UNIFAL-MG
Universidade Federal de Alfenas

Projeto e Análise de Algoritmos

Projeto de Algoritmos

Tentativa e Erro

Prof. Humberto Brandão
humberto@bcc.unifal-mg.edu.br

Laboratório de Pesquisa e Desenvolvimento
Universidade Federal de Alfenas
versão da aula: 0.3

PAA

UNIFAL-MG
Universidade Federal de Alfenas

- Na última aula....
  - Visão geral de projetos e algoritmos...

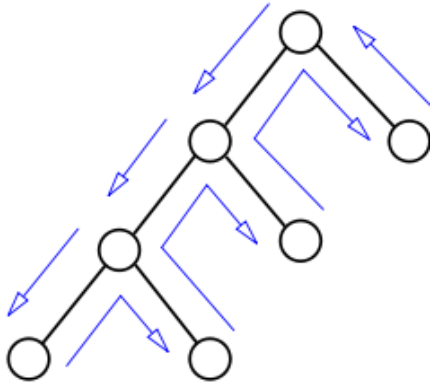
Algoritmos de Tentativa e Erro

UNIFAL-MG
Universidade Federal de Alfenas

- A recursividade pode ser utilizada para resolver problemas cuja solução é tentar todas as alternativas possíveis;

- A idéia para os algoritmos de tentativa e erro é decompor o processo em um número finito de subtarefas parciais que devem ser exploradas exaustivamente.

- O processo geral pode ser visto como um processo de pesquisa ou de tentativa que gradualmente constrói e percorre uma árvore de subtarefas;



Algoritmos de Tentativa e Erro

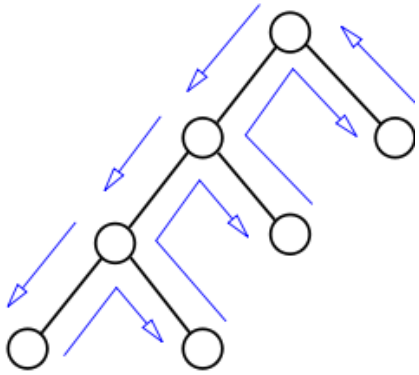
UNIFAL-MG
Universidade Federal de Alfenas

- Os algoritmos de tentativa e erro não seguem uma regra fixa para computação;

- Funcionam da seguinte maneira:

  – São efetuados **passos** em direção à solução final;

  – Caso esses passos tomados não levem à solução final, eles podem ser retirados e apagados do registro. (falha)



Algoritmos de Tentativa e Erro

UNIFAL-MG
Universidade Federal de Alfenas

- A natureza do problema é que define se uma ramificação da árvore não nos leva a nenhuma solução:
  - Em alguns problemas, o **limite primal** pode ser utilizado para efetuar tais podas;
    - Caixeiro Viajante;
  - Em outros, a violação de restrições pode inutilizar toda uma ramificação;
    - Coloração de mapas
  - Em outros, a violação de restrições em ramos interiores da árvore não necessariamente condena todos os nós filhos da subárvores.
    - *Pickup and Delivery Problem*.

Algoritmos de Tentativa e Erro

UNIFAL-MG
Universidade Federal de Alfenas

- Muitas vezes, a pesquisa na árvore de soluções cresce rapidamente, mas em uma grandeza polinomial;
- Em outras, este crescimento é exponencial;
  - Nestes casos, é recomendado que a pesquisa utilize algoritmos aproximados ou heurísticas para resolver problemas de médio e grande porte;
- Formulações incorretas podem levar ao crescimento exponencial da árvore;
  - Problema é simples, mas o algoritmo é complexo!
- Ou pode estar relacionada de fato com a natureza do problema.

UNIFAL-MG
Universidade Federal de Alfenas

Exemplo de Tentativa e Erro

Problemas de Satisfação de Restrições (PSR)

Algoritmos de Tentativa e Erro

UNIFAL-MG
Universidade Federal de Alfenas

- Pesquisadores em IA desenvolveram uma *framework* para resolução de problemas quaisquer que tenham restrições;

- A classe se chama:
  - Problemas de Satisfação de Restrições (PSR)

- Um PSR é definido por:
  - um conjunto de variáveis de decisão $\{x_1, x_2, x_3, ..., x_n\}$
  - um conjunto de restrições $\{c_1, c_2, c_3, ..., c_m\}$
  - um domínio de valores possíveis D_i, para cada variável, com $i = 1, ..., n$;
    - Este conjunto é não vazio;
    - Ou seja, cada variável pode assumir pelo menos um valor;

Algoritmos de Tentativa e Erro

UNIFAL-MG
Universidade Federal de Alfenas

- Um estado do problema é definido por uma atribuição de valores a no mínimo uma variável de decisão;

- Exemplos de estados:
  - `x1 = 3, x2 = ?, x3 = 6, x4 = 10;`
  - `x1 = ?, x2 = ?, x3 = ?, x4 = 15;`
  - `x1 = 3, x2 = 58, x3 = 6, x4 = 10;`

- Uma atribuição que não viola nenhuma restrição é chamada atribuição consistente;

- Uma solução ao PSR é quando todas as variáveis possuem valores dentro de seus domínios e nenhuma restrição é violada.

Algoritmos de Tentativa e Erro

UNIFAL-MG
Universidade Federal de Alfenas

- Alguns PSR exigem que a solução maximize ou minimize uma função objetivo;
  - Estes são nomeados de problemas de otimização;
- Em outros, é necessário apenas que encontremos uma atribuição completa que não viole restrições;
  - Geralmente problemas mais restritos;
  - Tais problemas também podem ser modelados como problemas de otimização;
    - A técnica mais utilizada é a de penalização da *f.o.* quando restrições são violadas.

Exemplo

Coloração de Mapas

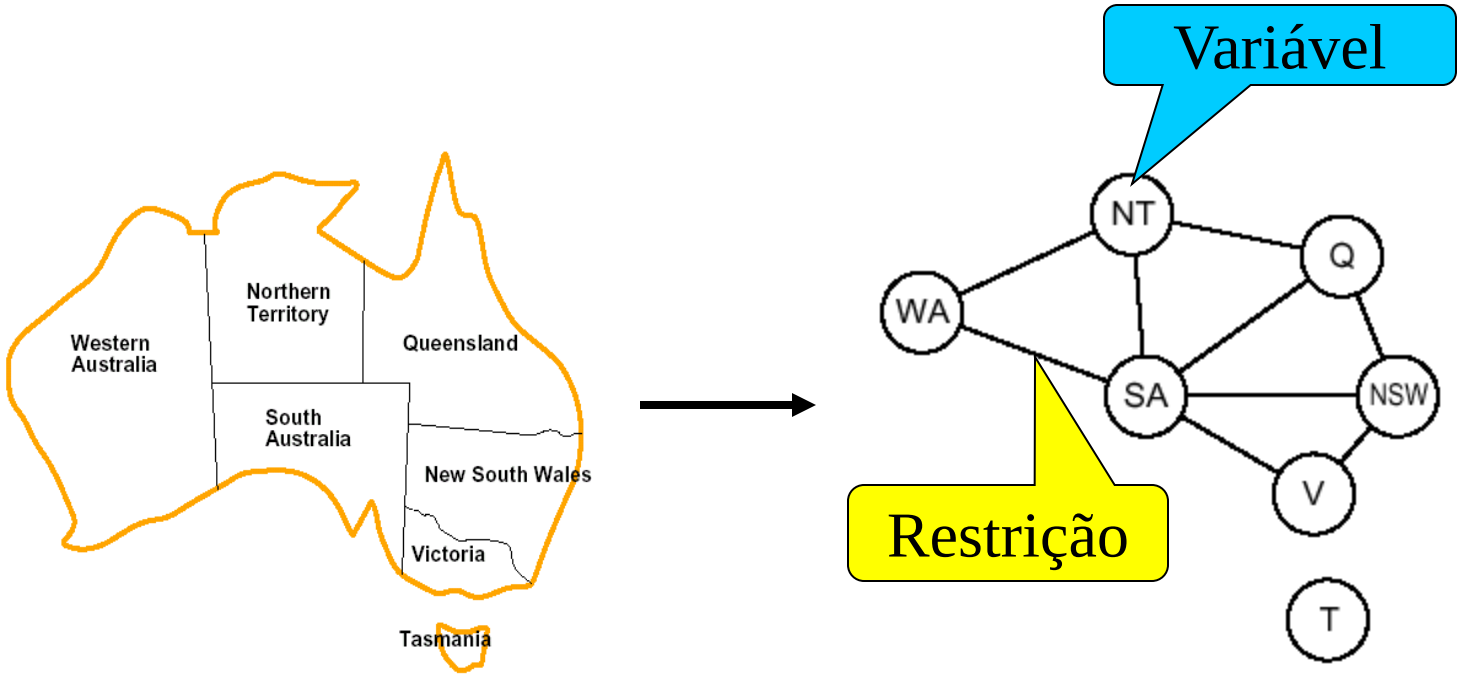
Problema de Colorir um Mapa

Variáveis: WA, NT, Q, NSW, V, SA, T

Domínio: D_i = { vermelho, verde, azul } $\forall$ I = 1, …, 7

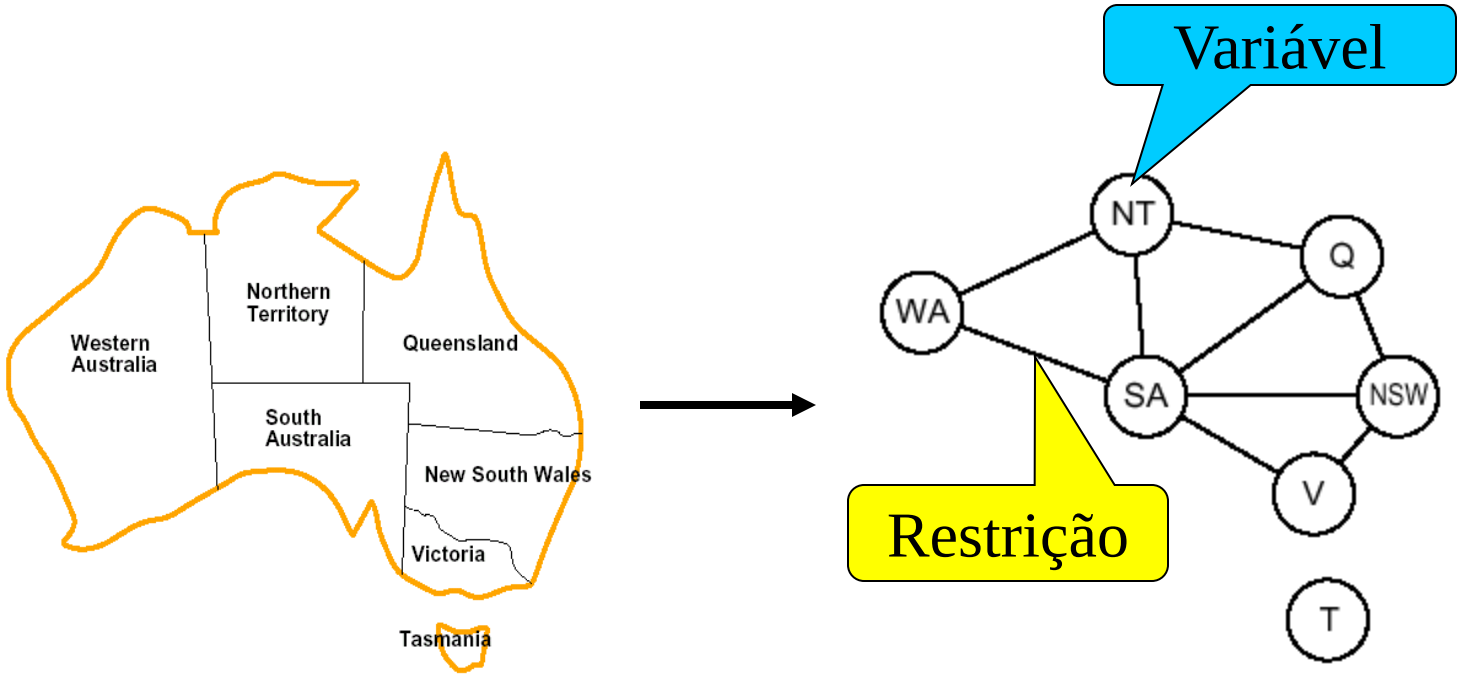
Restrições: Regiões vizinhas devem ter cores diferentes

Grafo de Restrições de um PSR



- Obs.: Tasmânia é um subproblema independente;
- Como identificar que existem problemas independentes?

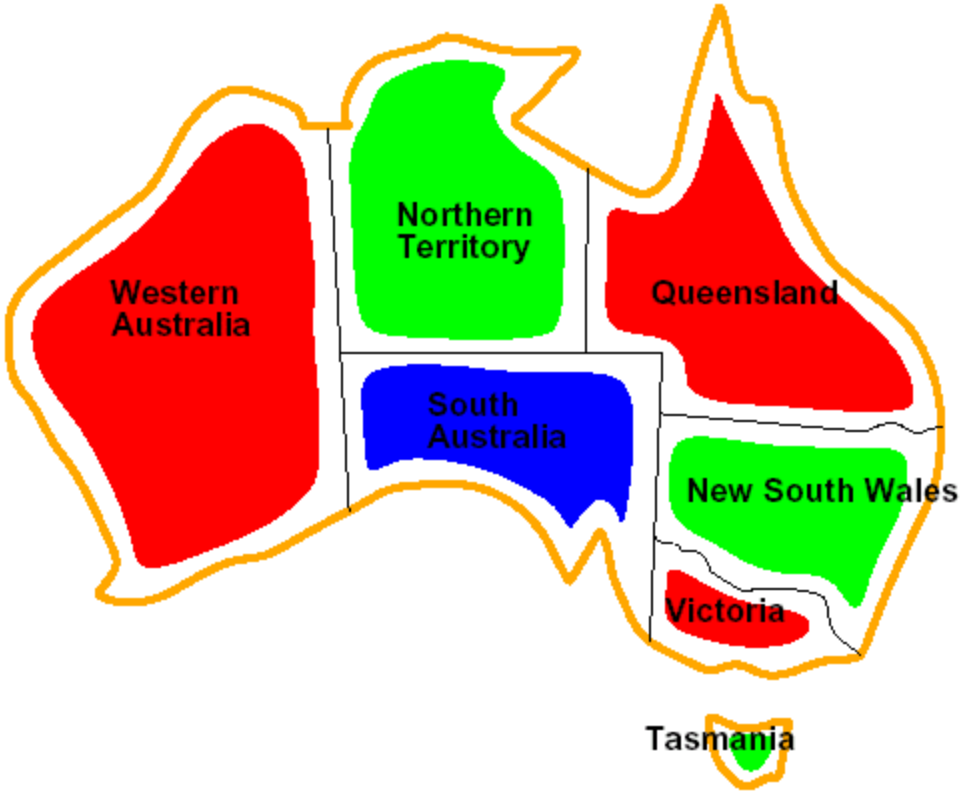
Grafo de Restrições de um PSR



- Obs.: Tasmânia é um subproblema independente;

- Como identificar que existem problemas independentes?
  - Componentes fortemente conectados...
    Vocês lembram o algoritmo???

Problema de Colorir um Mapa

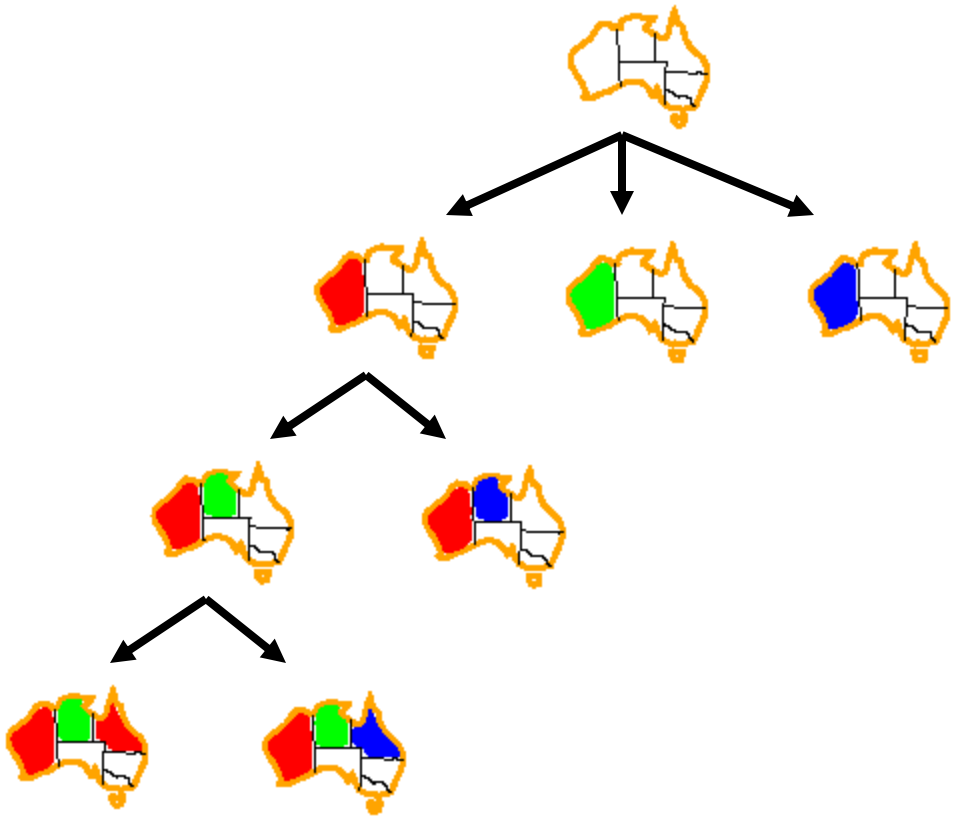


Soluções: São atribuições que satisfazem todas as restrições

Ex.: { WA=vermelho, NT=verde, Q=vermelho, NSW=verde,
Victoria=vermelho, SA=azul, T=verde }

Exemplo de Árvore de Busca do problema de coloração de mapas

UNIFAL-MG
Universidade Federal de Alfenas



Variedade de PSRs

UNIFAL-MG
Universidade Federal de Alfenas

- Variáveis Discretas
  - Domínios Finitos:
    - Ex.: PSRs Booleanos
      - inclusive Satisfabilidade Booleana (SAT)
        - » Problema NP-Completo
  - Domínios Infinitos:
    - Ex.: Que envolvem Inteiros, strings, etc.

Variedade de PSRs

UNIFAL-MG
Universidade Federal de Alfenas

- Variáveis Contínuas
  - Ex.: Que envolvem reais;
  - Exemplo de problema:
    - Resolve restrições lineares em tempo polinomial por métodos de Programação Linear (simplex, pontos interiores)

Tipos de Restrições

UNIFAL-MG
Universidade Federal de Alfenas

- **Unária**
  - Ex.: MG ≠ verde
- **Binária**
  - Ex.: MG ≠ SP
- **Ordem Maior** (Três ou mais variáveis)
- **Preferenciais** (Problemas de Otimização)
  - Ex.: Vermelho é melhor do que verde, isto é, há uma função de custo nas atribuições (função objetivo).

Exemplos de PSRs do Mundo Real

UNIFAL-MG
Universidade Federal de Alfenas

- Problemas de Alocação
  - Ex.: Qual será a sala de PAA no semestre que vem?
- Problemas de Oferta de Disciplinas
  - Ex.: Que matéria será oferecida?
- Configurações de Hardware
  - *Layout*;
- Agendamento para entregas
  - Um motoboy vai fazer entregas pela cidade:
    - Qual é a ordem para a entrega que minimiza a distância total percorrida?

Relacionado ao *Framework* do PSR

UNIFAL-MG
Universidade Federal de Alfenas

- **Estado Inicial**:
  - Atribuição vazia (raiz da árvore de busca): { };
- **Função Sucessor**:
  - Atribui um valor a uma variável não atribuída, desde que ela não entre em conflito com atribuições já realizadas;
- **Teste de Meta**:
  - Atingiu a atribuição completa;
  - Depende da natureza do problema;

Pilha de Execução

UNIFAL-MG
Universidade Federal de Alfenas

- Toda solução deve ser uma atribuição completa;
  - Portanto, a busca está na profundidade *n* se existem *n* variáveis;
- Não teremos problemas com o crescimento da pilha de execução, se for utilizado uma adaptação da busca em profundidade para resolver os PSRs.
  - A não ser que tenhamos uma quantidade exponencial de variáveis de decisão;
  - Mas neste caso, nosso problema se torna intratável por computadores atuais;

Pilha de Execução

UNIFAL-MG
Universidade Federal de Alfenas

- Um ponto importante é:
  - O caminho pela qual a solução é alcançada é irrelevante;
- Vocês poderão perceber isso na implementação do caixeiro viajante:
  - Através da poda "inteligente" que exige a ocorrência do consumidor x antes do y, por exemplo.

UNIFAL-MG
Universidade Federal de Alfenas

Busca com Retrocesso para PSRs

Busca com Retrocesso par PSRs

UNIFAL-MG
Universidade Federal de Alfenas

- Algo terrível pode acontecer ao gerar a árvore de busca de PSRs:
  - Suponhamos *n* variáveis de decisão;
  - E *d* valores que podem ser atribuídos a cada variável (domínio);
  - O fator de ramificação na raiz é *n.d*, pois qualquer valor de *d* pode ser atribuído a qualquer variável uma das *n* variáveis.
  - No próximo nível, o fator de ramificação é *(n-1).d*;
  - Assim, geramos uma árvore com $n!.d^n$ folhas, embora existam "apenas" d^n atribuições completas possíveis;

Busca com Retrocesso para PSRs

UNIFAL-MG
Universidade Federal de Alfenas

- A formulação de problema está correta;

- Mas é ingênua; Por quê?

Busca com Retrocesso para PSRs

UNIFAL-MG
Universidade Federal de Alfenas

- Por quê?
  - Ignora a propriedade comum a todos os PSRs;
    - A comutatividade;
- Um problema é comutativo se a ordem de aplicação nas atribuições não possui nenhum efeito sobre o resultado;
- Com esta consideração, nosso número de atribuições passa a ser d^n;
  - Apesar de representar um número astronômico para alguns problemas, a complexidade foi reduzida bruscamente;

Busca com Retrocesso para PSRs

UNIFAL-MG
Universidade Federal de Alfenas

- A expressão Busca com Retrocesso é utilizada para indicar uma busca em profundidade que:
  - escolhe valores para uma variável de cada vez e que;
  - efetua um retrocesso quando uma variável não tem valores válidos restantes para serem atribuídos;
- Quando resolvemos o Caixeiro Viajante com Tentativa e erro, temos informações específicas do domínio;

Busca com Retrocesso para PSRs

UNIFAL-MG
Universidade Federal de Alfenas

- A proposta do PSRs é em sentido oposto:
  - Resolver qualquer PSRs sem utilizar NENHUMA informação útil do domínio;
  - Está relacionado com aquela história de ferramentas específicas ou gerais;

Busca com Retrocesso para PSRs

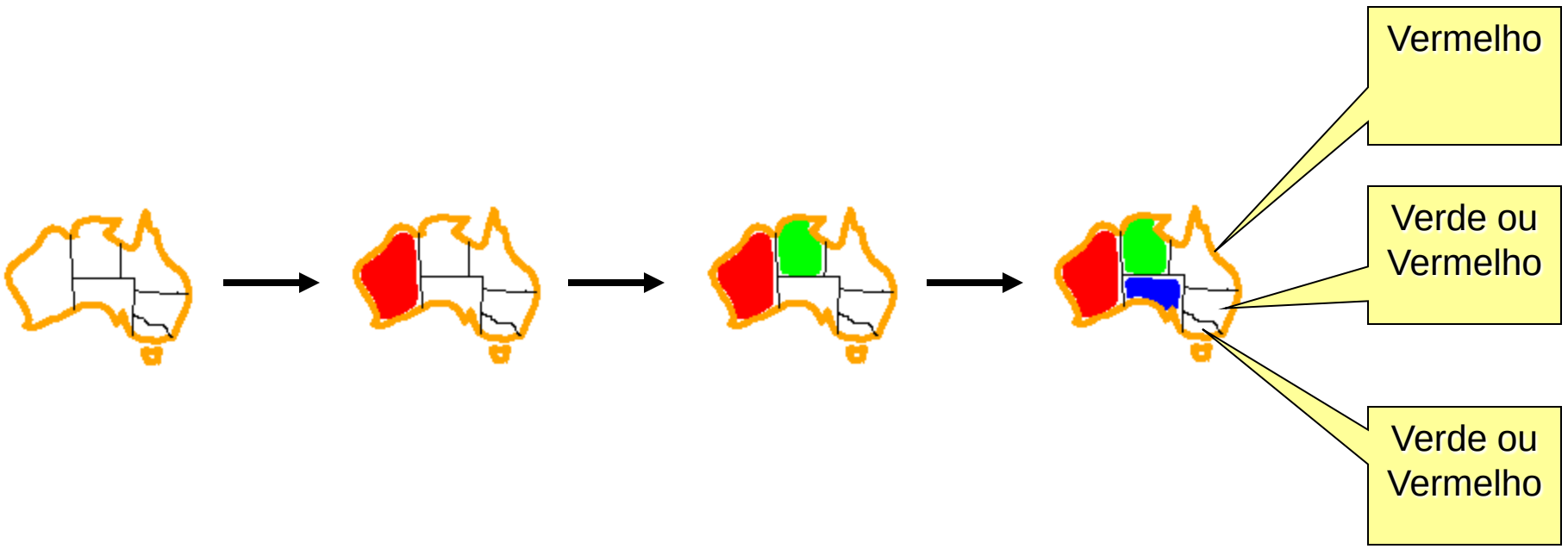
UNIFAL-MG
Universidade Federal de Alfenas

- Métodos de propósito geral podem obter ganhos em velocidade através dos seguintes questionamentos:
  - Que variável deve ser atribuída em seguida, e em que ordem seus valores devem ser experimentados?
  - Quais são as implicações das atribuições atuais para as outras variáveis ainda não atribuídas?
  - Quando um caminho falha a busca pode evitar a repetição da falha em caminhos subseqüentes?

Que variável será atribuída a seguir?

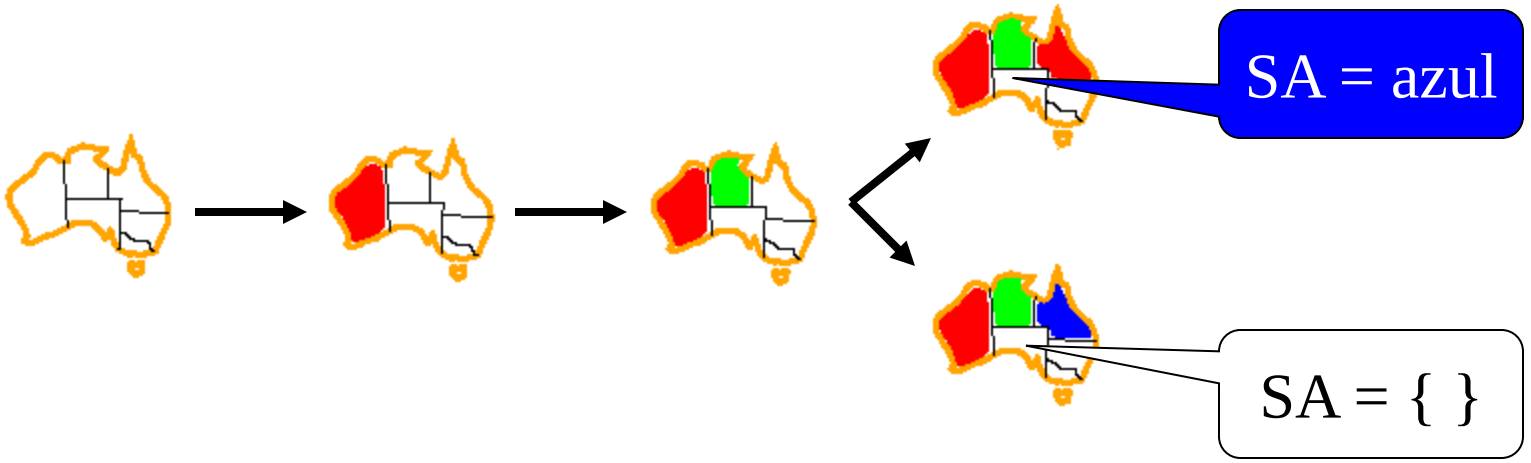
UNIFAL-MG
Universidade Federal de Alfenas

- A variável mais restringida (unária)
  - Escolha a variável com o menor número de valores legais (*Minimum Remaining Values*)



Em que ordem seus valores serão tentados?

- O valor com menos restrições
  - Escolha o valor que provoque o menor número de restrições possíveis nas variáveis restantes



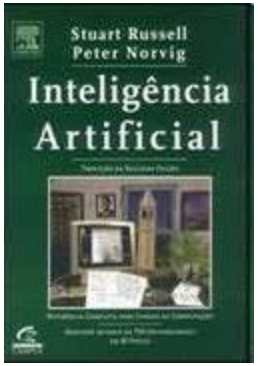
Tentativa e Erro

UNIFAL-MG
Universidade Federal de Alfenas

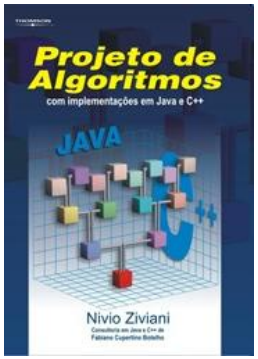
- Continuação na próxima aula...

Bibliografia

- RUSSEL, S.; NORVIG, P. (2004). Inteligência Artificial. Tradução da Segunda Edição. Editora Campus.



- ZIVIANI, N. (2007). Projeto e Algoritmos com implementações em Java e C++. São Paulo. Editora Thomson;



- Aula sobre PSR do aluno de doutorado Paulemir Campos do CIn-UFPE.
  - http://www.cin.ufpe.br/~in1006/2005/