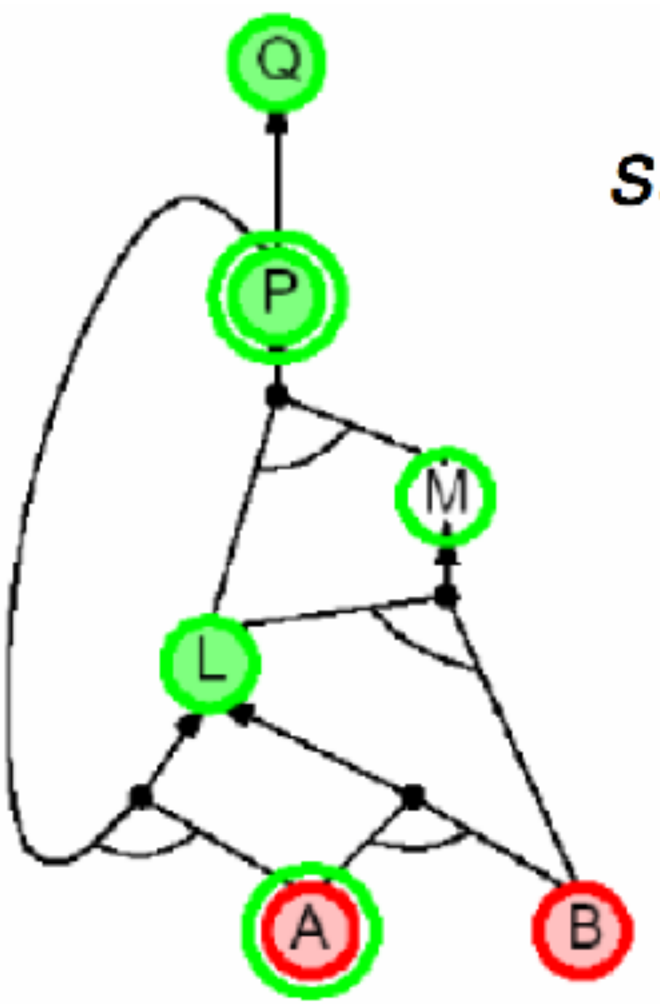
$A \wedge P \Rightarrow L$

$A \wedge B \Rightarrow L$

A

B

sub-meta



$P \Rightarrow Q$

$L \wedge M \Rightarrow P$

$B \wedge L \Rightarrow M$

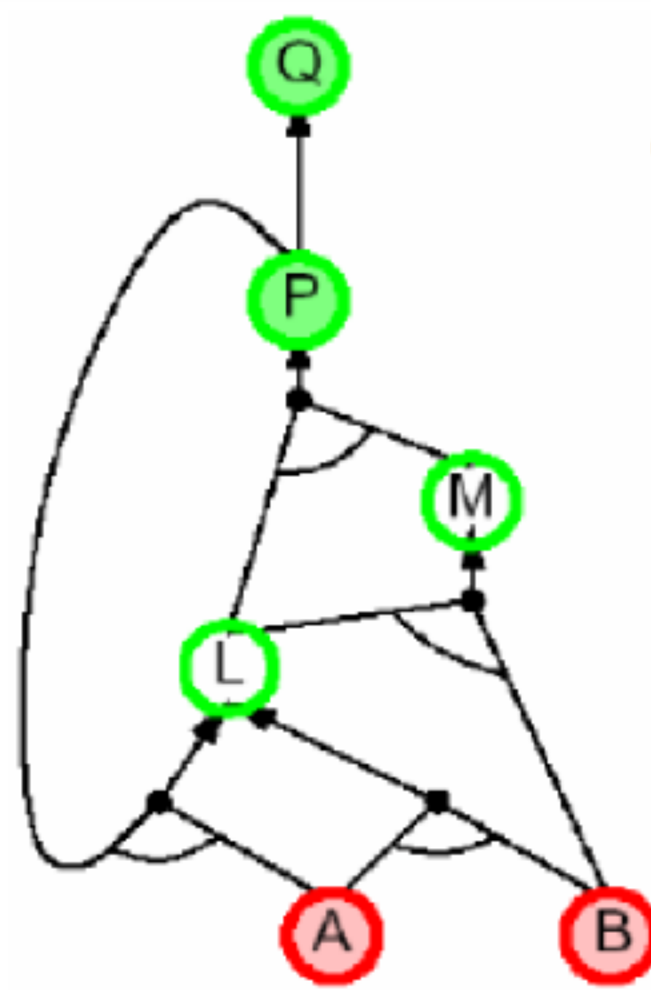
$A \wedge P \Rightarrow L$

$A \wedge B \Rightarrow L$

A

B

consulta L



$P \Rightarrow Q$

$L \wedge M \Rightarrow P$

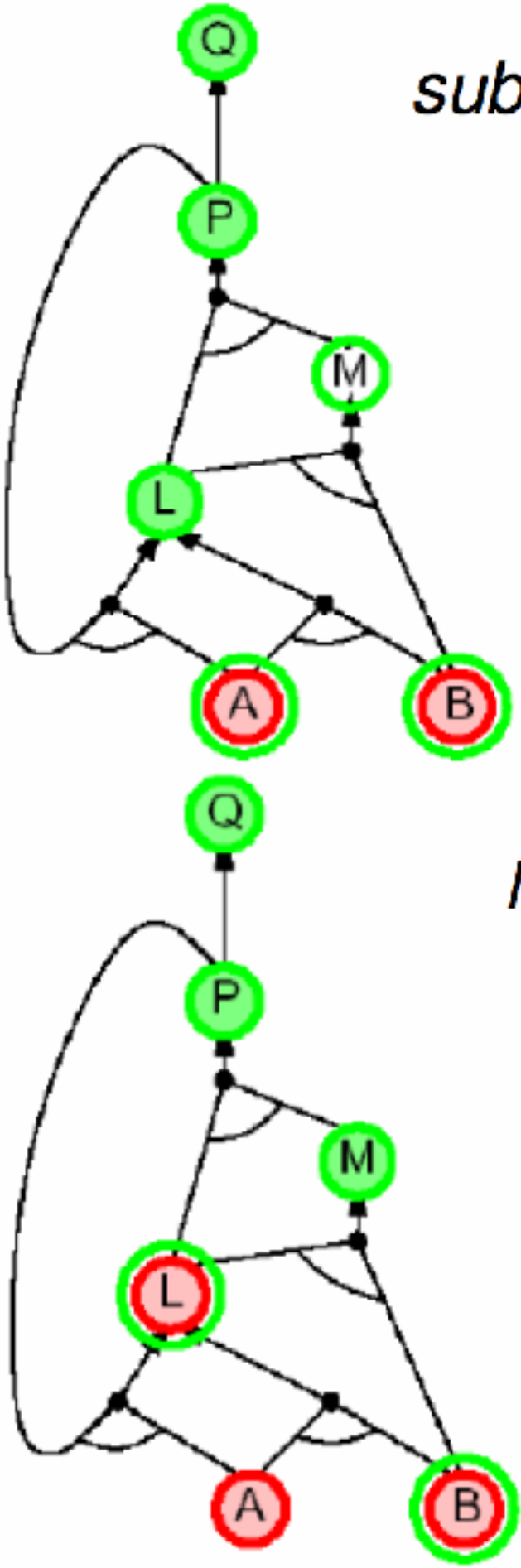
$B \wedge L \Rightarrow M$

$A \wedge P \Rightarrow L$

$A \wedge B \Rightarrow L$

A

B



sub-meta

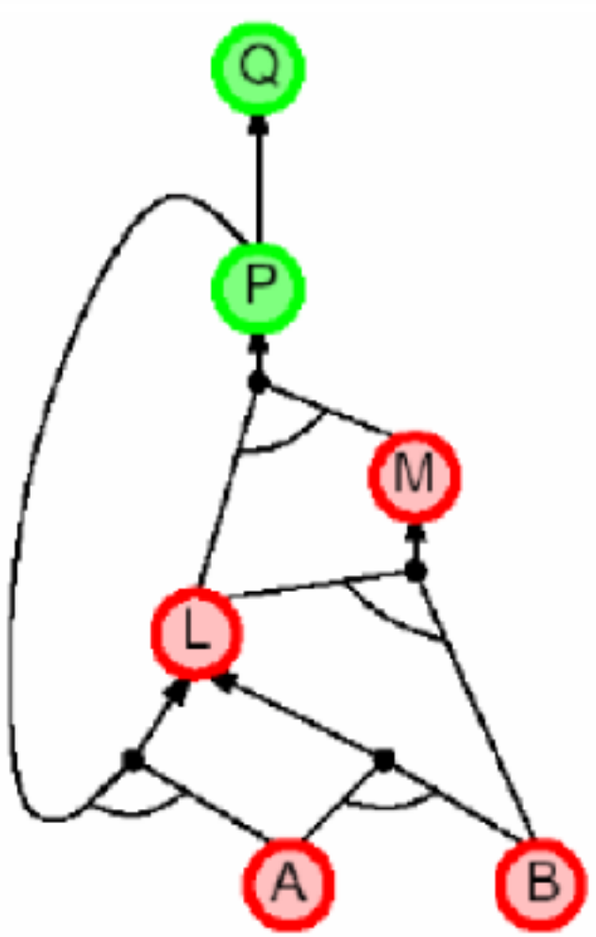
$P \Rightarrow Q$
$L \wedge M \Rightarrow P$
$B \wedge L \Rightarrow M$
$A \wedge P \Rightarrow L$
$\boxed{A \wedge B \Rightarrow L}$
A
B

L true

$P \Rightarrow Q$
$L \wedge M \Rightarrow P$
$B \wedge L \Rightarrow M$
$A \wedge P \Rightarrow L$
$A \wedge B \Rightarrow L$
A
B

M true

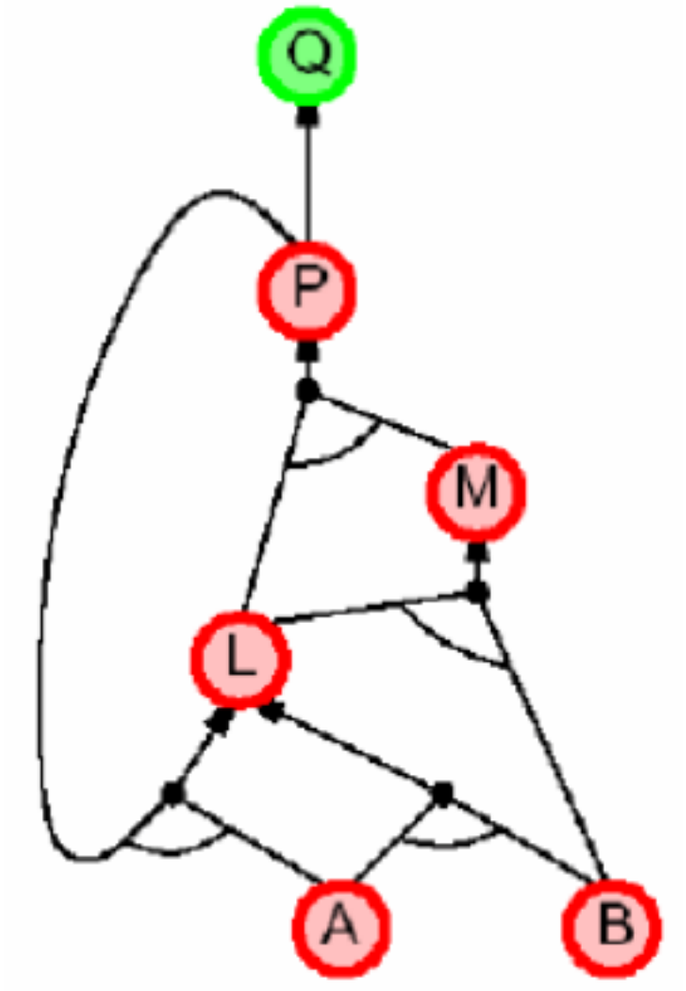
$P \Rightarrow Q$
$L \wedge M \Rightarrow P$
$\boxed{B \wedge L \Rightarrow M}$
$A \wedge P \Rightarrow L$
$A \wedge B \Rightarrow L$
A
B



P true

$P \Rightarrow Q$
$\boxed{L \wedge M \Rightarrow P}$
$B \wedge L \Rightarrow M$
$A \wedge P \Rightarrow L$
$A \wedge B \Rightarrow L$
A
B

Resposta: Q true



$P \Rightarrow Q$

$L \wedge M \Rightarrow P$

$B \wedge L \Rightarrow M$

$A \wedge P \Rightarrow L$

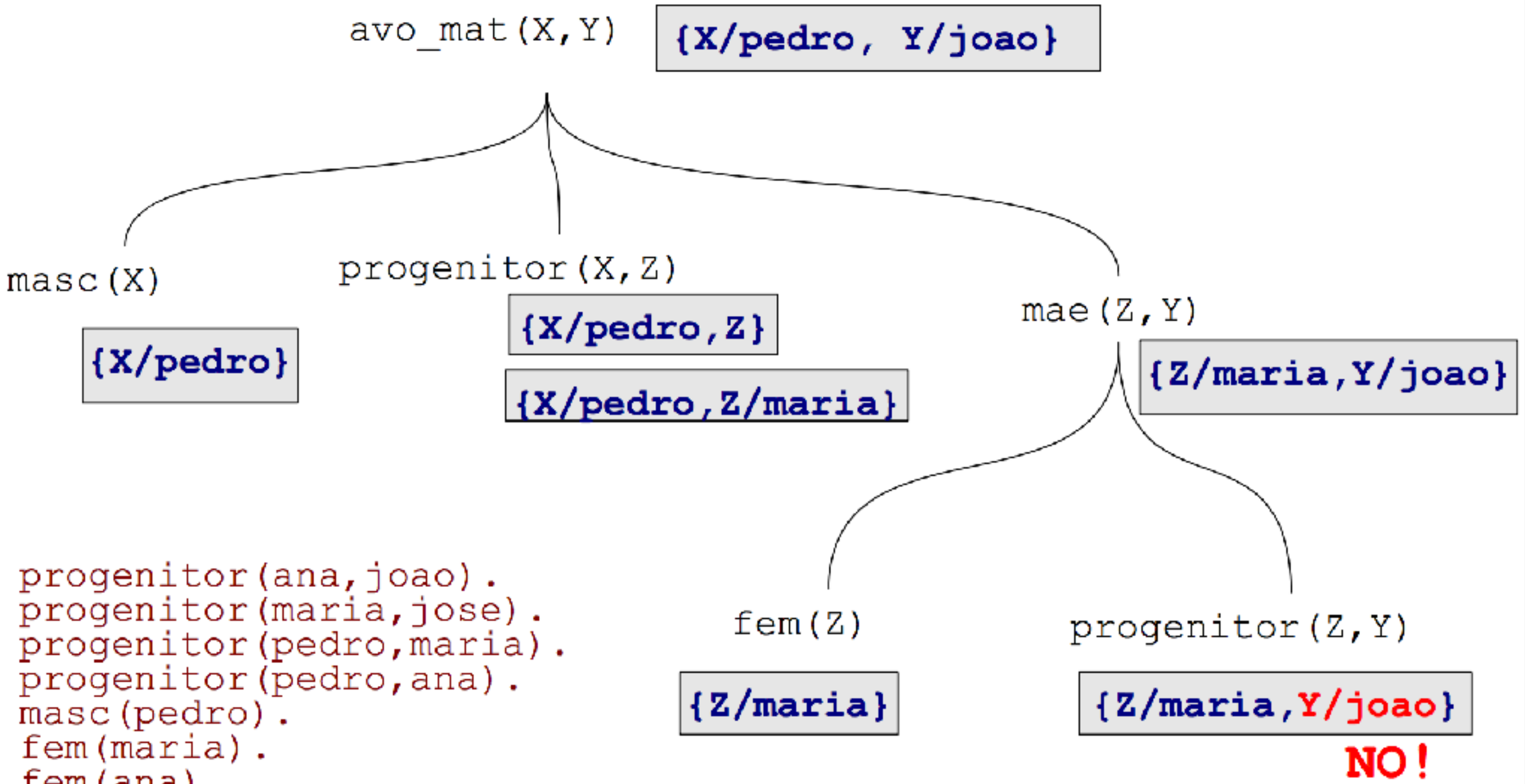
$A \wedge B \Rightarrow L$

A

B

Exemplo PROLOG

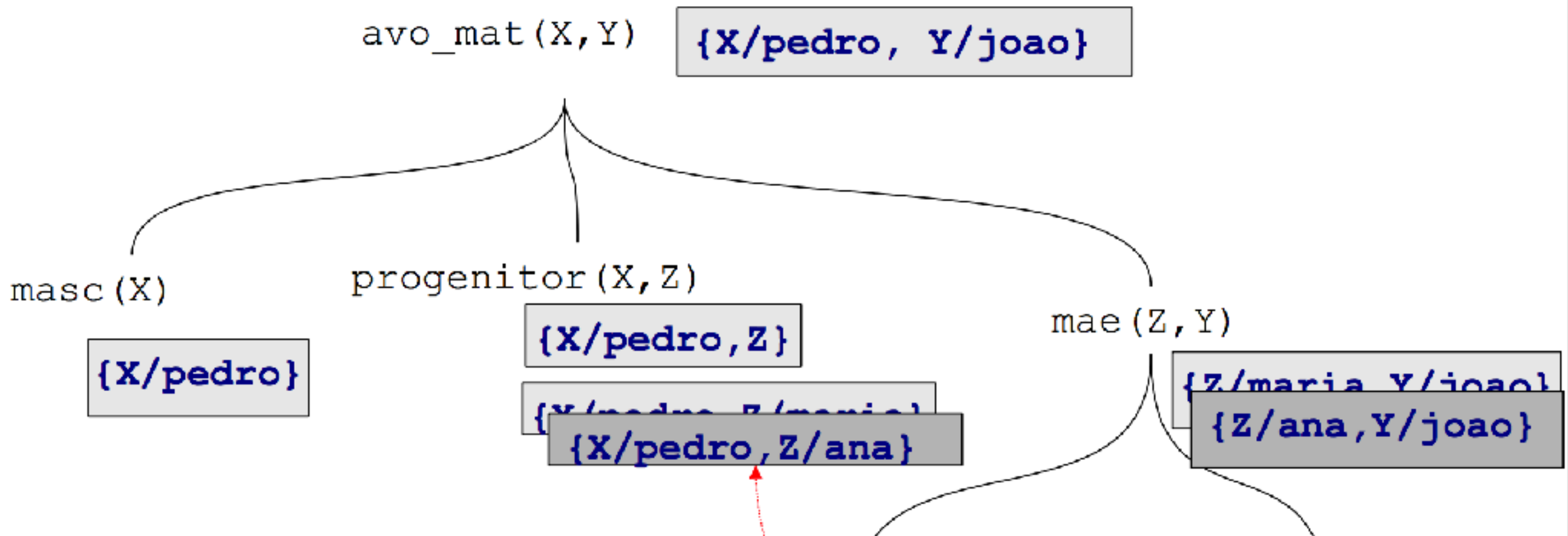
```
?- avo_mat(pedro,joao).
```




```
progenitor(ana,joao).
progenitor(maria,jose).
progenitor(pedro,maria).
progenitor(pedro,ana).
masc(pedro).
fem(maria).
fem(ana).

mae(A,B):- fem(A), progenitor(A,B).
avo_mat(X,Y):- masc(X),
               progenitor(X,Z),
               mae(Z,Y).
```

```
?- avo_mat(pedro,joao).
```




```
progenitor(ana,joao).
progenitor(maria,jose).
progenitor(pedro,maria).
progenitor(pedro,ana).
masc(pedro).
fem(maria).
fem(ana).

mae(A,B):- fem(A), progenitor(A,B).
avo_mat(X,Y):- masc(X),
               progenitor(X,Z),
               mae(Z,Y).
```

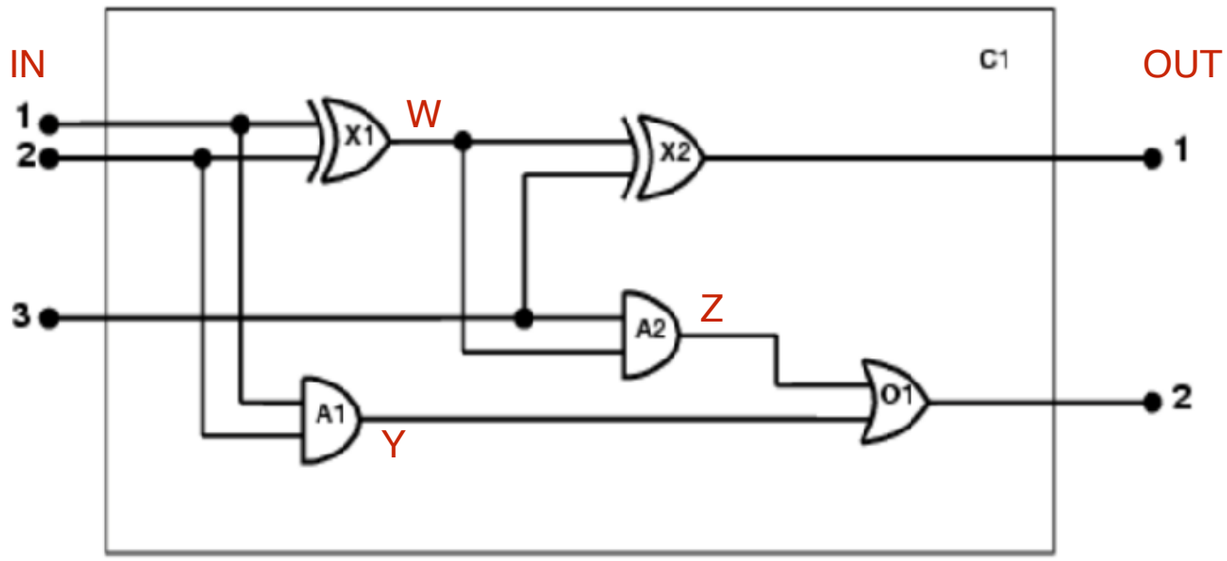
PROLOG

exemplo genealogica.pl

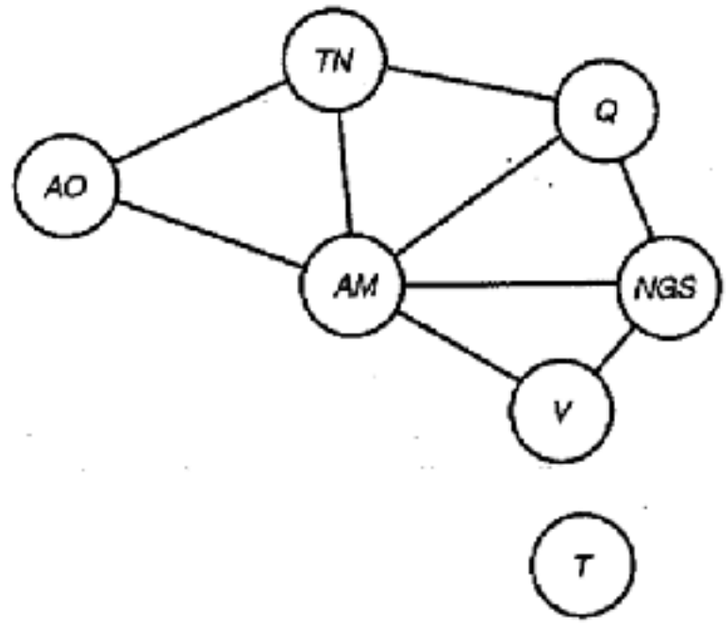
```
sobrinho(X, Y) :- ...
cunhado(X, Y)  :-  ...
```

PROLOG

exemplo circuito.pl



exemplo coloring.pl



PROLOG

exemplo fatorial.pl

exemplo fibonacci.pl

PROLOG

exemplo fatorial_dyn.pl

exemplo fibonacci_dyn.pl

(Forward Chaining)

Encadeamento para frente

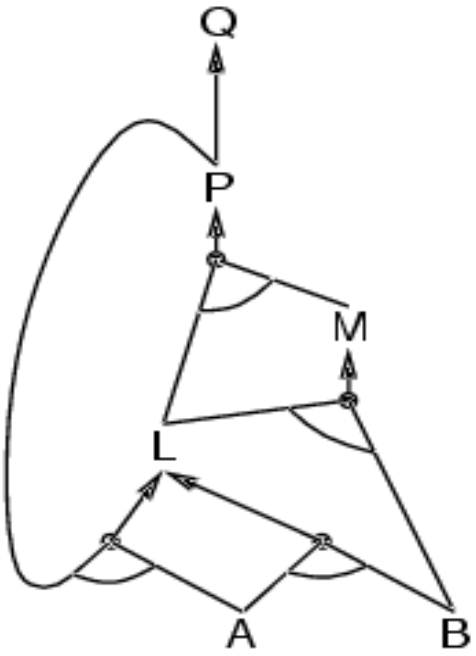
Raciocínio dirigido aos dados

Dos dados às conclusões.

Programa procura por respostas a partir de fatos conhecidos.

O que posso provar a partir de A e B ?

$$P \Rightarrow Q$$
$$L \wedge M \Rightarrow P$$
$$B \wedge L \Rightarrow M$$
$$A \wedge P \Rightarrow L$$
$$A \wedge B \Rightarrow L$$
$$A$$
$$B$$



$P \Rightarrow Q$

$L \wedge M \Rightarrow P$

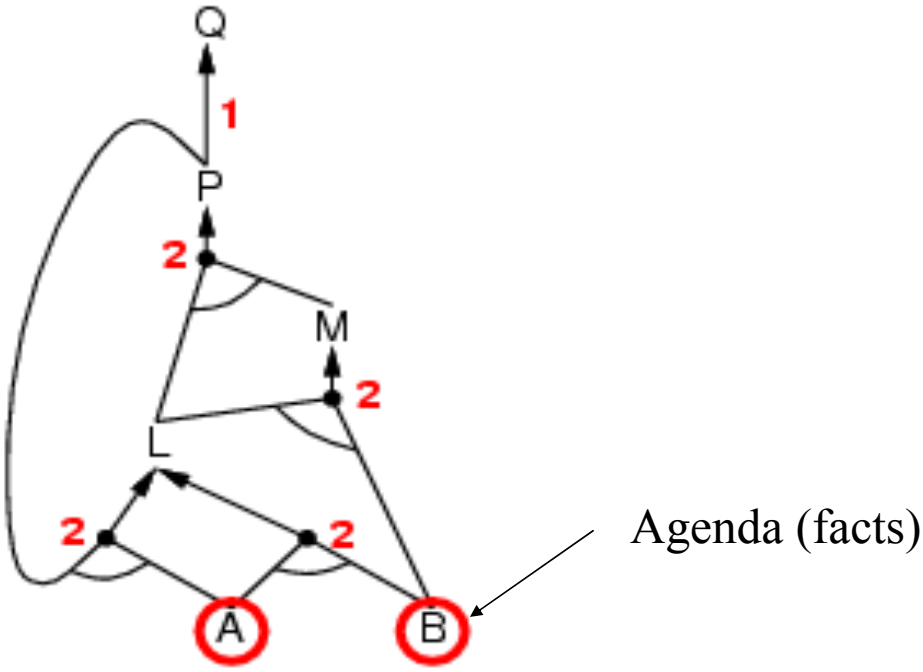
$B \wedge L \Rightarrow M$

$A \wedge P \Rightarrow L$

$A \wedge B \Rightarrow L$

A

B



$P \Rightarrow Q$

$L \wedge M \Rightarrow P$

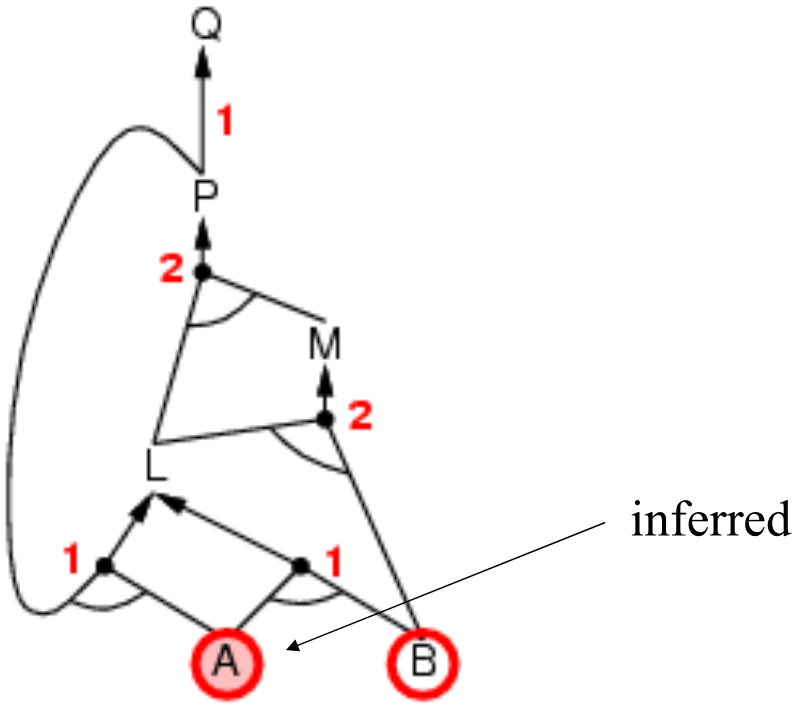
$B \wedge L \Rightarrow M$

$A \wedge P \Rightarrow L$

$A \wedge B \Rightarrow L$

A

B



$P \Rightarrow Q$

$L \wedge M \Rightarrow P$

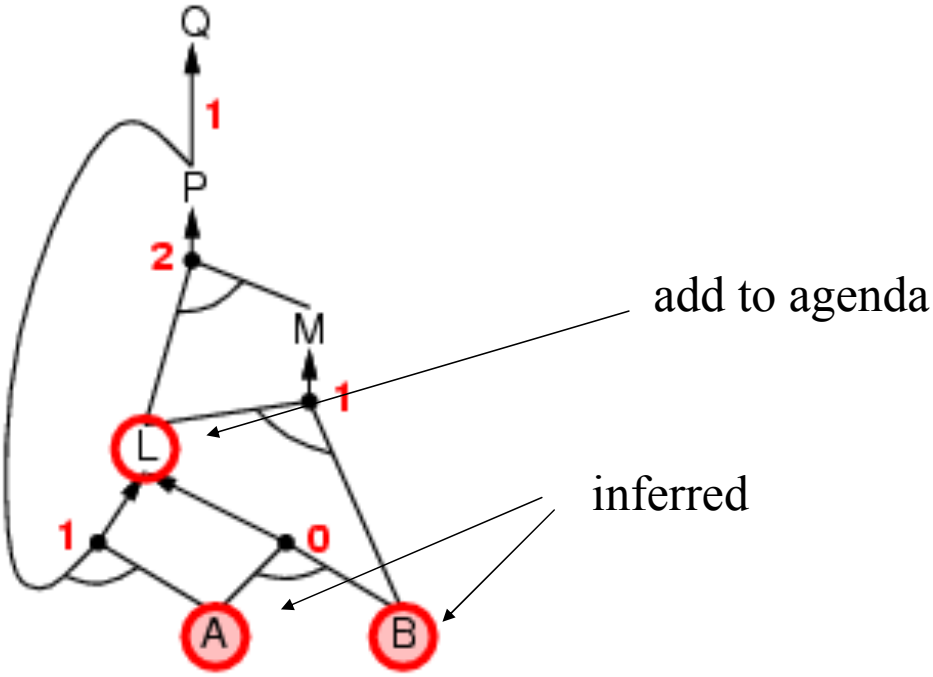
$B \wedge L \Rightarrow M$

$A \wedge P \Rightarrow L$

$A \wedge B \Rightarrow L$

A

B



$P \Rightarrow Q$

$L \wedge M \Rightarrow P$

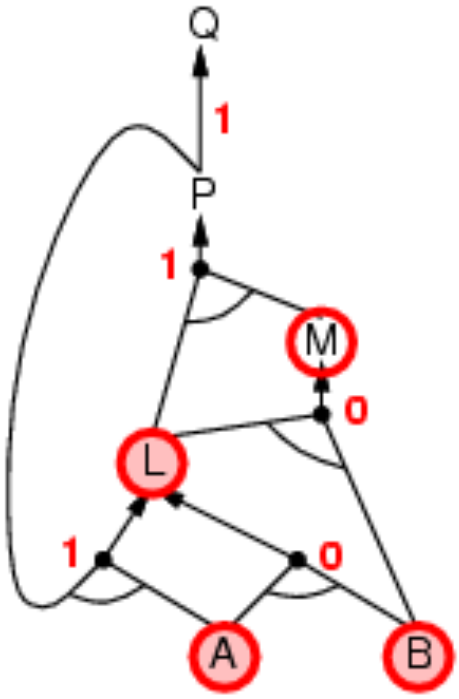
$B \wedge L \Rightarrow M$

$A \wedge P \Rightarrow L$

$A \wedge B \Rightarrow L$

A

B



$P \Rightarrow Q$

$L \wedge M \Rightarrow P$

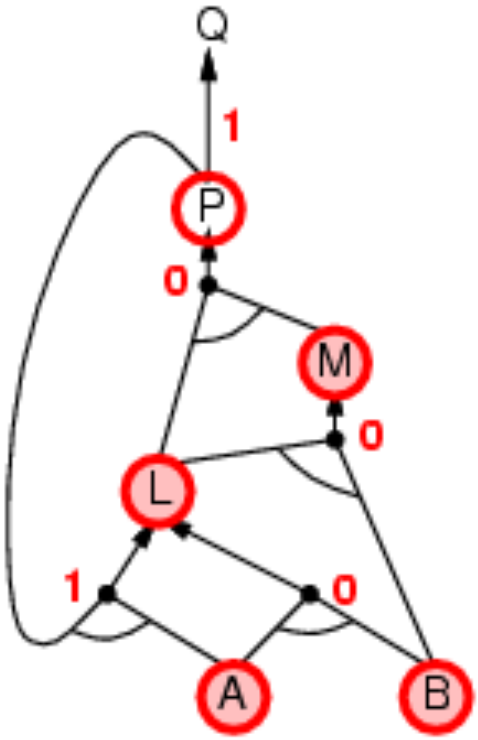
$B \wedge L \Rightarrow M$

$A \wedge P \Rightarrow L$

$A \wedge B \Rightarrow L$

A

B



$P \Rightarrow Q$

$L \wedge M \Rightarrow P$

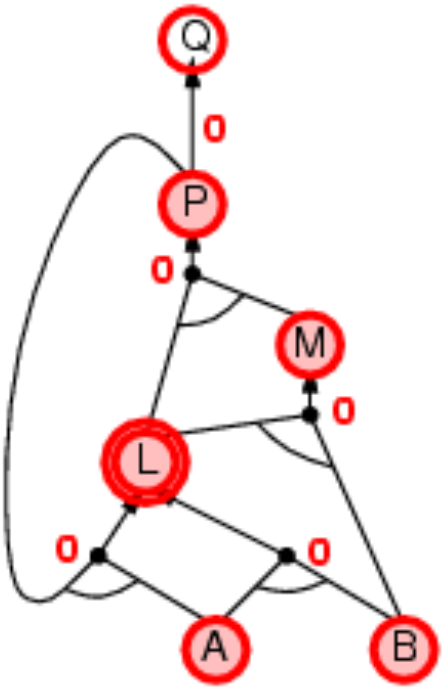
$B \wedge L \Rightarrow M$

$A \wedge P \Rightarrow L$

$A \wedge B \Rightarrow L$

A

B



Exercício 1:

9.9 This question considers Horn KBs, such as the following:

$$P(F(x)) \Rightarrow P(x)$$
$$Q(x) \Rightarrow P(F(x))$$
$$P(A)$$
$$Q(B)$$

Let FC be a breadth-first forward-chaining algorithm that repeatedly adds all consequences of currently satisfied rules; let BC be a depth-first left-to-right backward-chaining algorithm that tries clauses in the order given in the KB. Which of the following are true?

1. FC will infer the literal $Q(A)$.
2. FC will infer the literal $P(B)$.
3. If FC has failed to infer a given literal, then it is not entailed by the KB.
4. BC will return $true$ given the query $P(B)$.
5. If BC does not return $true$ given a query literal, then it is not entailed by the KB.

Exercício 2:

1. Represente as sentenças a seguir em Lógica de Primeira Ordem
2. Implemente-as em FolKB
3. Varie os fatos e realize execuções em Forward Chaining
4. Determine um objetivo e verifique se este pode ser provado via Backward Chaining

I1. If the animal has hair then it is a mammal
I2. If the animal gives milk then it is a mammal
I3. If the animal has feathers then it is a bird
I4. If the animal flies and it lays eggs then it is a bird
I5. If the animal is a mammal and it eats meat then it is a carnivore
I6. If the animal is a mammal and it has pointed teeth and it has claws and its eyes point forward then it is a carnivore
I7. If the animal is a mammal and it has hoofs then it is an ungulate
18. If the animal is a mammal and it chews cud then it is an ungulate and it is even-toed
I9. If the animal is a carnivore and it has a tawny color and it has dark spots then it is a cheetah
I10. If the animal is a carnivore and it has a tawny color and it has black strips then it is a tiger
I11. If the animal is an ungulate and it has long legs and it has a long neck and it has a tawny color and it has dark spots then it is a giraffe
I12. If the animal is an ungulate and it has a white color and it has black stripes then it is a zebra
Il3. If the animal is a bird and it does not fly and it has long legs and it has a long neck and it is black and white then it is an ostrich,
Il4. If the animal is a bird and it does not fly and it swims and it is black and white then it is a penguin
Il5. If the animal is a bird and it is a good flyer then it is an albatross.