

# CSP procura, estrutura

# após propagação das restrições

- resultado:

- uma solução possível
- nenhuma solução possível

há casos em que a propagação resolve o problema sem procura!

- várias soluções possíveis

→ que tipo de procura?

# procura com retrocesso

baseia-se na propriedade de **comutatividade** do CSP

↔ a ordem pela qual se faz a atribuição é irrelevante  
reduz a complexidade de  $O(n!d^n)$  para  $O(d^n)$

consiste em

procura em profundidade com escolha do valor de uma variável em cada nível e  
retrocesso quando uma variável não tem valores possíveis

**function** BACKTRACKING-SEARCH(*csp*) **returns** a solution or *failure*  
**return** BACKTRACK(*csp*, { })

**function** BACKTRACK(*csp*, *assignment*) **returns** a solution or *failure*  
**if** *assignment* is complete **then return** *assignment*  
*var*  $\leftarrow$  SELECT-UNASSIGNED-VARIABLE(*csp*, *assignment*)  
**for each** *value* **in** ORDER-DOMAIN-VALUES(*csp*, *var*, *assignment*) **do**  
    **if** *value* is consistent with *assignment* **then**  
        add {*var* = *value*} to *assignment*  
        *inferences*  $\leftarrow$  INFERENCE(*csp*, *var*, *assignment*)  
        **if** *inferences*  $\neq$  *failure* **then**  
            add *inferences* to *csp*  
            *result*  $\leftarrow$  BACKTRACK(*csp*, *assignment*)  
            **if** *result*  $\neq$  *failure* **then return** *result*  
            remove *inferences* from *csp*  
        remove {*var* = *value*} from *assignment*  
**return** *failure*

não necessita  
- ações  
- modelo de transição  
- teste objetivo  
- estado inicial  
específicas do problema

# procura CSP com retrocesso

melhorias face a uma pesquisa não informada:

- que variável deve ser escolhida a seguir?

SELECT-UNASSIGNED-VARIABLE

- em que ordem devem ser testados os valores dessa variável?

ORDER-DOMAIN-VALUES

- que inferências se devem fazer em cada passo?

INFERENCE

- se uma atribuição viola uma restrição, pode-se evitar repetir?

# heurísticas de ordenação

sobre as  
variáveis

escolher a **variável mais restringida** | 1ª a falhar |

*minimum remaining values (MRV)*

em geral,  
mais poderosa

escolher a variável envolvida em mais restrições com variáveis ainda livres

**heurística de grau** (ideia de reduzir o fator de ramificação)

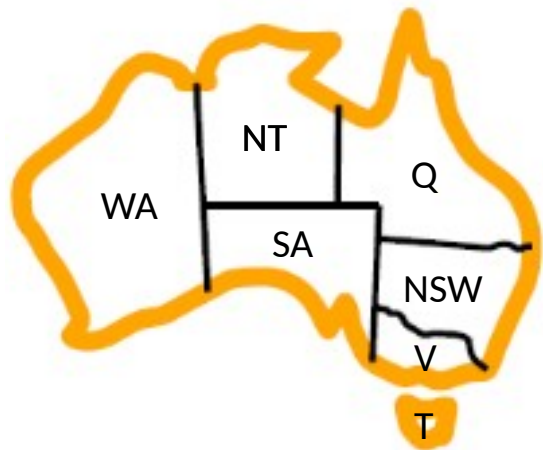
útil para  
desempatar

sobre os valores  
de uma variável

escolher o **valor menos restritivo**

valor que dá mais flexibilidade às variáveis seguintes

# heurísticas sobre as variáveis



empate na variável mais restringida

todas as cores possíveis em todas as variáveis

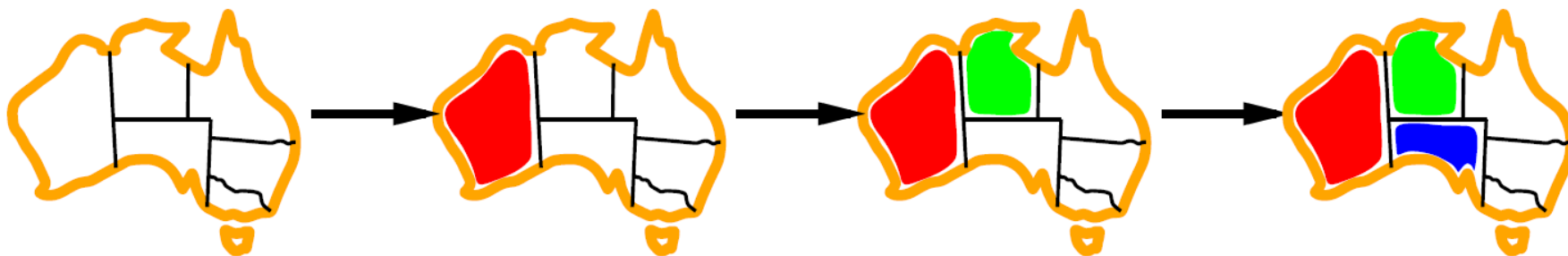
mas em grau...

SA tem grau 5

aplicando a heurística de grau resolve-se sem retrocesso!

# heurísticas sobre as variáveis

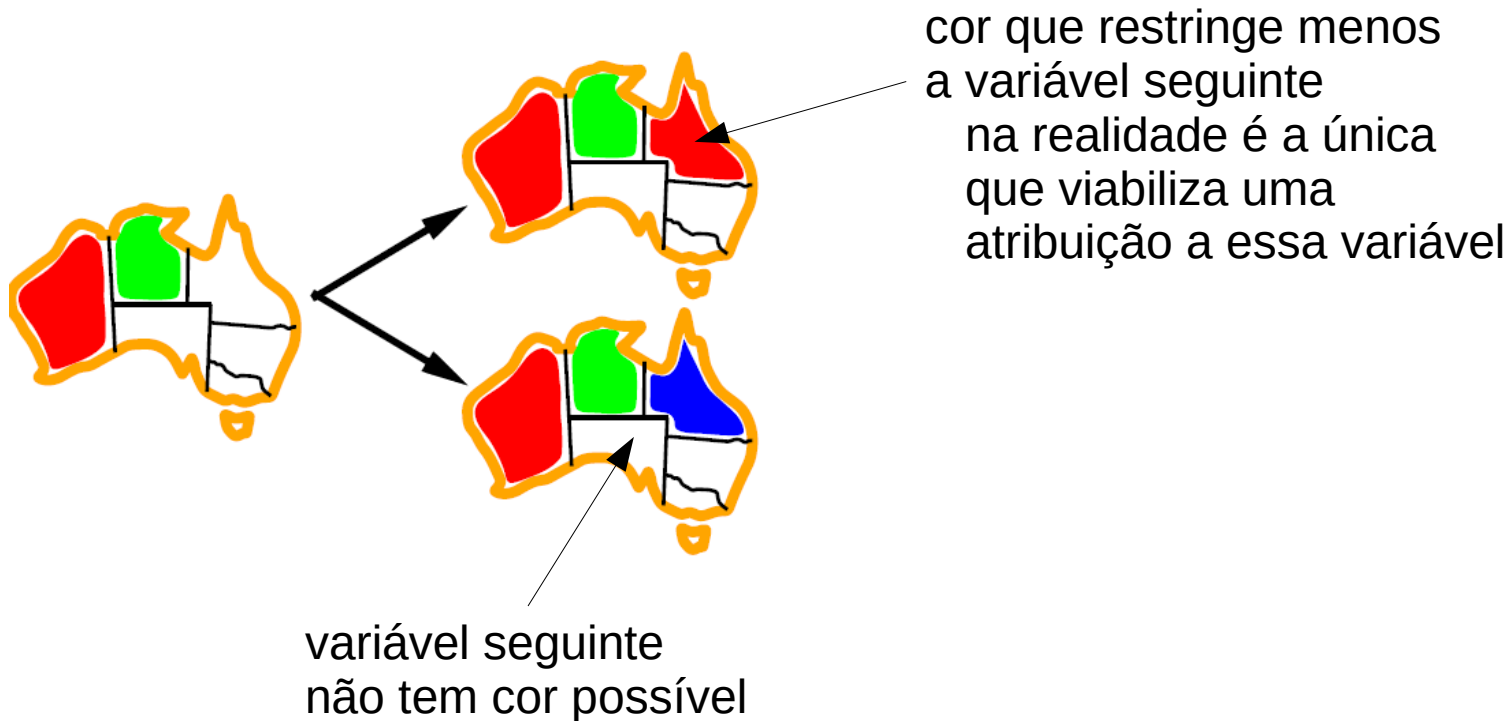
heurística de variável mais restringida



será que nesta ordenação garante evitar retrocesso?



# heurística sobre os valores



# intercalando procura e inferência

estratégia mais poderosa do aplicar só 1 vez infer+procura

**inferência adiante** (uma das formas mais simples)

de cada vez que se escolhe um valor para uma variável verifica-se consistência de arco

inútil se já foi usada consistência de arco como passo prévio

pode ser combinada com a heurística MRV

mais poderosa se for propagada a consistência de arco

mas, verificação de todo o grafo do problema (para todos os casos) é NP completa  
complexidade aumenta igualmente, com  $k$ , para verificações de  $k$ -consistência

# melhoramento iterativo

## métodos de procura local

- sobre estados completos – todas as variáveis com valores
- reparam iterativamente estados inconsistentes
- tipicamente começam de um atribuição aleatória

## enquanto houver conflitos

- seleciona uma variável aleatoriamente
- se tiver inconsistências, usa heurística para escolher o novo valor

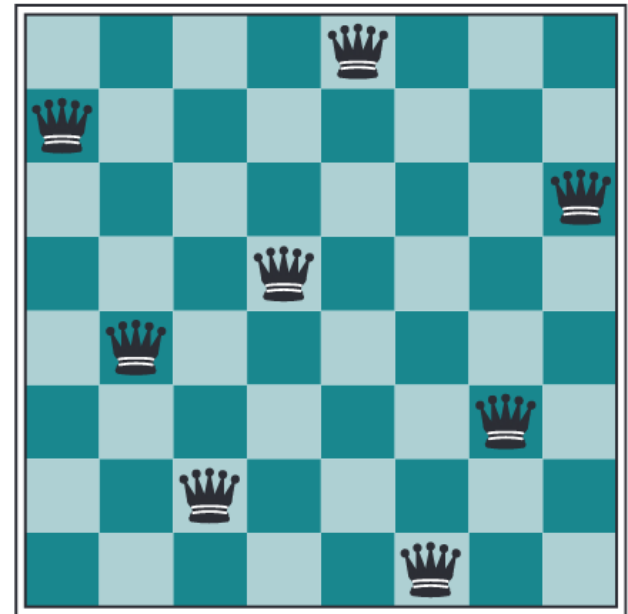
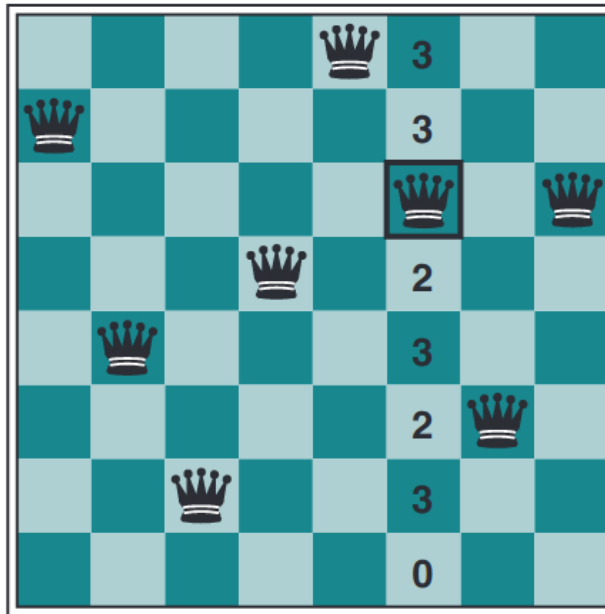
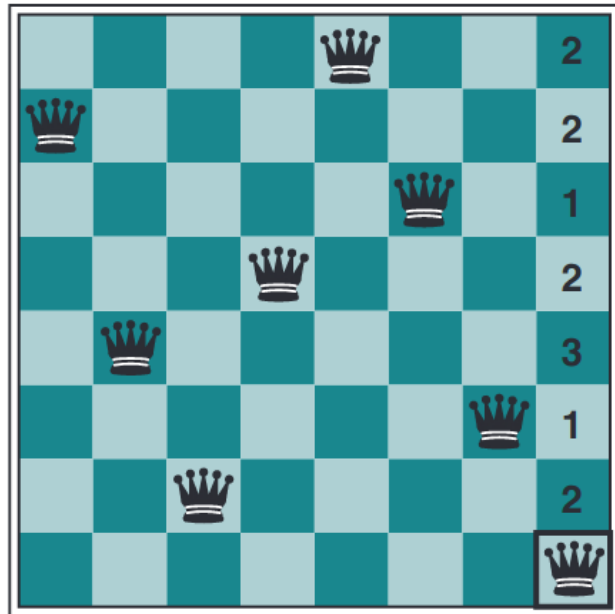
# min-conflitos

heurística para escolher o novo valor de uma variável  
numa procura local para resolver um CSP

consiste em escolher o **valor que minimiza o nº de conflitos com outras variáveis**

em caso de empate escolhe um dos menores aleatoriamente

# as rainhas



(neste caso) a escolha da outra casa com 1 conflito para a 1ª rainha também permitia resolver o problema com um movimento da seguinte

# live demo

Conflitos:				20
Q				
Q				
Q				
Q				
Q				

# eficiência do min-conflitos

## problemas das rainhas

a partir de estado inicial usando min-conflitos no posicionamento de uma rainha em cada linha

~  $10^6$  rainhas em ~ 50 movimentos  
(~independente da dimensão)

escalonamento de 1 semana de observações do Hubble

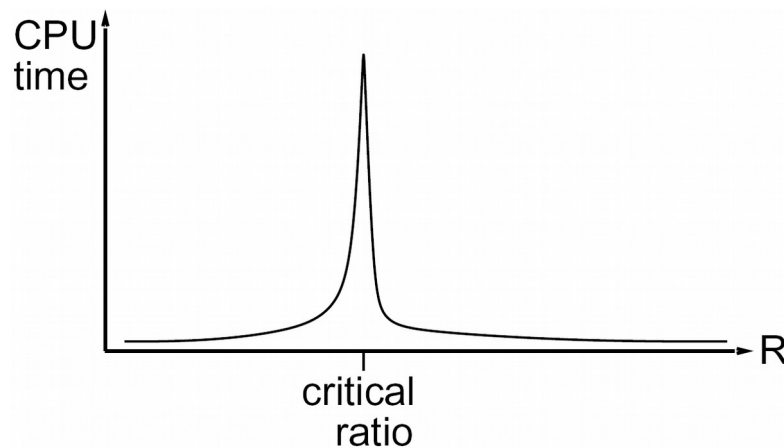
anteriormente em 3 semanas(!) atualmente em ~10 min

# eficiência do min-conflitos

parece poder generalizar-se a CSP complexos gerados aleatoriamente

*exceto* numa região estreita do rácio

$$R = \frac{\text{n}^\circ \text{ de restrições}}{\text{n}^\circ \text{ de variáveis}}$$





# estrutura de problemas

decomposição em problemas independentes reduz exponencialmente a complexidade

mas não são frequentes

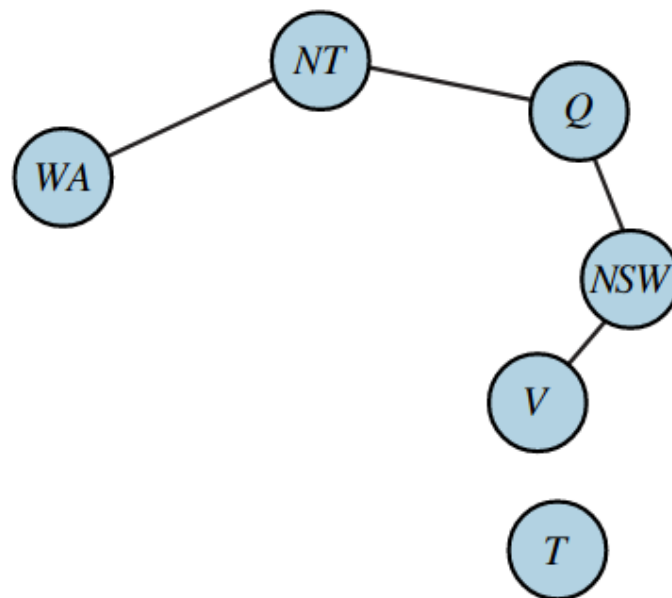
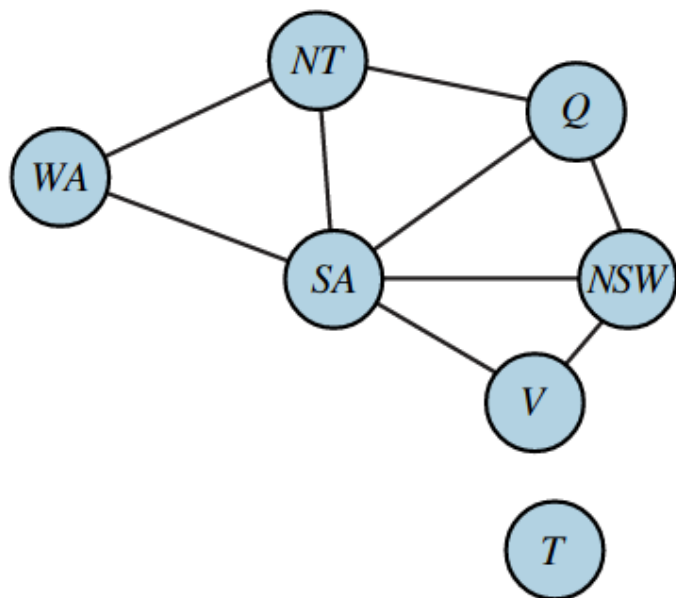
problemas com **estrutura em árvore** resolvidos em **tempo linear** com o  $n^o$  de variáveis

começa a resolver-se pela raiz e segue-se ordem topológica

**grafos** se reduzidos a árvores...

ex: atribuindo valores a algumas variáveis de modo que as restantes tenham uma estrutura em árvore

# transformação de grafo em árvore

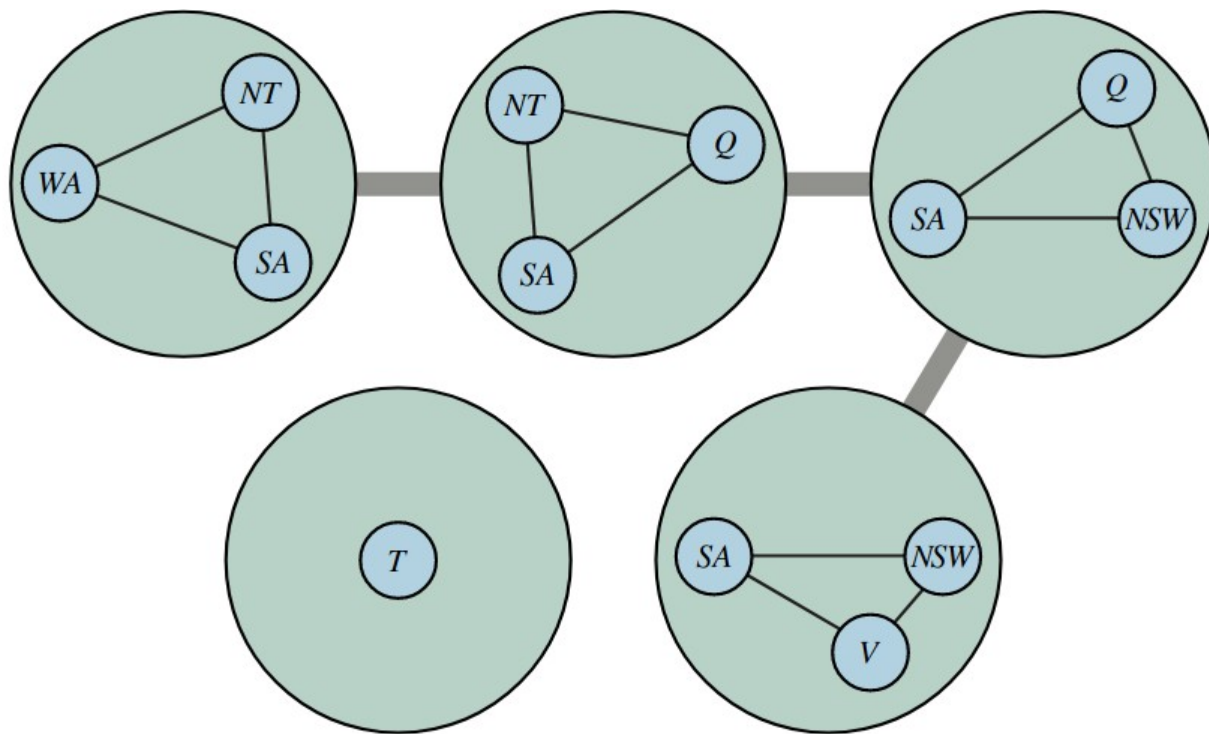
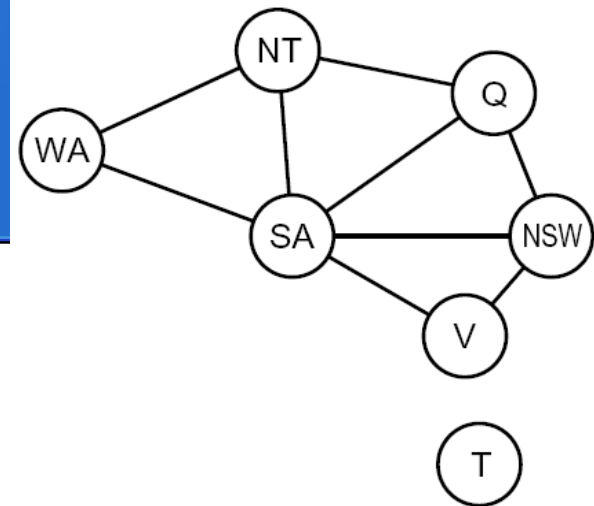


fixando um valor para *SA* e eliminando esse valor dos domínios das variáveis em restrições binárias com *SA*  
(no caso geral, caso esse valor esteja errado experimenta-se outro)

# *cycle cutset*

1. escolher um subconjunto  $S$  das variáveis do CSP de modo que o grafo de restrições se transforme numa árvore após a remoção de  $S$
2. para cada atribuição consistente de valores às variáveis de  $S$ 
  - a) remover do domínios das variáveis da árvore os valores inconsistentes com os escolhidos para  $S$
  - b) se o CSP em árvore tem solução retorna-a em conjunto com a atribuição de  $S$

# decomposição em árvore



resolve-se cada  
subproblema  
independentemente

e depois resolvem-se  
as restrições que ligam  
os subproblemas