



PARADIGMAS E LINGUAGENS DE PROGRAMAÇÃO

ENGENHARIA DA COMPUTAÇÃO – UFC/SOBRAL

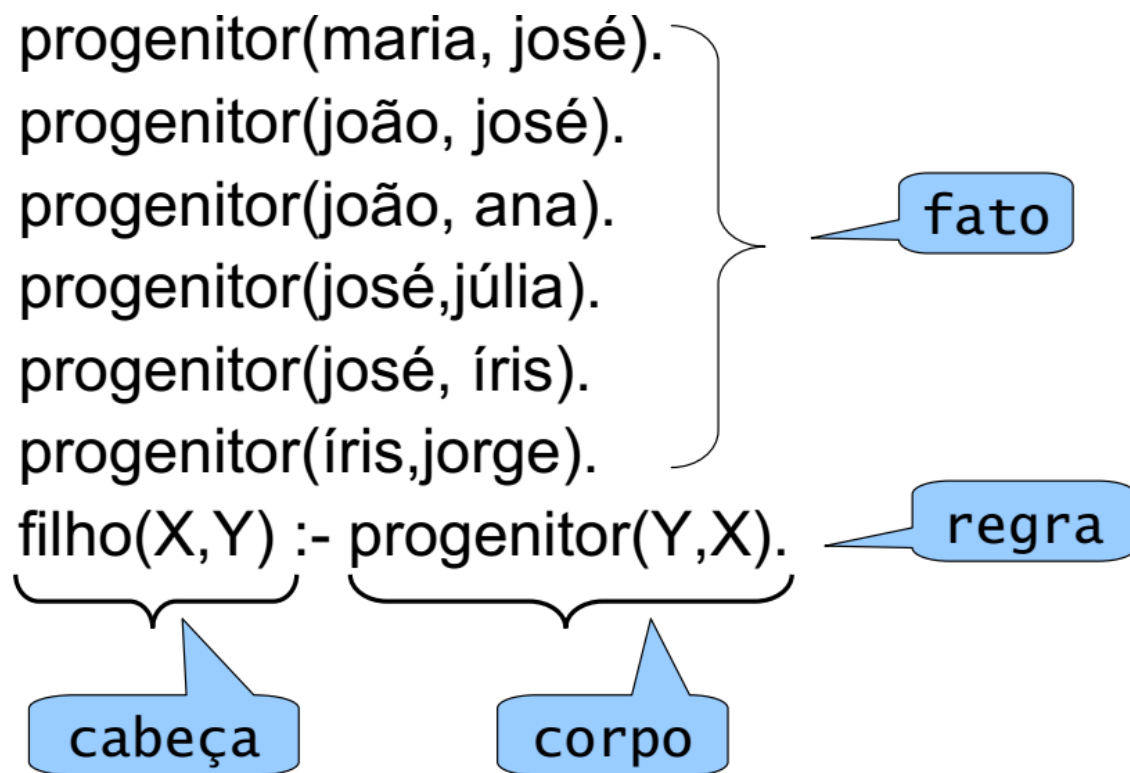
Prof. Danilo Alves

`danilo.alves@alu.ufc.br`

BASE DE CONHECIMENTO 2



UNIVERSIDADE
FEDERAL DO CEARÁ



BASE DE CONHECIMENTO 2



UNIVERSIDADE
FEDERAL DO CEARÁ

- ?- tocaGuitarra(maria).
- true
- ?- tocaGuitarra(iolanda).
- true

- feliz(iolanda).
- escuta_musica(maria).
- escuta_musica(iolanda):- feliz(iolanda).
- tocaGuitarra(maria):- escuta_musica(maria).
- tocaGuitarra(iolanda):- escuta_musica(iolanda).

CLÁUSULAS



UNIVERSIDADE
FEDERAL DO CEARÁ

- feliz(iolanda).
- escuta_musica(maria).
- escuta_musica(iolanda):- feliz(iolanda).
- tocaGuitarra(maria):- escuta_musica(maria).
- tocaGuitarra(iolanda):- escuta_musica(iolanda).

- Existem cinco **cláusulas** nesta base de conhecimento:
 - dois fatos e três regras.
- O final de uma cláusula é marcado com um ponto final.

PREDICADOS



UNIVERSIDADE
FEDERAL DO CEARÁ

- feliz(iolanda).
- escuta_musica(maria).
- escuta_musica(iolanda):- feliz(iolanda).
- tocaGuitarra(maria):- escuta_musica(maria).
- tocaGuitarra(iolanda):- escuta_musica(iolanda).

- Existem três predicados nesta base de conhecimento:
 - feliz, escuta_musica e tocaGuitarra

BASE DO CONHECIMENTO 4



UNIVERSIDADE
FEDERAL DO CEARÁ

- feliz(vicente).
- escuta_musica(bruno).
- tocaGuitarra(vicente):- escuta_musica(vicente), feliz(vicente).
- tocaGuitarra(bruno):- feliz(bruno).
- tocaGuitarra(bruno):- escuta_musica(bruno).

BASE DO CONHECIMENTO 4



UNIVERSIDADE
FEDERAL DO CEARÁ

- feliz(vicente).
- escuta_musica(bruno).
- tocaGuitarra(vicente):- escuta_musica(vicente), feliz(vicente).
- tocaGuitarra(bruno):- feliz(bruno).
- tocaGuitarra(bruno):- escuta_musica(bruno).

- A vírgula “,” expressa conjunção em Prolog

EXPRESSANDO DISJUNÇÃO



UNIVERSIDADE
FEDERAL DO CEARÁ

- feliz(vicente).
- escuta_musica(bruno).
- tocaGuitarra(vicente):- escuta_musica(vicente), feliz(vicente).
- **tocaGuitarra(bruno):- feliz(bruno).**
- **tocaGuitarra(bruno):- escuta_musica(bruno).**

feliz(vicente).
escuta_musica(bruno).
tocaGuitarra(vicente):- escuta_musica(vicente), feliz(vicente).
tocaGuitarra(bruno):- feliz(bruno); escuta_musica(bruno).

PROLOG E LÓGICA



UNIVERSIDADE
FEDERAL DO CEARÁ

- Prolog é baseado na lógica
- Operadores
 - Implicação :-
 - Conjunção ,
 - Disjunção ;

BASE DO CONHECIMENTO 5



UNIVERSIDADE
FEDERAL DO CEARÁ

?- mulher(X).

- X=maria;
- X=joana;
- X=iolanda.

?

- mulher(maria).
- mulher(joana).
- mulher(iolanda).

- ama(vicente, maria).
- ama(marcelo, maria).
- ama(abobrinha, coelhinho).
- ama(coelhinho, abobrinha).

BASE DO CONHECIMENTO 5



UNIVERSIDADE
FEDERAL DO CEARÁ

?- ama(marcelo,X), mulher(X).

■ X=maria.

?-

- mulher(maria).
- mulher(joana).
- mulher(iolanda).

- ama(vicente, maria).
- ama(marcelo, maria).
- ama(abobrinha, coelhinho).
- ama(coelhinho, abobrinha).

BASE DO CONHECIMENTO 5



UNIVERSIDADE
FEDERAL DO CEARÁ

?- ama(abobrinha,X), mulher(X).

■ false

?-

- mulher(maria).
- mulher(joana).
- mulher(iolanda).

- ama(vicente, maria).
- ama(marcelo, maria).
- ama(abobrinha, coelhinho).
- ama(coelhinho, abobrinha).

BASE DO CONHECIMENTO 6



UNIVERSIDADE
FEDERAL DO CEARÁ

?- tem_ciumes(marcelo,W).

■ W=vicente;

■ W=marcelo;

?-

- ama(vicente,maria).
- ama(marcelo,maria).
- ama(abobrinha, coelhinho).
- ama(coelhinho, abobrinha).
- tem_ciumes(X,Y):- ama(X,Z), ama(Y,Z).

BASE DO CONHECIMENTO 7



UNIVERSIDADE
FEDERAL DO CEARÁ

- Qual o resultado das seguintes perguntas?
- `gosta(joao, peixe).` `?- gosta(maria,X).`
- `gosta(joao,maria).` `?- gosta(X,livro).`
- `gosta(maria,livro).` `?- gosta(Quem,Oque).`
- `gosta(pedro,livro).` `?- gosta(X,Y).`
- `gosta(maria,flor).` `?- gosta(X,X).`
- `gosta(maria,vinho).` `?gosta(A,peixe).`

SWI-PROLOG



UNIVERSIDADE
FEDERAL DO CEARÁ

```
malbarbo@drcopper:~  
Welcome to SWI-Prolog (Multi-threaded, 64 bits, Version 6.4.1)  
Copyright (c) 1990-2013 University of Amsterdam, VU Amsterdam  
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software,  
and you are welcome to redistribute it under certain conditions.  
Please visit http://www.swi-prolog.org for details.  
  
For help, use ?- help(Topic). or ?- apropos(Word).  
?- emacs('arquivo.pl').  
true.  
  
?- % /home/malbarbo/arquivo compiled 0.00 sec, 3 clauses  
?- editor(joao, E).  
E = vim.  
?-
```

Editar arquivo.pl

Consulta

Resultado



- Editar o arquivo usando o editor de sua preferência
- Ler o arquivo no swipl
 - `?- consult('arquivo.pl').`
- Fazer consultas
- Depois de alterar o arquivo, ele deve ser lido novamente



- Prolog é uma linguagem de programação lógica de propósito geral associada com inteligência artificial e linguística que possui suas raízes na lógica de primeira ordem, uma lógica formal
- Prolog é declarativa, a lógica do programa é expressa em termos de relações, representada como fatos e regras; uma computação é iniciada executando uma pesquisa sobre essas relações
- Foi uma das primeiras linguagens de programação lógica e se mantém como a mais popular até hoje
- A linguagem é usada para prova de teoremas, sistemas especialistas, processamento de linguagens naturais, ...



- Valores
 - Constantes: letras (começa com minúscula), números, texto
 - Ex.: `maria`, `'maria'`, `123`, `"Texto"`
 - Estruturas (tuplas rotuladas)
 - Ex.: `ponto(5, 6)`, `data(01, 01, 2000)`
 - Listas
 - Ex.: `[1, 2, 3]`, `[maria, jose]`
- Variáveis (começam com letra maiúscula)
 - Ex.: `X`, `Y`, `Estudante`

PROLOG



UNIVERSIDADE
FEDERAL DO CEARÁ

- Comentário em Prolog

- Linha: %

- Bloco: /* */

CLÁUSULAS DE HORN



UNIVERSIDADE
FEDERAL DO CEARÁ

- Cláusulas de *Horn* são um subconjunto das cláusulas da lógica de predicados
 - Forma geral

$$B \text{ :- } A_1, A_2, \dots, A_n.$$

, onde cada A_i é uma relação da forma $B(\dots)$

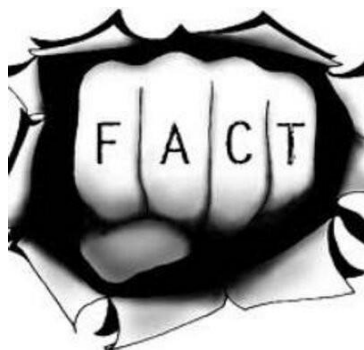
- Interpretação
 - se as condições A_1, A_2, \dots e A_n forem verdadeiras, então a conclusão B é verdadeira (na forma procedural: **se** condições **então** conclusão)
 - Se algum A_i for falso, não se pode deduzir que B é falso, apenas que não se pode inferir que B é verdadeiro

CLÁUSULAS DE HORN



UNIVERSIDADE
FEDERAL DO CEARÁ

- Nomenclatura:



- Se $n = 0$
 - a cláusula é chamada de **fato**



- Se $n \geq 1$
 - a cláusula é chamada de **regra**

Toda cláusula termina
com . (ponto).

EXEMPLO



UNIVERSIDADE
FEDERAL DO CEARÁ

- Exemplo somente com fatos:

```
gosta(maria, peixe).  
gosta(pedro, vinho).  
gosta(maria, vinho).  
gosta(pedro, maria).
```

- Quem gosta de peixe?
- ?- gosta(X, peixe).
- X = maria.
- Pedro e Maria se gostam?
- ?- gosta(maria, pedro), gosta(pedro, maria).
- false.
- Existe algo que Pedro e Maria (ambos) gostem?
- ?- gosta(pedro, X), gosta(maria, X).
- X = vinho

OUTRO EXEMPLO



UNIVERSIDADE
FEDERAL DO CEARÁ

?- orbita(venus, sol).

true.

?- orbita(B, marte).

B = phobos ;

B = deimos.

?- orbita(B, venus).

false.

?- planeta(mercurio).

false.

```
estrela(sol).  
estrela(sirius).
```

```
orbita(venus, sol).  
orbita(terra, sol).  
orbita(marte, sol).  
orbita(lua, terra).  
orbita(phobos, marte).  
orbita(deimos, marte).
```

```
planeta(B) :- orbita(B, sol).
```

```
satelite(B) :- orbita(B, P),  
               planeta(P).
```

OUTRO EXEMPLO



UNIVERSIDADE
FEDERAL DO CEARÁ

?- planeta(X).

X = venus ;

X = terra ;

X = marte.

?- satellite(X).

X = lua,

X = phobos,

X = deimos

```
estrela(sol).  
estrela(sirius).
```

```
orbita(venus, sol).  
orbita(terra, sol).  
orbita(marte, sol).  
orbita(lua, terra).  
orbita(phobos, marte).  
orbita(deimos, marte).
```

```
planeta(B) :- orbita(B, sol).
```

```
satelite(B) :- orbita(B, P),  
               planeta(P).
```


UNIFICAÇÃO



UNIVERSIDADE
FEDERAL DO CEARÁ

- Unificação é o nome do algoritmo usado por Prolog para determinar se existe uma maneira de instanciar as variáveis de dois predicados de modo a torná-los iguais.

Predicado 1	Predicado 2	Unificação
$p(X, Y)$	$q(X, Y)$	
$p(X, Y)$	$p(joao, jose)$	
$p(X, Y)$	$p(joao, Z)$	
$p(X, X)$	$p(1, 1)$	
$p(X, X)$	$p(1, W)$	
$p(X, X)$	$p(1, 2)$	
$p(X, X, 2)$	$p(1, W, W)$	
$p(G, jose)$	$p(X, Y)$	



- O algoritmo de execução
 - Faz uma busca em profundidade em uma árvore de pesquisa, com resolução de um objetivo, onde a lista de cláusulas é pesquisada de cima para baixo
- Se a unificação com o lado esquerdo de uma cláusula **for bem sucedida**, então tenta-se resolver os sub-objetivos do lado direito dessa cláusula (da esquerda para a direita)
- Pode haver retrocessos (backtracking) caso uma unificação **não seja bem sucedida**, tentando explorar outras unificações alternativas
 - Se todas as alternativas forem exploradas sem sucesso, então a resolução falha

A resolução é sujeita à ordem em que as cláusulas foram escritas.

EXEMPLO



UNIVERSIDADE
FEDERAL DO CEARÁ

- Verificar se João e Mario são irmãos
- ?- irmao(joao, mario).

```
homem(mario).
```

```
mae(joao, roberta).  
mae(mario, roberta).
```

```
pai(joao, paulo).  
pai(mario, paulo).
```

```
pais(X, M, P) :- mae(X, M),  
                 pai(X, P).
```

```
irmao(X, Y) :- homem(Y),  
              pais(X, M, P),  
              pais(Y, M, P).
```

EXEMPLO

```
irmão(joao, mario)
```

Com qual regra essa consulta
pode ser unificada?

EXEMPLO

`irmao(joao, mario)`

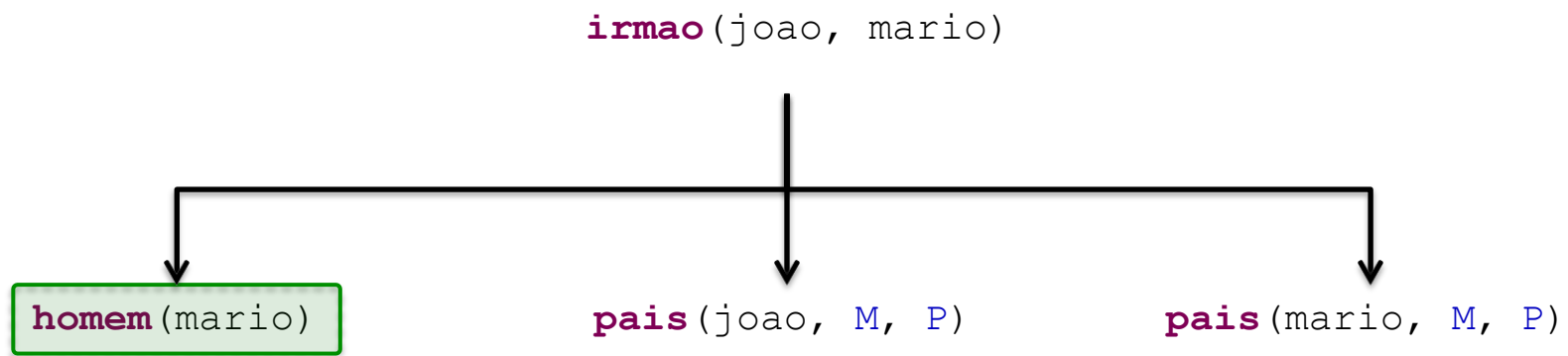


`X = joao, Y = mario`

`irmao(X, Y)`

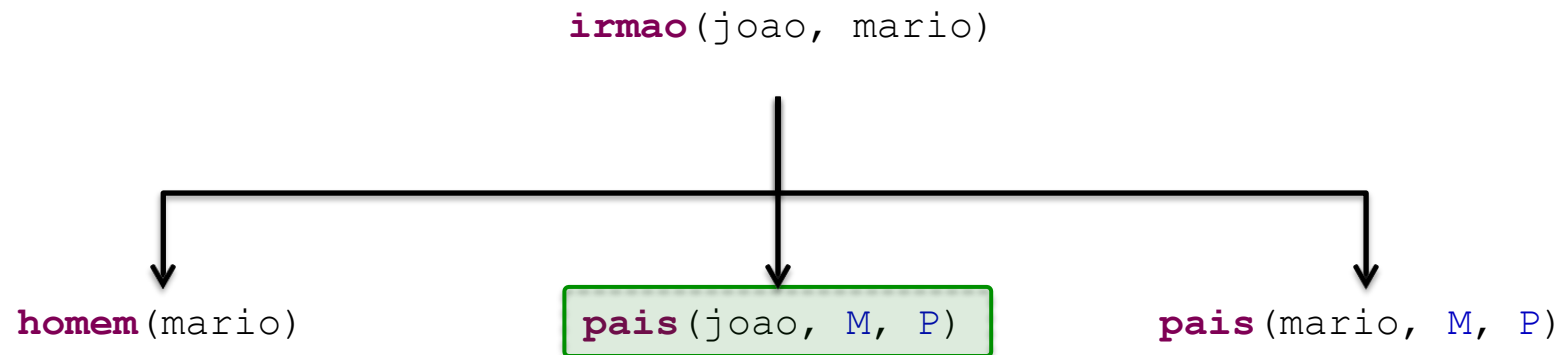
Expandir a regra `irmao(X, Y)`
com suas cláusulas à direita.

EXEMPLO



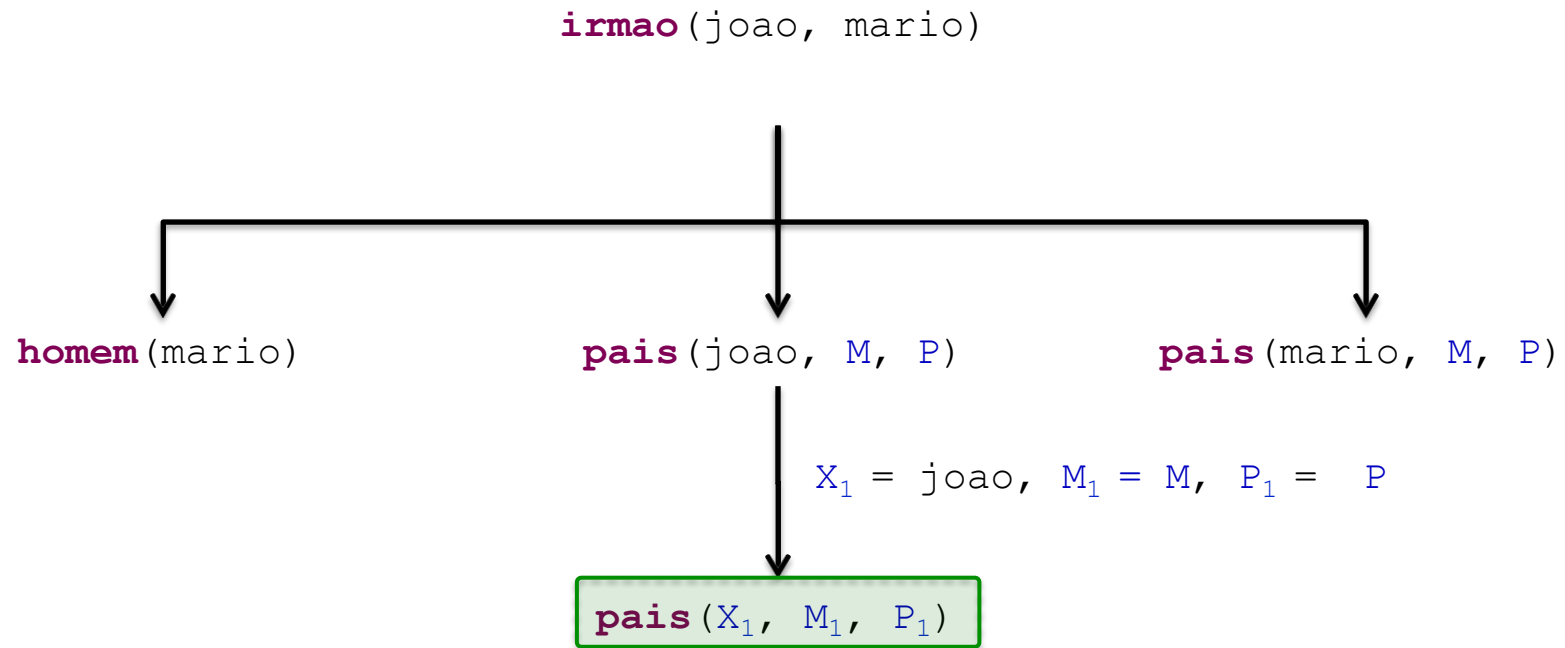
A cláusula `homem(mario)` é um fato.

EXEMPLO



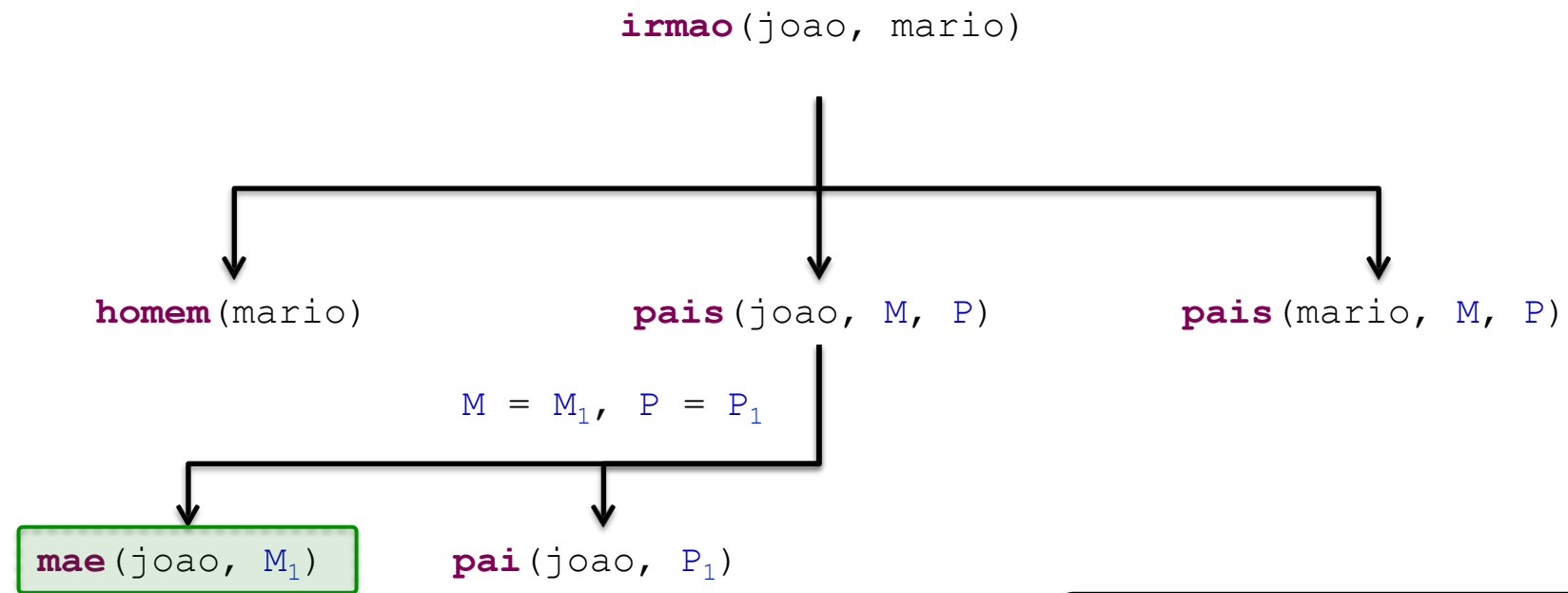
Qual regra pode ser unificada nesse momento?

EXEMPLO



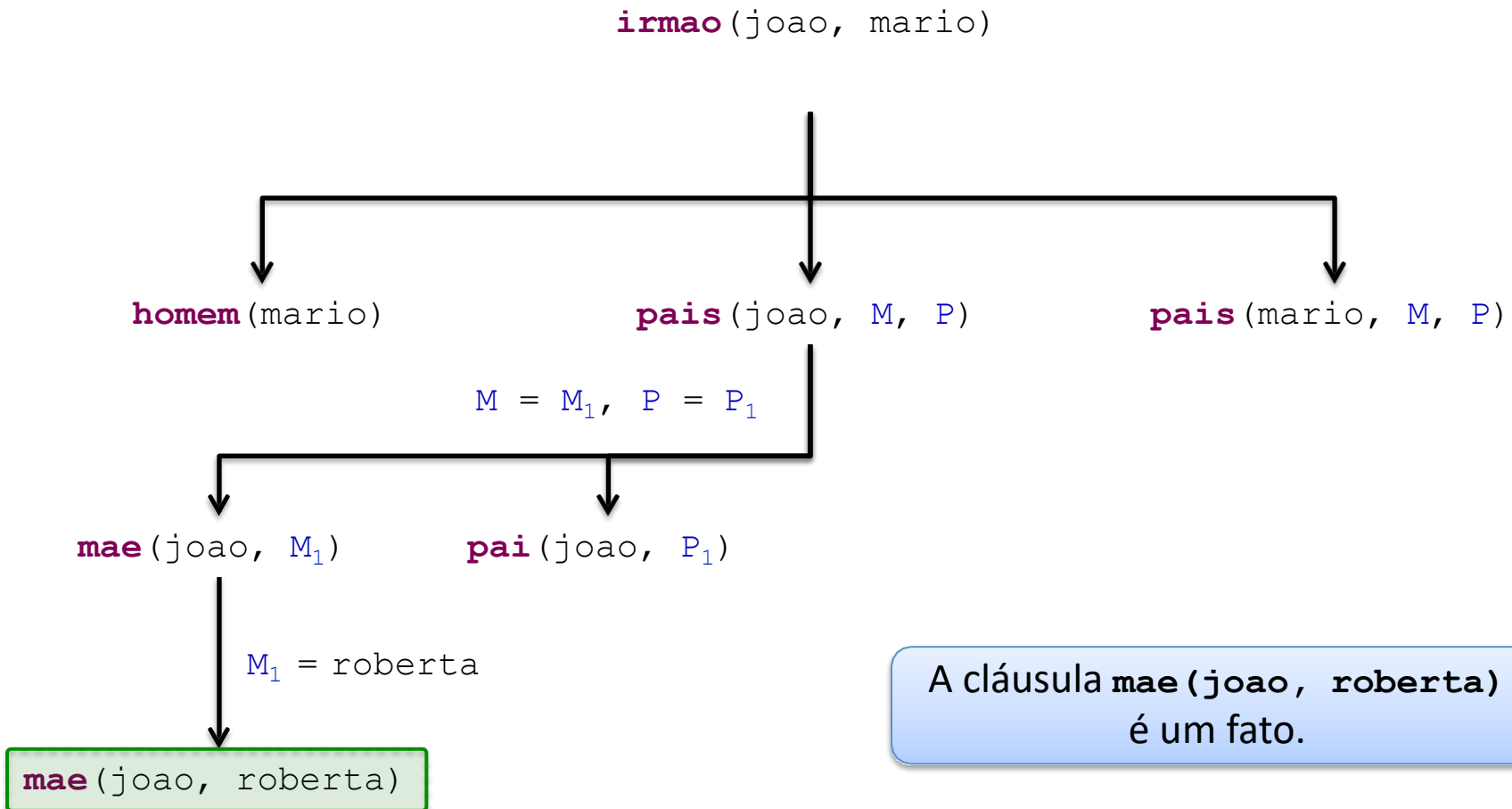
Expandir a regra `pais(X_1 , M_1 , P_1)`
com suas cláusulas à direita.

EXEMPLO



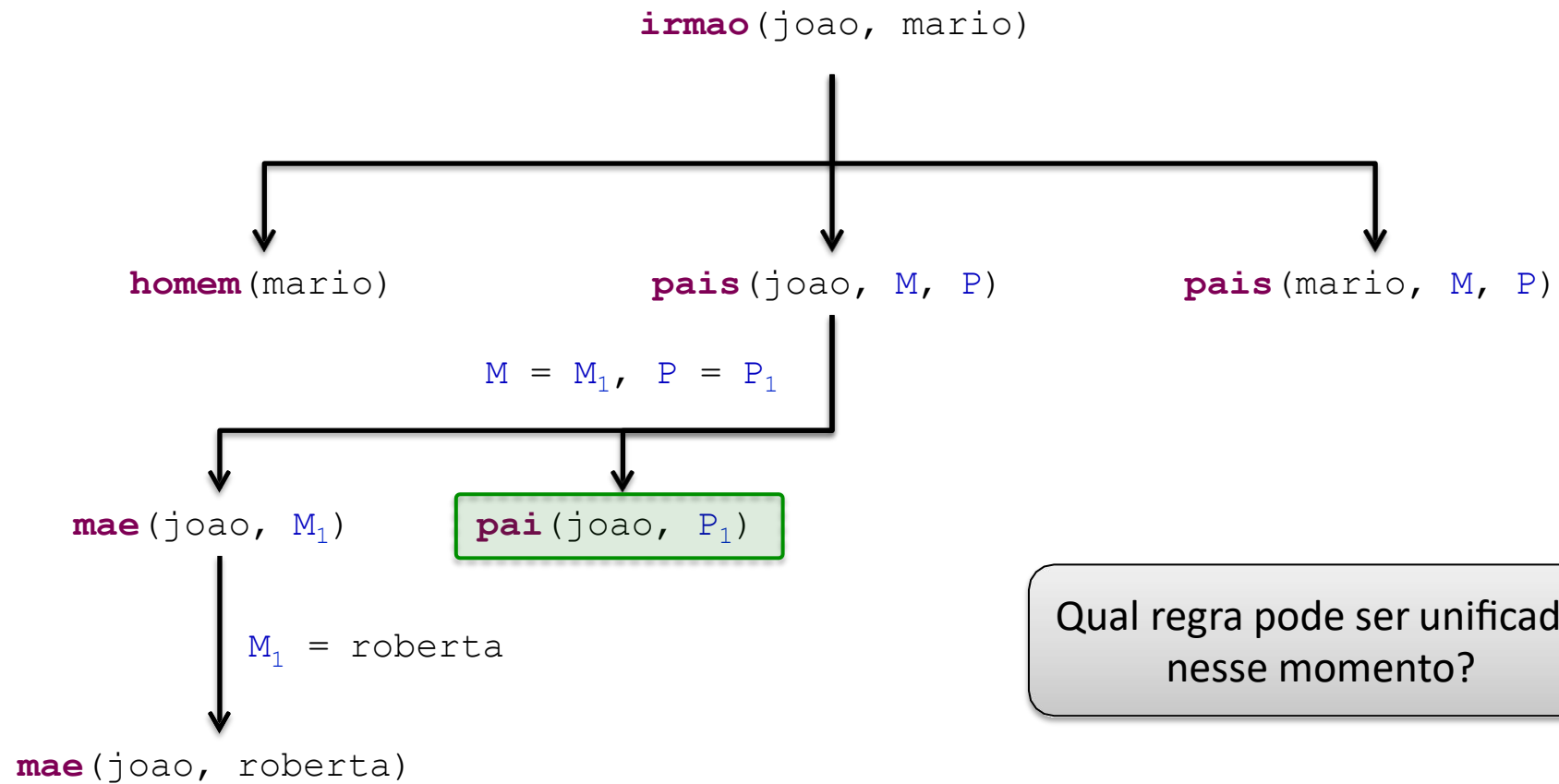
Qual regra pode ser unificada nesse momento?

EXEMPLO

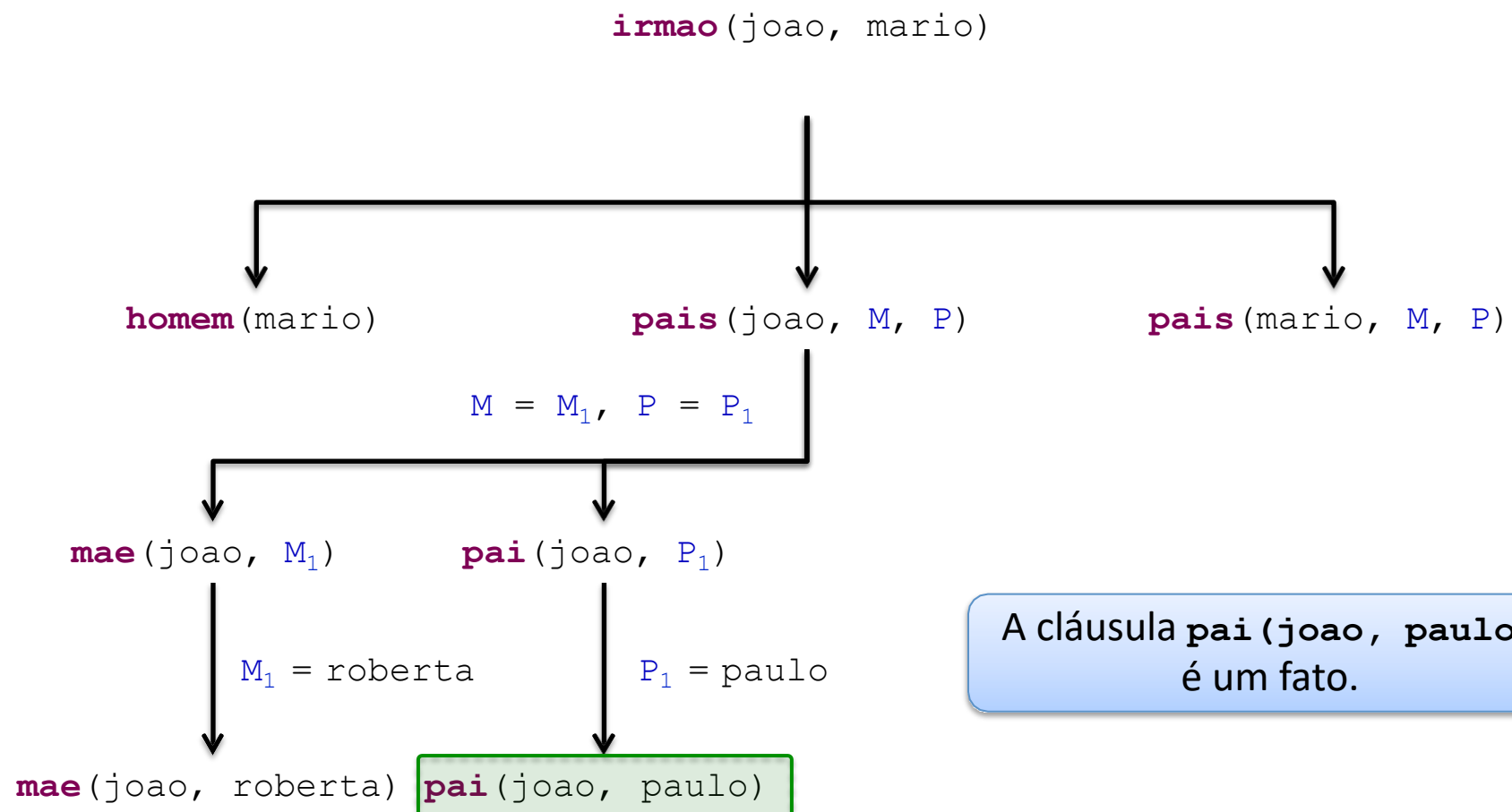


A cláusula `mae(joao, roberta)`
é um fato.

EXEMPLO

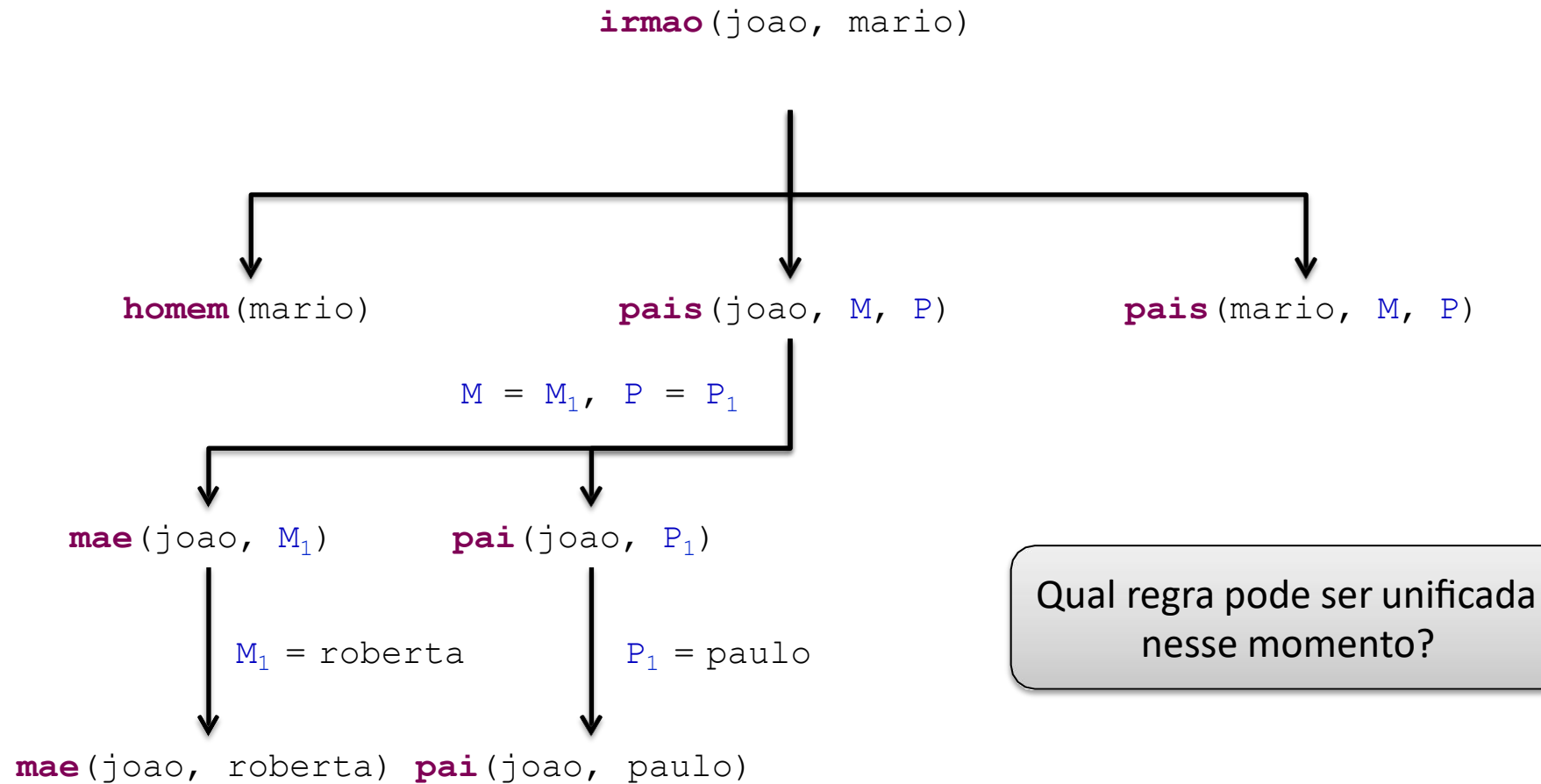


EXEMPLO

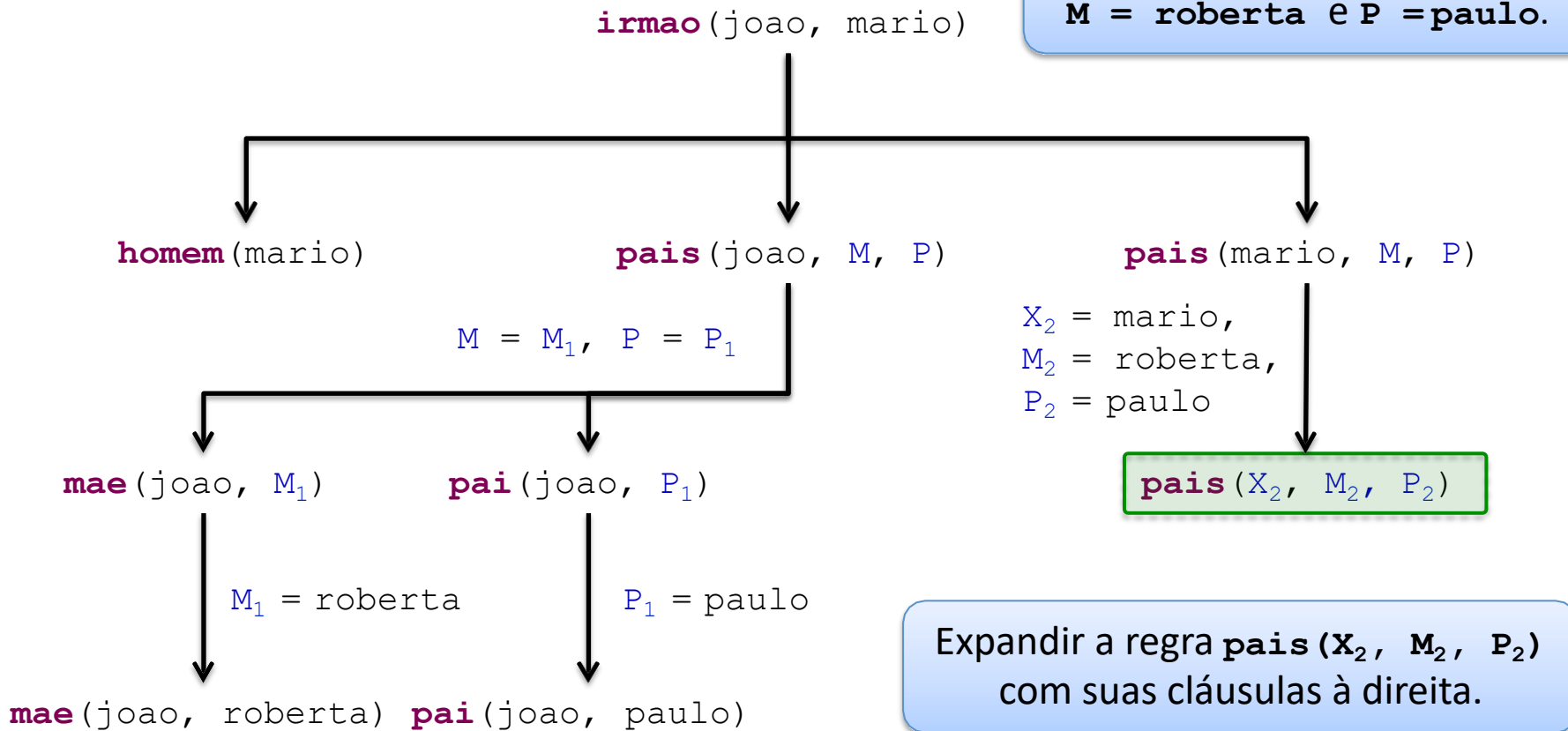


A cláusula `pai(joao, paulo)`
é um fato.

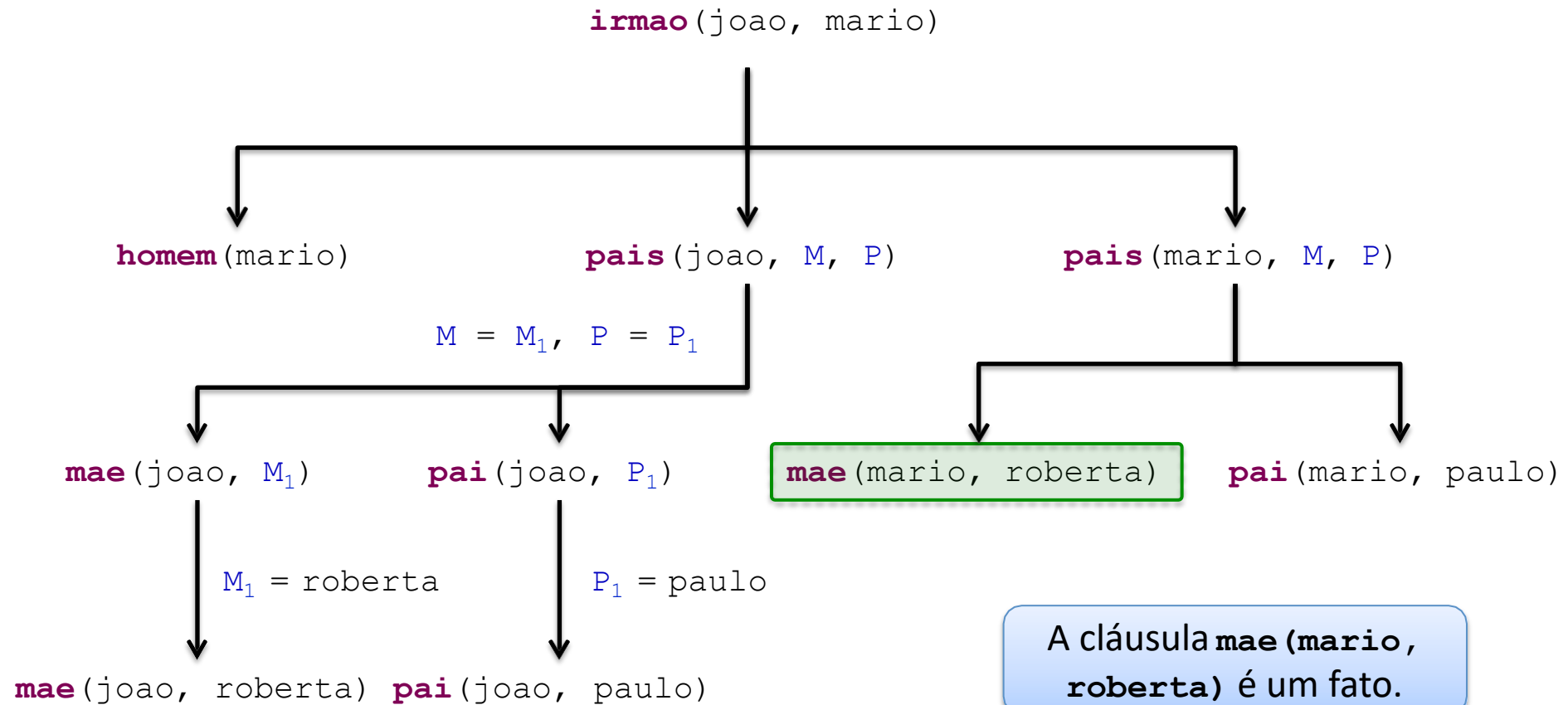
EXEMPLO



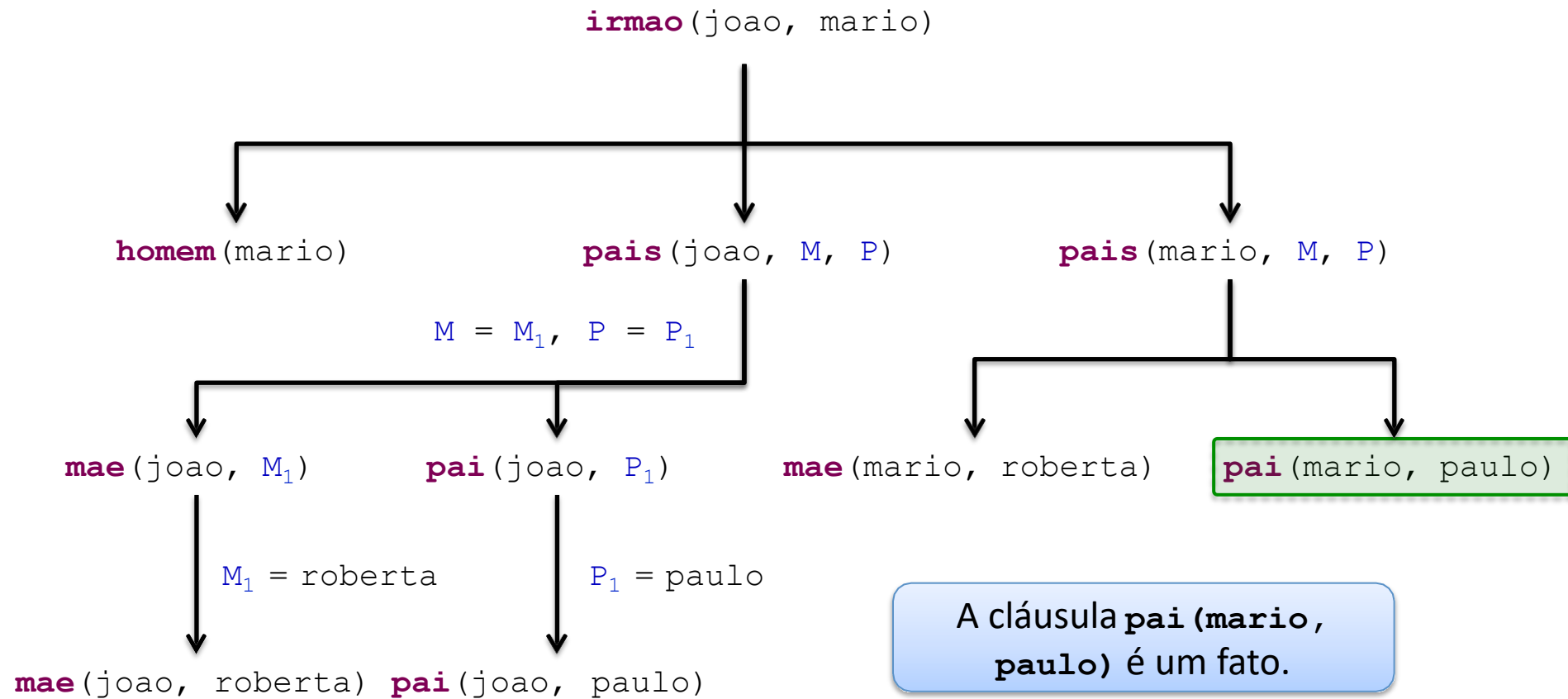
EXEMPLE



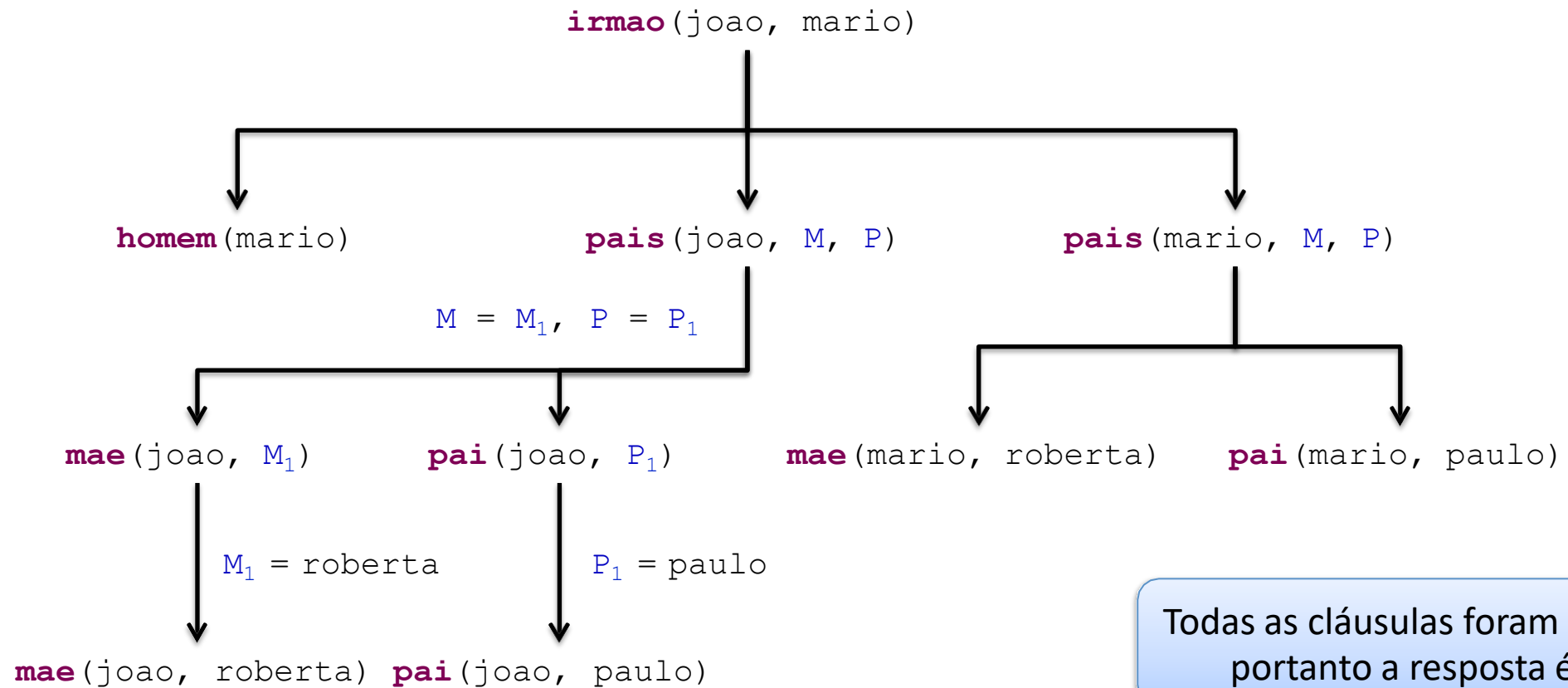
EXEMPLO



EXEMPLO



EXEMPLO



EXERCÍCIO



UNIVERSIDADE
FEDERAL DO CEARÁ

Enunciado:

Pouco se sabe da história passada da família Pinheiro. Existem alguns registros antigos que indicam que o casal José e Maria criou dois filhos, o João e a Ana. Que a Ana teve duas filhas, a Helena e a Joana, também parece ser verdade, segundo os mesmos registros. Além disso, o Mário é filho do João, pois muito se orgulha ele disso. Estranho também, foi constatar que o Carlos nasceu da relação entre a Helena, muito formosa, e o Mário.

EXERCÍCIO



UNIVERSIDADE
FEDERAL DO CEARÁ

- a) Utilizando o predicado progenitor(X,Y) (ou seja, X é progenitor de Y), represente em Prolog todos os progenitores da família Pinheiro.
- b) Represente em Prolog as relações: sexo (masculino ou feminino), irmã, irmão, descendente, mãe, pai, avô, tio, primo.
- c) Formule em Prolog as seguintes questões:
 - 1. O João é filho do José?
 - 2. Quem são os filhos da Maria?
 - 3. Quem são os primos do Mário?
 - 4. Quem são os sobrinhos/sobrinhas com um Tio existem na família Pinheiro?
 - 5. Quem são os ascendentes do Carlos?
 - 6. A Helena tem irmãos? E irmãs?

- São cadeias compostas pelos seguintes caracteres:
 - letras maiúsculas: A, B, ..., Z
 - letras minúsculas: a, b, ..., z
 - dígitos: 0, 1, ..., 9
 - caracteres especiais: + - * / < > = : . & _ ~
- Podem ser construídos por:
 - cadeias de letras, dígitos e o caractere '_', começando com uma letra minúscula: anna, nil, x25, x_25, x_25AB, x_, x__y, tem_filhos, tem_um_filho

NÚMEROS



UNIVERSIDADE
FEDERAL DO CEARÁ

- Números usados em Prolog incluem números inteiros e números reais

Operadores Aritméticos	
adição	+
subtração	-
multiplicação	*
divisão	/
divisão inteira	//
resto divisão inteira	mod
potência	**
atribuição	is

Operadores Relacionais	
$X > Y$	X é maior do que Y
$X < Y$	X é menor do que Y
$X \geq Y$	X é maior ou igual a Y
$X \leq Y$	X é menor ou igual a Y
$X =:= Y$	X é igual a Y
$X = Y$	X unifica com Y
$X \neq Y$	X é diferente de Y

NÚMEROS



UNIVERSIDADE
FEDERAL DO CEARÁ

- O operador `=` tenta unificar apenas
 - `?- X = 1 + 2.`
 - `X = 1 + 2`
- O operador `is` força a avaliação aritmética
 - `?- X is 1 + 2.`
 - `X = 3`
- Se a variável à esquerda do operador `is` já estiver instanciada, Prolog apenas compara o valor da variável com o resultado da expressão à direita de `is`
 - `?- X = 3, X is 1 + 2.`
 - `X = 3`
 - `?- X = 5, X is 1 + 2.`
 - `false`

- São cadeias de letras, dígitos e caracteres '_', sempre começando com letra maiúscula ou com o caractere '_'
 - X, Resultado, Objeto3, Lista_Alunos, ListaCompras, _x25, _32

O escopo de uma variável é dentro de uma mesma regra ou dentro de uma pergunta

- Isto significa que se a variável X ocorre em duas regras/perguntas, então são duas variáveis distintas
- Mas a ocorrência de X dentro de uma mesma regra/pergunta significa a mesma variável

UNIFICAÇÃO DE TERMOS



UNIVERSIDADE
FEDERAL DO CEARÁ

- Dois termos unificam (matching) se:
 - Eles são idênticos ou
 - As variáveis em ambos os termos podem ser instanciadas a objetos de maneira que após a substituição das variáveis por esses objetos os termos se tornam idênticos
- Por exemplo, há unificação entre os termos
 - `data(D,M,2003)` e `data(DI,maio,A)`
 - instanciando $D = DI$, $M = \text{maio}$, $A = 2003$

UNIFICAÇÃO DE TERMOS



UNIVERSIDADE
FEDERAL DO CEARÁ

- ?- $\text{data}(D, M, 2003) = \text{data}(DI, \text{maio}, A)$, $\text{data}(D, M, 2003) = \text{data}(I5, \text{maio}, AI)$.
- $D = I5$
- $M = \text{maio}$
- $DI = I5$
- $A = 2003$
- $AI = 2003$

UNIFICAÇÃO DE TERMOS



UNIVERSIDADE
FEDERAL DO CEARÁ

- Por outro lado, não há unificação entre os termos
 - `data(D,M,2003)` e `data(DI,MI,I948)`
 - `data(X,Y,Z)` e `ponto(X,Y,Z)`
- A unificação é um processo que toma dois termos e verifica se eles unificam
 - Se os termos não unificam, o processo falha (e as variáveis não se tornam instanciadas)
 - Se os termos unificam, o processo tem sucesso e também instancia as variáveis em ambos os termos para os valores que os tornam idênticos

UNIFICAÇÃO DE TERMOS



UNIVERSIDADE
FEDERAL DO CEARÁ

- As regras que regem se dois termos S e T unificam são:
 - Se S e T são constantes, então S e T unificam somente se são o mesmo objeto
 - Se S for uma variável e T for qualquer termo, então unificam e S é instanciado para T

COMPARAÇÃO DE TERMOS



UNIVERSIDADE
FEDERAL DO CEARÁ

Operadores Relacionais	
$X = Y$	X unifica com Y que é verdadeiro quando dois termos são o mesmo. Entretanto, se um dos termos é uma variável, o operador = causa a instanciização da variável porque o operador causa unificação
$X \neq Y$	X não unifica com Y que é o complemento de $X=Y$
$X == Y$	X é literalmente igual a Y (igualdade literal), que é verdadeiro se os termos X e Y são idênticos, ou seja, eles têm a mesma estrutura e todos os componentes correspondentes são os mesmos, incluindo o nome das variáveis
$X \neq Y$	X não é literalmente igual a Y que é o complemento de $X==Y$
$X @< Y$	X precede Y
$X @> Y$	Y precede X
$X @=< Y$	X precede ou é igual a Y
$X @>= Y$	Y precede ou é igual a X

COMPARAÇÃO DE TERMOS



UNIVERSIDADE
FEDERAL DO CEARÁ

- $?- f(a,b) == f(a,b).$
- true
- $?- f(a,b) == f(a,X).$
- false
- $?- f(a,X) == f(a,Y).$
- false
- $?- X == X.$
- true
- $?- X == Y.$
- false

COMPARAÇÃO DE TERMOS



UNIVERSIDADE
FEDERAL DO CEARÁ

- $?- X \text{ \texttt{==} } Y.$
- `true`
- $?- X \text{ \texttt{=} } Y.$
- `false`

PRECEDÊNCIA DE TERMOS



UNIVERSIDADE
FEDERAL DO CEARÁ

- A precedência entre termos simples é determinado para ordem alfabética ou numérica
- Variável livres @< números @< átomos @< estruturas

PRECEDÊNCIA DE TERMOS



UNIVERSIDADE
FEDERAL DO CEARÁ

- $?- X @< 10.$
- true
- $?- X @< \text{isaque}.$
- true
- $?- X @< f(X,Y).$
- true
- $?- 10 @< \text{sara}.$
- true
- $?- 10 @< f(X,Y).$
- true
- $?- \text{isaque} @< \text{sara}.$
- true

EXEMPLO



UNIVERSIDADE
FEDERAL DO CEARÁ

- Considere uma mesa com a seguinte configuração de pessoas:
- joao maria jose julia jorge ana íris
- Isto é, João está imediatamente à esquerda de maria, que está imediatamente à esquerda de José, e assim por diante.
- Crie um predicado para indicar quem está imediatamente a direita
- Crie um predicado para indicar quem está imediatamente a esquerda
- Crie um predicado que indica se duas pessoas são vizinhas de uma terceira
- Crie um predicado adjacente que indica se duas pessoas são vizinhas