

# APS MATEMÁTICA DISCRETA

**Cenário:** Empresa de Tecnologia "*SwordFish Consulting*" é uma *software house*/consultoria. A *SwordFish* desenvolve produtos de software. Ela possui funcionários que podem ser gerentes ou desenvolvedores. Projetos são criados para desenvolver produtos específicos. Gerentes supervisionam projetos e desenvolvedores trabalham neles.

## Constantes:

Funcionários: yuri, vinicius, carlos, mateus

Cargos: gerente, desenvolvedor

Projetos: FastTracker, CharlieProject

Produtos: CRM, micro SaaS

## Predicados:

$F(x)$  —  $x$  é um funcionário da empresa

$C(x, y)$  — O funcionário  $x$  ocupa o cargo  $y$ .

$Proj(x)$  —  $x$  é um projeto

$P(x)$  —  $x$  é um produto final

$G(x, y)$  — O funcionário  $x$  gerencia o projeto  $y$ .

$T(x, y)$  — O funcionário  $x$  trabalha no projeto  $y$

$D(x, y)$  — O projeto  $x$  desenvolve o produto  $y$

$R(x, y)$  — O funcionário  $x$  é responsável pelo produto  $y$

### 1. Todo projeto deve desenvolver pelo menos um produto:

$$\forall x (Proj(x) \rightarrow \exists y (P(y) \wedge D(x, y)))$$

### 2. Somente funcionários podem gerenciar projetos:

$$\forall x \forall y (G(x, y) \rightarrow F(x))$$

### 3. Todo projeto tem pelo menos um gerente:

$$\forall x (Proj(x) \rightarrow \exists y (F(y) \wedge G(y, x)))$$

### 4. Um funcionário só pode gerenciar projetos:

$$\forall x \forall y (G(x, y) \rightarrow Proj(y))$$

5. **Um projeto não pode desenvolver mais de um produto:**  
 $\forall x \forall y \forall z ((D(x, y) \wedge D(x, z)) \rightarrow y = z)$
6. **Todo gerente trabalha no projeto que gerencia:**  
 $\forall x \forall y (G(x, y) \rightarrow T(x, y))$
7. **Se alguém trabalha em um projeto que desenvolve um produto, essa pessoa é responsável (parcialmente) por esse produto:**  
 $\forall x \forall y \forall z (T(x, y) \wedge D(y, z) \rightarrow R(x, z))$
8. **Todo desenvolvedor deve trabalhar em pelo menos um projeto:**  
 $\forall x (C(x, \text{desenvolvedor}) \rightarrow \exists y (\text{Proj}(y) \wedge T(x, y)))$
9. **Um funcionário não pode ser gerente e desenvolvedor ao mesmo tempo:**  
 $\neg \exists x (C(x, \text{gerente}) \wedge C(x, \text{desenvolvedor}))$
10. **Apenas quem tem cargo é considerado funcionário:**  
 $\forall x (\text{Funcionario}(x) \leftrightarrow (\exists y (C(x, y))))$

### 2.3 Dedução Natural:

"Se um funcionário gerencia um projeto que desenvolve um produto, então esse funcionário é responsável por esse produto."

- **Premissa 1:**  $\forall x \forall y (G(x, y) \rightarrow T(x, y))$ 
  - (Todo gerente trabalha no projeto que gerencia)
- **Premissa 2:**  $\forall x \forall y \forall z (T(x, y) \wedge D(y, z) \rightarrow R(x, z))$ 
  - (Se alguém trabalha em um projeto que desenvolve um produto, essa pessoa é responsável por esse produto)

Queremos provar:  $\forall x \forall y \forall z (G(x, y) \wedge D(y, z) \rightarrow R(x, z))$

Passo	Fórmula	
1.	$\forall y \forall x (G(x, y) \rightarrow T(x, y))$	Premissa
2.	$\forall x \forall y \forall z (T(x, y) \wedge D(y, z) \rightarrow R(x, z))$	Premissa
3.	$G(a, b) \wedge D(b, c)$	Hip
4.	$G(a, b)$	$\wedge e$ 3
5.	$D(b, c)$	$\wedge e$ 3
6.	$G(a, b) \rightarrow T(a, b)$	$\forall e$ 1
7.	$T(a, b)$	$\rightarrow e$ MP 4, 6
8.	$T(a, b) \wedge D(b, c)$	$\wedge i$ de 7, 5
9.	$T(a, b) \wedge D(b, c) \rightarrow R(a, c)$	$\forall e$ 2
10.	$R(a, c)$	$\rightarrow e$ MP 8, 9
11.	$(G(a, b) \wedge D(b, c)) \rightarrow R(a, c)$	$\rightarrow i$ 3-10
12.	$\forall x \forall y \forall z (G(x, y) \wedge D(y, z) \rightarrow R(x, z))$	$\forall i$ 11

## 2.4 Implementação em Prolog

% Predicado: F(x) (x é um funcionário)

f(yuri).

f(vinicius).

f(carlos).

f(mateus).

% Predicado: C(x, y) ( O funcionário x ocupa o cargo y)

c(yuri, gerente).

c(vinicius, desenvolvedor).

c(carlos, desenvolvedor).

c(mateus, gerente).

% Predicado: Proj(x) (x é um projeto)

proj(fastTracker).

proj(charlieProject).

% Predicado: P(x) (x é um produto final)

p(crm).

p(microSaaS).

% Predicado: G(x, y) (O funcionário x gerencia o projeto y)

g(yuri, fastTracker).

g(mateus, charlieProject).

% Predicado: T(x, y) (O funcionário x trabalha no projeto y)

t(yuri, fastTracker).

t(mateus, charlieProject).

t(vinicius, fastTracker).

t(carlos, fastTracker).

t(carlos, charlieProject).

% Predicado: D(x, y) (O projeto x desenvolve o produto y)

d(fastTracker, crm).

d(charlieProject, microSaaS).

% Fórmula 2: Somente funcionários podem gerenciar projetos

%  $\forall x \forall y (G(x, y) \rightarrow F(x))$

f(X) :- g(X, \_Y).

% (Se X gerencia algo, X é um funcionário)

f(X) :- g(X, \_).

% Fórmula 6: Todo gerente trabalha no projeto que gerencia

%  $\forall x \forall y (G(x, y) \rightarrow T(x, y))$

t(X, Y) :- g(X, Y).

% Fórmula 7: Se alguém trabalha em um projeto que desenvolve um produto,  
% essa pessoa é responsável (parcialmente) por esse produto.

%  $\forall x \forall y \forall z (T(x, y) \wedge D(y, z) \rightarrow R(x, z))$

r(X, Z) :- t(X, Y), d(Y, Z).

% Fórmula 10: Apenas quem tem cargo é considerado funcionário

%  $\forall x (F(x) \leftrightarrow (\exists y C(x, y)))$

f(X) :- c(X, \_).

Consultas:

% 1. O vinicius é desenvolvedor?

?- c(vinicius, desenvolvedor).

% R: true.

% 2. (Inferência com Regra 6) O yuri trabalha no FastTracker?

% (Isso é um fato, mas também é inferido pela regra 6)

?- t(yuri, fastTracker).

% R: true.

% 3. (Inferência com Regra 7) O vinicius é responsável pelo CRM?

% (Ele T(vinicius, fastTracker) e D(fastTracker, crm) -> R(vinicius, crm))

?- r(vinicius, crm).

% R: true.

% 4. (Inferência com Regra 7) Quem é responsável pelo microSaaS?

% (Prolog buscará quem T(X, Y) e D(Y, microSaaS))

?- r(X, microSaaS).

% R: X = mateus ;

% R: X = carlos.

% 5. (Consulta com variável) Quais projetos o carlos trabalha?

?- t(carlos, Y).

```
% R: Y = fastTracker ;  
% R: Y = charlieProject.
```