

# Prolog

Programação Lógica

A series of horizontal lines in teal and light blue colors, with varying lengths and offsets, creating a modern, layered effect across the middle of the slide.

# Aritmética

Prolog oferece um predicado especial `is`, bem como um conjunto *operadores*, através dos quais podemos efetuar operações aritméticas.

```
?- X is 2+3.
```

```
X = 5
```

Os operadores aritméticos são `+` (*adição*), `-` (*subtração*), `*` (*multiplicação*), `mod` (*resto*), `/` (*divisão real*), `//` (*divisão inteira*) e `^` (*potenciação*).

# Aritmética

**Programa 2.1:** *Área e população dos países.*

```
% país (Nome, Área, População)
```

```
país (brasil, 9, 130) .
```

```
país (china, 12, 1800) .
```

```
país (eua, 9, 230) .
```

```
país (índia, 3, 450) .
```

# Aritmética

O Programa 2.1 representa uma tabela que relaciona a cada país sua área em Km<sup>2</sup> e sua população em milhões de habitantes. Note que a linha iniciando com % é um comentário e serve apenas para fins de documentação. Com base nesse programa, por exemplo, podemos determinar a densidade demográfica do Brasil, através da seguinte consulta:

```
?- país(brasil,A,P), D is P/A.  
A = 9  
P = 130  
D = 14.4444
```

Uma outra consulta que poderia ser feita é a seguinte: *"Qual a diferença entre a população da China e da Índia?"*.

# Comparação

Para realizar comparações numéricas podemos usar os seguintes predicados primitivos:  $=:=$  (*igual*) ,  $\neq$  (*diferente*),  $>$  (*maior*),  $\geq$  (*maior ou igual*),  $<$  (*menor*) e  $\leq$  (*menor ou igual*). Por exemplo, com esses predicados, podemos realizar consultas tais como:

- *A área do Brasil é igual à área dos Estados Unidos?*

```
?- país(brasil,X,_), país(eua,Y,_), X ::= Y.
```

```
X = 9
```

```
Y = 9
```

```
Yes
```

- *A população dos Estados Unidos é maior que a população da Índia?*

# Relacionamento entre tabelas

O quadro a seguir relaciona a cada funcionário de uma empresa seu *código*, seu *salário* e os seus *dependentes*.

Código	Nome	Salário	Dependentes
1	Ana	R\$ 1000,90	Ary
2	Bia	R\$ 1200,00	-----
3	Ivo	R\$ 903,50	Raí, Eva

# Relacionamento entre tabelas

Outra coisa que podemos fazer é descobrir de quem Ary é dependente:

Ou, descobrir quem depende de funcionário com salário inferior a R\$ 950,00:

Finalmente, poderíamos também consultar o sistema para encontrar funcionários que não têm dependentes:

# O Modelo de dados Relacional

Programas lógicos são uma poderosa extensão do modelo de dados relacional. Conjuntos de fatos correspondem às tabelas do modelo relacional e as operações básicas da álgebra relacional (*seleção, projeção, união, diferença simétrica e produto cartesiano*) podem ser facilmente implementadas através de regras. Como exemplo, considere o Programa 2.3.

## **Programa 2.3:** *Uma tabela de filmes.*

```
% filme(Título, Gênero, Ano, Duração)
filme('Uma linda mulher', romance, 1990, 119).
filme('Sexto sentido',      suspense, 2001, 108).
filme('A cor púrpura',      drama,    1985, 152).
filme('Copacabana',         comédia, 2001,  92).
filme('E o vento levou',    drama,   1939, 233).
filme('Carrington',         romance, 1995, 130).
```



# Exercícios

Inclua no Programa 2.1 uma regra para o predicado  $\text{dens}(P, D)$ , que relaciona cada país  $P$  à sua densidade demográfica correspondente  $D$ . Em seguida, faça consultas para descobrir:

- a) *qual a densidade demográfica de cada um dos países;*
- b) *se a Índia é mais populosa que a China.*

Inclua no Programa 2.2 as informações da tabela abaixo e faça as consultas indicadas a seguir:

Código	Nome	Salário	Dependentes
4	Leo	R\$ 2500,35	Lia, Noé
5	Clô	R\$ 1800,00	Eli
6	Gil	R\$ 1100,00	-----

- a) *Quem tem salário entre R\$ 1500,00 e R\$ 3000,00?*
- b) *Quem não tem dependentes e ganha menos de R\$ 1200,00?*
- c) *Quem depende de funcionário que ganha mais de R\$ 1700,00?*

# Exercícios

Inclua no Programa 2.3 as seguintes regras:

- a) *Um filme é longo se tem duração superior a 150 minutos.*
- b) *Um filme é lançamento se foi lançado a menos de 1 ano.*

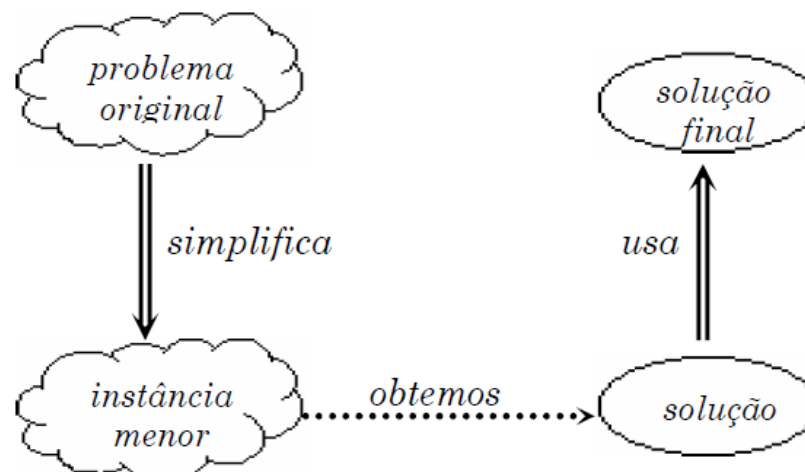
Codifique um programa contendo as informações da tabela abaixo e faça as consultas indicadas a seguir:

Nome	Sexo	Idade	Altura	Peso
Ana	fem	23	1.55	56.0
Bia	fem	19	1.71	61.3
Ivo	masc	22	1.80	70.5
Lia	fem	17	1.85	57.3
Eva	fem	28	1.75	68.7
Ary	masc	25	1.72	68.9

- a) *Quais são as mulheres com mais de 20 anos de idade?*
- b) *Quem tem pelo menos 1.70m de altura e menos de 65kg?*
- c) *Quais são os possíveis casais onde o homem é mais alto que a mulher?*

# Recursividade

A *recursividade* é um princípio que nos permite obter a solução de um problema a partir da solução de uma instância menor dele mesmo. Para aplicar esse princípio, devemos assumir como hipótese que a solução da instância menor é conhecida. Por exemplo, suponha que desejamos calcular  $2^{11}$ . Uma instância menor desse problema é  $2^{10}$  e, para essa instância, "sabemos" que a solução é 1024. Então, como  $2 \times 2^{10} = 2^{11}$ , concluímos que  $2^{11} = 2 \times 1024 = 2048$ .



# Predicados Recursivos

A definição de um *predicado recursivo* é composta por duas partes:

1º *base*: resolve diretamente a instância mais simples do problema.

2º *passo*: resolve instâncias maiores, usando o princípio de recursividade.

## Programa 4.1: Cálculo de potência.

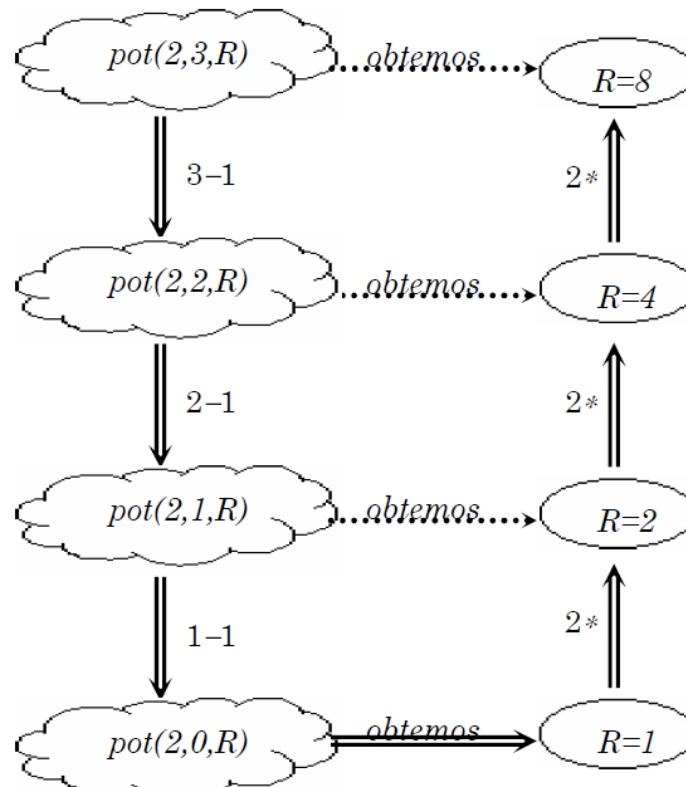
```
% pot (Base, Expoente, Potência)
pot (_, 0, 1) .           % base
pot (B, N, P) :-         % passo
    N > 0,                % condição do passo
    M is N-1,             % simplifica o problema
    pot (B, M, R),        % obtém solução da instância menor
    P is B*R.             % constrói solução final
```

# Predicados Recursivos

O Programa 4.1 mostra a definição de um predicado para calcular potências. A base para esse problema ocorre quando o expoente é 0, já que qualquer número elevado a 0 é igual a 1. Por outro lado, se o expoente é maior que 0, então o problema deve ser simplificado, ou seja, temos que chegar um pouco mais perto da base. A chamada recursiva com  $M$  igual a  $N-1$  garante justamente isso. Portanto, após um número finito de passos, a base do problema é atingida e o resultado esperado é obtido.

# Predicados Recursivos

Um modo de entender o funcionamento dos predicados recursivos é desenhar o *fluxo de execução*.



# Exercícios

- Construa uma recursividade para o fatorial
- Construa o fluxo de execução para o fatorial 3