

Projeto e Análise de Algoritmos

Projeto de Algoritmos

Tentativa e Erro

Parte 2

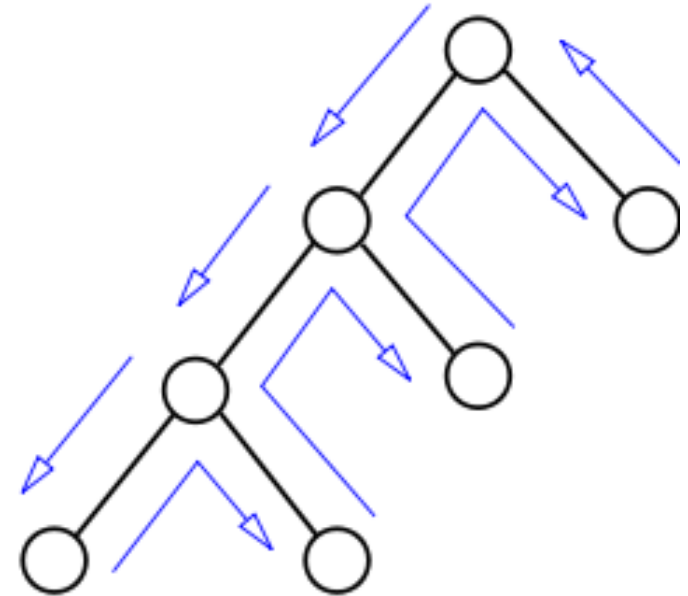
Prof. Humberto Brandão

humberto@bcc.unifal-mg.edu.br

versão da aula: 0.3

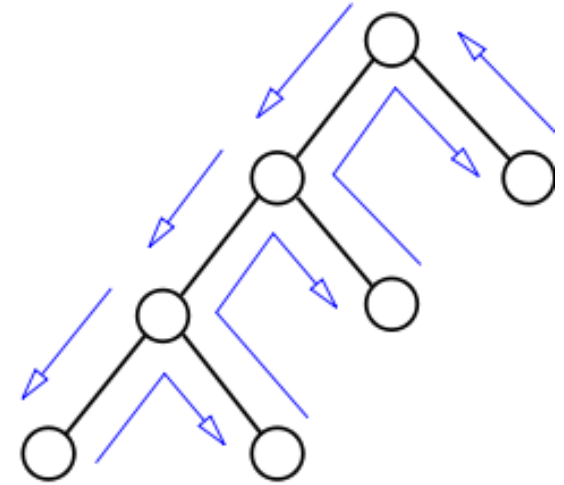
Última aula teórica

- Algoritmo de Tentativa e Erro:
 - *Backtracking* (Busca com retrocesso);
 - Refinamento da busca por força bruta;
 - Existem implementações de algoritmos gerais (*frameworks*) que trabalham com qualquer problema de satisfação de restrições (**PSR**)



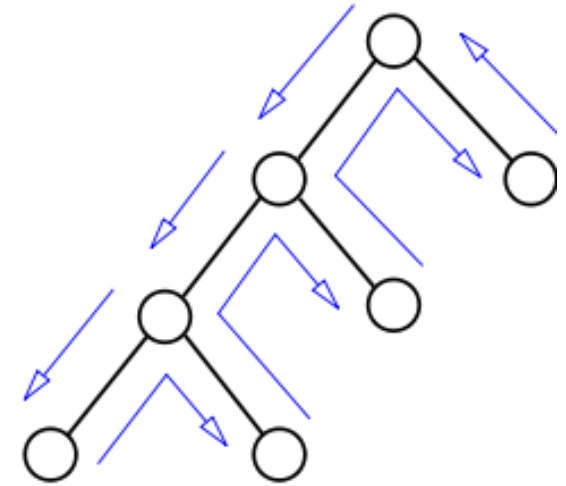
Última aula teórica

- Algoritmo de Tentativa e Erro:
 - PSR:
 - Não se pode explorar informações específicas do domínio;
 - Diferentemente do *backtracking* padrão;



Última aula teórica

- Algoritmo de Tentativa e Erro:
 - PSR:
 - Escolha para exploração:
 - Variável mais restringida;
 - » É uma heurística que tenta privilegiar **caminhos mais promissores**, escolhendo as variáveis com o menor número de valores legais.



Aula de hoje

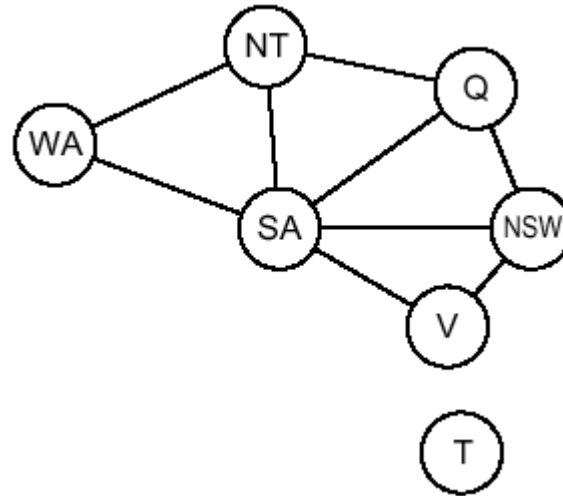
- Questionamentos:
 - É possível eliminar sub-árvores que já possuem ramificações sem nenhuma atribuição legal para alguma variável ainda não atribuída?
 - É possível prever falhas futuras?

Verificação Prévia (*Forward-Checking*)

Especialização do *Backtracking*

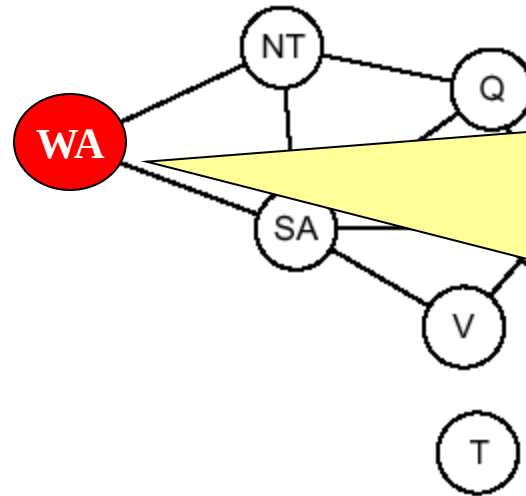
- *Forward-checking* (Verificação Prévia)
 - Em um PSR, uma maneira de **utilizar melhor as restrições durante a busca é a chamada verificação prévia** (*forward-checking*);
 - Sempre que uma variável X é atribuída, o processo de verificação prévia examina cada variável não atribuída Y que está conectada a X por uma restrição;
 - A partir daí, é excluído do domínio de Y , qualquer valor que esteja inconsistente com o valor escolhido para X ;

Especialização do *Backtracking* adição do *forward-checking*



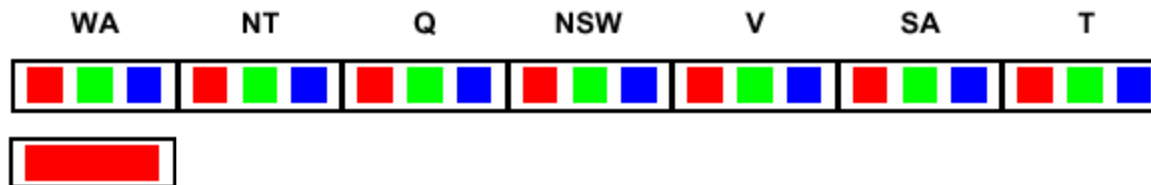
As variáveis não possuem nenhuma atribuição. Todas possuem no domínio Verde, Vermelho e Azul como possibilidades

Especialização do *Backtracking* adição do *forward-checking*

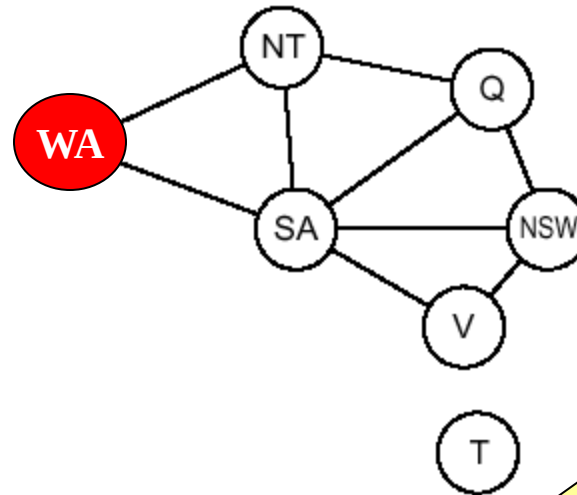


A heurística da variável menos restringida não possui escolha no início da busca, pois todas as variáveis possuem 3 opções de cores.

Vamos supor que WA recebe a cor vermelha a princípio no *backtracking*.



Especialização do *Backtracking* adição do *forward-checking*



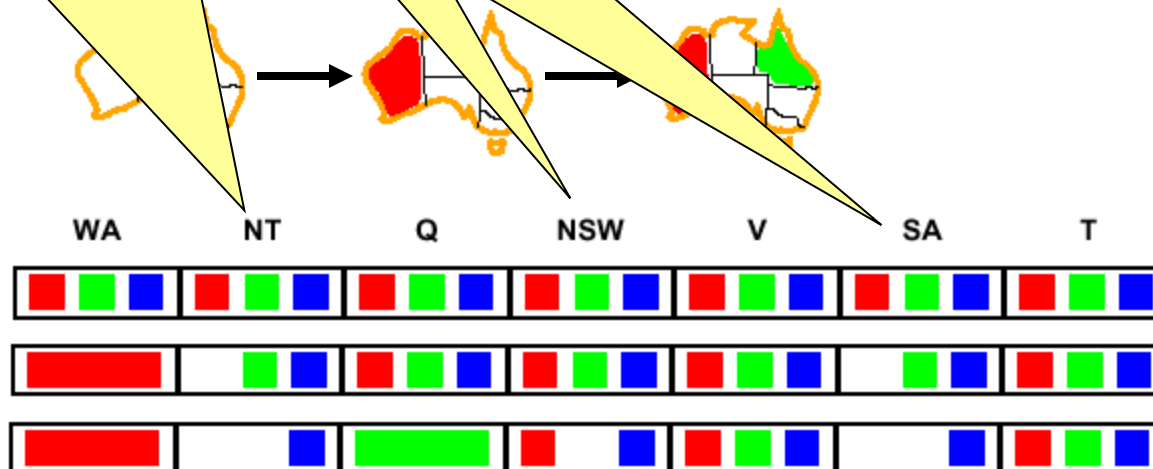
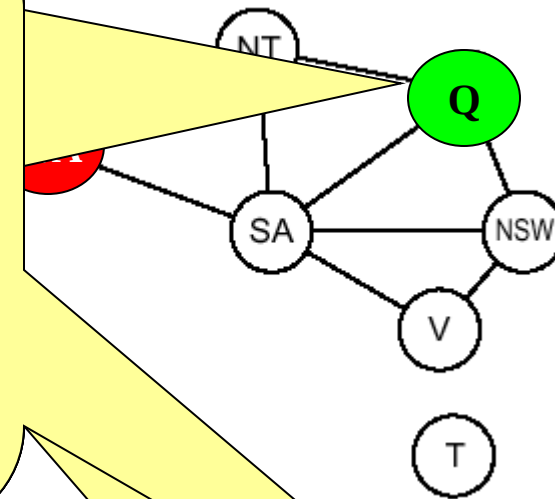
WA assumindo a cor vermelha, a política *Forward-checking* já anula possíveis conflitos com as variáveis vizinhas (lembrando que a vizinhança é definida pela restrição binária).



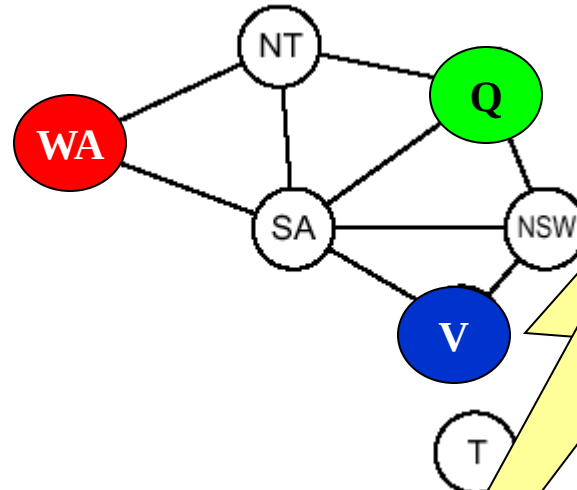
Especialização do Backtracking

Vamos assumir que a próxima variável a ser fixada no *backtracking* é o Estado Q.

Neste instante, o sistema já elimina do domínio das variáveis vizinhas a cor Verde: de NSW, SA e NT.

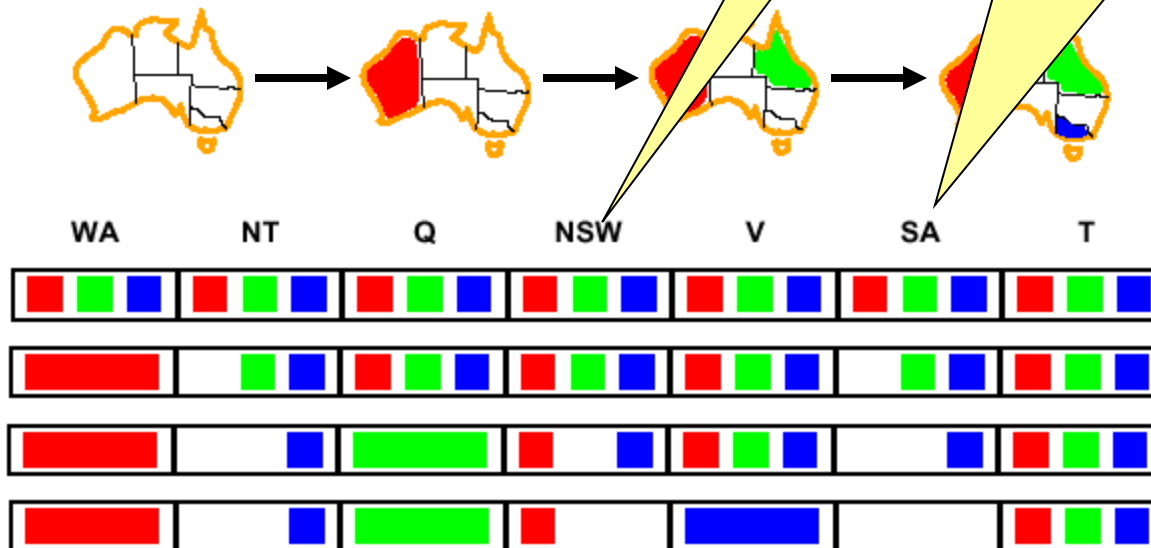


Especialização do *Backtracking* adição do *forward-checking*

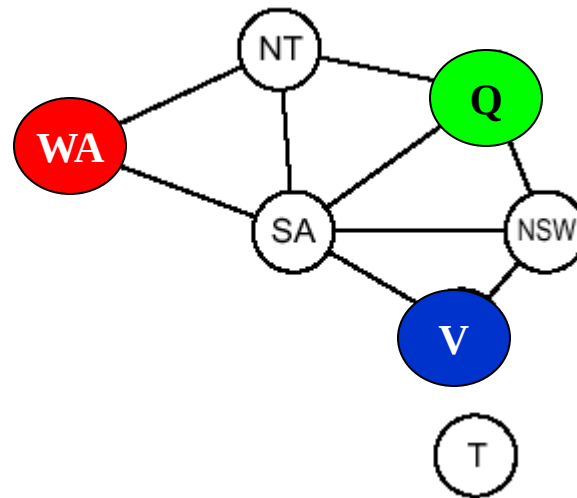


Vamos supor que na próxima escolha do *backtracking* seja fixada a variável V com a cor azul.

Neste ponto, já será removida a cor azul do domínio de NSW e de SA.

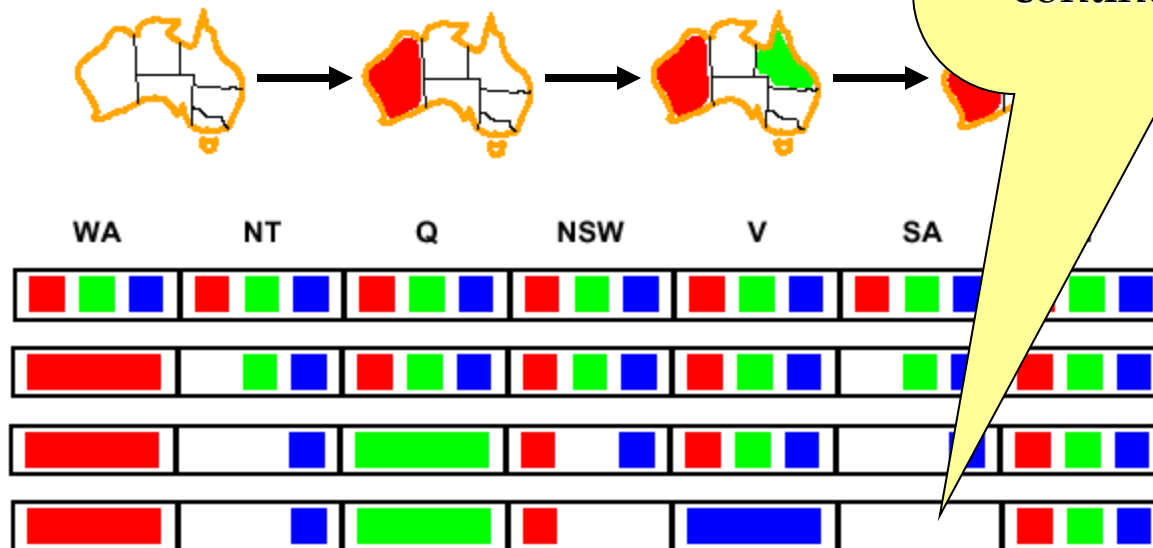


Especialização do *Backtracking* adição do *forward-checking*



A checagem para frente (*forward-checking*) também age neste instante, eliminando o ramo de busca, já que este é inviável por enumeração implícita.

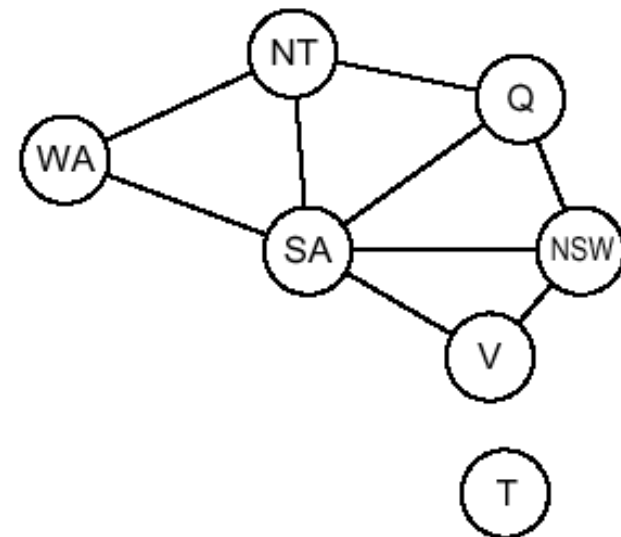
O *backtracking* pode desempilhar e continuar sua força bruta.



Propagação de Restrições (*Constraint propagation*)

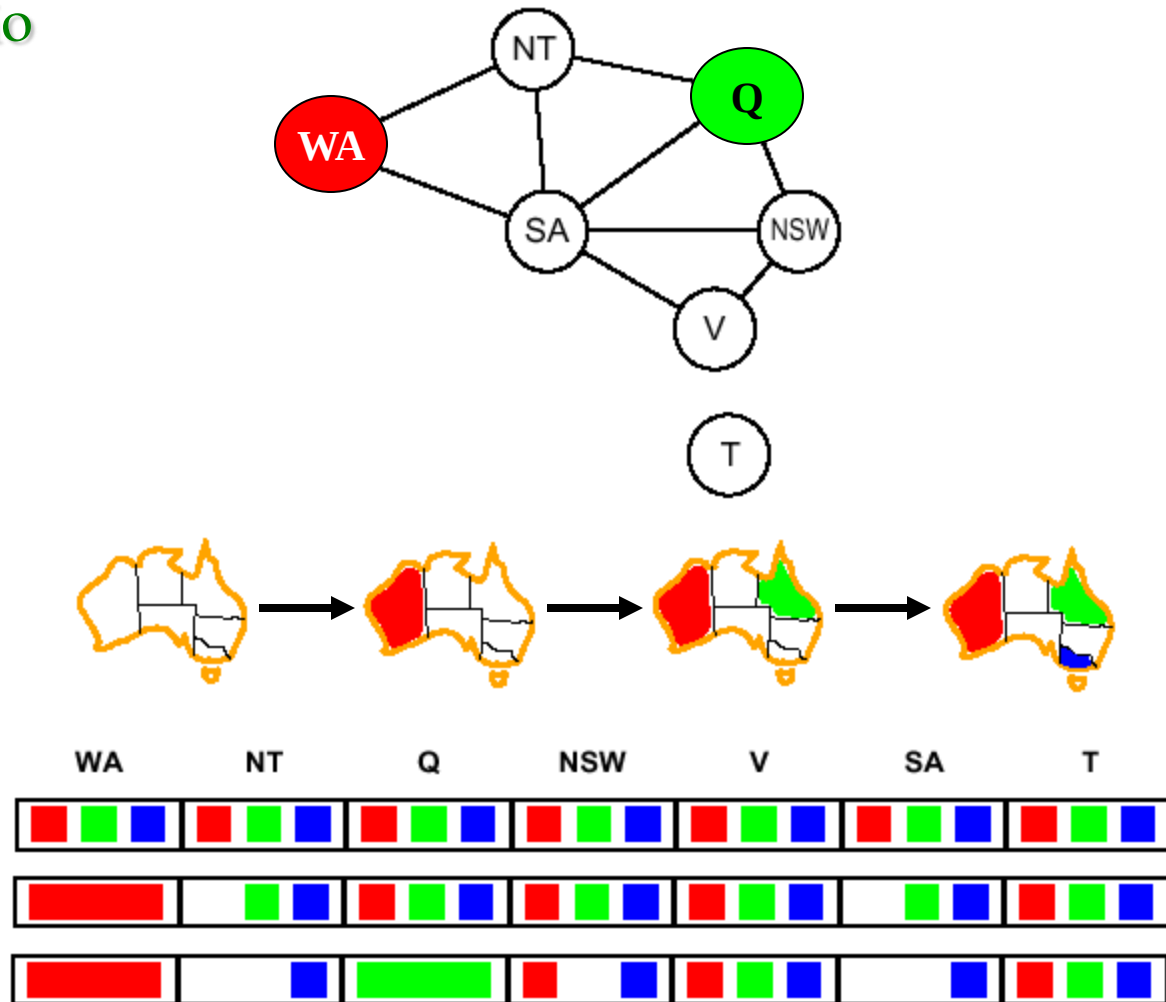
Propagação de restrições

- Embora detecte muitas inconsistências, o *forward-checking* não detecta todas elas;
- Por exemplo:
 - Quando WA for VERMELHO e Q for VERDE, tanto NT como SA serão forçados a serem AZUIS;
 - Mas NT e SA são adjacentes; Como ambos podem ser azuis?



Propagação de restrições

- Foi o que aconteceu na segunda atribuição do exemplo anterior;
- O ramo já estava condenado.
- Mas apenas com *forward-checking*, ele não poderia ser eliminado pois todas as variáveis ainda possuíam valores legais.



-
- ```

graph TD
 WA((WA)) --- NT((NT))
 WA --- SA((SA))
 NT --- SA
 SA --- Q((Q))
 SA --- NSW((NSW))
 SA --- V((V))
 Q --- NSW
 NSW --- V
 T((T))

```

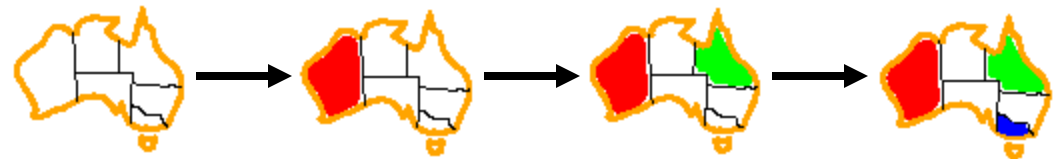
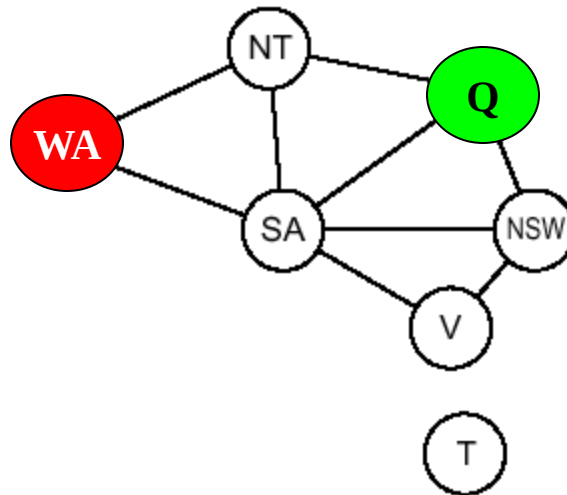


Figure 1 illustrates the experimental design with three rows of colored blocks representing different conditions (WA, NT, Q, NSW, V, SA, T) across three time points (1, 2, 3). The blocks are color-coded: red for 'No', green for 'Yes', and blue for 'No'.

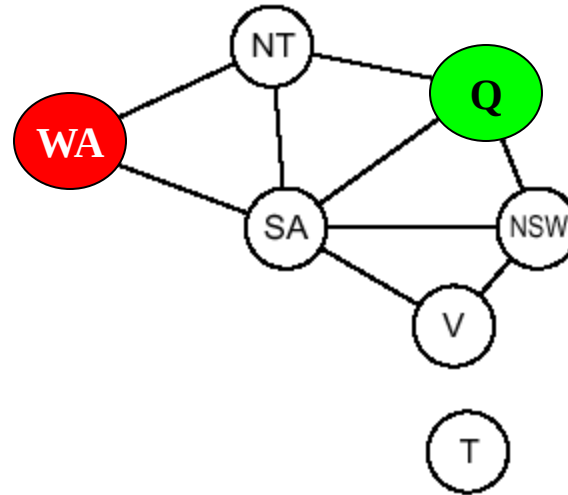
| Condition | Time 1           | Time 2           | Time 3           |
|-----------|------------------|------------------|------------------|
| WA        | Red, Green, Blue | Red, Green, Blue | Red, Green, Blue |
| NT        | Red, Green, Blue | Red, Green, Blue | Red, Green, Blue |
| Q         | Red, Green, Blue | Red, Green, Blue | Red, Green, Blue |
| NSW       | Red, Green, Blue | Red, Green, Blue | Red, Green, Blue |
| V         | Red, Green, Blue | Red, Green, Blue | Red, Green, Blue |
| SA        | Red, Green, Blue | Red, Green, Blue | Red, Green, Blue |
| T         | Red, Green, Blue | Red, Green, Blue | Red, Green, Blue |

# Propagação de restrições

- Precisamos propagar esta informação a partir de WA e Q sobre NT e SA;
  - Como foi feito no *forward-checking*;
- E depois sobre a restrição entre NT e SA para detectar a inconsistência;



# Propagação de restrições



- Mas devemos fazer de modo rápido;
- É inútil reduzir a quantidade de busca se gastamos mais tempo propagando restrições do que gastaríamos fazendo uma busca simples;
- Ai entra a pergunta: Até quando propagar?

# Propagação de restrições

- A idéia de consistência de arco fornece uma idéia substancialmente mais forte do que o *forward-checking*;
- Aqui o termo arco se refere a um arco orientado no grafo de restrições;
- Como o arco de SA para NSW

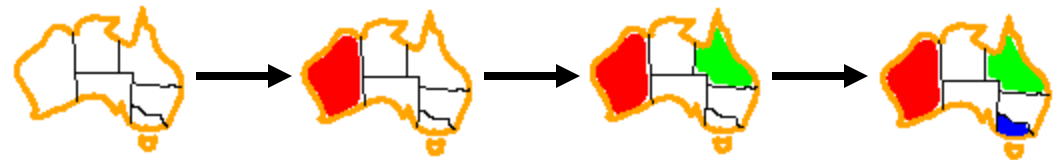
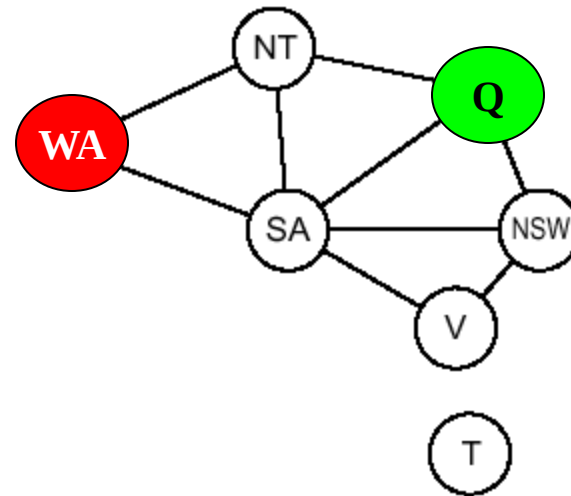
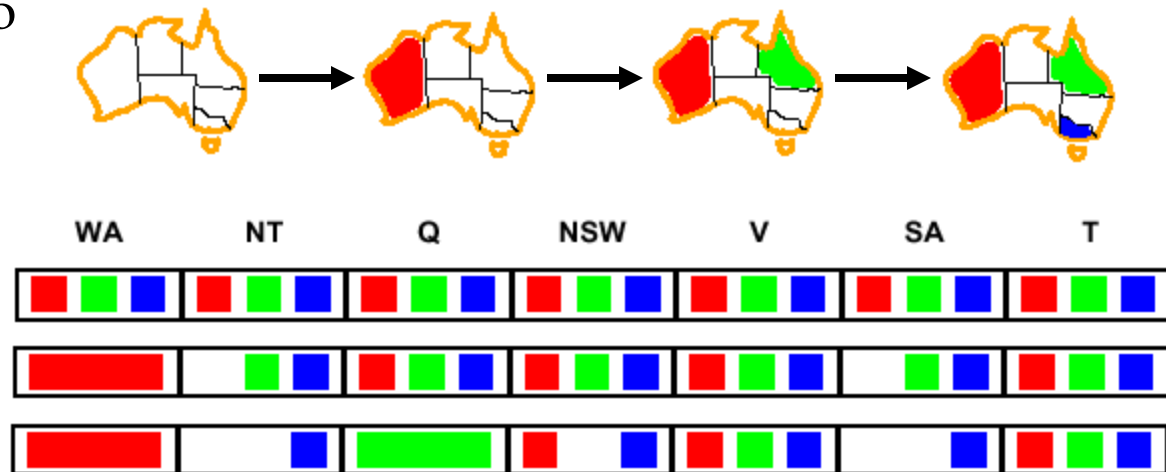


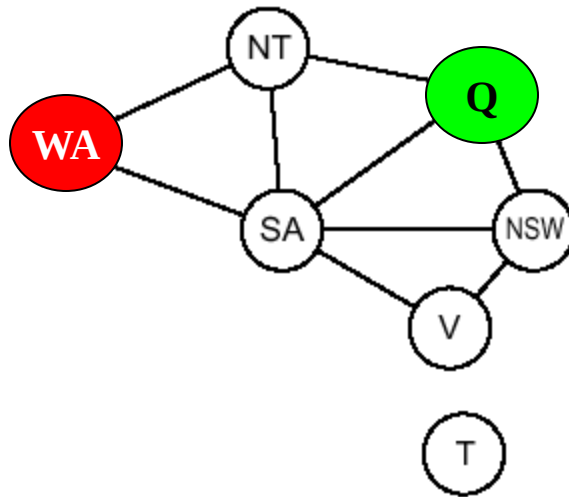
Figure 1: Schematic representation of the experimental design. The figure shows a 3x7 grid of colored squares representing trials. The columns are labeled WA, NT, Q, NSW, V, SA, and T. The rows represent different conditions. The first row shows a sequence of colored squares (red, green, blue) for each state. The second row shows a sequence of colored squares (red, green, blue) for each state. The third row shows a sequence of colored squares (red, green, blue) for each state.

- 
- ```

graph TD
    WA((WA)) --- NT((NT))
    WA --- SA((SA))
    NT --- SA
    SA --- Q((Q))
    SA --- NSW((NSW))
    SA --- V((V))
    Q --- NSW
    NSW --- V
    T((T))
  
```



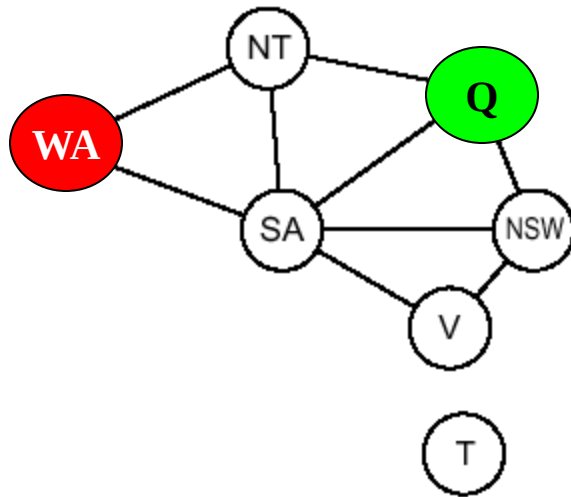
Propagação de restrições



WA	NT	Q	NSW	V	SA	T

- Analisando o arco em sentido contrário: (NSW,SA):
- O arco é inconsistente;
- Quando NSW=AZUL, não existe atribuição válida para SA;
- Para tornar o arco consistente, basta eliminar a cor AZUL do domínio de NSW;

Propagação de restrições



WA	NT	Q	NSW	V	SA	T

- Em um processo semelhante, podemos propagar a restrição do arco (NT,SA);
- Domínio(NT) = {AZUL}
- Domínio(SA) = {AZUL}
- AZUL deve ser eliminado do domínio de NT, o que deixa o domínio vazio;
- Desde modo, a aplicação da propagação de restrições via consistência de arcos detectou uma inconsistência que não é detectada pelo *forward-checking*.

Propagação de restrições

- A verificação de consistência de arco pode ser aplicada como uma etapa de propagação;
- O processo deve ser aplicado repetidamente até não restar mais nenhuma inconsistência;
- Isso porque, sempre que um valor é excluído do domínio de alguma variável; para remover uma inconsistência de arco, pode surgir uma nova inconsistência de arco em arcos que apontam para esta variável;

Propagação de restrições

- Embora o processo seja bastante dispendioso (se comparado ao *forward-checking*), **em geral o custo extra é compensador**;
- O que vai determinar se é ou não é interessante efetuar a propagação de restrições via consistência de arcos?

Leitura para próxima aula

- RUSSEL, S.; NORVIG, P. (2004). Inteligência Artificial. Tradução da Segunda Edição. Editora Campus.
 - Capítulo 5
 - <http://aima.cs.berkeley.edu/newchap05.pdf>
- Programação Dinâmica
 - Livro do Nivio
 - Livro do Cormen

Bibliografia

- RUSSEL, S.; NORVIG, P. (2004). Inteligência Artificial. Tradução da Segunda Edição. Editora Campus.
 - Capítulo 5 é gratuitamente distribuído na web (em inglês):
 - <http://aima.cs.berkeley.edu/newchap05.pdf>
- Aula sobre PSR do aluno de doutorado Paulemir Campos do CIn-UFPE.
 - <http://www.cin.ufpe.br/~in1006/2005/>

