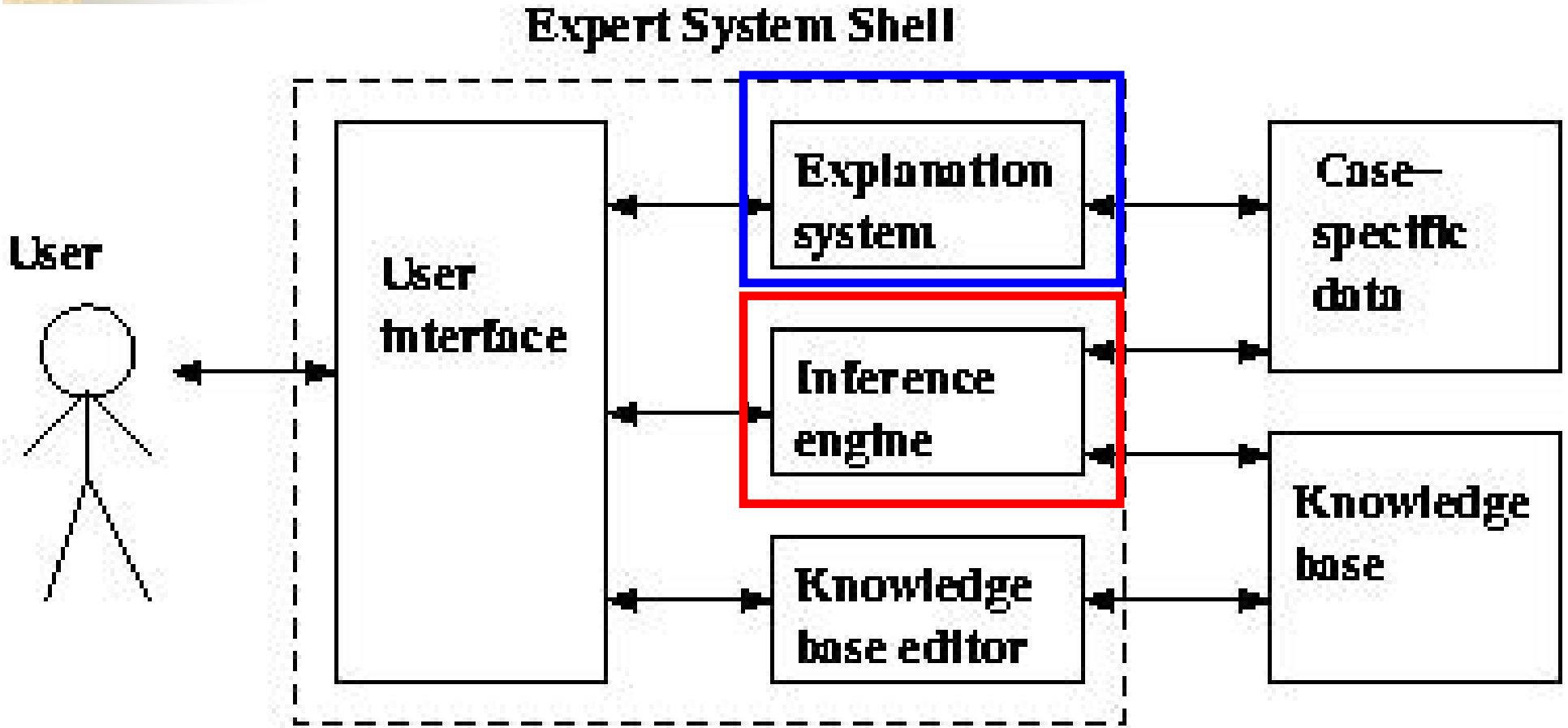
Sistemas especialistas



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) forall x,y,z Amer(x) E Arma(y) E Nacao(z) E Hostil(z) E Vende(x,z,y) → 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) → Vende(Oeste,Nono,x)

(4) Para todo x Missil(x) → Arma(x)

(5) Para todo x Inimigo(x,EUA) → Hostil(x)

Fatos:

(6) Americano(Oeste)

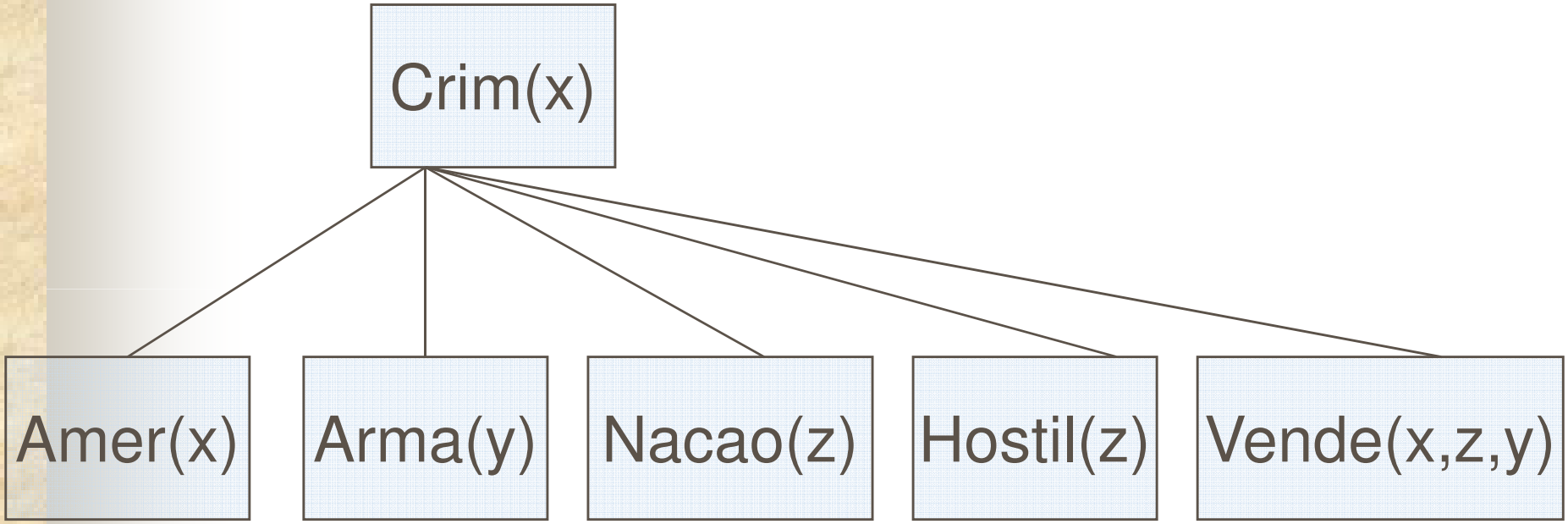
(7) Nacao(Nono)

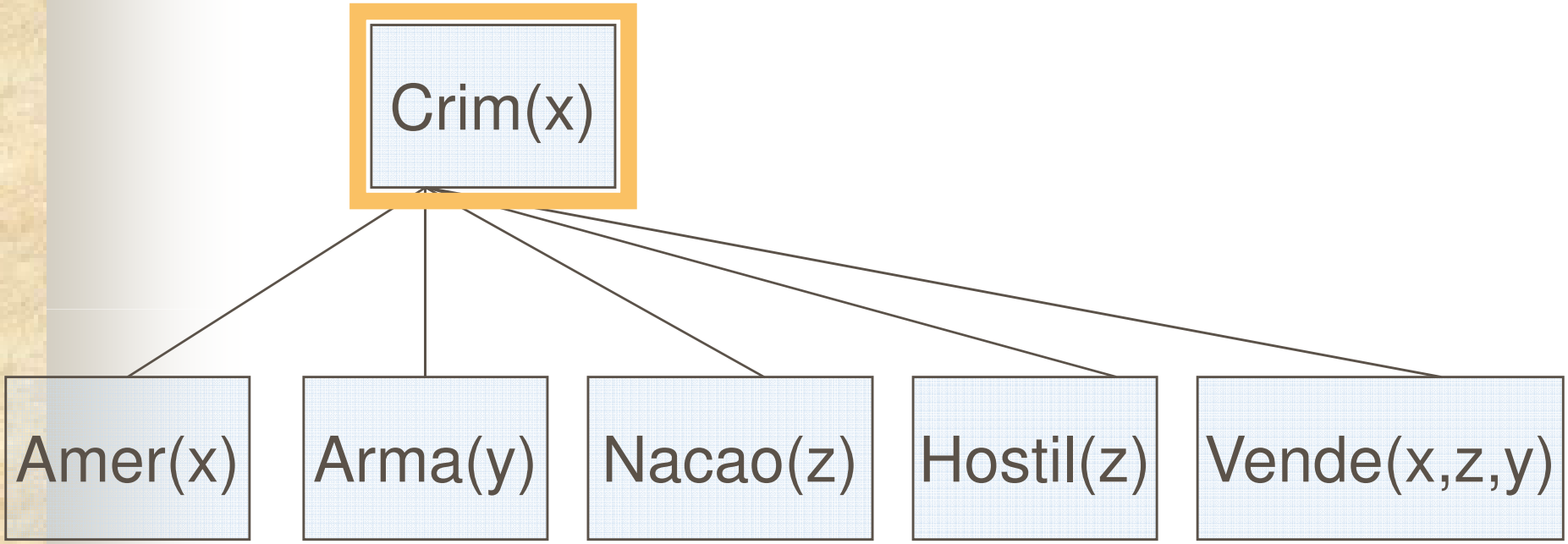
(8) Inimigo(Nono,EUA)

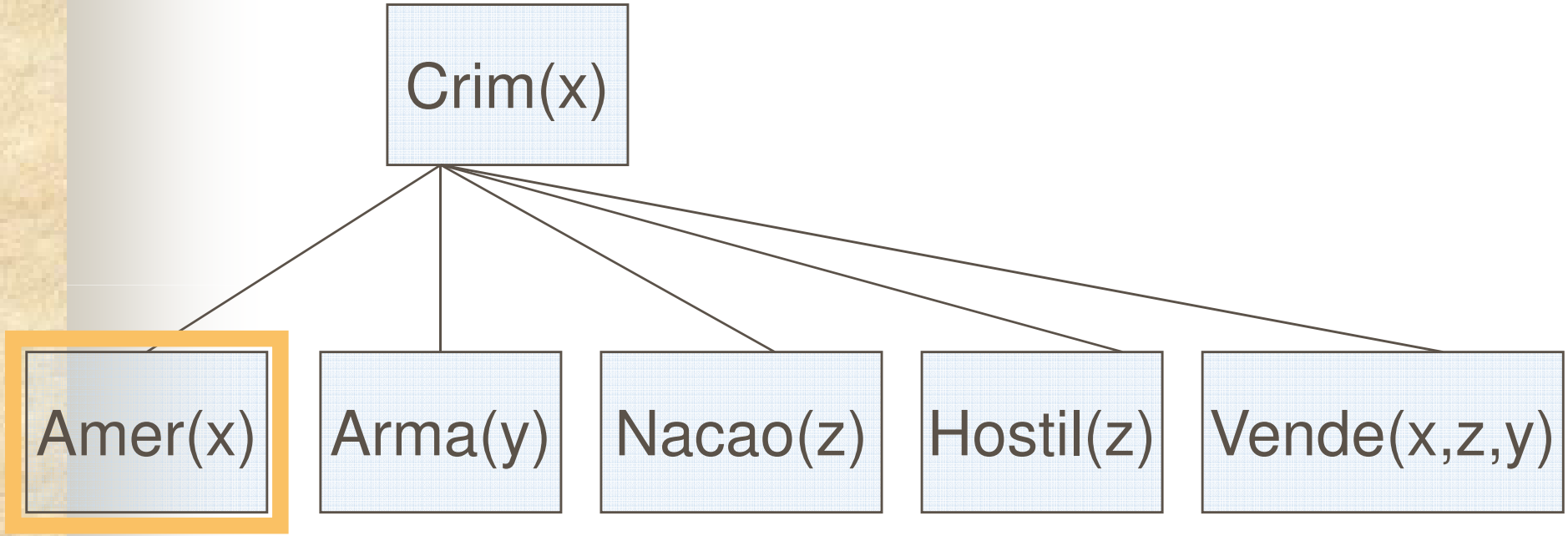
(9) Nacao(EUA)

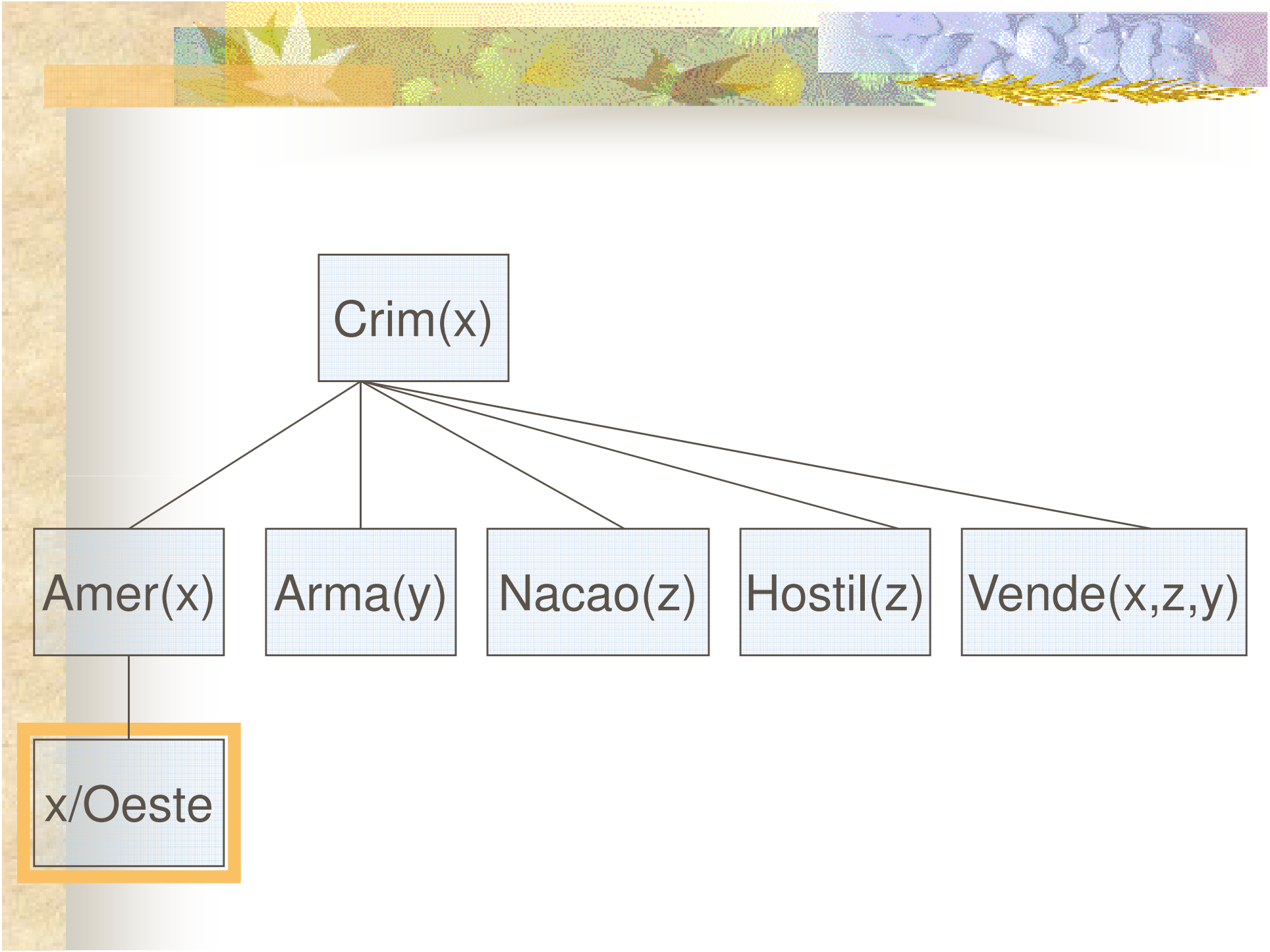
(10) Arma(M1)

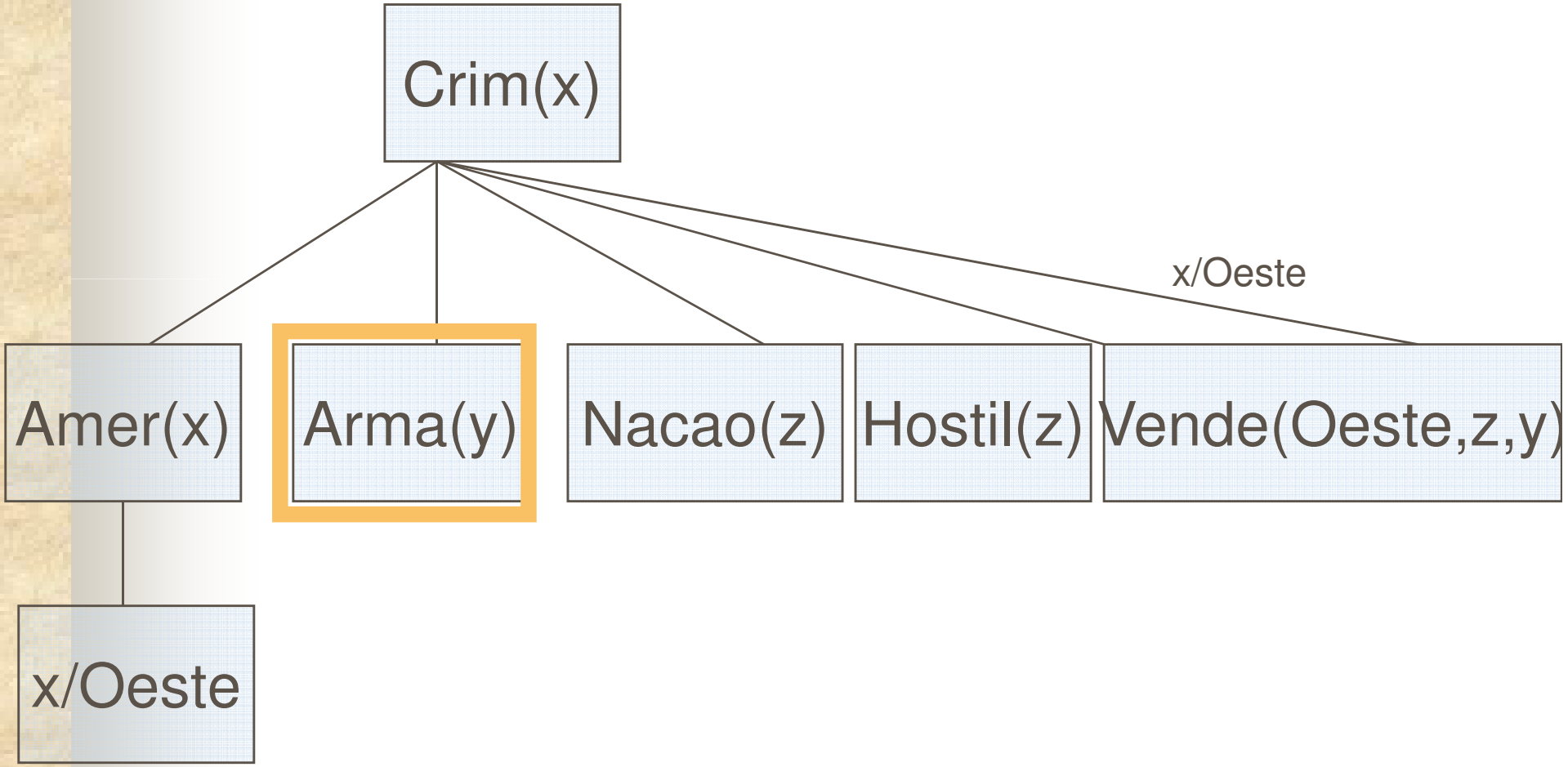
Passo 2: inferência...

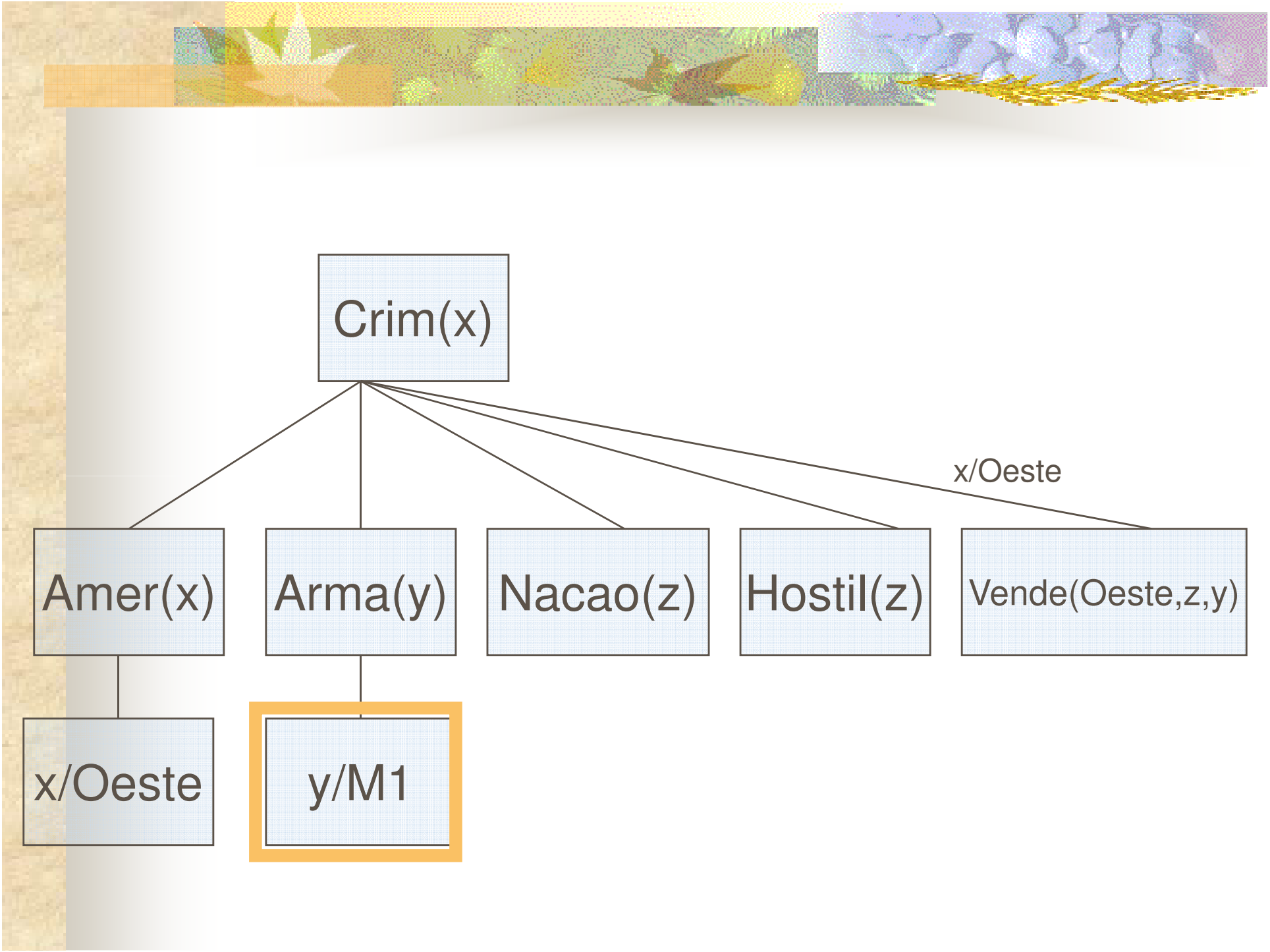


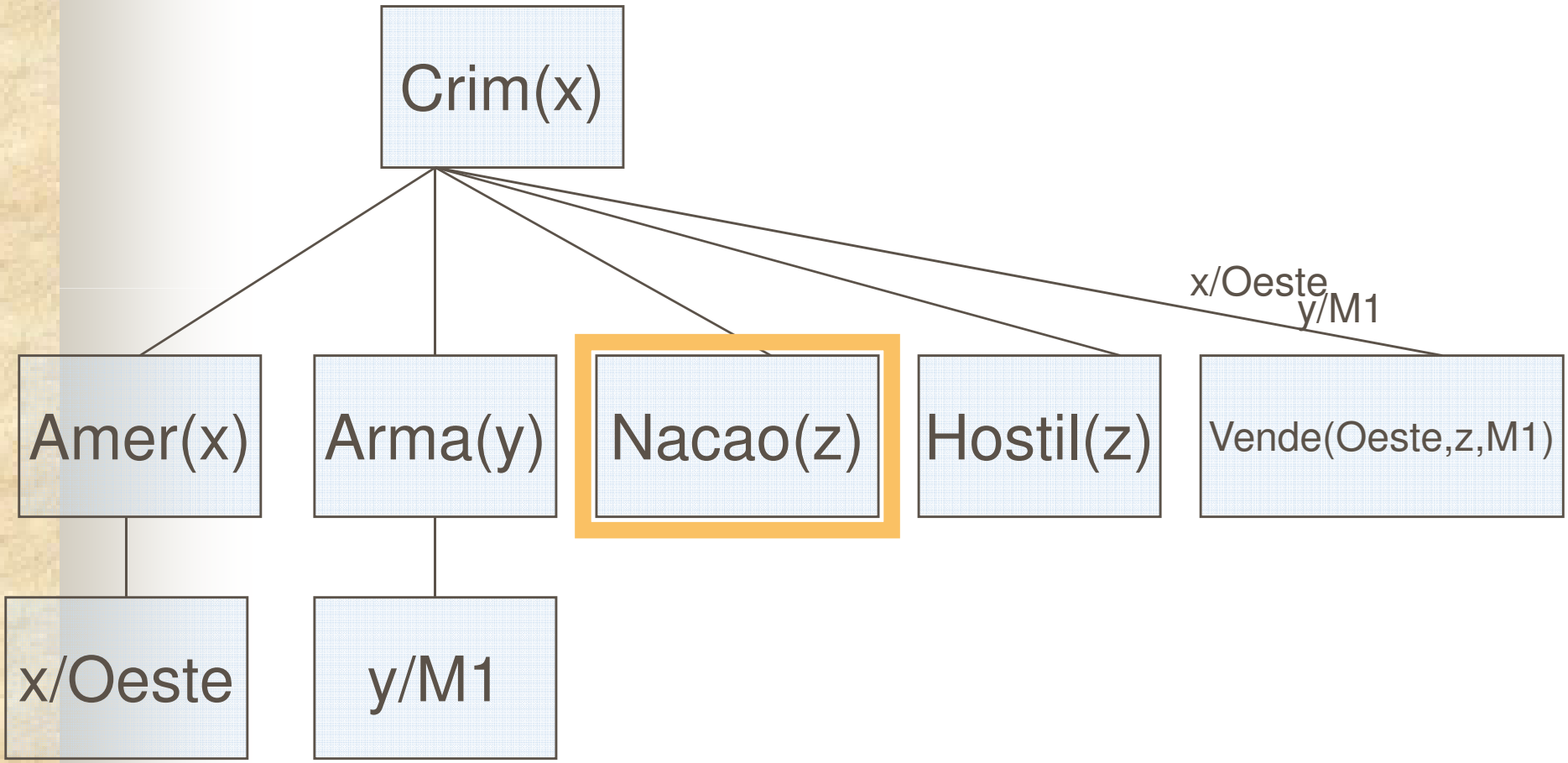

Crim(x)
Amer(x)
Arma(y)
Nacao(z)
Hostil(z)
Vende(x,z,y)

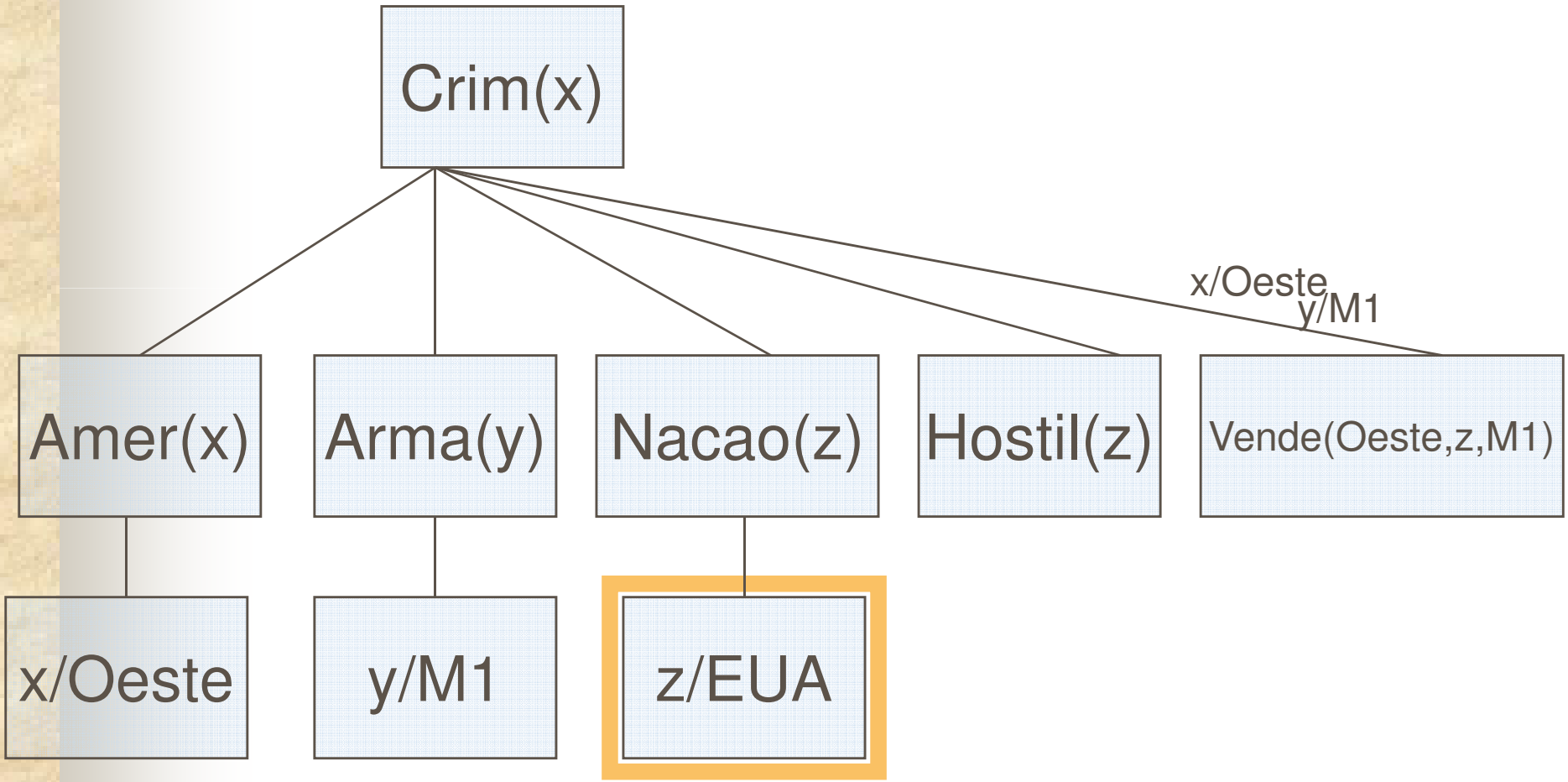

Crim(x)
Amer(x)
Arma(y)
Nacao(z)
Hostil(z)
Vende(x,z,y)

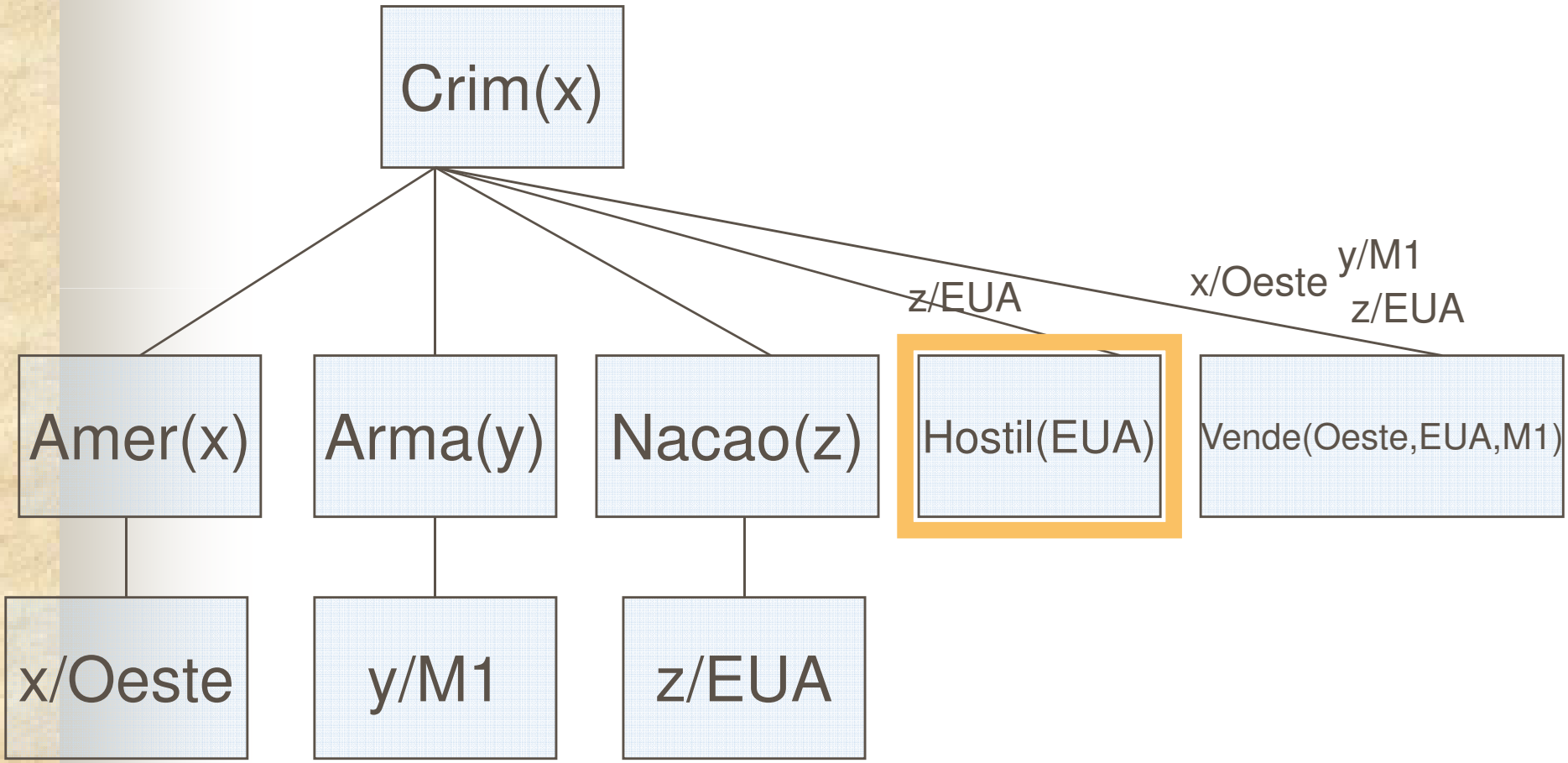

Crim(x)
Amer(x)
Arma(y)
Nacao(z)
Hostil(z)
Vende(x,z,y)
x/Oeste

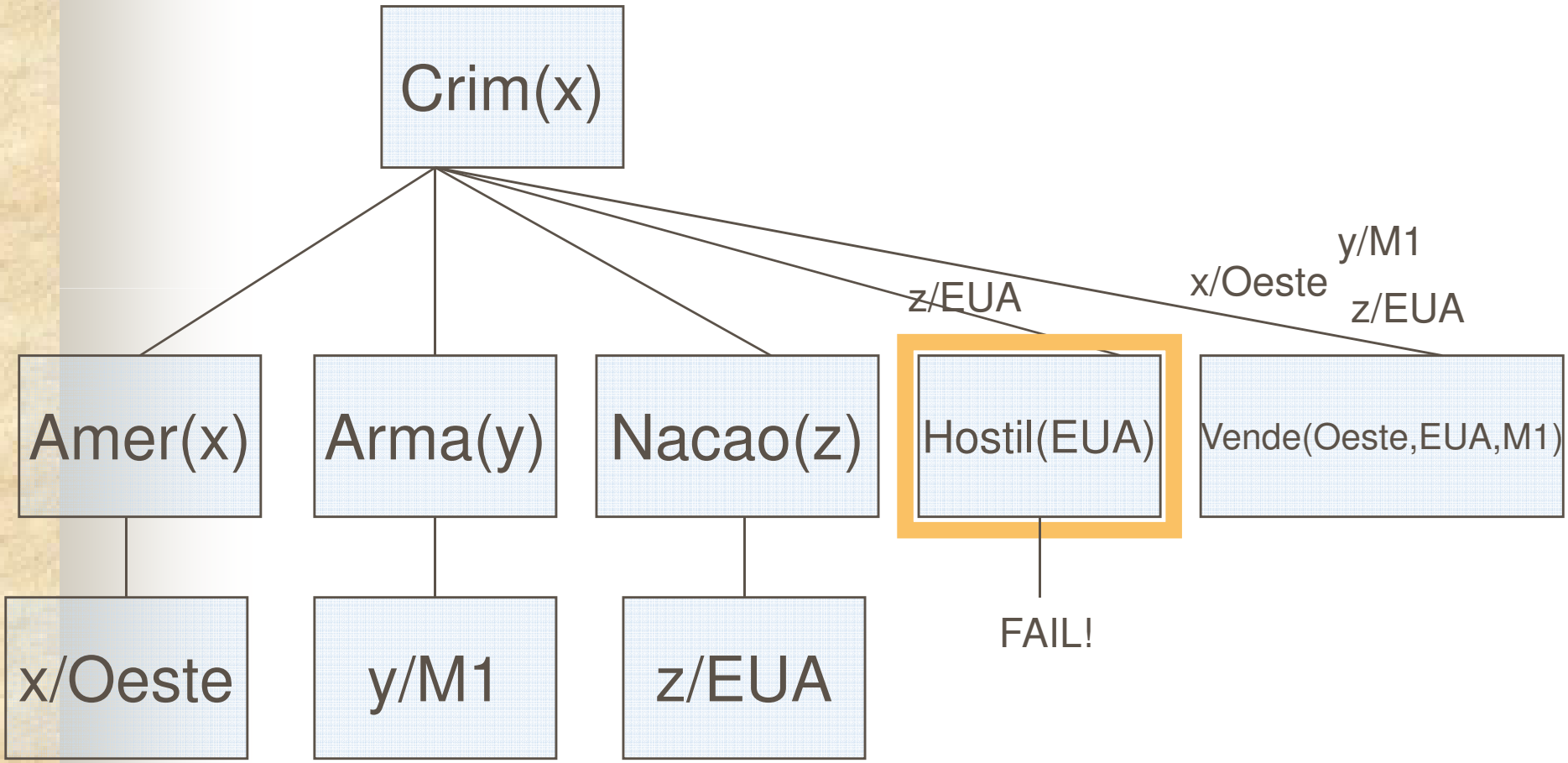

Crim(x)
x/Oeste
Amer(x)
Arma(y)
Nacao(z)
Hostil(z)
Vende(Oeste,z,y)
x/Oeste

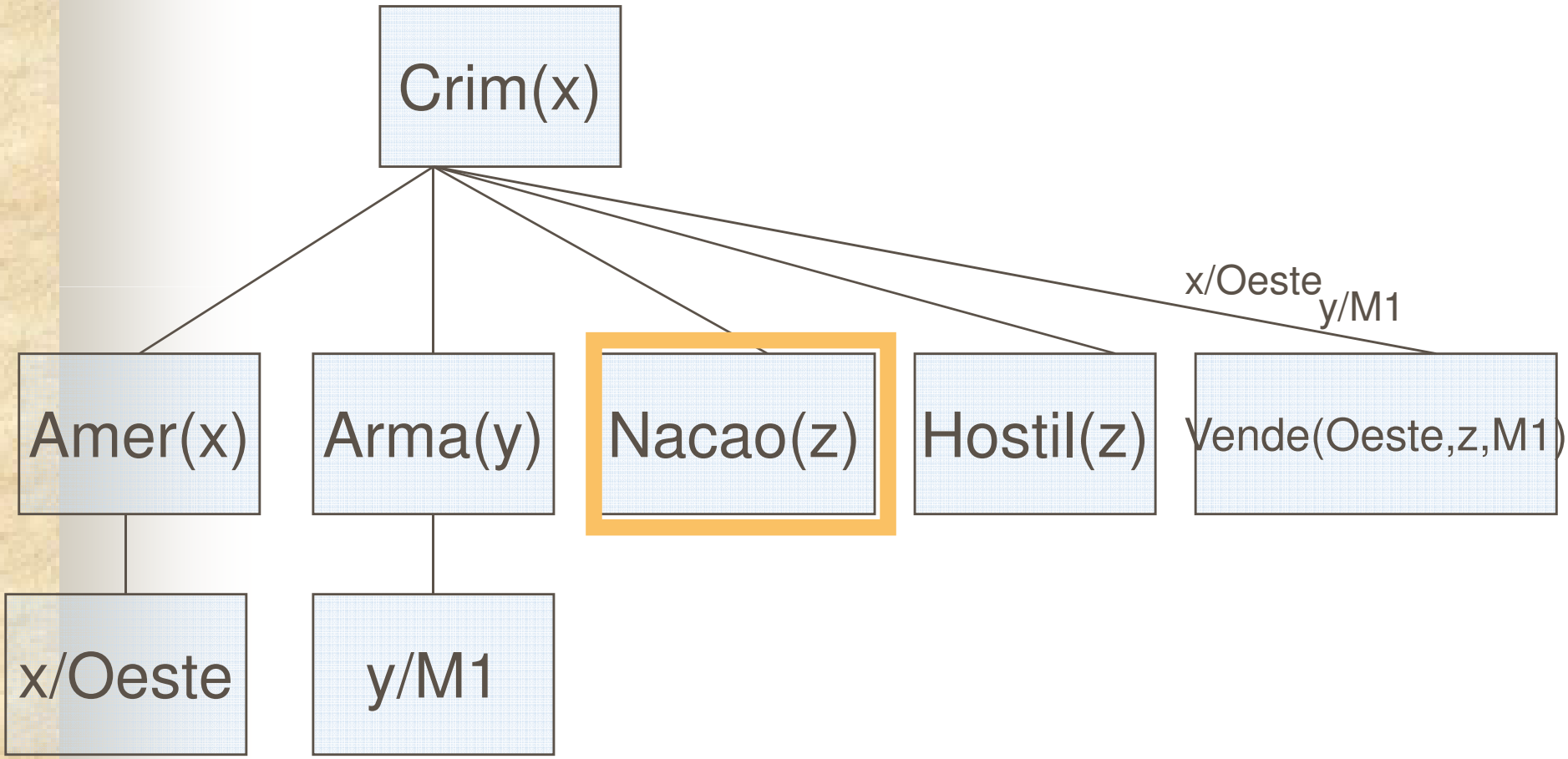

Crim(x)
x/Oeste
Amer(x)
Arma(y)
Nacao(z)
Hostil(z)
Vende(Oeste,z,y)
x/Oeste
y/M1

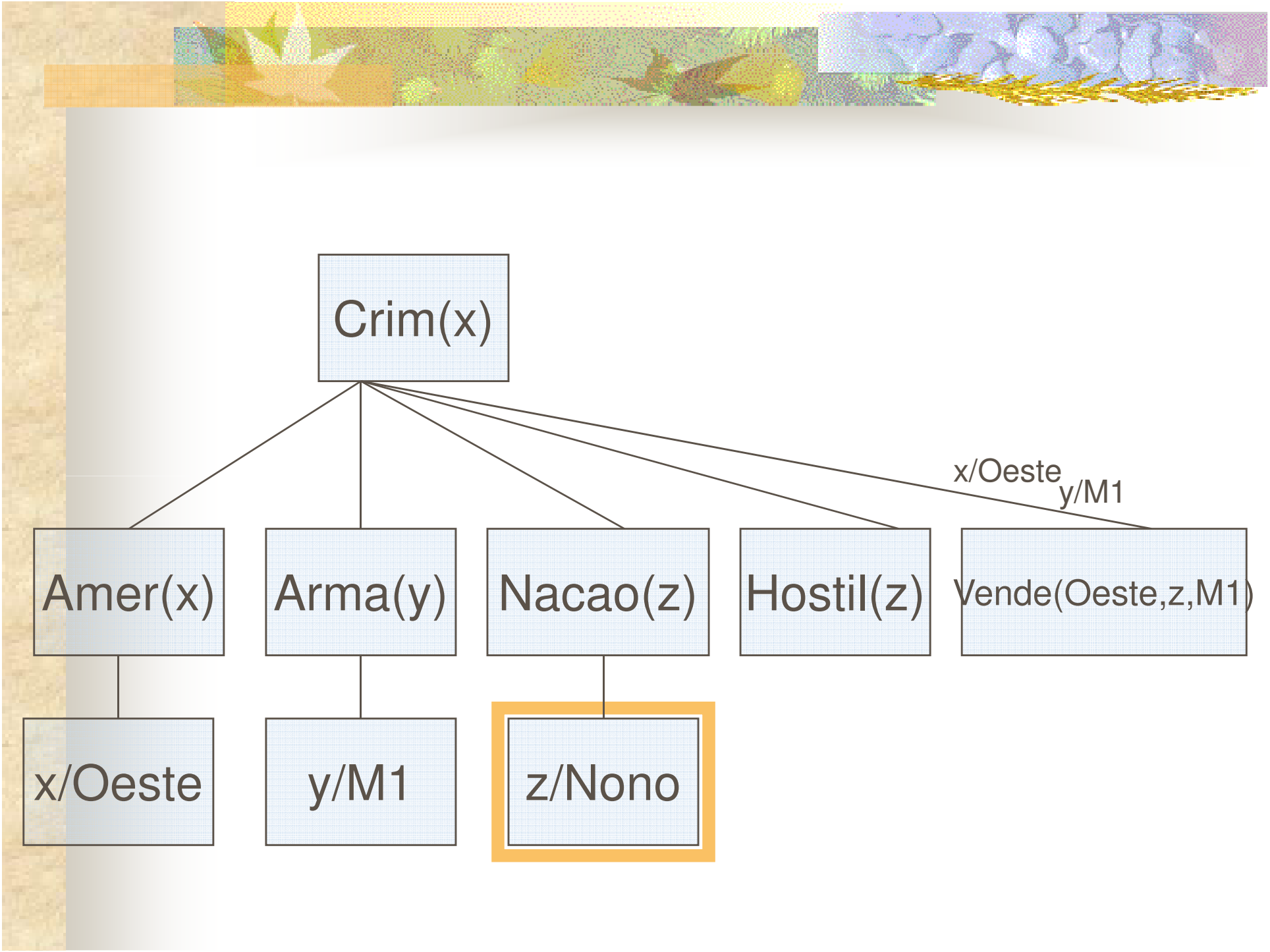

Crim(x)
x/Oeste
y/M1
Amer(x)
Arma(y)
Nacao(z)
Hostil(z)
Vende(Oeste,z,M1)
x/Oeste
y/M1

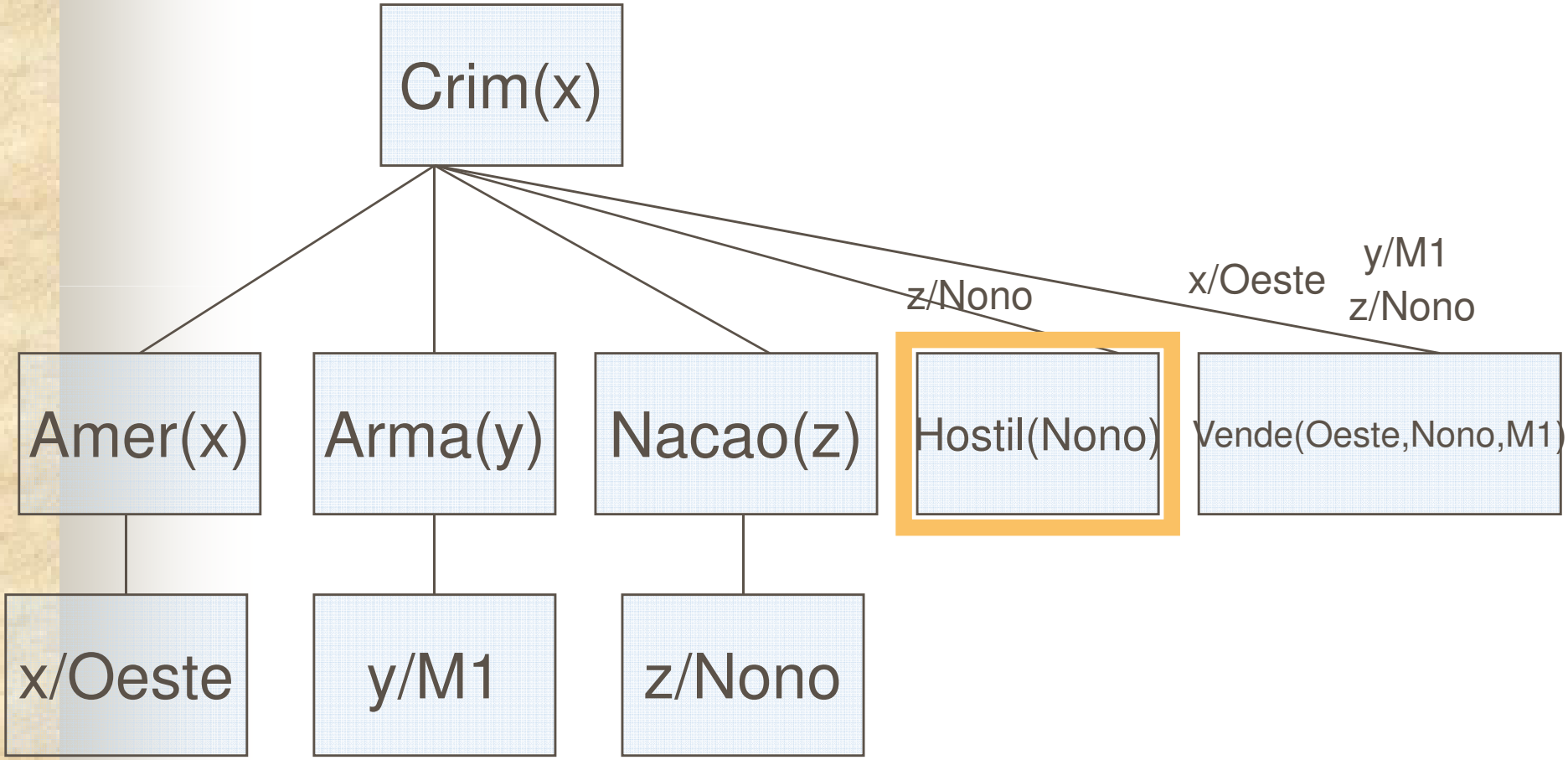

Crim(x)
x/Oeste
y/M1
Amer(x)
Arma(y)
Nacao(z)
Hostil(z)
Vende(Oeste,z,M1)
x/Oeste
y/M1
z/EUA

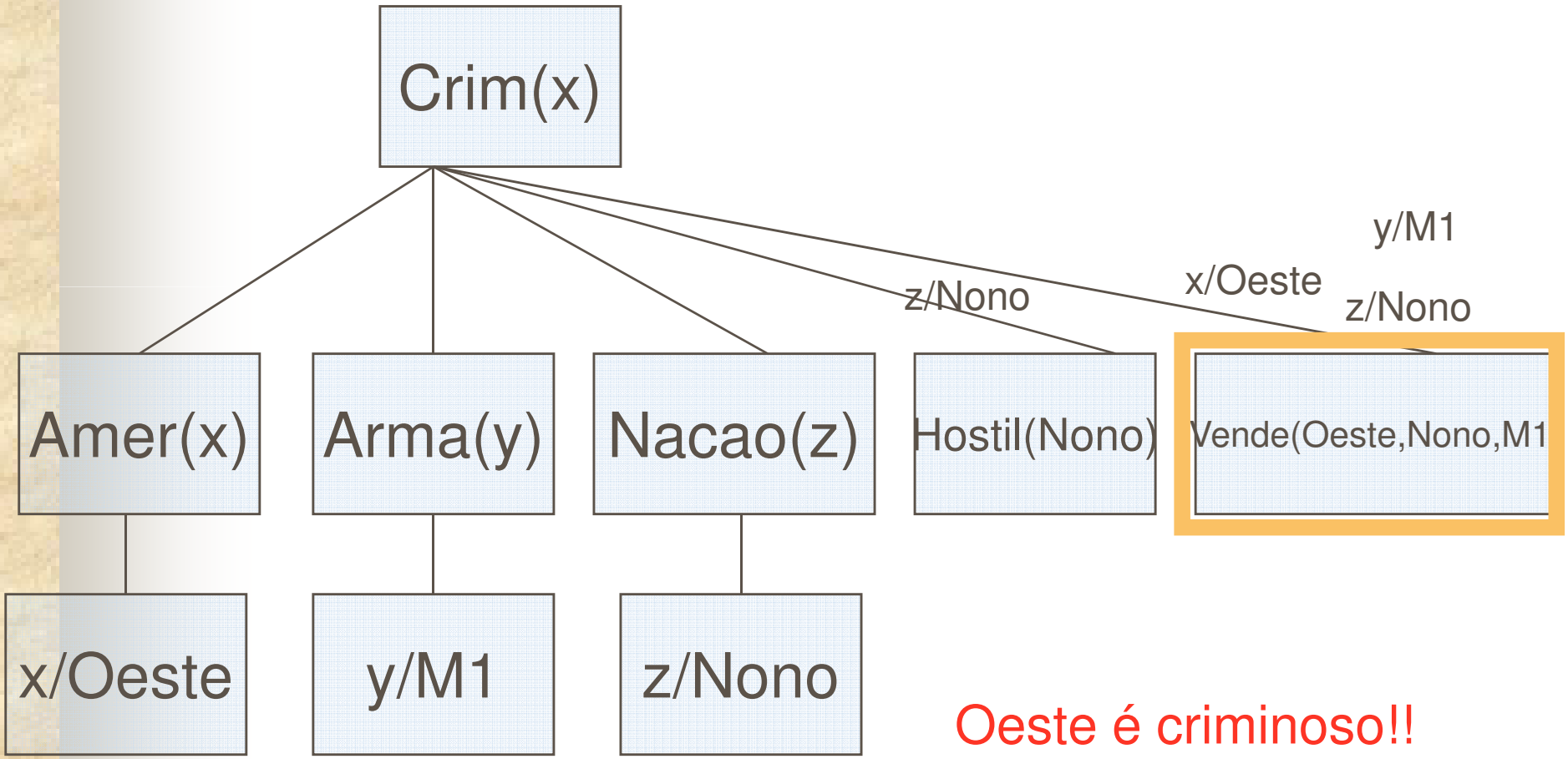

Crim(x)
x/Oeste
y/M1
z/EUA
z/EUA
Amer(x)
Arma(y)
Nacao(z)
Hostil(EUA)
Vende(Oeste,EUA,M1)
x/Oeste
y/M1
z/EUA


Crim(x)
y/M1
x/Oeste
z/EUA
z/EUA
Amer(x)
Arma(y)
Nacao(z)
Hostil(EUA)
Vende(Oeste,EUA,M1)
x/Oeste
y/M1
z/EUA
FAIL!


Crim(x)
x/Oeste
y/M1
Amer(x)
Arma(y)
Nacao(z)
Hostil(z)
Vende(Oeste,z,M1)
x/Oeste
y/M1


Crim(x)
x/Oeste
y/M1
Amer(x)
Arma(y)
Nacao(z)
Hostil(z)
Vende(Oeste,z,M1)
x/Oeste
y/M1
z/Nono


Crim(x)
z/Nono
x/Oeste
y/M1
z/Nono
Amer(x)
Arma(y)
Nacao(z)
Hostil(Nono)
Vende(Oeste,Nono,M1)
x/Oeste
y/M1
z/Nono


Crim(x)
y/M1
x/Oeste
z/Nono
z/Nono
Amer(x)
Arma(y)
Nacao(z)
Hostil(Nono)
Vende(Oeste,Nono,M1
x/Oeste
y/M1
z/Nono
Oeste é criminoso!!

MYCIN (Shortliffe *et al.*)

- Examplo 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:

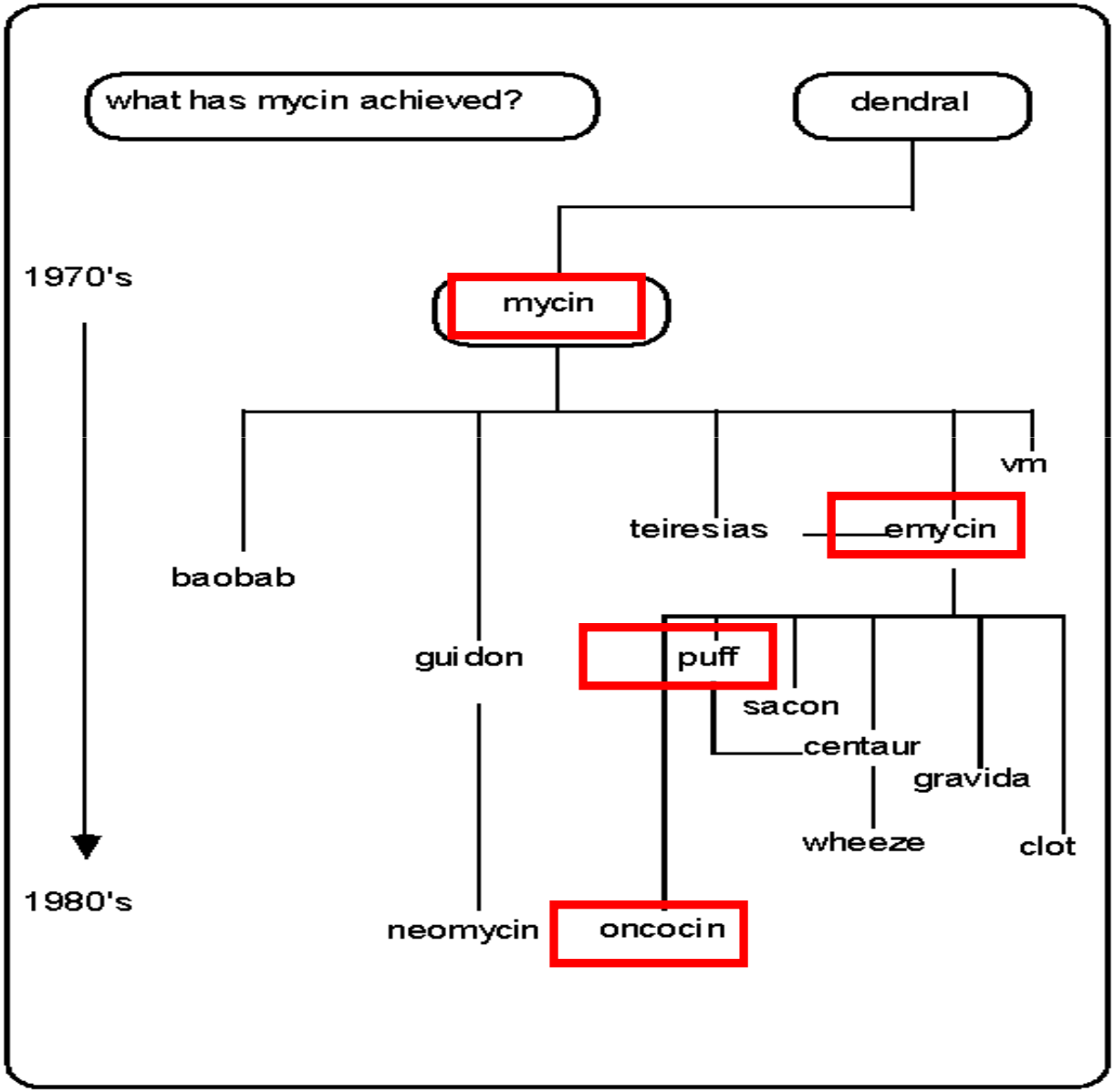
** **NEG**

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...)
- .....

Linguagem mais popular: Prolog

- Linguagem de alto nível.
- Poder de expressão.
- Formal.
- Manipulação fácil de símbolos.
- Prova de Teoremas e Processamento de Linguagem Natural.

Prova de Teoremas ou Dedução Automática

- Representação do conhecimento de forma *declarativa*, em lógica matemática.
- Mecanismo de inferência (provador de teoremas) deduz soluções para os problemas.
- Exceto para classes de problemas muito restritas, o espaço de busca pode crescer exponencialmente.
- Requer estratégias de busca inteligentes.
- Pouco sucesso após muito trabalho nos anos 60.

O que faz programação em lógica ser diferente de prova de teoremas?

- Programação em Lógica **é** Programação!
- Programador se preocupa com a eficiência e praticidade (roda e termina em tempo polinomial?)
- Utilização de uma lógica restrita (Cláusulas de Horn).
- Mecanismo de inferência simples e eficiente.
- Algoritmo = Lógica + Controle.
- Controle pode ser ignorado quando ``lemos'' o programa de forma declarativa.

Conceitos Básicos em Lógica

- Computação: método de ``raciocínio formal".
- ``objetos" do raciocínio: sentenças sobre o mundo (*fatos* ou *regras*).
- ``Computação lógica": usa o conjunto de sentenças para predizer ou computar a validade ou falsidade de outras sentenças.

Conceitos Básicos em Lógica

- Modelo computacional básico: *máquina de inferência*.
- *Fatos*: entidades básicas em lógica, assume-se que são verdadeiros (axiomas).
- Exs: o preço deste livro é 49 euros, às 5 h do dia 13/03/98 está chovendo, o fatorial de 3 é 6.
- Forma de expressar fatos: *relações*.

Conceitos Básicos em Lógica

- Relação: conjunto de tuplas.
- Cada tupla: conjunto de objetos que compartilham as mesmas características ou possuem a mesma propriedade.
- Ex: relação cor-do-cabelo(Ines,cast-escuro)

Conceitos Básicos em Lógica

- Outras formas de relações: ``Se A é um fato, então B também é um fato".
- Contradições não são permitidas: A é um fato e A não é um fato.
- Uma tupla não pode estar numa relação e ao mesmo tempo não estar.

Conceitos Básicos em Lógica

- *Inferência*: conclui que uma sentença é verdadeira através da verificação de que outras sentenças são verdadeiras, sem ter que procurar exaustivamente pelo conjunto total de sentenças. Ex: Se x é pai de alguma criança, então x é pai. Vítor é pai de Mariana.
- Não é necessário procurar por todas as tuplas e-pai para provar que Vítor é pai.

Sistemas de Lógica Formal

- Sintaxe + Semântica.
- *Fórmulas bem formadas*: expressões sintaticamente válidas na linguagem.

Sistemas de Lógica Formal

- Conjunto de símbolos permitidos:
  - constantes;
  - funções;
  - predicados;
  - variáveis lógicas;
  - conectivos lógicos: implicação, conjunção, disjunção, negação, relacionais;
  - quantificadores.

Sistemas de Lógica Formal

- Regras de inferência utilizadas para o raciocínio automático
- Resolução por refutação: método computacional correto e completo (se existir solução) para raciocínio automático

Programação Prolog

- Programas Prolog são representados por *cláusulas de Horn*, um subconjunto de *lógica de primeira ordem*, onde cada cláusula só pode conter no máximo um literal positivo na cabeça da cláusula (máximo um consequente positivo).
- Prolog: linguagem declarativa.
- Programa: conjunto de *fatos* e/ou *regras* que definem relações entre objetos.

Programação Prolog

- Fatos:

```
valioso(ouro).
sexo_feminino(jane).
pai(john,mary).
humano(socrates).
ateniense(socrates).
```

- Atenção à sintaxe!

- Regras:

```
gosta(john,X) :-
    gosta(X,vinho).

passaro(X) :- animal(X),
    tem_penas(X).

irma(X,Y) :-
    sexo_feminino(X),
    pais(M,F,X),
    pais(M,F,Y).
```

Programação Prolog

- Computação de um programa em lógica é a dedução dos consequentes do programa.
- *Fatos*: relações consideradas sempre verdadeiras (axiomas).
- *Regras*: Relações que são verdadeiras ou falsas dependendo de outras relações.

Programação Prolog: A Linguagem - Sintaxe

- Termos:
  - Variáveis: X, Y, C1, _ABC, Input
  - Constantes: prolog, a, 123, 'rio de janeiro'
  - Estruturas (termos compostos): dono(john,livro(ulysses,autor(james,joyce)))
- Caracteres: letras maiúsculas, letras minúsculas, dígitos, sinais do teclado.
- Símbolos especiais: **:- ; , .**
- Comentários:
  - linha: % isto e' um comentario.
  - texto: /* este tambem e' um comentário */

Programação Prolog: A Linguagem - Sintaxe

- Operadores: **+**, **-**, *, **/** etc.
- Igualdade e ``matching'':

  a(b,c,d(e,F,g(h,i,j))) = a(B,C,d(E,f,g(H,i,j)))
- Aritmética números: =, \=, <, >, >=, =<
- Aritmética strings/termos: ==, \== @<, @>

Programação Prolog: A Linguagem - Sintaxe

- Observação: Prolog **não avalia** expressões que não apareçam explicitamente no corpo da cláusula no contexto do operador especial **is**.
  - p(2+3,4*5).
  - Estas operações não são avaliadas!!!
  - Para obrigar a avaliação:

    p(X,Y) :- X is 2+3, Y is 4*5.

Programação Prolog: Exemplo simples

```
parent(C,M,F) :-
        mother(C,M),
        father(C,F).
```

```
mother(john,ann).
mother(mary,ann).
father(mary,fred).
father(john,fred).
female(mary).
```

Consulta:

```
?-female(mary),parent(mary,M,F),parent(john,M,F).
```

Programação Prolog: Outras estruturas de Dados

- Listas: estrutura de dados especial em Prolog.
- Exs:
  - []: lista vazia.
  - [the,men, [like,to,fish]]
  - [a,V1,b, [X,Y]]

Programação Prolog: Outras estruturas de Dados

- Estrutura geral de lista não vazia:
  [Head|Tail]
- Head: primeiro elemento da lista (pode ser de qualquer tipo).
- Tail: lista restante (tipo é **obrigatoriamente** uma lista).

Programação Prolog: Outras estruturas de Dados

- Exemplos:

Lista	Head	Tail
[a,b,c]	a	[b,c]
[a]	a	[]
[[the,cat],sat]	[the,cat]	[sat]
[the,[cat,sat]]	the	[[cat,sat]]
[X+Y,x+y]	X+Y	[x+y]
[]	no head	no tail