

# Aula 08 - Satisfação de Restrições

Profa. Gabrielly Queiroz

# Satisfação de Restrições - Constraint Satisfaction Problems - CSP

No lugar de encontrar um caminho em um espaço de busca, queremos **encontrar uma solução que satisfaça um conjunto de restrições específicas.**

CSP é um problema onde o objetivo é encontrar um estado ou conjunto de estados que satisfaça todas as restrições impostas a um conjunto de variáveis.

# Estrutura de um Problema CSP

## Componentes Básicos:

Em problemas de Satisfação de Restrições, temos:

- **Variáveis:** O que queremos resolver ou encontrar. Ex.: X, Y, Z.
- **Domínios:** Os valores possíveis que cada variável pode assumir. Ex.:  $X \in \{1, 2, 3\}$ ,  $Y \in \{1, 2, 3\}$ ,  $Z \in \{1, 2, 3\}$ .
- **Restrições:** Regras que determinam quais combinações de valores são aceitáveis. Ex.:  $X \neq Y$ ,  $Y < Z$ .

## Formalização Matemática:

Um CSP é definido como um triplo  $(X, D, C)$ , onde:

- $X = \{X_1, X_2, \dots, X_n\}$  é o conjunto de variáveis.
- $D = \{D_1, D_2, \dots, D_n\}$  é o conjunto de domínios de cada variável  $X_i$ .
- $C = \{C_1, C_2, \dots, C_m\}$  é o conjunto de restrições.

## Exemplo:

Problema de Colorização de Mapas. Variáveis são regiões a serem coloridas, domínios são cores disponíveis, e restrições garantem que regiões adjacentes tenham cores diferentes.

# Exemplo

Temos o Mapa ao lado, mas podemos pintar cada região com apenas três cores: verde, vermelho e azul.

As cores não podem ser a mesma cor para os seus vizinhos.



# Exemplo

## Variáveis:

- N,NE,SE,CO,S.

## Domínios:

- {verde,vermelho,azul}

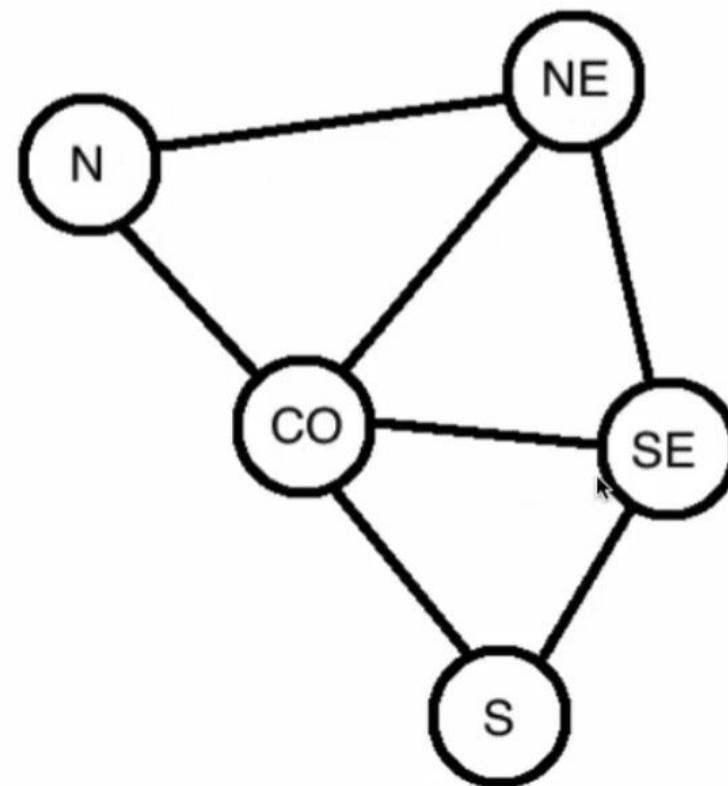
## Restrições:

- Regiões adjacentes devem ter cores diferentes.
- Por exemplo,  $N \neq NE$
- Relações válidas entre variáveis adjacentes (N,NE)

## Soluções:

- Conjuntos de atribuições que sejam completos e consistentes, ou seja, todas as variáveis são atribuídas a cores, e todas as restrições são satisfeitas.

# Exemplo



# Tipos de CSPs

## CSPs Discretos:

- Domínios finitos
- Domínios infinitos

## CSPs Contínuos:

Variáveis que assumem valores contínuos, como problemas de engenharia.

## CSPs Estruturados:

- CSPs de grande escala que podem ser divididos em subproblemas menores.

# CSPs Discretos (Discrete CSPs)

Aqui, as variáveis têm um conjunto limitado (finito) ou ilimitado (infinito) de valores possíveis.

- ◆ Domínios Finitos (Comumente resolvidos por técnicas como Backtracking, Arc-Consistency, etc.)
  - Coloração de Mapas (Map Coloring):
    - Problema clássico onde queremos colorir um mapa usando o menor número possível de cores.
    - Variáveis: Regiões do mapa (ex.: A, B, C, D).
    - Domínios: {Vermelho, Verde, Azul}.
    - Restrições: Regiões adjacentes não podem ter a mesma cor.

Domínios Infinitos (Normalmente tratados por discretização ou heurísticas específicas)

- Quebra-cabeças Numéricos (Ex.: Equações Algébricas):
  - Variáveis: Variáveis numéricas (ex.: x,y,z).
  - Domínios: Todos os números inteiros ou reais possíveis.
  - Restrições: Regras matemáticas que devem ser satisfeitas (ex.:  $x^2+y^2=25$ ).

# CSPs Contínuos (Continuous CSPs)

Nestes problemas, as variáveis podem assumir **qualquer valor dentro de um intervalo contínuo**.

- ♦ **Exemplos:**

- **Otimização em Engenharia:**
  - Problemas onde queremos minimizar ou maximizar funções contínuas.
  - Variáveis: Parâmetros físicos (ex.: pressão, temperatura, corrente elétrica).
  - Domínios: Intervalos contínuos (ex.:  $0 \leq x \leq 100$ ).
  - Restrições: Fórmulas que devem ser respeitadas (ex.:  $f(x,y) \leq 10$ ).
- **Robótica:**
  - Planejamento de movimento de um robô.
  - Variáveis: Coordenadas contínuas do espaço ( $x, y, z, \theta$ ).
  - Restrições: Evitar colisões, respeitar limites físicos.
- **Aprendizado de Máquina (Otimização de Funções):**
  - Ajuste de pesos em redes neurais.
  - Variáveis: Pesos contínuos.
  - Domínios: Intervalos específicos (ex.:  $[0, 1]$ ).
  - Restrições: Funções de custo ou perda a serem minimizadas.

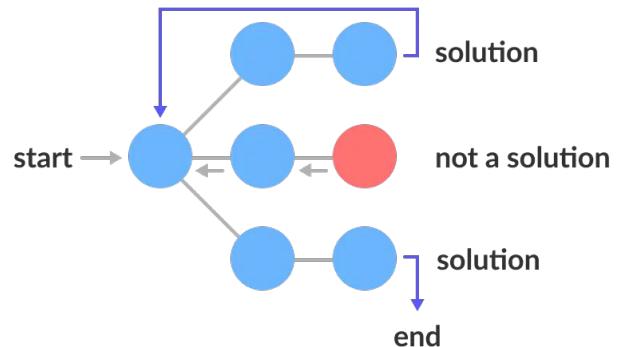
# Algoritmo para resolução - Backtracking

## Algoritmo Backtracking

O **Backtracking (Retrcesso)** é um algoritmo de **busca sistemática** usado para resolver problemas de decisão, otimização e satisfação de restrições, como os **Problemas de Satisfação de Restrições (CSPs)**.

### Passos do Algoritmo:

1. **Escolher uma variável para atribuir.**
2. **Atribuir um valor válido à variável** do seu domínio.
3. **Verificar se a atribuição é válida:**
  - o Se for válida, continuar para a próxima variável.
  - o Se não for válida, **desfazer a atribuição (retroceder)** e tentar outro valor.
4. **Repetir até que:**
  - o Uma solução válida seja encontrada (todas as variáveis atribuídas sem violar as restrições).
  - o Todas as possibilidades sejam tentadas e nenhuma solução seja encontrada.



# Exemplo - Mapa

**Escolher uma variável para atribuir.**

- Vamos começar pela região A.

**Atribuir um valor válido à variável do seu domínio.**

- Tentamos a cor **Vermelho** para A.
- Atribuição atual:
  - A = Vermelho
  - B = ?
  - C = ?
  - D = ?

**Verificar se a atribuição é válida:**

- Como A é a primeira variável, é válida até o momento.
- Avançamos para B.

**Atribuir um valor para B:**

- Tentamos a cor **Vermelho** para B.
- Como A e B são vizinhos, isso é **inválido!** (Retrocede)
- Tentamos a cor **Verde** para B (válido).
- Atribuição atual:
  - A = Vermelho
  - B = Verde
  - C = ?
  - D = ?

**Continuar o processo:**

- Atribuímos cores válidas para C e D.
- Testamos as cores possíveis e aplicamos **retrocesso sempre que ocorrer um conflito.**

# Exercício

Cinco amigos vão organizar os livros de uma biblioteca, mas cada um só pode pegar **um tipo específico de livro**. A tarefa é descobrir **quem ficou com qual tipo de livro** baseado nas seguintes pistas.

## Amigos:

- Ana, Bruno, Carlos, Daniela, Eduardo.

## Tipos de Livros:

- Romance, Mistério, Ficção Científica, História, Poesia.

## Pistas:

1. Ana não pegou o livro de **Ficção Científica** nem de **História**.
2. Carlos ficou com o livro de **Mistério**.
3. Bruno não ficou com o livro de **Romance**.
4. Eduardo ficou com o livro de **Ficção Científica**.
5. Daniela NÃO FICOU COM **Mistério**.

## Objetivo:

Descobrir **quem ficou com qual tipo de livro**.

# Resolução

## Amigos (Variáveis):

- Ana, Bruno, Carlos, Daniela, Eduardo.

## Tipos de Livros (Domínio):

- Romance, Mistério, Ficção Científica, História, Poesia.

## Pistas (Restrições):

1. Ana não pegou o livro de Ficção Científica nem de História.
2. Carlos ficou com o livro de Mistério.
3. Bruno não ficou com o livro de Romance.
4. Eduardo ficou com o livro de Ficção Científica.
5. Daniela não gosta de Mistério.

## Escolher uma variável para atribuir.

- Vamos começar atribuindo livros para **cada amigo**, um de cada vez.

## Atribuir um valor válido à variável do seu domínio.

- Testamos **todos os livros possíveis** para cada amigo.

## Verificar se a atribuição é válida:

- Cada atribuição é testada contra as **restrições** (pistas) dadas.

## Retroceder se necessário:

- Se alguma atribuição não é válida, desfazemos e tentamos outra.

# Resolução

## Etapa 1: Atribuir livro para Carlos (Pista 2).

- Carlos = **Mistério** (✓ Válido).
- Livros disponíveis: **Romance, Ficção Científica, História, Poesia.**

## Etapa 2: Atribuir livro para Eduardo (Pista 4).

- Eduardo = **Ficção Científica** (✓ Válido).
- Livros disponíveis: **Romance, História, Poesia.**

## Etapa 3: Atribuir livro para Ana (Pista 1).

- Ana NÃO pode ter: **Ficção Científica, História.**
- Testar:
  - Ana = **Romance** (✓ Válido).
- Livros disponíveis: **História, Poesia.**

## Etapa 4: Atribuir livro para Bruno (Pista 3).

- Bruno NÃO pode ter: **Romance.**
- Testar:
  - Bruno = **História** (✓ Válido).
- Livros disponíveis: **Poesia.**

## Etapa 5: Atribuir livro para Daniela (Pista 5).

- Daniela NÃO pode ter: **Mistério.**
- Sobram os livros: **Poesia.**
- Daniela = **Poesia** (✓ Válido).
  -

# Resolução

<b>Amigo</b>	<b>Livro</b>
Ana	Romance
Bruno	História
Carlos	Mistério
Daniela	Poesia
Eduardo	Ficção Científica