Problemas de Satisfação de Restrições (capítulos 3 e 4)

O que é um CSP:

- **Estado:** definido por variáveis V_i, que podem tomar valores de um domínio D_i.

  Domínio: *discreto* ou *contínuo*.

- **Teste de meta:** conjunto de restrições que devem ser satisfeitas pelos valores das variáveis.

  Restrição: *unária*, *binária*, etc.

Solução de um CSP:

- Atribuição de valores para todas as variáveis, satisfazendo o teste de meta (ou seja, todas as restrições).

$\Rightarrow$ Algoritmos específicos para CSP são mais poderosos do que algoritmos de busca padrões.

Exemplo: 4 rainhas no tabuleiro:

- **Variáveis:** $V_i, i = 1, 2, 3, 4$: linha da rainha na col. i.
- **Domínios:** $D_i = \{1, 2, 3, 4\}$.
- **Restrições:** Para $i \neq j$:
  - $V_i \neq V_j$ (linhas distintas)
  - $|V_i - V_j| \neq |i - j|$ (diagonais distintas)

Para domínios discretos:

- Cada restrição pode ser representada enumerando-se os valores permissíveis.

  Exemplo: $V_1 \neq V_2$ e $|V_1 - V_2| \neq |1 - 2|$

  $(V_1, V_2) \in \{(1, 3), (1, 4), (2, 4), (3, 1), (4, 1)(4, 2)\}$.

Grafo de restrições: cada variável é um nodo; cada restrição (binária) um arco.

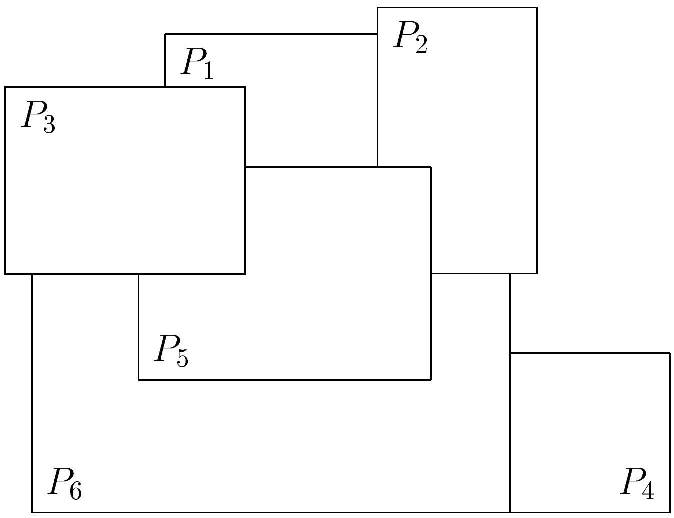
Exemplo: Criptoaritmética

```
   S E N D
 + M O R E
----------
 M O N E Y
```

- **Variáveis:** D, E, M, N, O, R, S, Y.
- **Domínios:** $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$.
- **Restrições:**
  - $M \neq 0$, $S \neq 0$ (unárias)
  - $Y = D + E$ ou $Y = D + E - 10$, etc.
  - $D \neq E$, $D \neq M$, $D \neq N$, etc.

Exemplo: Coloração de Mapas

Colorir um mapa de forma que países adjacentes não tenham a mesma cor.



- **Variáveis:** países P_i.
- **Domínios:** $\{Vermelho, Azul, Laranja\}$.
- **Restrições:** $P_1 \neq P_2$, $P_1 \neq P_5$, etc.

Busca "padrão" para CSP's:

- Estado inicial: conjunto das variáveis sem valores.
- Operadores: atribuir um valor a uma variável livre.
- Teste de meta: toda variável com valor e nenhuma restrição violada.

Representação de uma restrição:

- *Explicitamente* como conjunto de valores permissíveis.
- *Implicitamente* como uma função que a testa.

Exemplo: Coloração de Mapas

Complexidade da busca "padrão" cega:

- Profundidade máxima da árvore de busca, $m = n$ (número de variáveis).
- Profundidade de estado meta, $d = n$.
- Algoritmo a usar: busca por profundidade.
- Fator de ramificação, $b = \Sigma_i^n |D_i|$ (no topo da árvore).

Isto pode ser melhorado dramaticamente, levando em conta que:

1) Ordem de atribuição de valores é irrelevante: muitos caminhos equivalentes.

2) Atribuição de valores não corrigem uma restrição já violada.

Busca por profundidade é o algoritmo base para CSP's. Mas:

1) Fixar uma ordem de atribuição; com isto, $b = |D_i|$.

2) Checar por violação de restrições.

$\Rightarrow$ Resolve Prob. das n rainhas para $n \approx 15$.

Checagem à frente (*forward checking*)

Idéia:

- Atualizar valores legais restantes para variáveis que ainda não receberam valores.
- Terminar a busca se alguma variável não tem valor legal possível.

Exemplo: coloração de mapas

	V	A	L
P_1			
P_2			
P_3			
P_4			
P_5			
P_6			

$\longrightarrow$

	V	A	L
P_1	$\surd$		
P_2	$\times$		
P_3	$\times$		
P_4			
P_5	$\times$		
P_6			

$\longrightarrow$

	V	A	L
P_1	$\surd$		
P_2	$\times$	$\surd$	
P_3	$\times$		
P_4			
P_5	$\times$	$\times$	
P_6		$\times$	

	V	A	L
P_1	$\surd$		
P_2	$\times$	$\surd$	
P_3	$\times$	$\surd$	
P_4			
P_5	$\times$	$\times$	
P_6		$\times$	

$\longrightarrow$

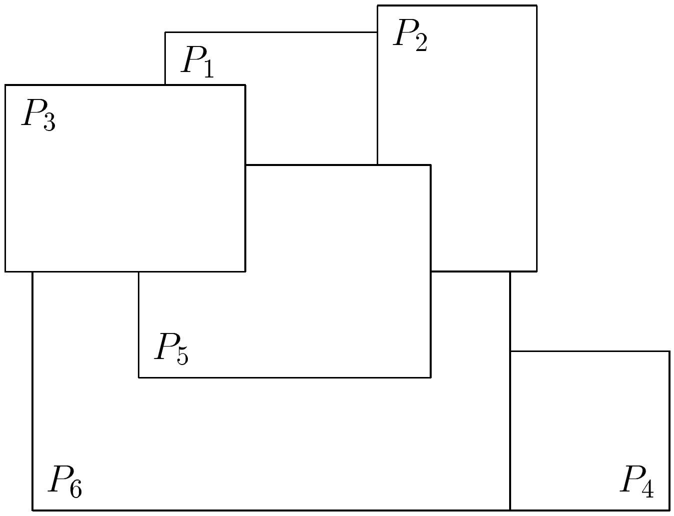
	V	A	L
P_1	$\surd$		
P_2	$\times$	$\surd$	
P_3	$\times$	$\surd$	
P_4	$\surd$		
P_5	$\times$	$\times$	
P_6	$\times$	$\times$	

$\longrightarrow$

	V	A	L
P_1	$\surd$		
P_2	$\times$	$\surd$	
P_3	$\times$	$\surd$	
P_4	$\surd$		
P_5	$\times$	$\times$	$\surd$
P_6	$\times$	$\times$	$\times$

$\Rightarrow$ Resolve Prob. das n rainhas para $n \approx 30$.

Heurísticas para CSP's



Dados:

- $P_1 =$ V, $P_2 =$ L. $P_3 =?$

  $P_3 = L \Rightarrow$ *valor menos restringente*

- $P_1 =$ V, $P_2 =$ L. Que variável escolher em seguida?

  $P_5 \Rightarrow$ *variável mais restrita*

$\Rightarrow$ Resolve Prob. das n rainhas para $n \approx 1000$.

Algoritmos Iterativos para CSP's

Hill-climbing e *Simulated annealing* são aplicados a estados completos (todas as variáveis com valor).

Para aplicar tais algoritmos a CSP's:

- Permitir estados com restrições violadas.
- Operadores *reatribuem* valores de variáveis.

Escolha de variável:

Variável que viola restrição.

Escolha de novo valor para variável:

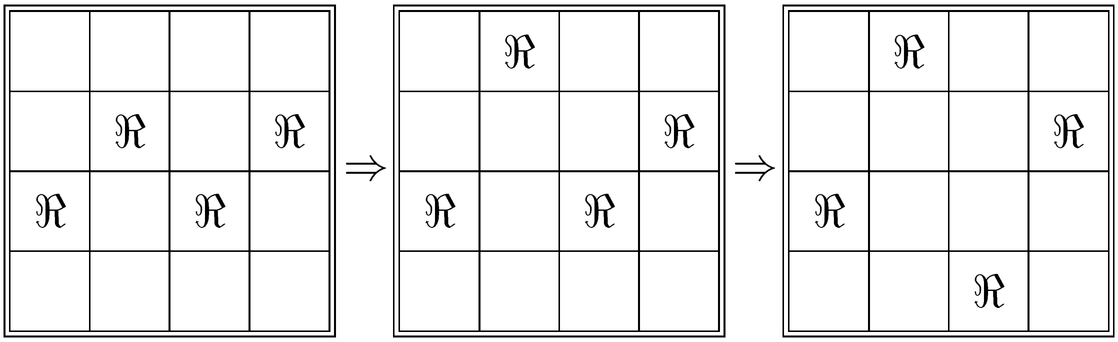
Heurística de **conflitos mínimos** (*min-conflicts*):

- Escolha valor que viola menos restrições. Ou seja, $h(n)$ = número de restrições violadas.

Exemplo: 4 rainhas

- **Estados:** 4 rainhas em 4 colunas (4^4=256 estados).
- **Operadores:** mover rainha em uma coluna.
- **Teste de meta:** nenhuma rainha ameaça outra.
- **Função de avaliação:**

  $h(n)$ = número de ameaças.



$h = 5$ $\qquad$ $h = 2$ $\qquad$ $h = 0$

$\Rightarrow$ Resolve Prob. das n rainhas para qualquer n com alta probabilidade (em tempo quase constante!).