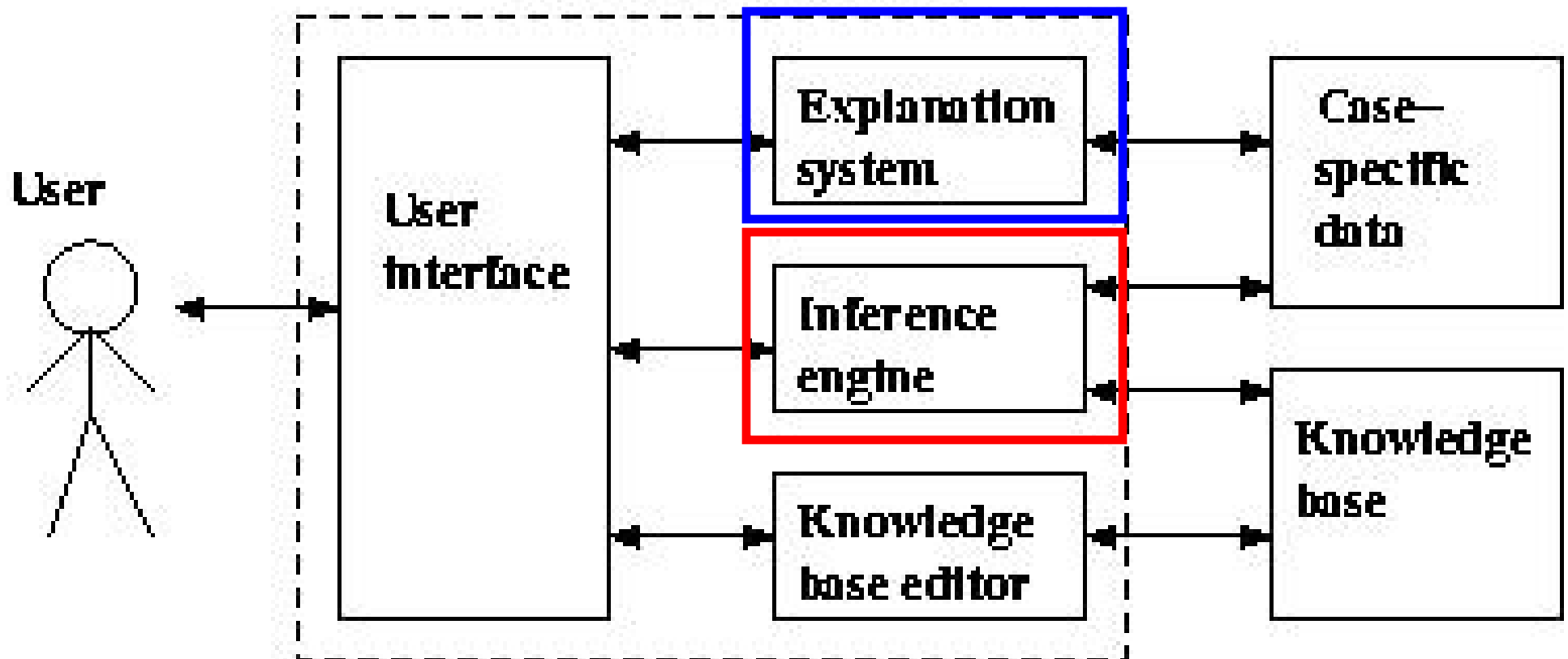


# Sistemas especialistas

## Expert System Shell





# Inferência em lógica de primeira ordem

- Método mais utilizado:  
resolução por refutação
- Passos largos baseados em eliminação do E e Modus Ponens, como em lógica proposicional
- Precisa lidar com as variáveis lógicas:  
substituição e unificação



# Sistemas dedutivos: exemplo

“A lei americana diz que é crime um americano vender armas para nações hostis.

Nono, um país inimigo dos EUA, tem alguns mísseis, e todos estes mísseis foram vendidos pelo Coronel Oeste, que é americano”.

**Como provar que o coronel é criminoso?**



# Passo 1: representação...

...é um crime um americano vender armas para nações hostis...

(1) for all  $x,y,z$  Amer( $x$ ) E Arma( $y$ ) E Nacao( $z$ ) E Hostil( $z$ ) E  
Vende( $x,z,y$ )  $\rightarrow$  Crim( $x$ )

...Nono...tem alguns mísseis...

(2) exists  $x$  Dono(Nono, $x$ ) E Missil( $x$ )

...todos estes mísseis foram vendidos pelo Coronel Oeste...

(3) Para todo  $x$  Dono(Nono, $x$ ) E Missil( $x$ )  $\rightarrow$   
Vende(Oeste,Nono, $x$ )

(4) Para todo  $x$  Missil( $x$ )  $\rightarrow$  Arma( $x$ )

(5) Para todo  $x$  Inimigo( $x$ ,EUA)  $\rightarrow$  Hostil( $x$ )

Fatos:

(6) Americano(Oeste)

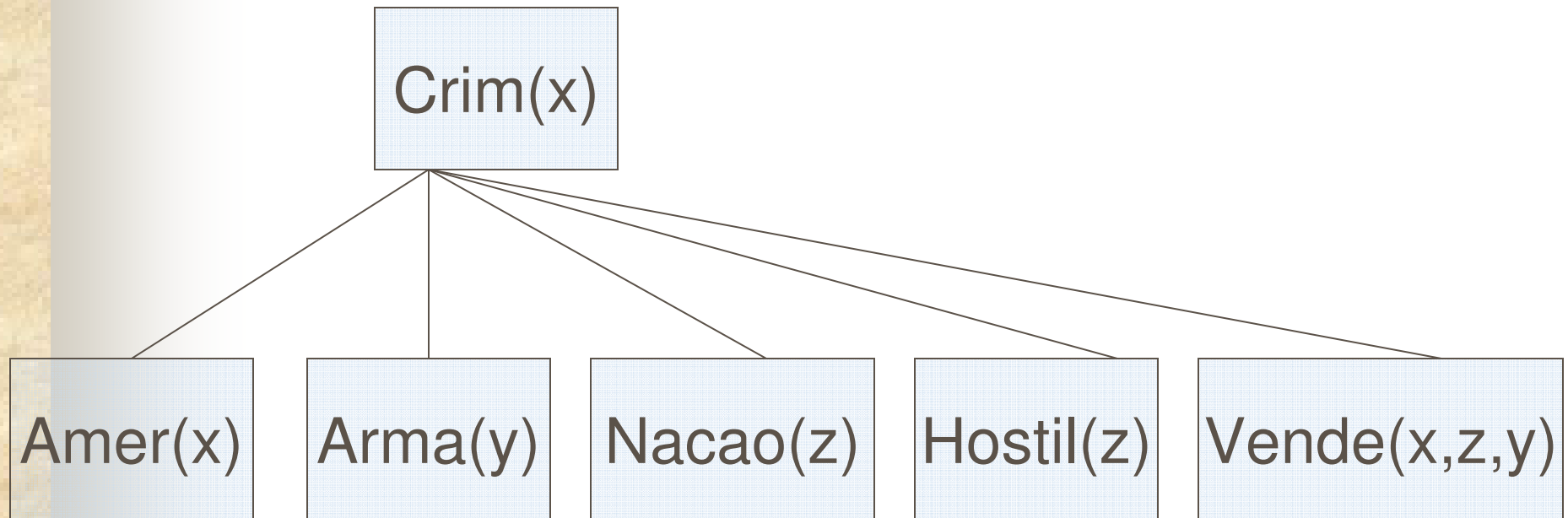
(9) Nacao(EUA)

(7) Nacao(Nono)

(10) Arma(M1)

(8) Inimigo(Nono,EUA)

## Passo 2: inferência...



```
graph TD; Crim(x) --- Amer(x); Crim(x) --- Arma(y); Crim(x) --- Nacao(z); Crim(x) --- Hostil(z); Crim(x) --- Vende(x,z,y);
```

Crim(x)

Amer(x)

Arma(y)

Nacao(z)

Hostil(z)

Vende(x,z,y)

```
graph TD; Crim(x) --- Amer(x); Crim(x) --- Arma(y); Crim(x) --- Nacao(z); Crim(x) --- Hostil(z); Crim(x) --- Vende(x,z,y);
```

Crim(x)

Amer(x)

Arma(y)

Nacao(z)

Hostil(z)

Vende(x,z,y)

```
graph TD; Crim(x) --- Amer(x); Crim(x) --- Arma(y); Crim(x) --- Nacao(z); Crim(x) --- Hostil(z); Crim(x) --- Vende(x,z,y); Amer(x) --- xOeste[x/Oeste]; style xOeste stroke:#f96,stroke-width:2px
```

Crim(x)

Amer(x)

Arma(y)

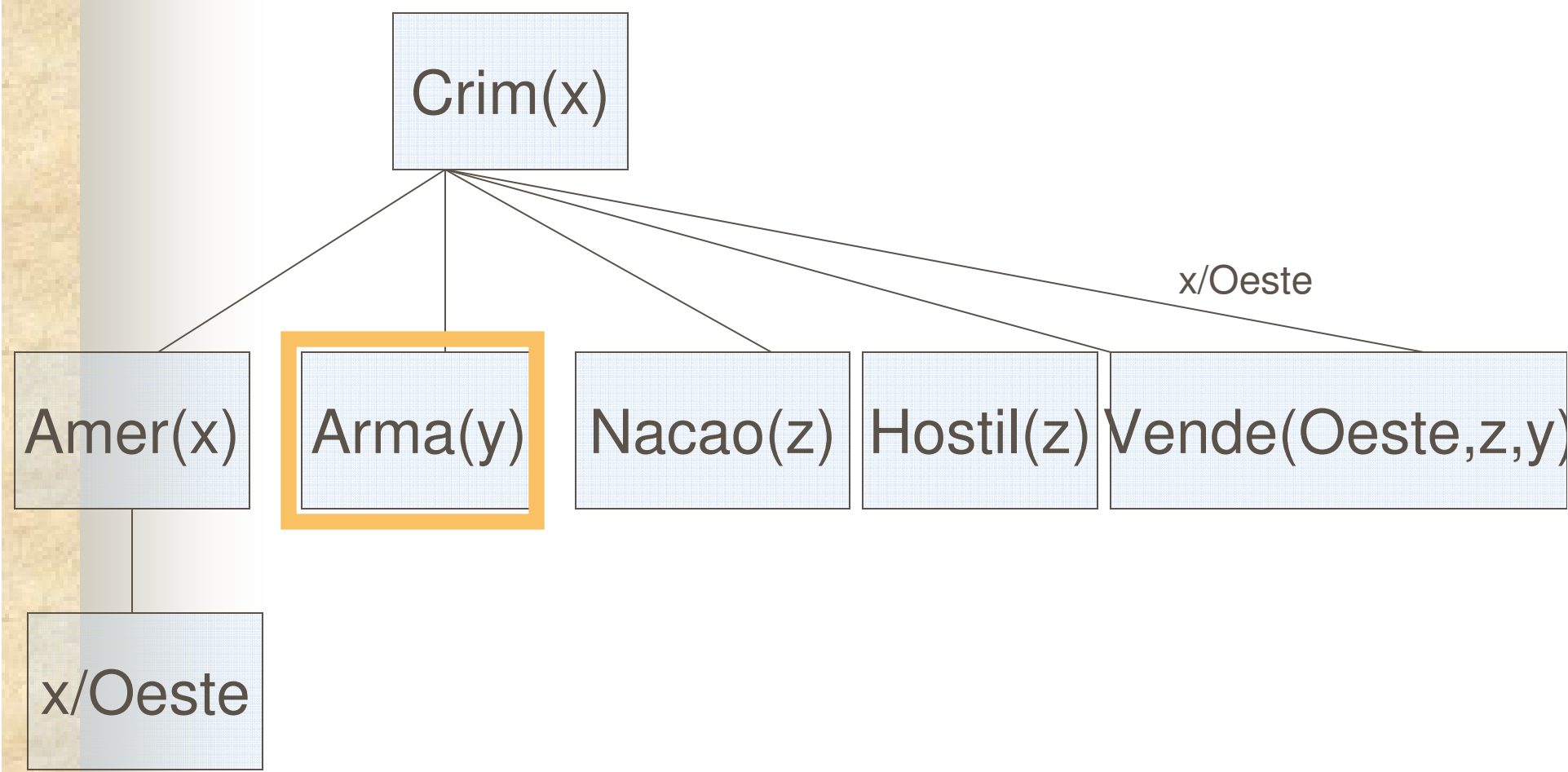
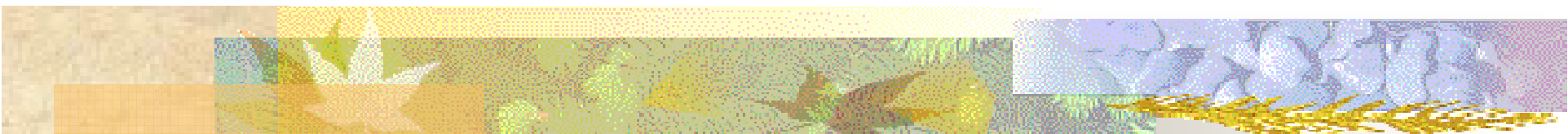
Nacao(z)

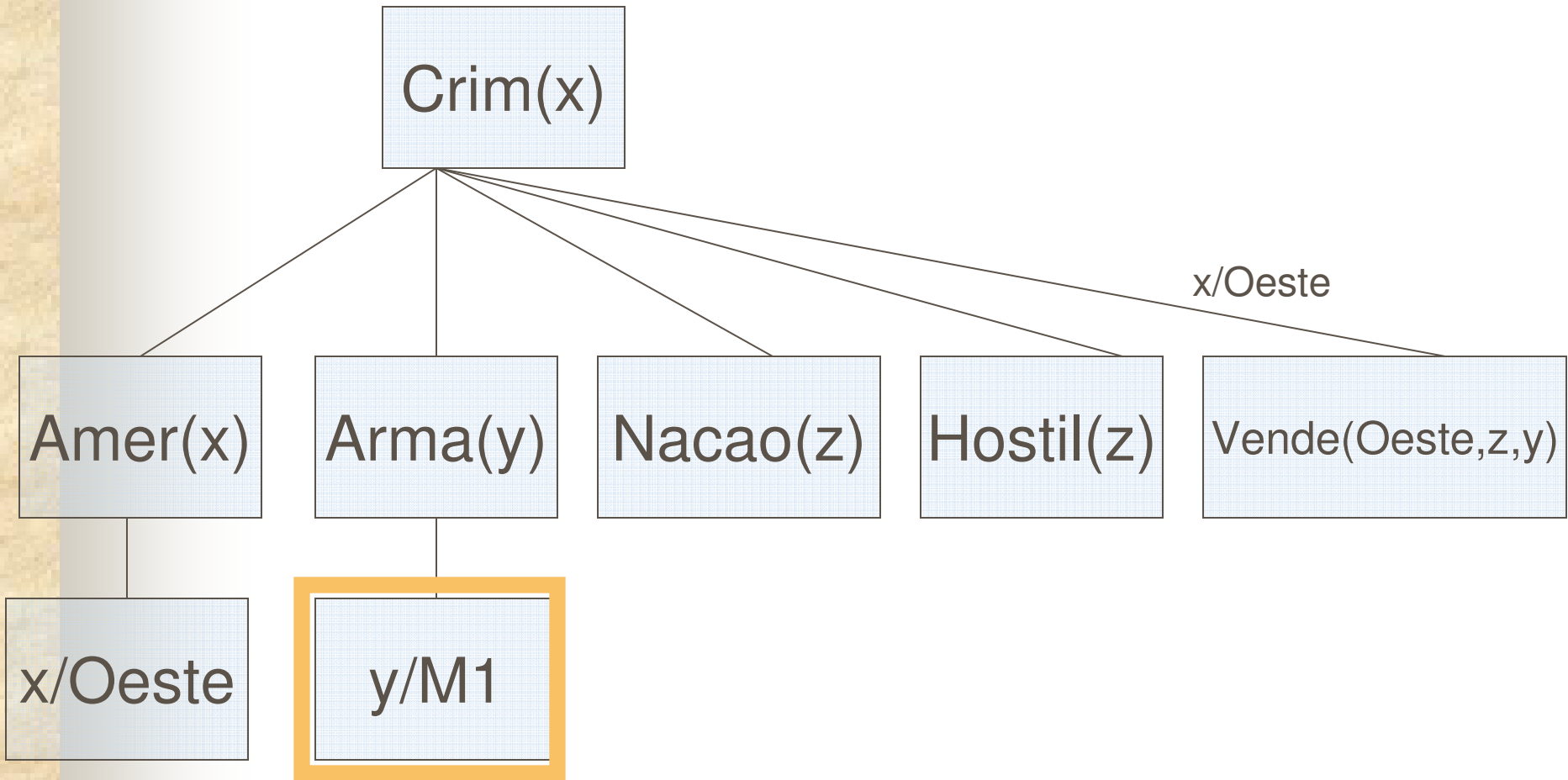
Hostil(z)

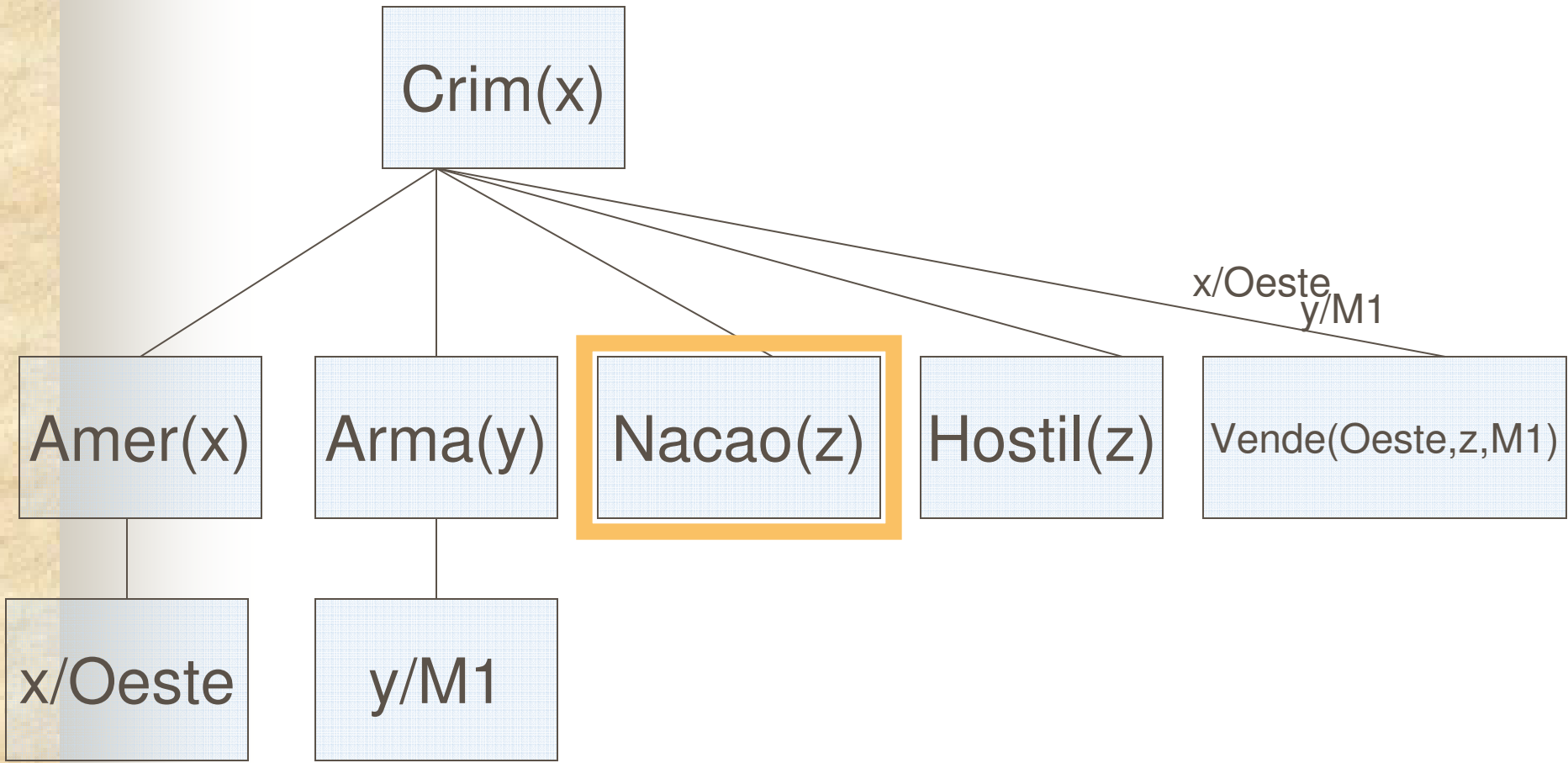
Vende(x,z,y)

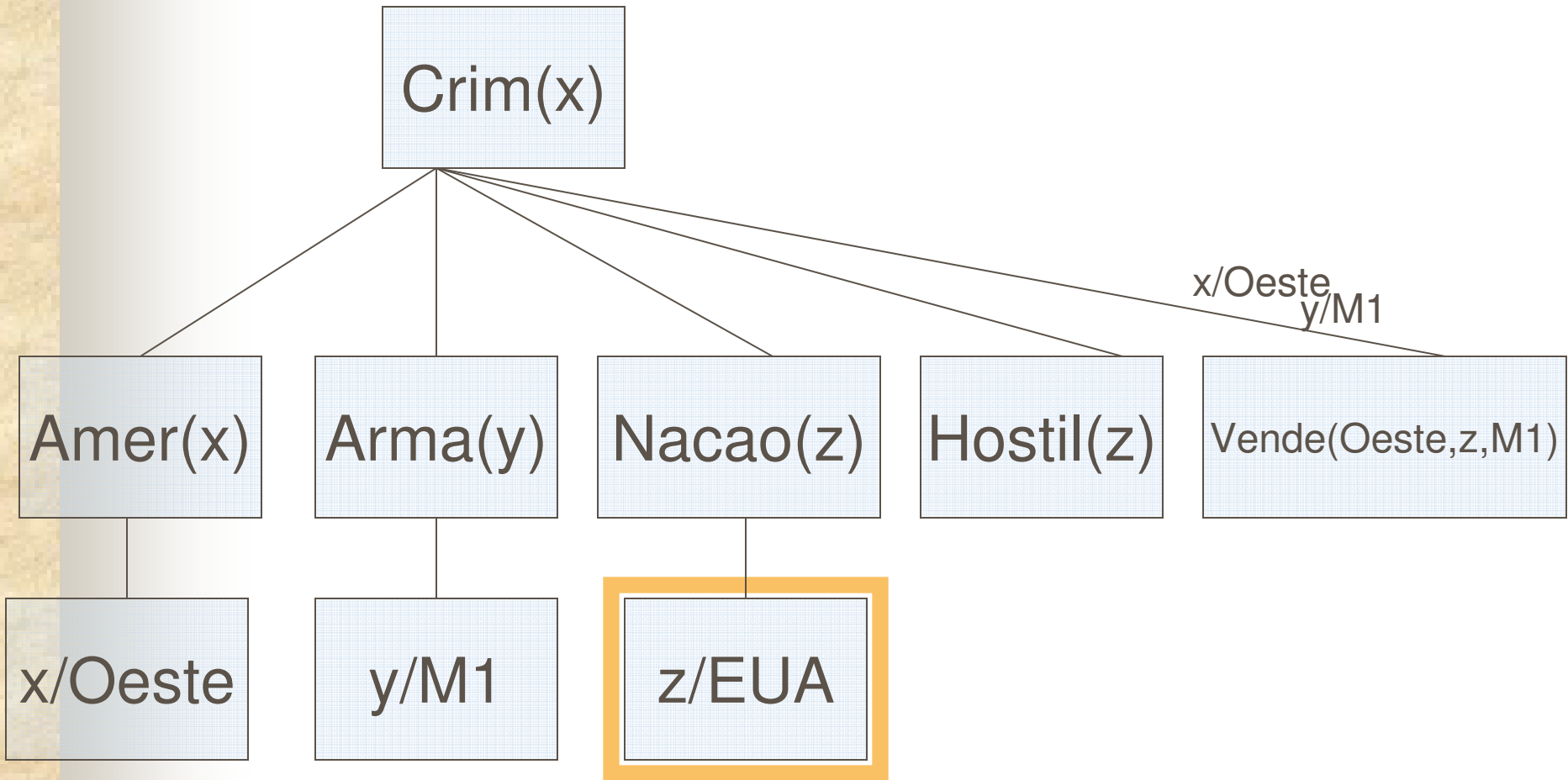
x/Oeste

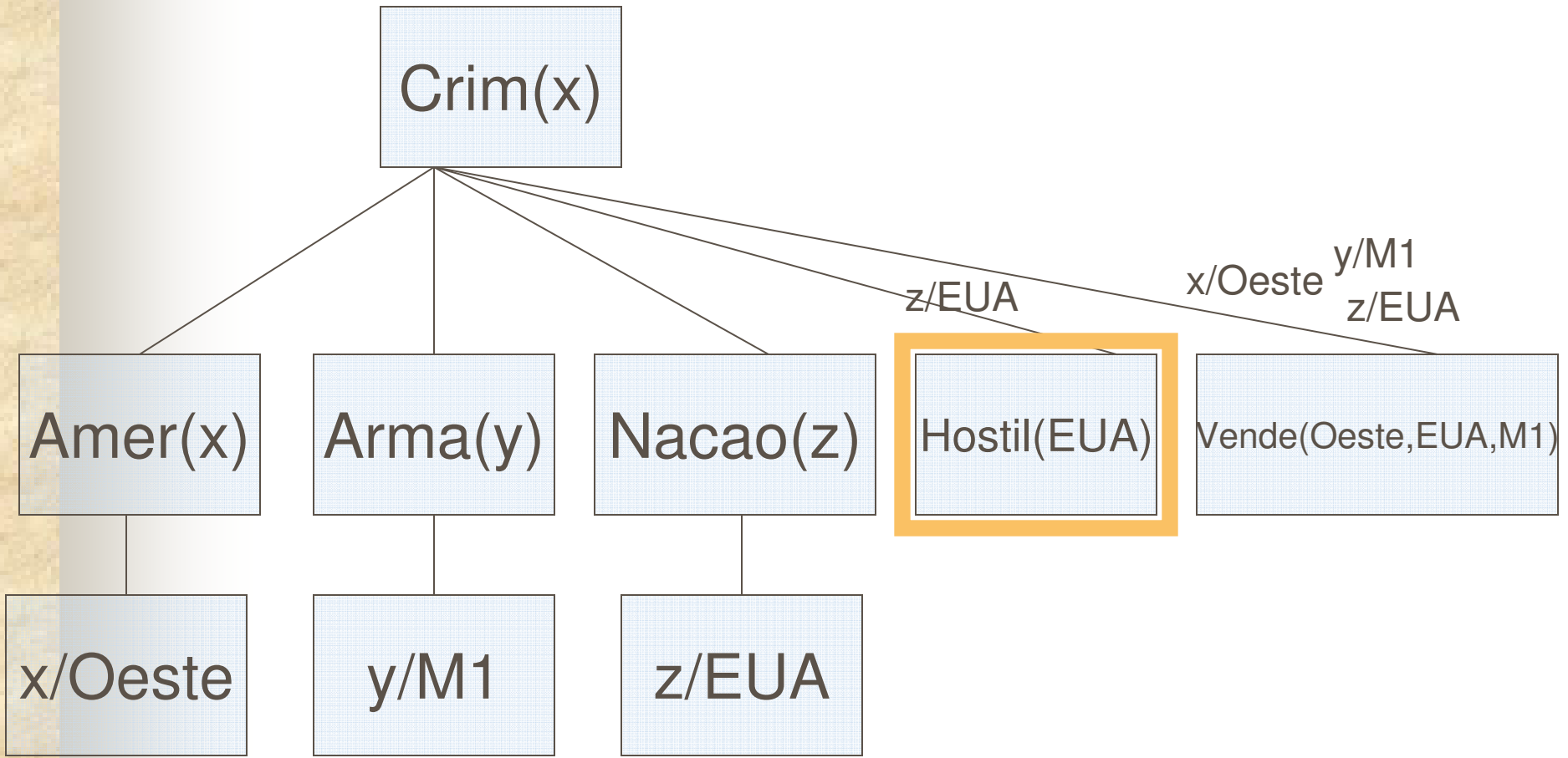


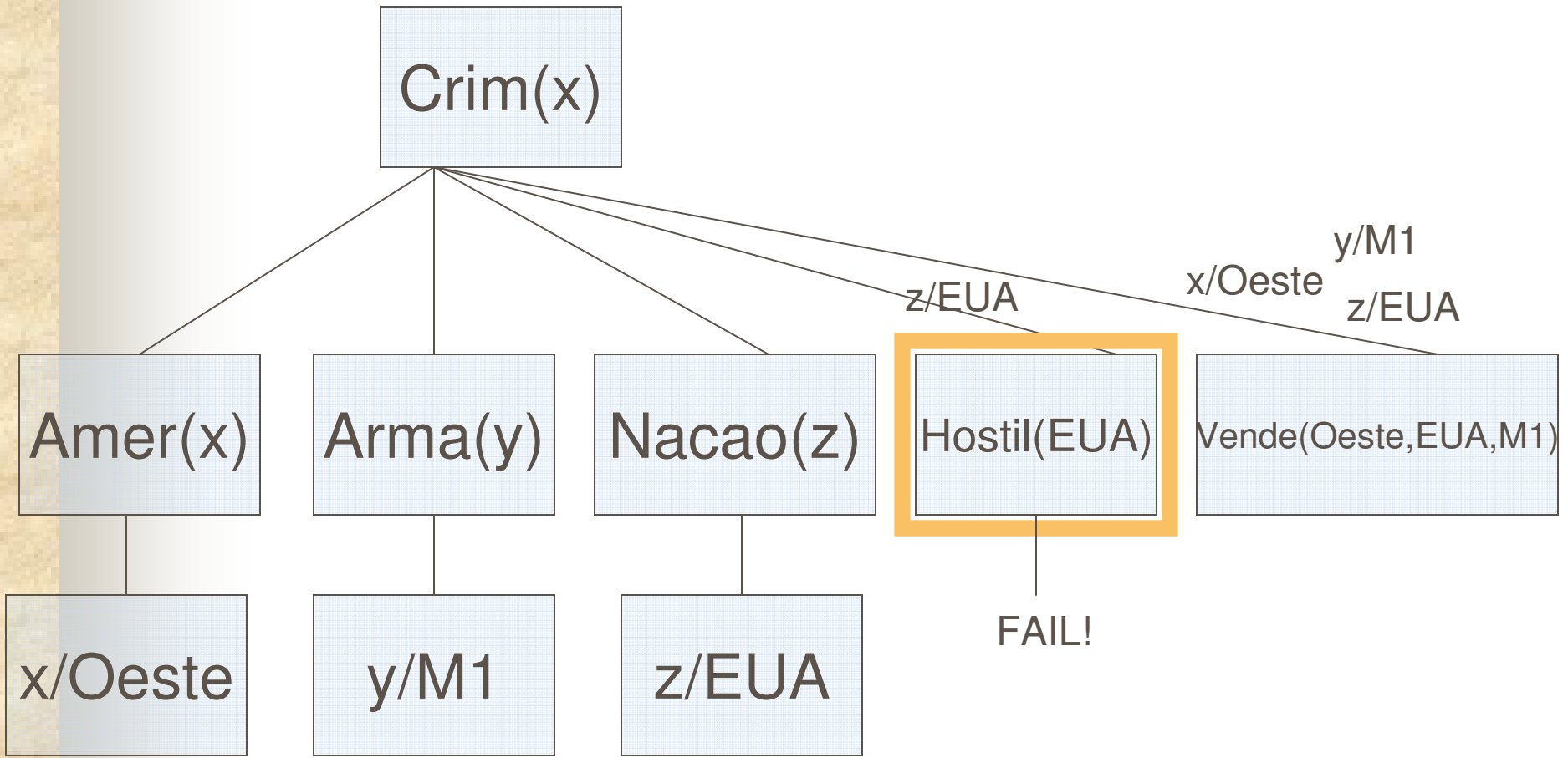


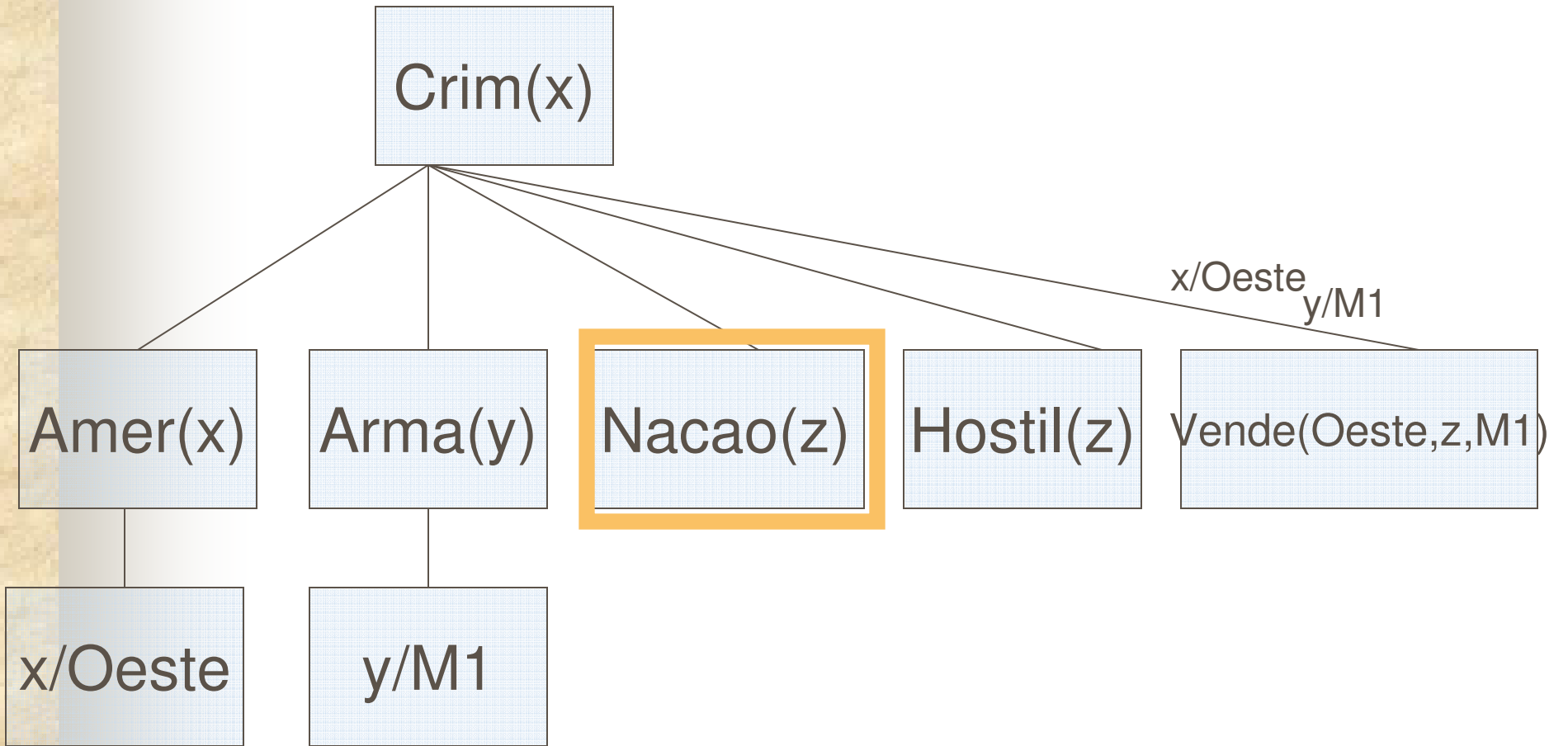


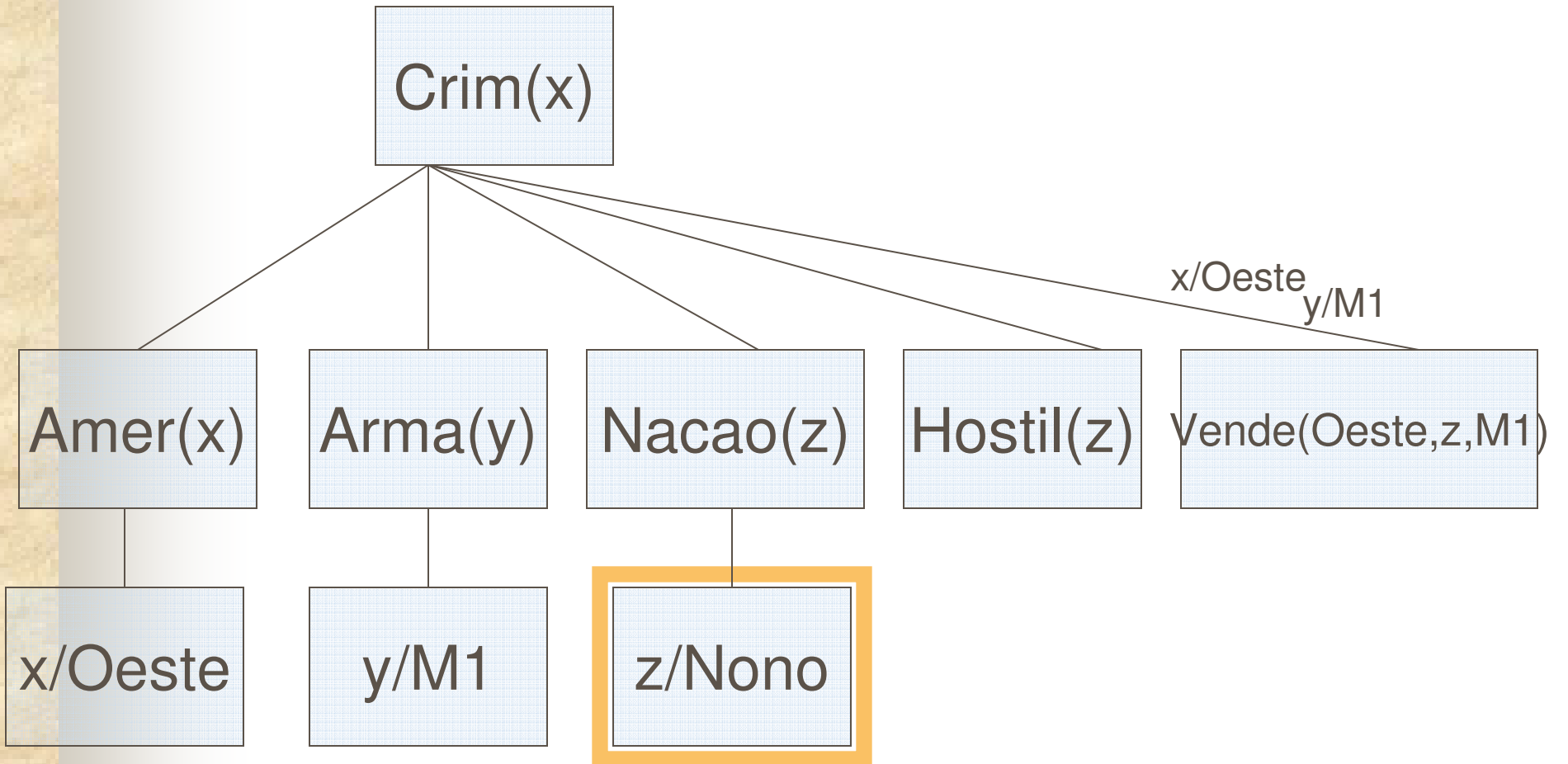




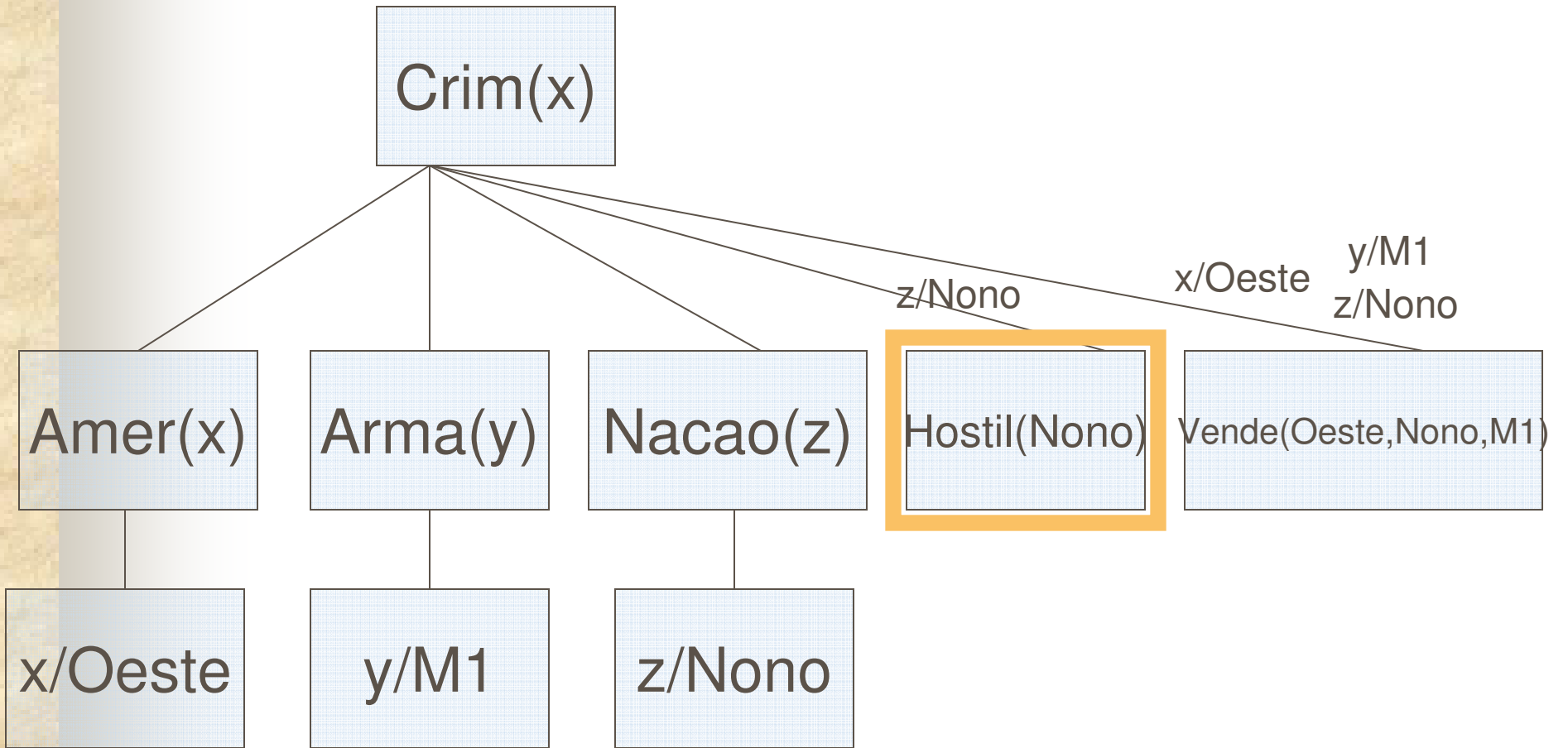


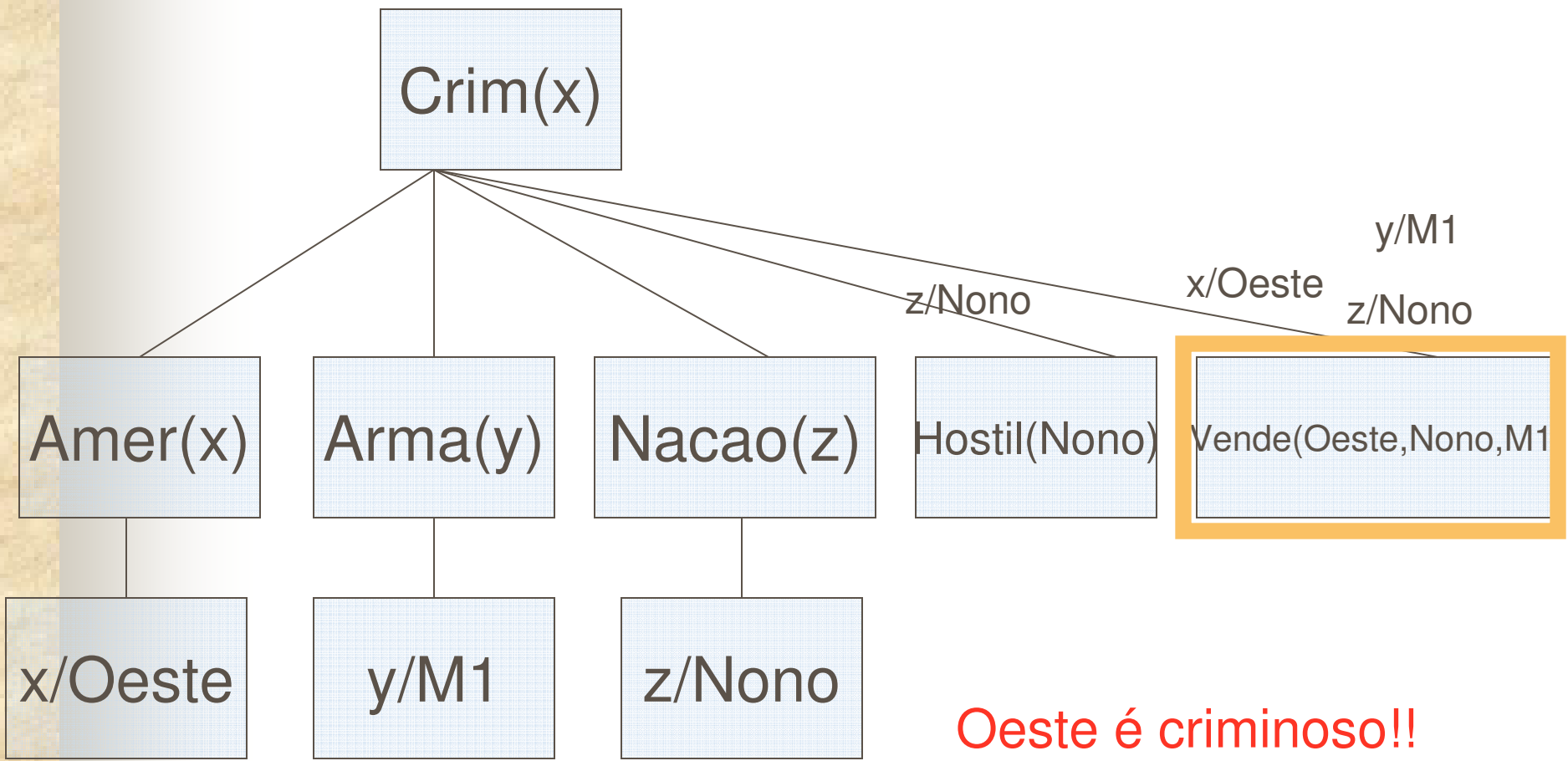












Oeste é criminoso!!



## MYCIN (Shortliffe *et al.*)

- Exemplo de regra:  
**IF** the infection is primary-bacteremia  
**AND** the site of the culture is one of the sterile sites  
**AND** the suspected portal of entry is the gastrointestinal tract  
**THEN** there is suggestive evidence (0.7) that infection is bacteroid.
- MYCIN usa uma notação que não é exatamente em língua inglesa



# MYCIN

## MYCIN -- A typical consultation

1. Patient's name: (first - last)

\*\* **FRED BRAUN**

(Questions 1-3 request background patient data).

2. Sex:

\*\* **M**

3. Age:

\*\* **55**

4. Are there any cultures for Fred Braun which may be related to the present illness, and from which organisms have been grown successfully in the microbiology laboratory?

\*\* **Y**

--- CULTURE - 1 ---

5. From what site was the specimen for CULTURE - 1 taken?

\*\* **BLOD**  
= **BLOOD**

(System corrects typing error).

6. Please give the date and time when CULTURE-1 was obtained.

(mo/da/yr time)

\*\* **JUN 20, 1977**

The first organism isolated from the blood culture of 20-JUN-77 (CULTURE-1) will be referred to as:

--- ORGANISM - 1 ---

7. Enter the laboratory - reported identity of ORGANISM-1:

\*\* **UNKNOWN**

(Typically identity is not known yet).

8. The strain (gram or Ziehl-Neelsen acid-fast) of ORGANISM-1:

\*\* **NEG**

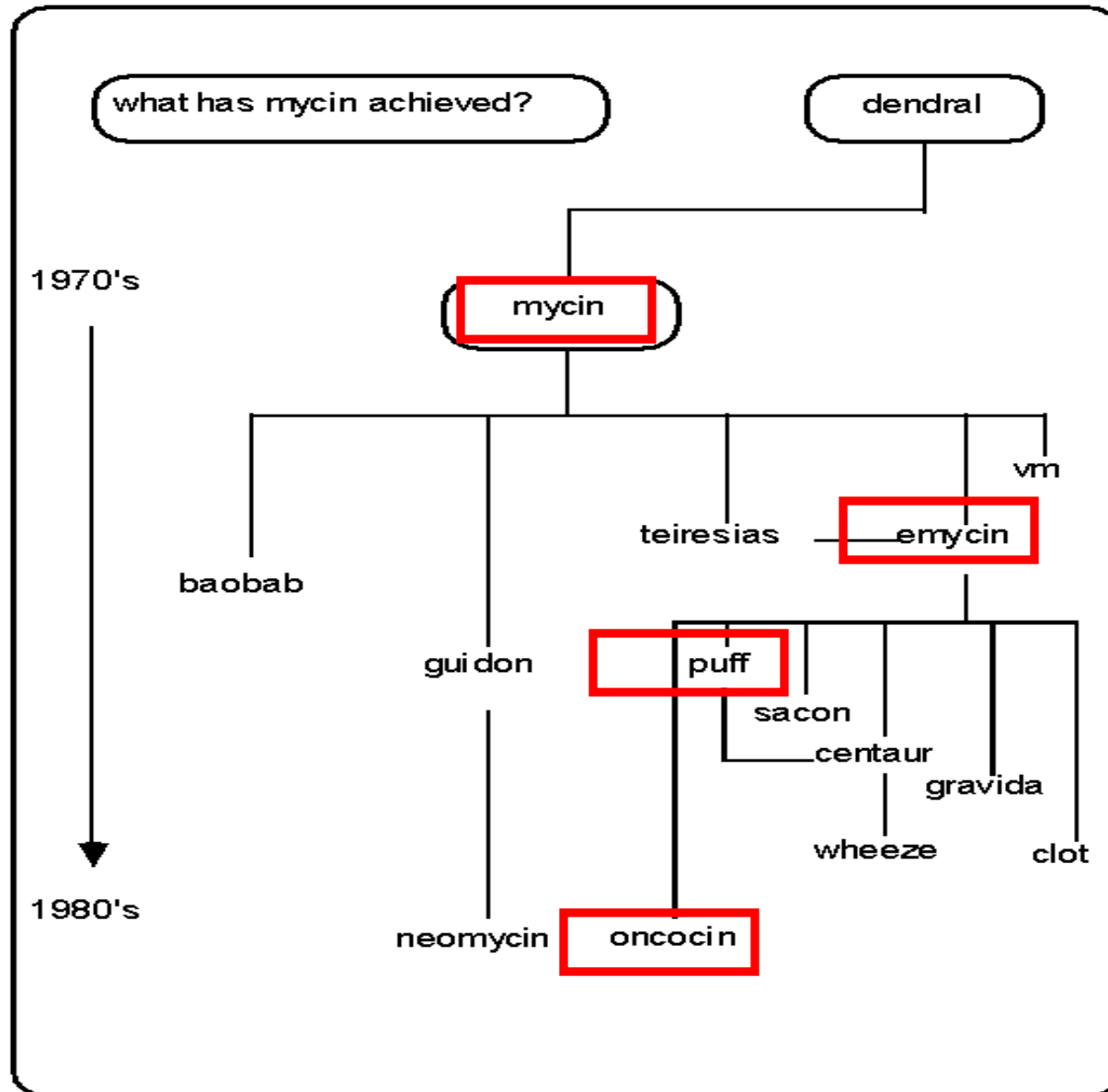


# MYCIN

- Sistema dedutivo:
  - Médico e paciente fornecem dados (anamnese e sintomas) que fazem MYCIN iniciar uma busca em profundidade, orientada ao objetivo para encontrar o “melhor” diagnóstico e forma de tratamento

# MYCIN

- Fonte: <http://www.computing.surrey.ac.uk/ai/PROFILE/mycin.html>





# Sistema especialista (apoio à decisão)

## ■ Vantagens

- Provê respostas consistentes para perguntas repetitivas, processos e tarefas
- Mantém níveis significantes de informação
- Motiva o esclarecimento da lógica da tomada de decisões
- Nunca esquece de perguntar alguma coisa 😊



# Sistema especialista

## ■ Desvantagens


- Falta de senso comum necessário em processos de tomada de decisão
- Não cria respostas em circunstâncias não usuais, como os humanos fazem
- Especialista nem sempre consegue explicar seu raciocínio
- erros no banco de dados podem levar a conclusões erradas
- Não conseguem se adaptar às modificações do ambiente, a menos que se modifique o banco de dados





# Outros métodos

- Árvores de decisão (decision trees)
- Clusterização (agrupamento - clustering)
- Baseados em explicação (explanation-based)
- Baseados em casos (case-based reasoning)
- Aprendizagem por reforço (reinforcement learning)
- Redes neuronais (neural networks)
- Algoritmos genéticos (genetic algorithms)
- Programação evolutiva (evolutionary programming)
- Estatísticos (statistical methods)
- Híbridos (mixture of the above...)
- .....

- 
- Programação lógica indutiva
  - Raciocínio com incertezas



# Sistemas de aprendizagem

- Aprendizagem de máquina?
  - Extração de informação relevante de dados, de forma automática, utilizando métodos computacionais ou estatísticos
  - Métodos podem ser dedutivos ou indutivos
- Dedução versus Indução?
- Indução é o raciocínio a partir de observações

# Raciocínio Dedutivo

$\mathcal{T}$

$\mathcal{U}$

$\mathcal{B}$

$\models$

$\mathcal{E}$

parent(X,Y) :- mother(X,Y)  
parent(X,Y) :- father(X,Y)

$\cup$

mother(penelope,victoria)  
mother(penelope,artur)  
father(christopher,victoria)  
father(christopher,artur)

$\models$

parent(penelope,victoria)  
parent(penelope,artur)  
parent(christopher,victoria)  
parent(christopher,artur)

# Raciocínio Indutivo

$\mathcal{E}$

$\mathcal{U}$

$\mathcal{B}$

$\models$

$\mathcal{T}$

parent(penelope,victoria)  
parent(penelope,artur)  
parent(christopher,victoria)  
parent(christopher,artur)



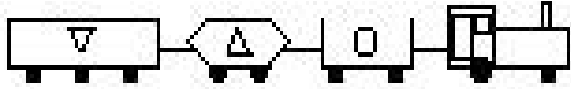


$\cup$

mother(penelope,victoria)  
mother(penelope,artur)  
father(christopher,victoria)  
father(christopher,artur)



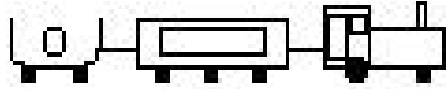
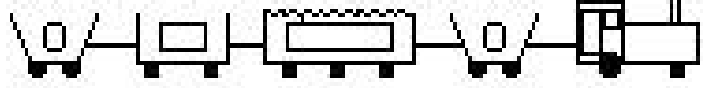

$\models$

parent(X,Y) :- mother(X,Y)  
parent(X,Y) :- father(X,Y)

## TRAINS GOING EAST

1. 
2. 
3. 
4. 
5. 

## TRAINS GOING WEST

1. 
2. 
3. 
4. 
5. 

# Programação Lógica Indutiva: exemplo



short(car\_12).

closed(car\_12).

long(car\_11).

long(car\_13).

short(car\_14).

open\_car(car\_11).

open\_car(car\_13).

open\_car(car\_14).

shape(car\_11,rectangle).

shape(car\_12,rectangle).

shape(car\_13,rectangle).

shape(car\_14,rectangle).

load(car\_11,rectangle,3).

load(car\_12,triangle,1).

load(car\_13,hexagon,1).

load(car\_14,circle,1).

wheels(car\_11,2).

wheels(car\_12,2).

wheels(car\_13,3).

wheels(car\_14,2).

has\_car(east1,car\_11).


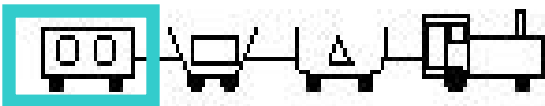

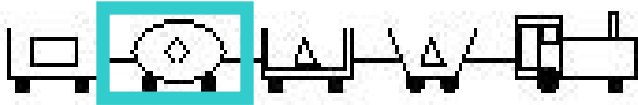
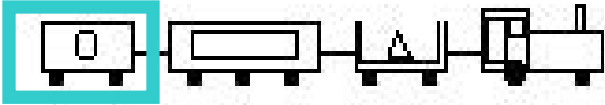
has\_car(east1,car\_12).

has\_car(east1,car\_13).

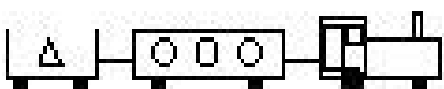

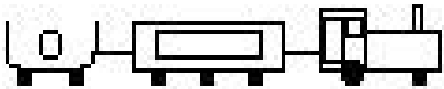
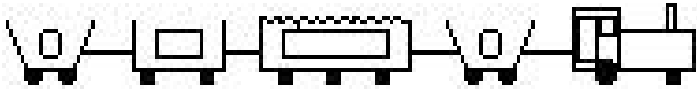
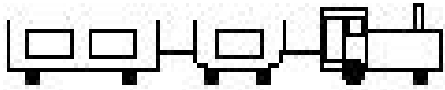
has\_car(east1,car\_14).

# Programação Lógica Indutiva: exemplo


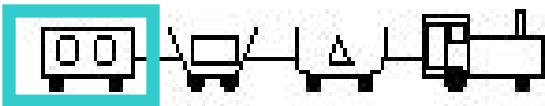

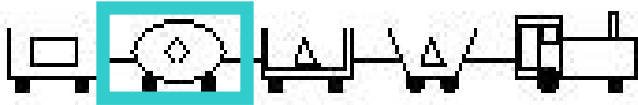
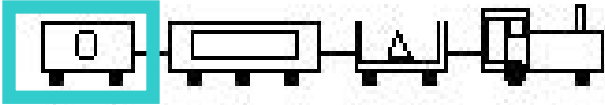
## TRAINS GOING EAST

1. 
2. 
3. 
4. 
5. 



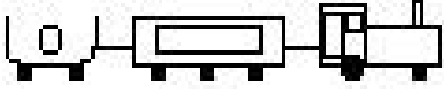
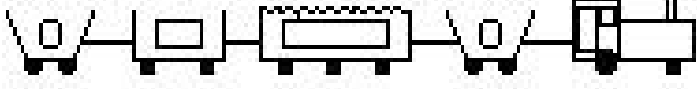
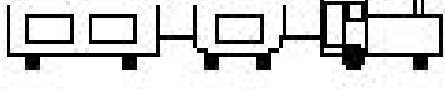
## TRAINS GOING WEST

1. 
2. 
3. 
4. 
5. 

## TRAINS GOING EAST

1. 
2. 
3. 
4. 
5. 

## TRAINS GOING WEST

1. 
2. 
3. 
4. 
5. 

eastbound(T) IF has\_car(T,C) AND short(C) AND closed(C)



## Outro exemplo menos trivial: extração de conhecimento relevante de mamografias

```
is_malignant(A) if
  'BIRADS_category' (A,b5), 'MassPA0' (A,present), 'Age' (A,age6570),
  previous_finding(A,B), 'MassesShape' (B,none),
  'Calc_Punctate' (B,notPresent),
  previous_finding(A,C), 'BIRADS_category' (C,b3).
```

Esta regra diz que A é um caso maligno SE:

**A** is classified as BI-RADS 5 AND had a mass present  
in a patient who:

- was between the ages of 65 and 70
- had two prior mammograms (**B**, **C**)

AND prior mammogram (**B**):

- had no mass shape described
- had no punctate calcifications

AND prior mammogram (**C**) was classified as BI-RADS 3

BI-RADS: **B**reast **I**maging **R**eporting **A**nd **D**ata **S**ystem



# Programação Lógica Indutiva

- Mais formalmente:
- Dados:
  - Conjuntos de exemplos **e** (observações, casos) rotulados como positivos ou negativos (classe **c**)
  - Uma linguagem
  - Possivelmente, um conjunto de restrições
- Encontrar:
  - Uma hipótese **h**, tal que  $\mathbf{h}(\mathbf{e}_i) = \mathbf{c}_i$
  - Para o maior número possível de exemplos



# Programação Lógica Indutiva

## ■ Vantagens:

- Utilização de uma linguagem fácil de interpretar, mais próxima do especialista
- Classificadores mais concisos
- Poder de representação: representa relações

## ■ Devantagens:

- Tamanho do espaço de busca para alguns problemas
- Classificação não probabilística



# Algoritmo?

- **Algoritmo**: lista de instruções bem definidas utilizadas para executar uma determinada tarefa
- Dado um estado inicial, o algoritmo passa por uma série de estados sucessivos bem definidos, **eventualmente** terminando
- A transição de um estado para outro não precisa ser **determinística**
- Alguns algoritmos são **probabilísticos** e incorporam aleatoriedade



# ILP: A Common Approach

- Use a greedy covering algorithm.
  - Repeat while some positive examples remain uncovered (not entailed):
    - Find a *good clause* (one that covers as many positive examples as possible but no/few negatives).
    - Add that clause to the current theory, and remove the positive examples that it covers.
- ILP algorithms use this approach but vary in their method for finding a *good clause*.



## Some ILP Systems

- PROGOL, ALEPH (top-down): saturates first uncovered positive example, and then performs top-down admissible search of the lattice above this saturated example.
- GOLEM (bottom-up), FOIL (top-down), LINUS/DINUS.
- Tilde, Claudien, IndLog, ...



# ILP Saturation

- Consists of building a *bottom clause* (seed)
- Incorporates background knowledge to an atomic formula
- Example:

metabolism(**A**) :-

essential(**A**, 'Non-Essential'), motif(**A**, 'PS00510'), chromosome(**A**, '14'),  
**interaction**(**A**, **B**, **C**, **E**),  
essential(**B**, 'Non-Essential'), motif(**B**, 'PS00188'), chromosome(**B**, '2'),  
**interaction**(**A**, **F**, **D**, **G**),  
intertype(**C**, 'Genetic'), intertype(**D**, ?),  
**interaction**(**B**, **A**, **C**, **E**),  
**interaction**(**B**, **H**, **C**, **I**),  
**interaction**(**F**, **A**, **D**, **G**),  
**interaction**(**H**, **B**, **C**, **I**), **interaction**(**H**, \_, \_, \_).



# ILP: Aleph

- Procedure to extract theories from examples
- Complete (branch-and-bound) search for best clause in the *whole* space
- Search subject to several user control settings
  - Max clause length
  - Max chaining length
  - Minacc
  - Max nodes
  - Search strategy, etc.





# ILP: Aleph

- Aleph

- Desenvolvido na Universidade de Oxford por Ashwin Srinivasan

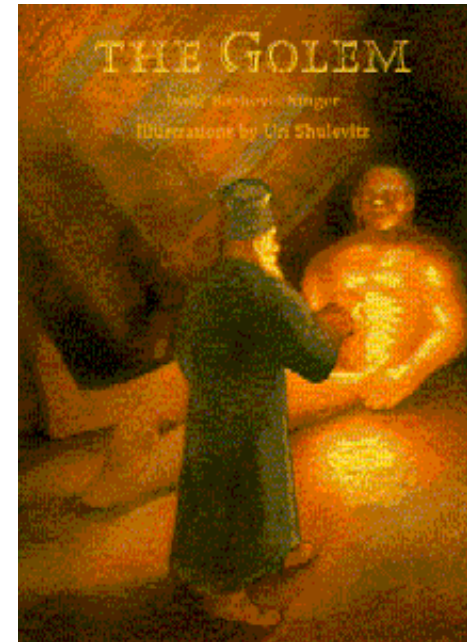
<http://www.comlab.ox.ac.uk/oucl/research/areas/machlearn/Aleph/>

# ILP: Aleph

Then the Rabbi said,

*“Golem, you have not been completely formed, but I am about to finish you now... You will do as I will tell you.”*

Saying these words, Rabbi Leib finished engraving the letter **Aleph**. Immediately the golem began to rise.





# Aleph: algoritmo

- Estado inicial:
  - Exemplos ou observações
  - Descrições: conhecimento prévio ou background knowledge (BK)
- Estado final: hipótese ou teoria ou modelo
- Transições: hipóteses intermediárias



# Aleph: algoritmo

- Select example
- Build most-specific-clause (**bottom clause**)
- Search. Find a clause more general than the bottom clause
- Remove redundant. The clause with the best score is added to the current theory, and all examples made redundant are removed. This step is sometimes called the "**cover removal**" step. Note here that the best clause may make clauses other than the examples redundant
- Return to first step



# Aleph: Knowledge Representation

**Input Files: Prolog Syntax**

**dtp.b: BK**

**dtp.f: pos examples**

**dtp.n: neg examples**



## Representation: BK

**chromosome('G234064','1').**  
**chromosome('G234065','1').**  
**chromosome('G234070','1').**  
**chromosome('G234073','1').**  
**chromosome('G234074','1').**  
**chromosome('G234076','1').**  
**chromosome('G234084','2').**  
**chromosome('G234085','2').**  
**chromosome('G234089','2').**



## Representation: BK

**interaction('G234062','G235011','Physical',?).**

**interaction('G234064','G234126','Genetic-Physical','0.9141').**

**interaction('G234064','G235065','Genetic-Physical','0.7515').**

**interaction('G234064','G235571','Physical','0.9691').**

**interaction('G234065','G234073','Physical','0.7492').**

**interaction('G234065','G235042','Physical','-0.4659').**



# Representation: Examples

**metabolism('G239098').**

**metabolism('G234980').**

**metabolism('G235245').**

**metabolism('G234108').**

**metabolism('G238387').**

**metabolism('G240504').**

**metabolism('G236733').**





# Example of clause learned

*metabolism(A) :-*

*chromosome(A, '15'),*

*interaction(A, B, \_, \_),*

*complex(B, 'Transcription*

*complexes/Transcriptosome').*

*A* and *B* are variables that represent **genes**



# Aleph: algoritmo

- Exemplo: trens que vão para leste e trens que vão para oeste



# Aleph: algoritmo

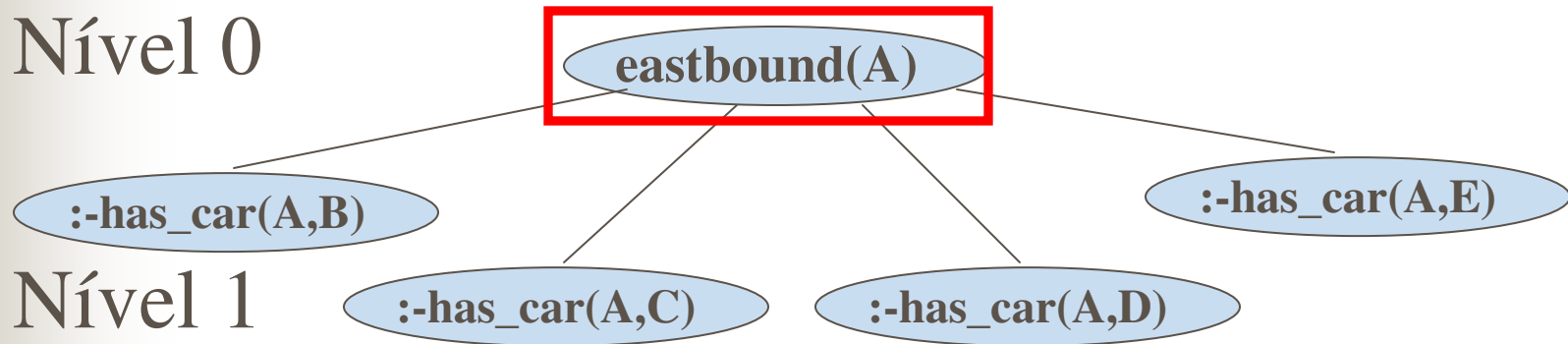
## ■ Saturação:

eastbound(A) :-

```
has_car(A,B), has_car(A,C), has_car(A,D), has_car(A,E),  
short(B), short(D), closed(D), long(C),  
long(E), open_car(B), open_car(C), open_car(E),  
shape(B,rectangle), shape(C,rectangle), shape(D,rectangle),  
shape(E,rectangle),  
wheels(B,2), wheels(C,3), wheels(D,2), wheels(E,2),  
load(B,circle,1), load(C,hexagon,1), load(D,triangle,1),  
load(E,rectangle,3).
```

# Aleph: Busca

Nível 0



Nível 1

# Aleph: Busca

Nível 0

eastbound(A)

**:-has\_car(A,B)**

**:-has\_car(A,E)**

Nível 1

**:-has\_car(A,C)**

**:-has\_car(A,D)**

short(B)

open\_car(B)

shape(B,rectangle)

wheels(B,2)

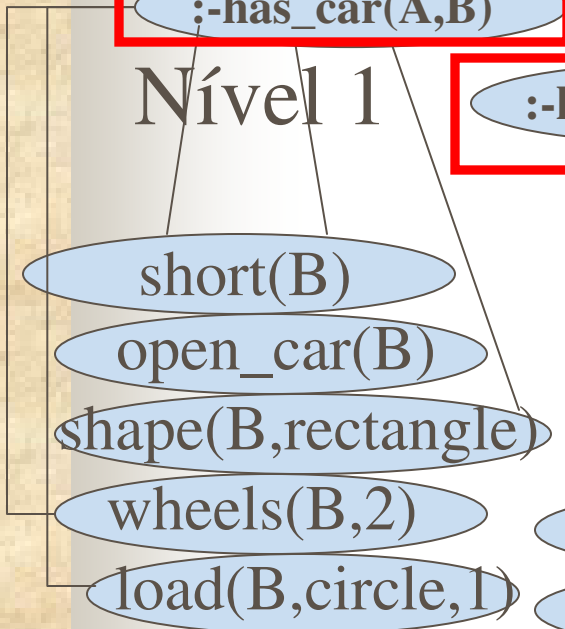
load(B,circle,1)

Nível 2

has\_car(A,C)

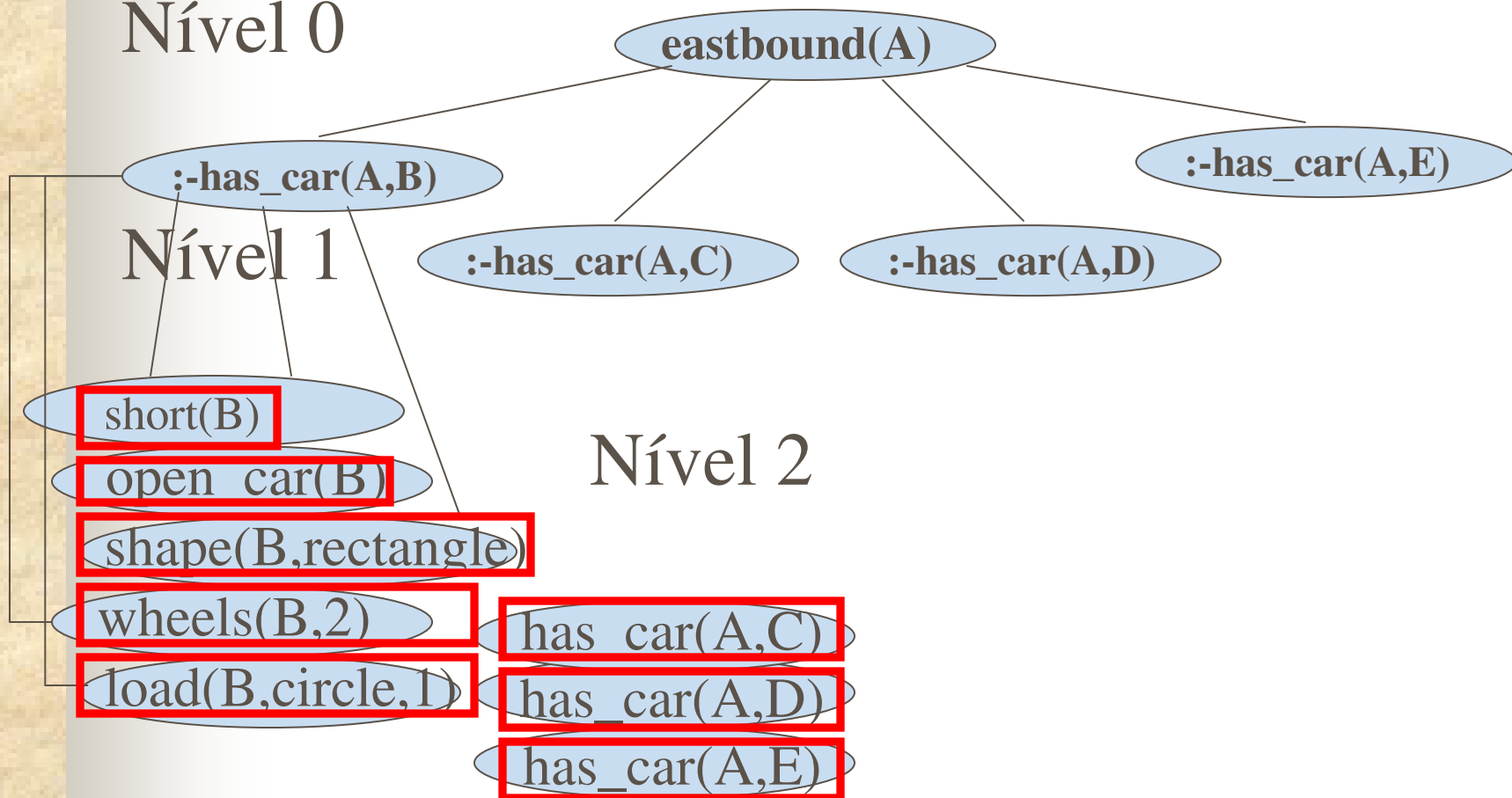
has\_car(A,D)

has\_car(A,E)



# Aleph: Busca

Nível 0





# Aleph: algoritmo

- Busca: cláusula mais geral

**eastbound(A) :-**

```
has_car(A,B), has_car(A,C), has_car(A,D), has_car(A,E),  
short(B), short(D), closed(D), long(C),  
long(E), open_car(B), open_car(C), open_car(E),  
shape(B,rectangle), shape(C,rectangle), shape(D,rectangle),  
shape(E,rectangle),  
wheels(B,2), wheels(C,3), wheels(D,2), wheels(E,2),  
load(B,circle,1), load(C,hexagon,1), load(D,triangle,1),  
load(E,rectangle,3).
```



# Aleph: algoritmo

- Busca: adiciona “filhos” possíveis (literais candidatos)

eastbound(A) :-

```
has_car(A,B), has_car(A,C), has_car(A,D), has_car(A,E),  
short(B), short(D), closed(D), long(C),  
long(E), open_car(B), open_car(C), open_car(E),  
shape(B,rectangle), shape(C,rectangle), shape(D,rectangle),  
shape(E,rectangle),  
wheels(B,2), wheels(C,3), wheels(D,2), wheels(E,2),  
load(B,circle,1), load(C,hexagon,1), load(D,triangle,1),  
load(E,rectangle,3).
```





# Aleph: algoritmo

- Busca: adiciona “filhos” possíveis ao primeiro filho

eastbound(A) :-

has\_car(A,B), has\_car(A,C), has\_car(A,D), has\_car(A,E),  
short(B), short(D), closed(D), long(C),  
long(E), open\_car(B), open\_car(C), open\_car(E),  
shape(B,rectangle), shape(C,rectangle), shape(D,rectangle),  
shape(E,rectangle),  
wheels(B,2), wheels(C,3), wheels(D,2), wheels(E,2),  
load(B,circle,1), load(C,hexagon,1), load(D,triangle,1),  
load(E,rectangle,3).



# Aleph: algoritmo

- Busca: segundo filho de nível 1

**eastbound(A) :-**

```
has_car(A,B), has_car(A,C), has_car(A,D), has_car(A,E),  
short(B), short(D), closed(D), long(C),  
long(E), open_car(B), open_car(C), open_car(E),  
shape(B,rectangle), shape(C,rectangle), shape(D,rectangle),  
shape(E,rectangle),  
wheels(B,2), wheels(C,3), wheels(D,2), wheels(E,2),  
load(B,circle,1), load(C,hexagon,1), load(D,triangle,1),  
load(E,rectangle,3).
```



# Aleph: algoritmo

- Busca: filhos do segundo filho de nível 1

eastbound(A) :-

has\_car(A,B), has\_car(A,C), has\_car(A,D), has\_car(A,E),  
short(B), short(D), closed(D), long(C),  
long(E), open\_car(B), open\_car(C), open\_car(E),  
shape(B,rectangle), shape(C,rectangle), shape(D,rectangle),  
shape(E,rectangle),  
wheels(B,2), wheels(C,3), wheels(D,2), wheels(E,2),  
load(B,circle,1), load(C,hexagon,1), load(D,triangle,1),  
load(E,rectangle,3).



# Aleph: example of run

→ aleph\_trains



# Aleph: how to run?

- You need to have a Prolog system
  - Yap: <http://yap.sourceforge.net> OU
  - SWI: <http://www.swi-prolog.org>
- Aleph:  
<http://www.comlab.ox.ac.uk/oucl/research/areas/machlearn/Aleph/>
- Files: .b, .f, .n
- To make things easier: everything in the same directory!



# Aleph: Comandos básicos

- `read_all`
- `reduce`
- `induce`



# Aleph: Parameters

```
:- set(clauselength,5).
:- set(depth, 200).
:- set(i,3).
:- set(noise,0).
:- set(minacc,0.7).
:- set(nodes,1000000).
:- set(m,20).
:- set(evalfn,mestimate).
:- set(test_pos,'/u/dutra/Protein/prot_test_set.f').
:- set(test_neg,'/u/dutra/Protein/prot_test_set.n').
:- set(optimise_clauses,true).

:- set(record,true).
:- set(recordfile,'prot_train_set.out').
:- set(samplesize,0).
```

$Strength\ estimate = (support + m * prior) / (coverage + m)$

$M \rightarrow 0, strength \rightarrow precision$

Support = True positives

Coverage = True positives + false negatives



# Aleph: Modes and Types

- :- **modeh**(1, eastbound(+**train**)).
- :- **modeb**(1, short(+car)).
- :- modeb(1, closed(+car)).
- :- modeb(1, long(+car)).
- :- modeb(1, open\_car(+car)).
- :- modeb(1, double(+car)).
- :- modeb(1, jagged(+car)).
- :- modeb(1, shape(+car, **#shape**)).
- :- modeb(1, load(+car, **#shape**, **#int**)).
- :- modeb(1, wheels(+car, **#int**)).
- :- modeb(\*, has\_car(+train, **-car**)).

- :- **determination**(eastbound/1, short/1).
- :- determination(eastbound/1, closed/1).
- :- determination(eastbound/1, long/1).
- :- determination(eastbound/1, open\_car/1).
- :- determination(eastbound/1, double/1).
- :- determination(eastbound/1, jagged/1).
- :- determination(eastbound/1, shape/2).
- :- determination(eastbound/1, wheels/2).
- :- determination(eastbound/1, has\_car/2).
- :- determination(eastbound/1, load/3).





# Aleph: Modes and Types

- `:- modeh(1,metabolism(+gene)).`
- `:- modeb(1,essential(+gene,#essential)).`
- `:- modeb(1,class(+gene,#class)).`
- `:- modeb(1,complex(+gene,#complex)).`
- `:- modeb(1,phenotype(+gene,#phenotype)).`
- `:- modeb(1,motif(+gene,#motif)).`
- `:- modeb(1,chromosome(+gene,#chromosome)).`
- `:- modeb(*,gte(+number,#number)).`
- `:- modeb(*,interaction(+gene,-gene,-intertype,-number)).`
- `:- modeb(1,intertype(+intertype,#intertype)).`



# Case study 1: Learning rules for early diagnosis of rheumatic diseases

- Correct diagnosis in the early stage of a rheumatic disease is a difficult problem [Pirnat et al. 1989]
- Having passed all investigations, many patients can not be reliably diagnosed after their first visit to the specialist
- Two reasons:
  - symptoms, clinical manifestations, laboratory and radiological findings of various rheumatic diseases are very similar and not specific
  - subjective interpretation of anamnestic, clinical, laboratory and radiological data



# Case study 1: rheumatic disease

- Application of LINUS to the problem of learning rules for early diagnosis of rheumatic diseases.
- Given: attribute-value descriptions of patient data, bk provided by a medical specialist in the form of typical co-occurrences of symptoms
- Experiments: LINUS with CN2
- Showed that the noise-handling mechanism of CN2 and the ability of LINUS to use bk affect the performance (classification accuracy and information content) and the complexity of the induced diagnostic rules



## Case study 1: rheumatic disease

- Data about 462 patients (Univ medical center of ljubljana)
- Over 200 rheumatic diseases that can be grouped into 3, 6, 8 or 12 diagnostic classes
- 8 classes: suggested by a specialist



# Case study 1: rheumatic disease

Class	Name	Num patients
A1	Degenerative spine diseases	158
A2	Degenerative joint diseases	128
B1	Inflammatory spine diseases	16
B234	Other inflammatory diseases	29
C	Extra-articular rheumatism	21
D	Crystal-induced synovitis	24
E	Non-specific rheumatism manifestations	32
F	Non rheumatic diseases	54



# Case study 1: rheumatic disease

- Experiments on anamnestic data without patient's clinical manifestations, laboratory and radiological findings
- 16 anamnestic attributes:
  - sex, age, family anamnesis, duration of present symptoms, duration of rheumatic diseases, joint pain (arthrotic or arthritic), number of painful joints, number of swollen joints, spinal pain, other pain, duration of morning stiffness, skin manifestations, mucosal manifestations, eye manifestations, other manifestations and therapy.
- From 462 patients, 8 were incomplete, 12 attribute values missing (sex and age) (not a problem since LINUS with CN2 handles missing data)



# Case study 1: rheumatic disease

- Medical bk: augmented the patient data with typical co-occurrences of symptoms (diagnostic knowledge)
- 6 typical groups suggested by the specialist:

# Case study 1: rheumatic disease

Joint pain	Morning stiffness
No pain	$\leq 1h$
arthrotic	$\leq 1h$
arthritic	$> 1h$

spinal pain	Morning stiffness
No pain	$\leq 1h$
spondylotic	$\leq 1h$
spondylitic	$> 1h$

sex	Other pain
male	thorax
male	heels

Joint pain	Spinal pain
No pain	spondylotic
arthrotic	No pain
No pain	spondylitic
arthritic	spondylitic
arthritic	No pain
No pain	No pain



# Case study 1: rheumatic disease

Joint pain	Spinal pain	Painful joints
No pain	spondylotic	0
arthrotic	No pain	$1 \leq \text{joints} \leq 30$
No pain	spondylitic	0
arthrotic	spondylitic	$1 \leq \text{joints} \leq 5$
arthritic	No pain	$1 \leq \text{joints} \leq 30$
No pain	No pain	0

Swollen joints	Painful joints
0	0
0	$1 \leq \text{joints} \leq 30$
$1 \leq \text{joints} \leq 10$	$0 \leq \text{joints} \leq 30$



# Case study 1: rheumatic disease

- Example of rules:

```
% grouping3( sex/input, location/input, grouping3_value/output)  
grouping3( male, thorax, male_thorax ) :- !.  
grouping3( male, heels, male_heels ) :- !.  
grouping3( -, -, irrelevant ).
```

# Case study 1: rheumatic disease

bk	Signif test	Acc (%)	Relative inf score (%)	Num of rules	Num of literals
no	no	62.8	49	96	302
no	yes	51.7	22	30	102
yes	no	72.9	59	96	301
yes	yes	52.4	30	38	120



# Medical evaluation

- Specialist evaluated the entire set of induced rules
- For each of the conditions in a rule:
  - +1 if the condition favours the diagnosis made by the rule
  - -1 if the condition was against the diagnosis
  - 0 if the condition is irrelevant
- Mark of a rule: sum of the points for all conditions in the rule
- Actual marks range from -1 to 3
  - 3: rules which are very characteristic for a disease
  - 2: good, correct rules
  - 1: not wrong, but not too characteristic for the disease
  - 0: by chance
  - -1: misleading rules

# Medical evaluation: sem BK

class	Num rules with mark					rules	avgm
	3	2	1	0	-1		
A1			4	1	2	7	0.29
A2			3	2	1	6	0.33
B1		1	2			3	1.33
B2			3	1		4	0.75
C			1	2		3	0.33
D		1	2			3	1.33
E				3		3	0.00
F				1		1	0.00

# Medical evaluation: com BK

class	Num rules with mark					rules	avgm
	3	2	1	0	-1		
A1		3	2	2		7	1.14
A2	1	1	3	1	1	7	1.00
B1		1	2			3	1.33
B2	1	2	4			7	1.57
C				3		3	0.00
D		1	2			3	1.33
E				4		4	0.00
F	1	1	1	1		4	1.50

# Medical evaluation: com BK

*Class = degenerative\_spine\_diseases* **if**

**Mark: 2**

*Duration\_of\_present\_symptoms > 6.5\_months,*

*Duration\_of\_rheumatic\_diseases < 5.5\_years,*

*Number\_of\_painful\_joints > 16,*

*grouping2(Spinal\_pain, Duration\_of\_morning\_stiffness, Value),*

*Value = 'spondylotic & dms =< 1\_hour'.*



# Medical evaluation: com BK

*Class = nonrheumatic-diseases* **if**

**Mark: 3**

*Age < 53.5,*

*Duration\_of\_present\_symptoms > 2\_months,*

*Number\_of\_swollen\_joints < 0.5,*

*Other\_pain = no,*

*Eye\_manifestations = no,*

*grouping5(Joint\_pain, Spinal\_pain, Number\_of\_painful\_joints, Value),*

*Value = 'no\_pain & no\_pain & pj = 0'.*





# Medical evaluation

- Use of bk provided by specialist helps to guide the search to obtain new knowledge
- System can work and infer the specialist's knowledge **plus** new knowledge, but it will probably take much more time ☹



## Case study 2: drug discovery

- Given:
  - Molecules active and inactive for dtp
  - Their description in terms of coordinates and bonds
- Find small structures that model active molecules



## Case study 2: drug discovery

- Examples of dtp groups:

**hydrophobic**(m752,

hyphob([a2, a3, a5, a8, a7, a4, a2],  
2.16452, -0.833917, 3.6379)).

**hacc**(m9706,

hacc(a10, -6.2969, -1.3684, -0.4631)).



# Case study 2: drug discovery

## ■ Utilisation of refinement operator

refine(false,Clause):-

```
member(Point1, [hydrophobic(M,P1), hdonor(M,P1),halogen(M,P1),hacc(M,P1)]),  
member(Point2,[hydrophobic(M,P2),hdonor(M,P2),halogen(M,P2),hacc(M,P2)]),  
Clause = (active(M) :- Point1, Point2, dist(M,P1,P2,D1,E)).
```

refine(Clause1,Clause2):-

```
Clause1 = (active(M) :- Point1,Point2, dist(M,P1,P2,D1,E)),  
member(Point3,[hydrophobic(M,P3),hdonor(M,P3),halogen(M,P3),hacc(M,P3)]),  
Clause2 = (active(M) :- Point1, Point2, dist(M,P1,P2,D1,E),  
          Point3, dist(M,P1,P3,D2,E), dist(M,P2,P3,D3,E)).
```

## ■ Reduce search space!!!



# Como avaliar resultados?

- Conjunto de treino?
- Como verificar se o classificador encontrado (teoria) comporta-se bem para novos exemplos (que nunca foram vistos antes?)
- Conjunto de ajuste (tuning set)
- Métricas:
  - Accuracy
  - Receiver operating characteristic (ROC)
  - Precision-recall (PR)
  - Area under the curve (AUC)

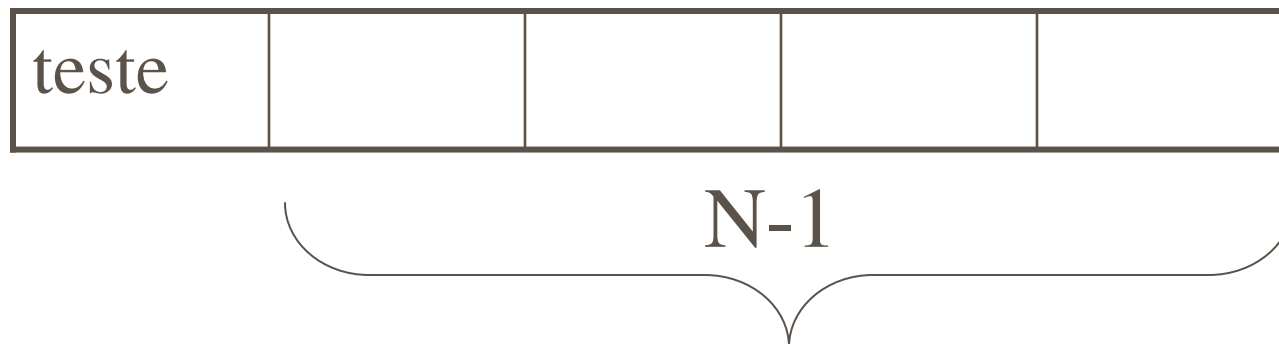


# Como avaliar resultados?

- Classificadores separam:
  - TP: True positives
  - TN: True negatives
  - FP: False positives
  - FN: False negatives

# Como avaliar resultados?

- Para minimizar erro do classificador em exemplos nunca vistos: cross-validation
- Particiona o conjunto de treino em  $n$  partes iguais. Treina em  $n-1$  e testa no  $n$ -ésimo conjunto. Repete  $n$  vezes





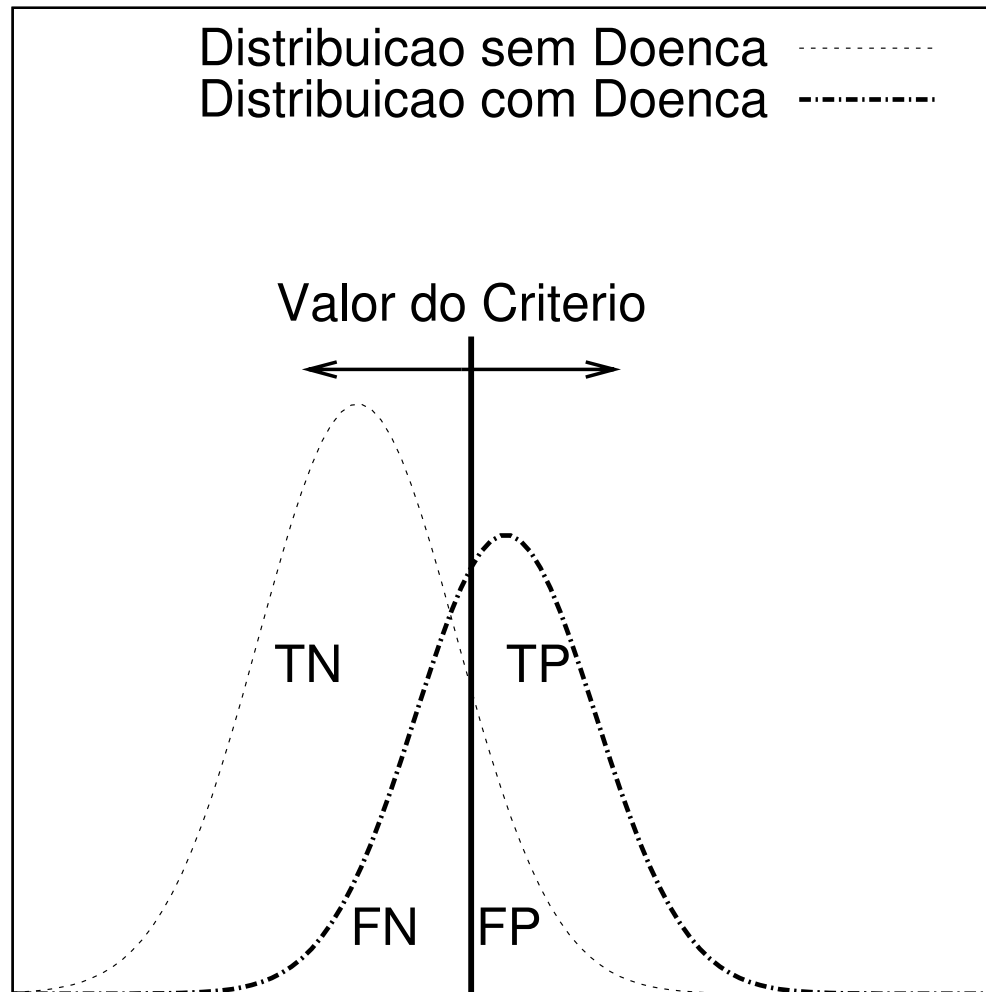
## Como avaliar resultados?

- Leave-one-out: cross-validation onde temos  $n$  exemplos, treinamos em  $n-1$  e deixamos 1 único exemplo para teste
- Problemas com cross-validation: sobreposição de exemplos em cada conjunto de treino
- Segundo Dietterich: 5 times 2-fold cross-validation should be used



# Avaliação

Densidade de Probabilidade para os Resultados



Resultado dos Testes

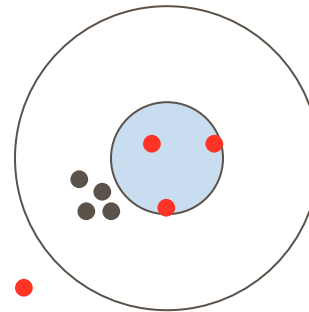


# Como avaliar resultados?

- Tuning set?
- Geralmente utilizado para estimar parâmetros

# Métricas

## ■ Accuracy x Precision



- Accuracy
- Precision

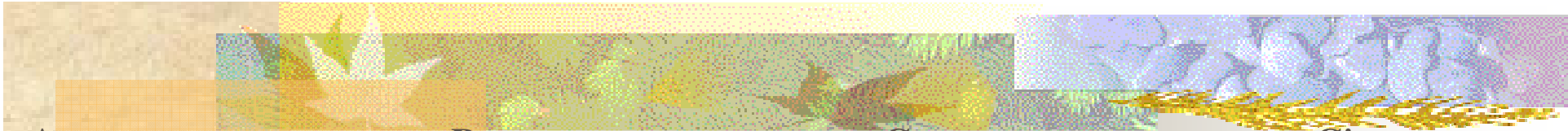
$$P = TP / (TP+FP)$$

$$Acc1 = (TP+TN)/Totex$$

$$Acc2 = (TP/(TP+FP) + TN/(TN+FN)) / 2$$

Tx acerto pos

Tx acerto neg



A			B			C			C'		
TP=63	FP=28	91	TP=77	FP=77	154	TP=24	FP=88	112	TP=88	FP=24	112
FN=37	TN=72	109	FN=23	TN=23	46	FN=76	TN=12	88	FN=12	TN=76	88
100	100	200	100	100	200	100	100	200	100	100	200
TPR = 0.63			TPR = 0.77			TPR = 0.24			TPR = 0.88		
FPR = 0.28			FPR = 0.77			FPR = 0.88			FPR = 0.24		
ACC = 0.68			ACC = 0.50			ACC = 0.18			ACC = 0.82		

