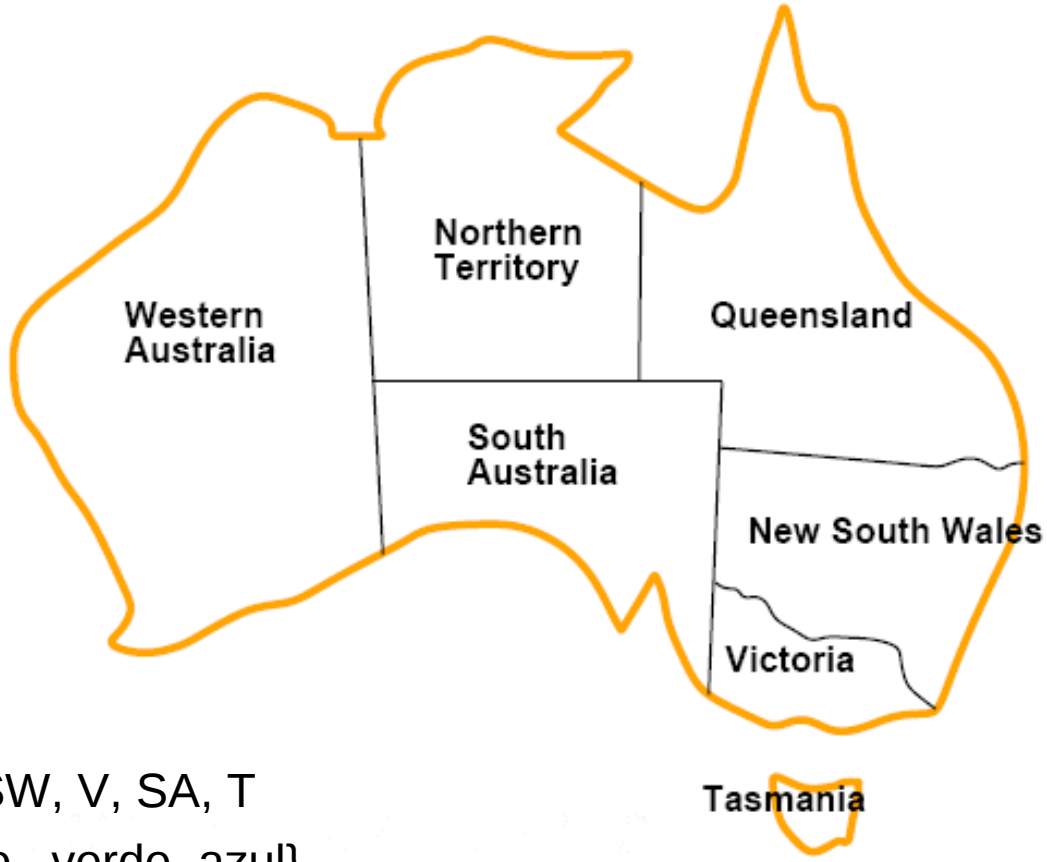
Problemas de satisfação de restrições (CSP)

- Problema de pesquisa habitual:
  - *Estado* é uma "caixa preta" – estrutura de dados que contém o teste de objectivo, função de avaliação e função sucessor

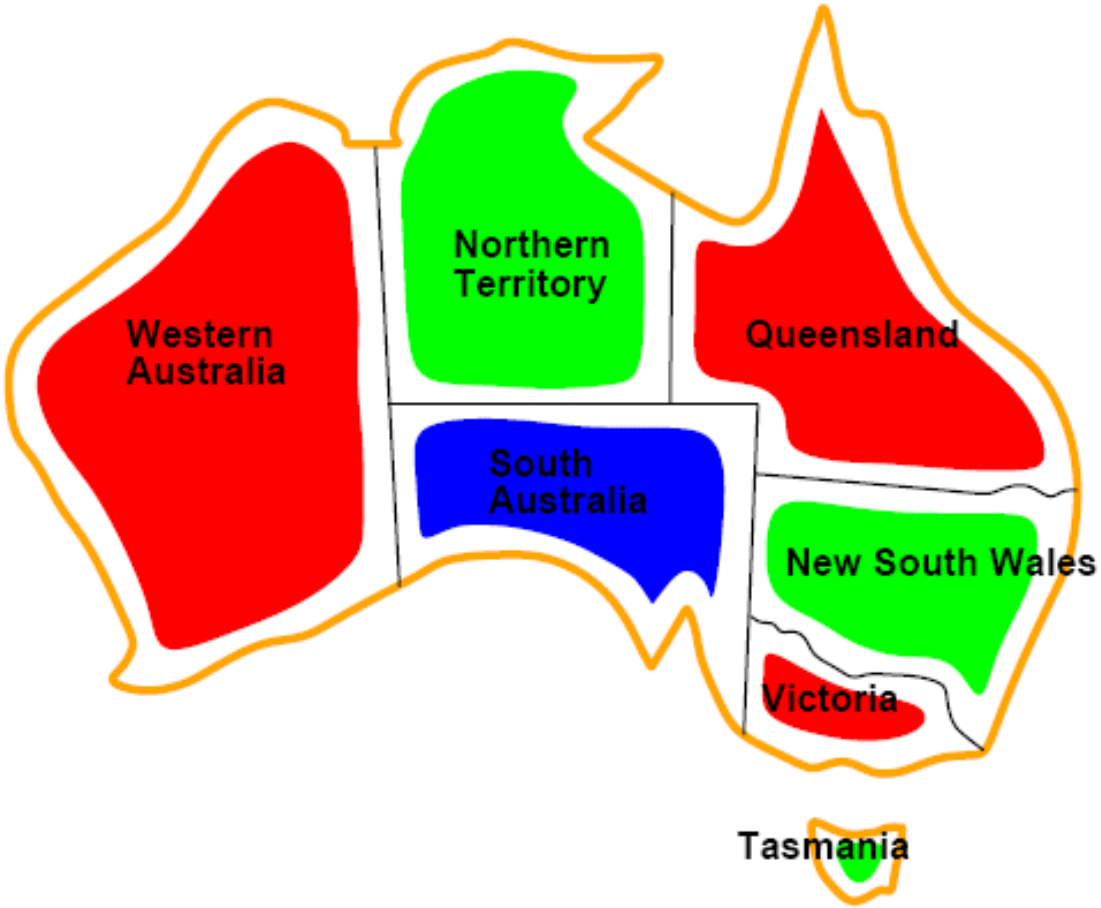
- CSP:
  - *Estado* é definido por variáveis *Xi* com valores do domínio *Di*
  - *Teste de objectivo* é um conjunto de restrições especificando combinações permitidas de valores para subconjuntos de variáveis

Problemas de satisfação de restrições (cont.)



- Variáveis WA, NT, Q, NSW, V, SA, T
- Domínios Di = { vermelho, verde, azul}
- Restrições : regiões adjacentes devem ter cores diferentes
  p.ex. WA ≠ NT

Problemas de satisfação de restrições (cont.)



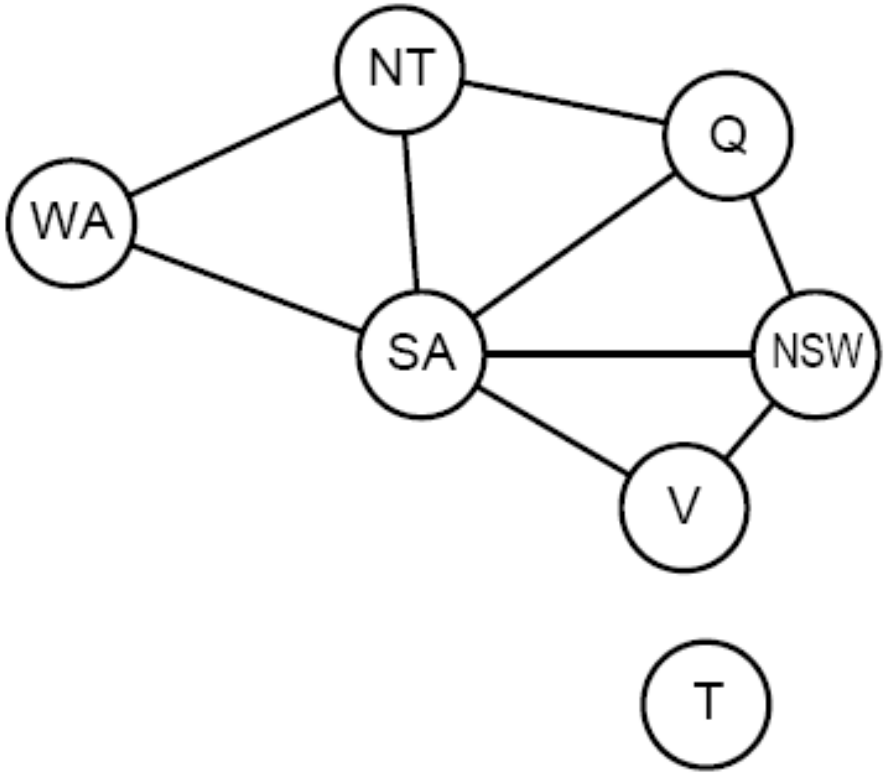
- Soluções são atribuições que satisfazem todas as restrições

Tipos de restrições

- Restrições unárias
  - envolvem uma só variável, p.ex., SA ≠ verde
- Restrições binárias
  - envolvem pares de variáveis, p.ex., SA ≠ WA
- Restrições de ordem mais elevada
  - envolvem mais três ou mais variáveis, p.ex., criptaritmética
- Preferências
  - p.ex., verde é melhor do que vermelho

Grafo de restrições

- CSP binário: cada restrição relaciona no máximo duas variáveis
- Grafo de restrições: nós representam variáveis, ramos restrições



- Algoritmos gerais usam esta estrutura, um grafo, para acelerar a pesquisa. Por exemplo, Tasmania é um problema independente.

Tipos de CSPs

- Variáveis discretas
  - Domínios finitos
    - dimensão do domínio d com n variáveis ==> O(d^n) atribuições completas
  - Domínios infinitos
    - inteiros, strings, etc.

      exemplo: escalonamento de tarefas

      é necessária uma linguagem de restrições, p.ex., Início1 + 5 ≤ Início2
- Variáveis contínuas

Criptaritmética

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

- Variáveis:
  M, S, O, E, N, R, Y, D
- Domínios:
  {0,1,2,3,4,5,6,7,8,9}
- Restrições
  todas diferentes
  D + E = Y + 10 * T1
  N + R + T1 = E + 10 * T2
  ...

Problemas concretos de CSPs

- Problemas de atribuição
  - Quem dá quais aulas?
- Problemas de horários
  - As aulas funcionam quando e onde?
- Configurações de hardware
- Escalonamento de transportes
- Gestão de produção
- Gestão de operações de fabrico (floorplanning)

Definição gradual de uma solução

- Estados são definidos pelos valores atribuídos até ao momento
- Estado inicial: { }
- Função sucessor: atribuir um valor a uma variável que ainda não tem valor ==> falha se não houver atribuições válidas
- Teste de objectivo: a atribuição actual é completa

- É a mesma solução para todos os problemas

- Cada solução com n variáveis tem profundidade n
  - permite usar pesquisa em profundidade

- O caminho é irrelevante

Definição gradual de uma solução (cont.)

Atribuições de variáveis são comutativas

- [WA = verde e NT = azul] é o mesmo que [WA = azul e NT = verde]

- Em cada nó só é necessário considerar atribuições a uma das variáveis

  ==> há d^n folhas (nós terminais da árvore)

  d é a dimensão do domínio

  n é o número de variáveis (profundidade do nó final)

- Pesquisa backtracking
  - A pesquisa em profundidade em CSPs com atribuições a uma variável em cada nó é chamada pesquisa backtracking
  - A pesquisa backtracking é o algoritmo não informado fundamental para CSPs

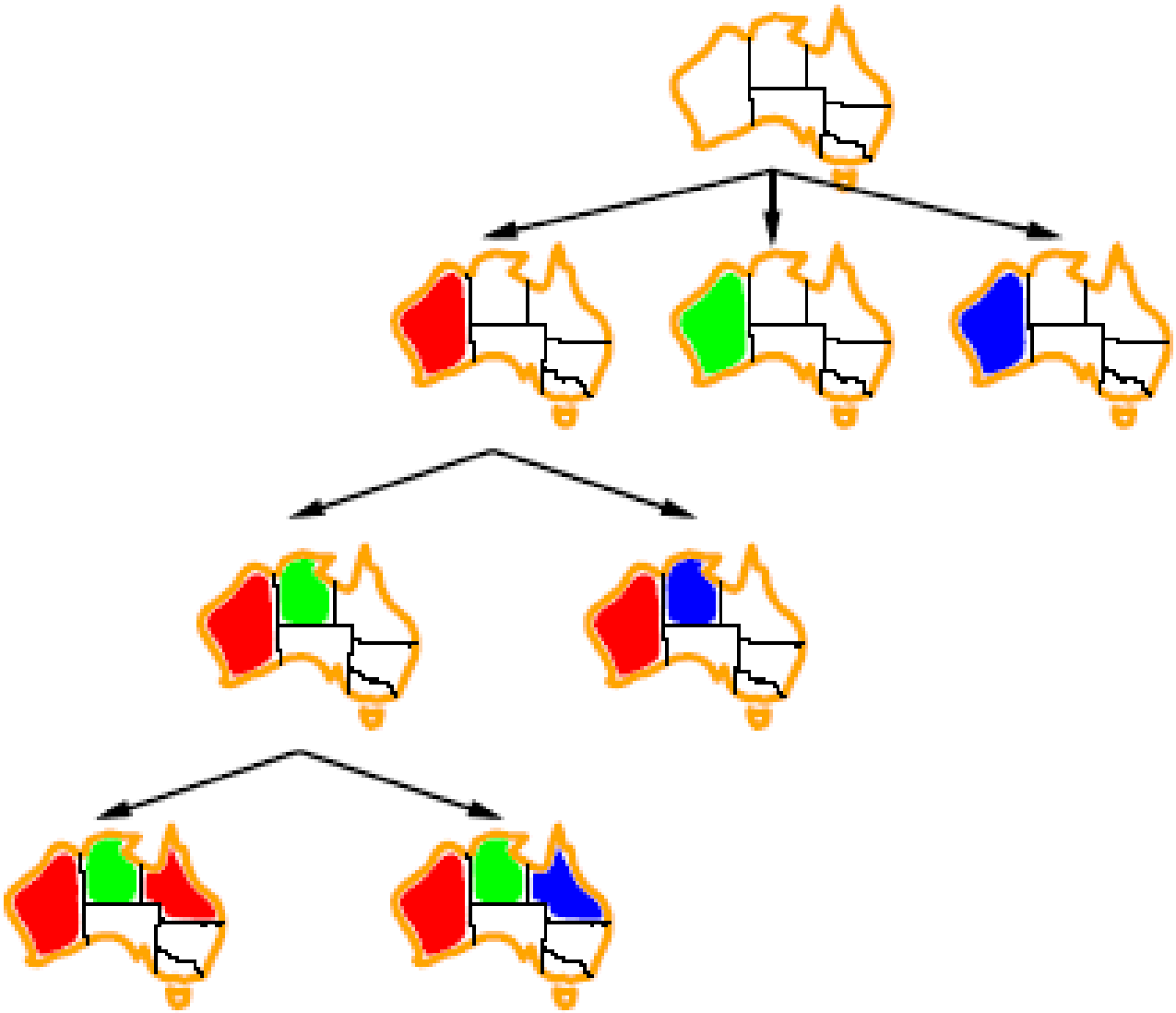
- Implementado com pesquisa em profundidade

Pesquisa backtracking

```
função Pesquisa-Backtracking( csp) retorna solução / falhanço
   retorna Backtracking-Recursivo( [ ], csp)

/* atrib: lista de atribuições a variáveis */
função Backtracking-Recursivo( atrib, csp) retorna solução / falhanço
  se atrib está completa então retorna atrib
  var ← SELECCIONA-VAR-NÃO-ATRIB( Variáveis[csp], atrib, csp)
  para cada valor em Ordena-Valores-Dominio( var, atrib, csp)
    se valor consistente com atrib de acordo com Restrições[csp] então
      resultado ← Backtracking-Recursivo( [var=valor|atrib],csp)
      se resultado ≠ falhanço então retorna resultado
  fim para cada
  retorna falhanço
```

Pesquisa backtracking (exemplo)

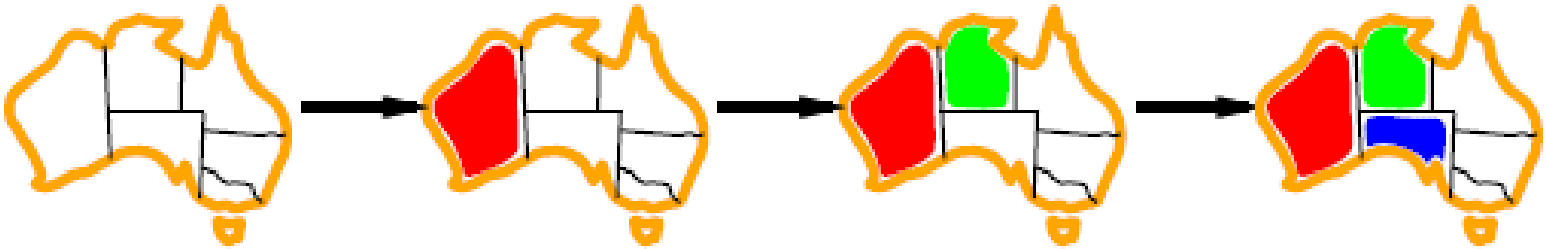


Pesquisa backtracking (cont.)

- Aumentando a eficiência do backtracking
  - Há técnicas gerais que podem aumentar a eficiência da pesquisa
    - Qual a próxima variável à qual atribuir um valor
    - Qual a ordem pela qual os valores devem ser atribuídos
    - Detecção precoce de um falhanço inevitável
    - Aproveitamento da estrutura do problema

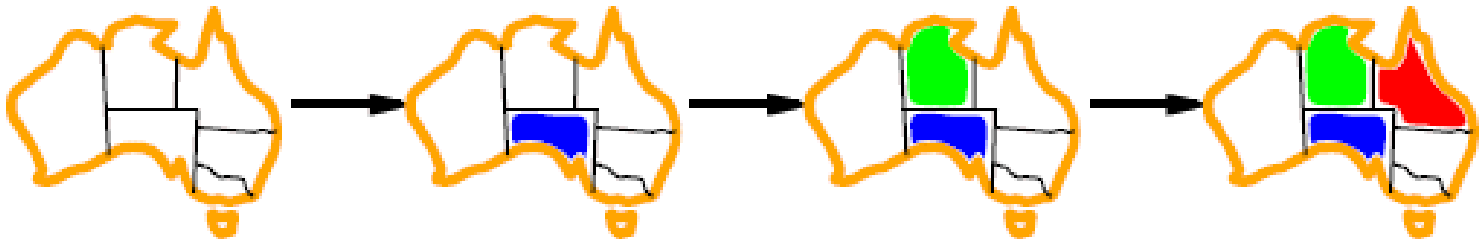
Pesquisa backtracking (cont.)

- Variável mais restringida
  - escolher primeiro a variável com menos valores válidos possíveis



Pesquisa backtracking (cont.)

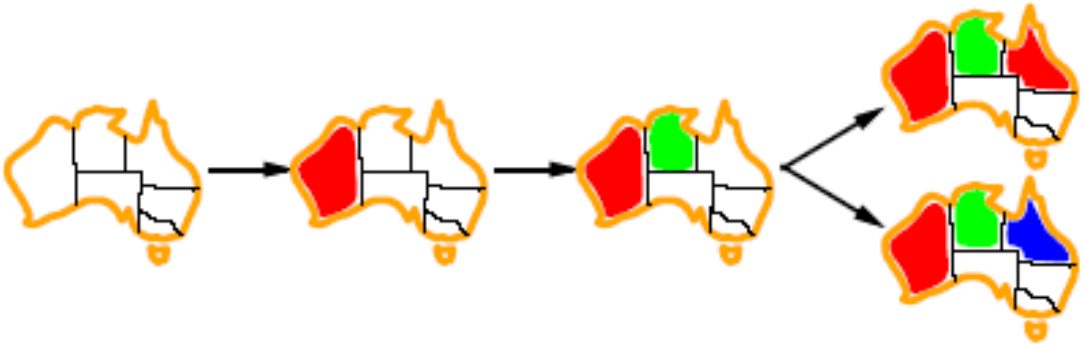
- Variável mais restritora
  - Para desempate entre variáveis mais restringidas
  - Escolher a variável com mais restrições com as variáveis restantes



Pesquisa backtracking (cont.)

- Valor menos restritor
  - Dada uma variável, escolher o valor menos restritor:

    aquele que elimina menos valores válidos possíveis para as variáveis restantes



permite 1 valor para SA

permite 0 valores para SA

Teste antecipado (forward checking)

- Ideia:
  - manter um registo dos valores válidos possíveis para as variáveis restantes
  - Interromper esse caminho da pesquisa quando uma ou mais das variáveis restantes não tem valores válidos possíveis