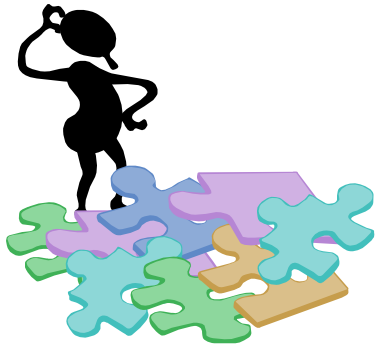
Inteligência Artificial

Novrig e Russel - Buscas

www.joinville.udesc.br/coca/

Agentes solucionadores de problemas (leia-se: resolução de problemas)

- O que é um problema em I.C.?
- Como formulá-lo?
- Como buscar a solução do problema?
  - Busca cega
  - Busca heurística
- Quais são os tipos de problemas?
- Quais são as aplicações?



Sistemas de Produção (SP), Espaços de estados e buscas

- Sistemas de Produção
  - Consiste em transformar o problema em um grafo de estados.
  - Mecanismo genérico para modelar certos tipos de problemas.

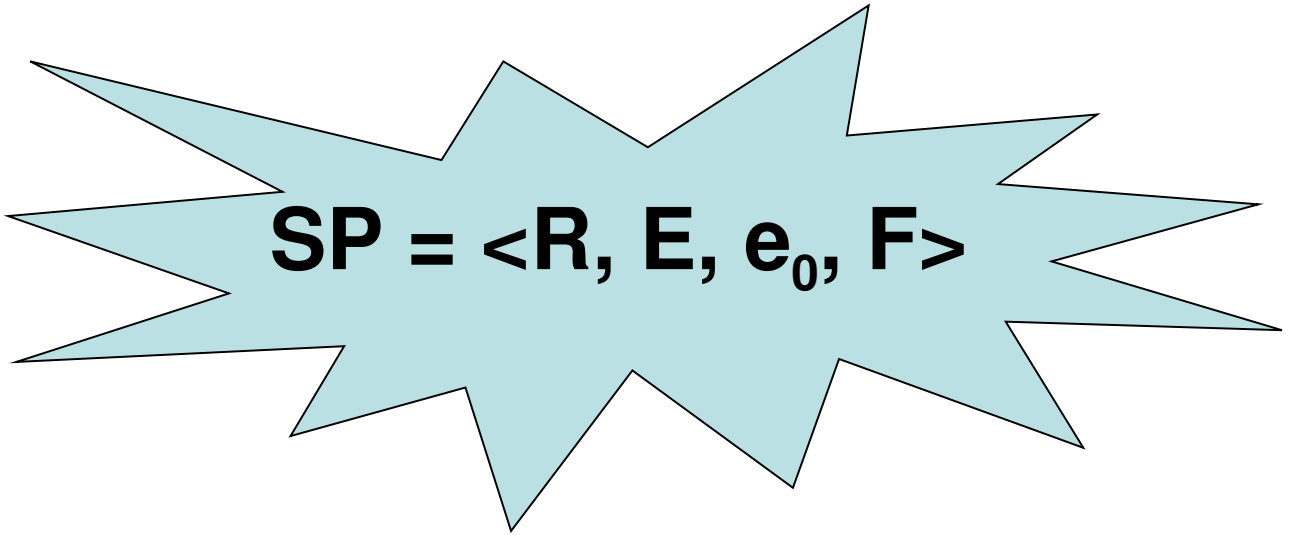
- Sistemas de Produção = Memórias a Curto Prazo + Longo Prazo
- Ver Diagrama Operacional do Prolog
- É sistemas de POST mesmo... aplicado a especificação de problemas
- Conceitos básicos: estados, espaço de estados, árvores de buscas, Est. Inicial, Est. Final, otimizar, etc.
- Complemente com as transparências do Claudio

- Para descrever um modelo formal de um problema através de um SP é necessário:
  - Definir um ***espaço de estados*** que contenha todas as possíveis configurações dos objetos relevantes ao problema.
  - Especificar um ou mais estados que descrevam situações possíveis a partir das quais o processo de resolução do problema poderá começar. Esses estados são denominados ***estados iniciais***.

- Para descrever um modelo formal de um problema através de um SP é necessário: (*continuação*)
  - Especificar um ou mais estados que seriam aceitáveis como soluções para o problema. Esses estados são denominados *estados meta* ou ***estados finais***.
  - Especificar um ***conjunto de regras*** que descrevam as ações (operadores) disponíveis. As regras são do tipo: <*padrão*, *ação*>. O padrão define quais estados podem sofrer a aplicação da regra e a ação define como são construídos novos estados a partir dos estados pertencentes ao padrão.

- Definições a saber:
  - **Solução** para o problema: caminho (seqüência de ações ou operadores) que leva do estado inicial a um estado final (objetivo).
  - **Função de custo** de caminho: atribui um custo numérico a cada caminho gerado.

Definição Formal: Sistema de Produção



- ***R*** é um conjunto de regras
- ***E*** é um conjunto de estados
- $\boldsymbol{e_0} \in E$ é o estado inicial
- ***F*** é o conjunto de estados finais.

Definição Formal: Regra de Produção

- É constituída por um par

$$\langle p, f \rangle$$

Onde: ***p*** é o *padrão* da regra e ***f*** constitui a *ação*.

- O padrão *p* define como verdadeiros os estados aos quais a regra é aplicável.

$$p: E \rightarrow \{V, F\}$$

- A aplicação da regra consiste em aplicar a operação/ação *f* a um destes estados, gerado um novo estado.

$$f: E \rightarrow E$$

- Exemplo 1 de **SP**:

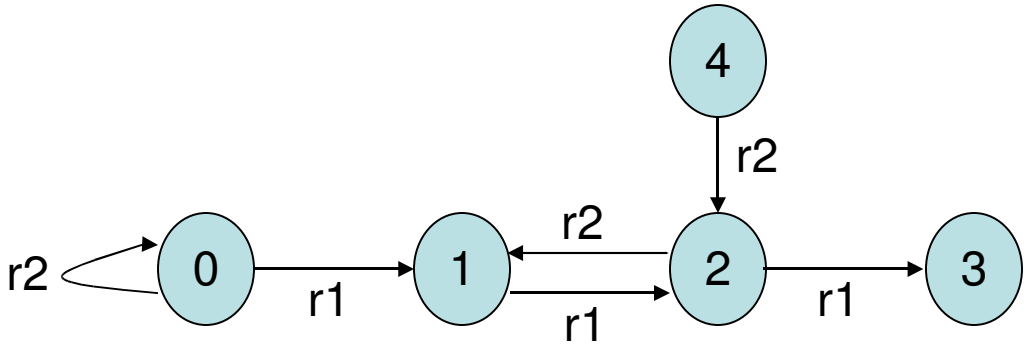
$E = \{ 0, 1, 2, 3, 4 \}$

$e_0 = 0$

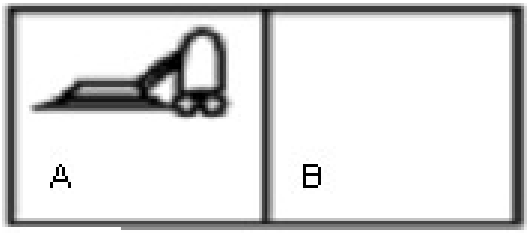
$F = \{ 3, 4 \}$

$R = \{ \quad r_1 = (x \mid x < 3) \rightarrow x + 1$

$r_2 = (x \mid \acute{e}Par(x)) \rightarrow x / 2 \quad \}$



- Exemplo 2: Mundo do aspirador de pó



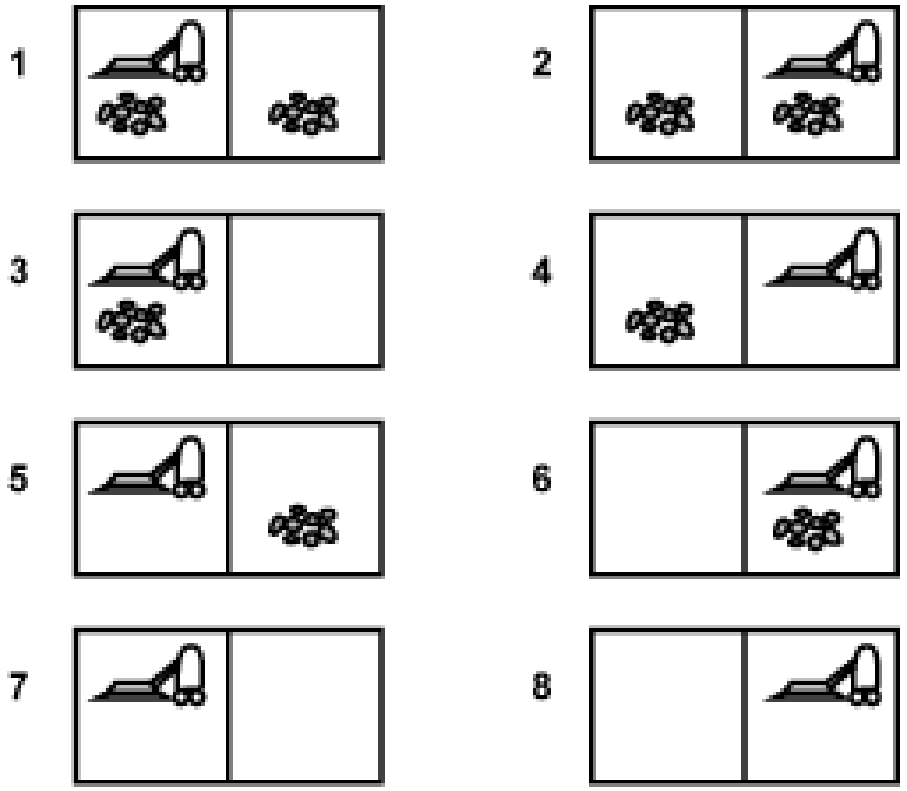
Existem dois quartos: quadrado A e B. O aspirador de pó percebe em que quadrado está e se existe sujeira. Ele pode optar por mover-se para a esquerda (L), direita (R) e aspirar (S).

Defina um Sistema de Produção para este cenário.

Defina o agente solucionador de problemas e o ambiente onde o agente está inserido.

- Exemplo 2: Mundo do aspirador de pó

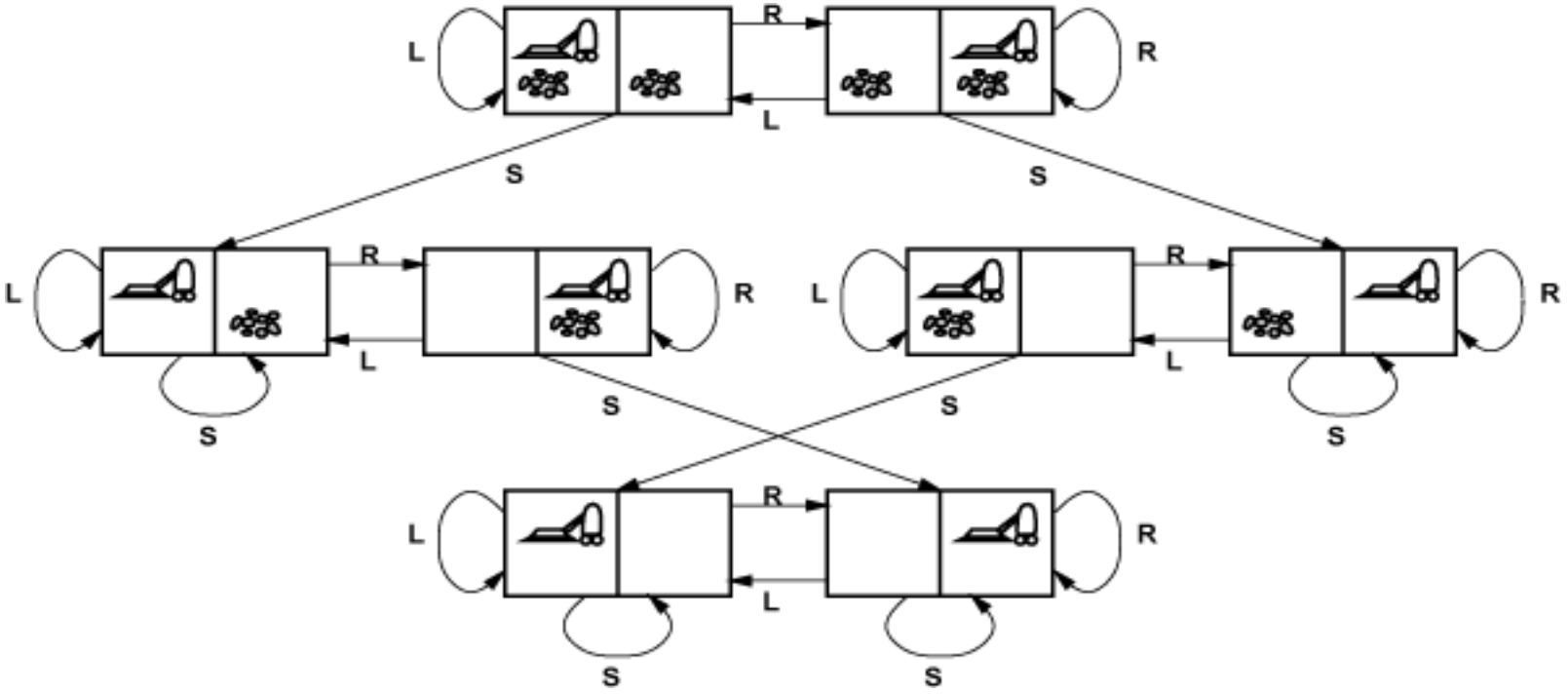
- Formulação do problema:
  - estados = mostrados na figura
  - estado inicial = qualquer um dos estados possíveis
  - teste de término = os dois quartos limpos
  - operadores = mover direita, mover esquerda, aspirar
  - custo do caminho = quantidade de ações realizadas



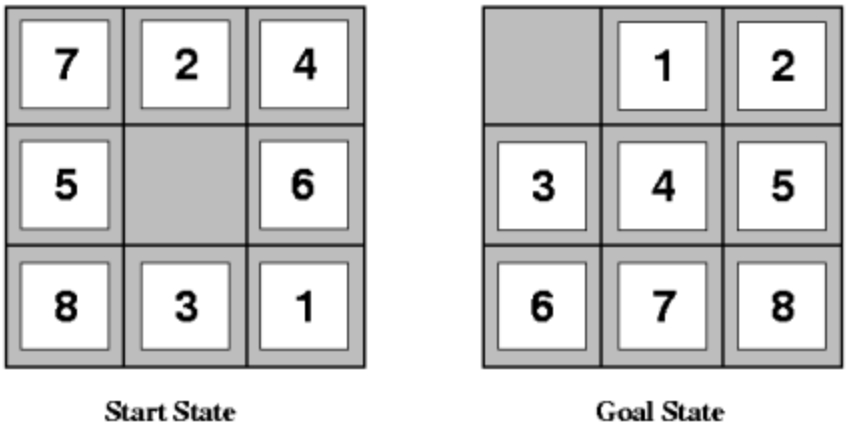
- Exemplo 2: Mundo do aspirador de pó

- Regras de Produção:
  - R = { r_1= [A, sujo] → S; r_2= [A, limpo] → R;

    r_3= [B, sujo] → S; r_4= [B, limpo] → L}



- Exemplo 3: Quebra-cabeça de 8 peças



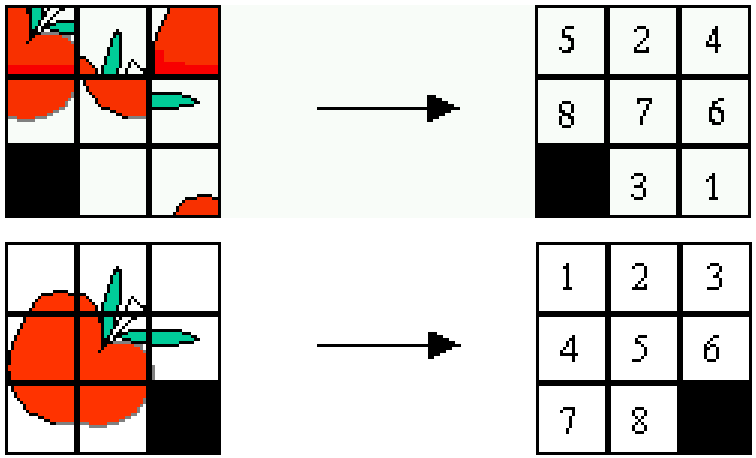
Uma peça adjacente ao espaço vazio pode deslizar para o espaço. O objetivo é alcançar um estado objetivo especificado.

Defina um Sistema de Produção para este cenário.

Defina o agente solucionador de problemas e o ambiente onde o agente está inserido.

- Exemplo 3: Quebra-cabeça de 8 peças

- Formulação do problema:
  - estados = qualquer descrição dos números e do espaço vazio. 9!/2 = **181.440** estados acessíveis
  - estado inicial = qualquer um dos estados possíveis
  - teste de término = estado desejado
  - operadores = vazio para a direita, esquerda, acima e abaixo
  - custo do caminho = cada passo custa 1

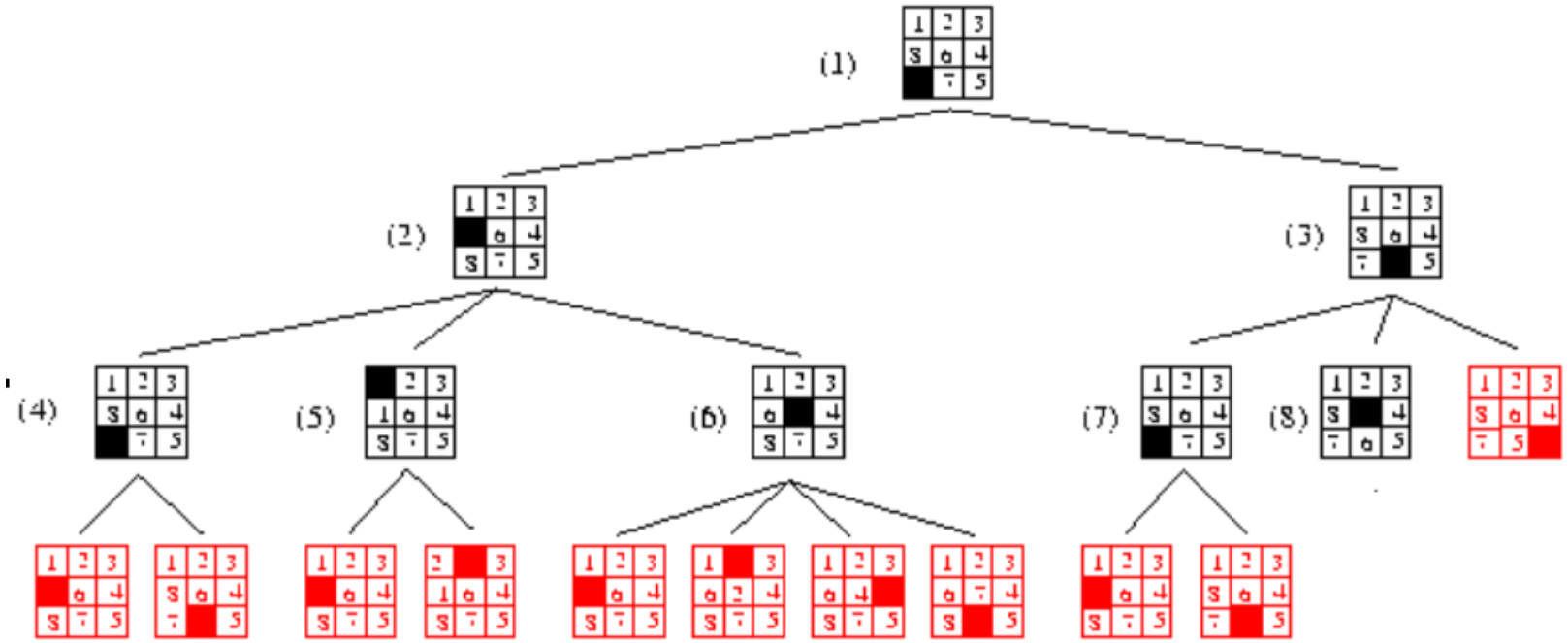


Aplicações: Problemas Reais

- Cálculo de rotas
  - rotas em redes de computadores
  - sistemas de planejamento de viagens
  - planejamento de rotas de aviões
  - caixeiro viajante
- Alocação (*Scheduling*)
  - salas de aula
  - máquinas industriais (*job shop*)
- Projeto de VLSI
  - *cell layout*
  - *channel routing*
- Navegação de robôs
  - generalização do problema da navegação
  - robôs movem-se em espaços contínuos, com um conjunto (infinito) de possíveis ações e estados
    - controlar os movimentos do robô no chão, e de seus braços e pernas requer espaço multi-dimensional
- Montagem de objetos complexos por robôs
  - ordenar a montagem das diversas partes do objeto
- etc...

Busca em Espaço de Estados

- Uma vez o problema bem formulado... o estado final deve ser “buscado”.
- O **espaço de estado** é a árvore de todos os estados que podem ser produzidos a partir do estado inicial.



Busca em Espaço de Estados

- Deve-se usar um método de busca para saber a ordem correta de aplicação dos operadores/ ações que lavará do estado inicial ao final.
- Uma vez a busca terminada com sucesso, é só executar a solução (conjunto ordenado de operadores a aplicar).

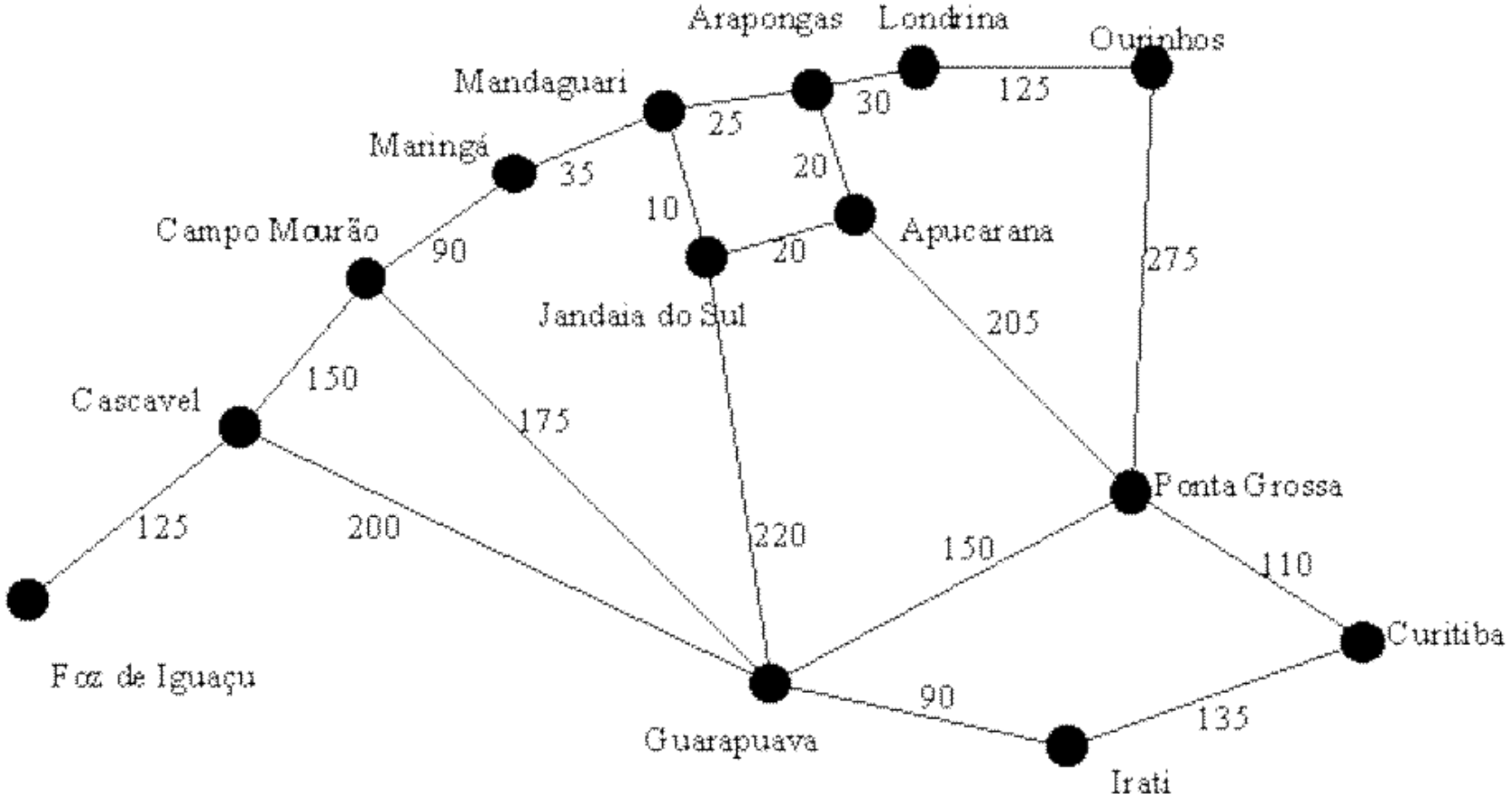
Busca em Espaço de Estados: Geração e Teste

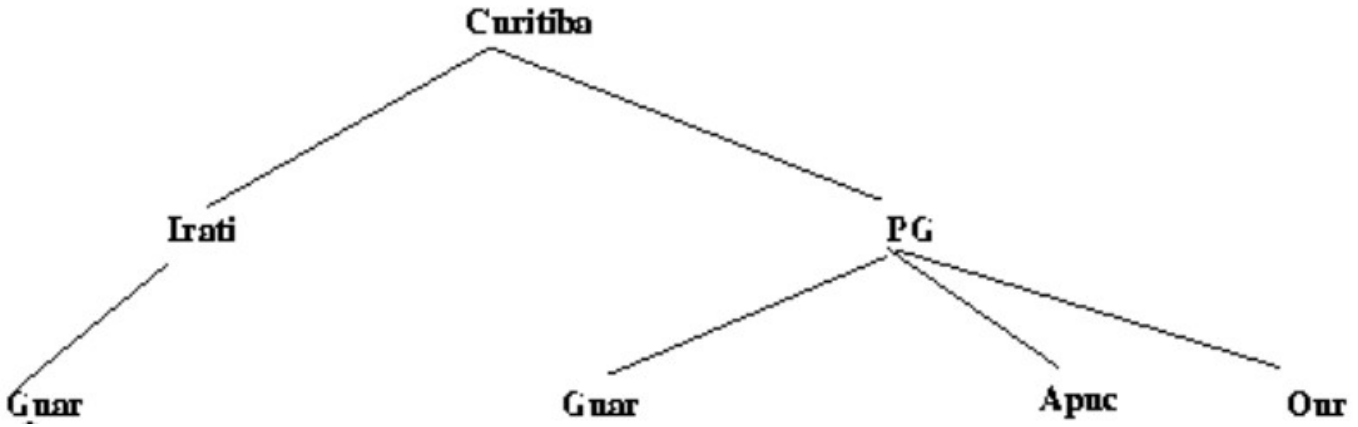
- *Fronteira* do espaço de estados
  - **nós** (estados) a serem expandidos no momento.
- Algoritmo:

Obs: o algoritmo começa com a fronteira contendo o estado inicial do problema.

1. Selecionar o primeiro nó (estado) da *fronteira* do espaço de estados:
   - se a fronteira está vazia, o algoritmo termina com falha.
2. Testar se o nó é um estado final (solução):
   - se "sim", então retornar nó - a busca termina com sucesso.
3. Gerar um novo conjunto de estados pela aplicação dos operadores ao estado selecionado.
4. Inserir os nós gerados na *fronteira*, de acordo com a estratégia de

Caminho mais curto entre Curitiba e Maringá:



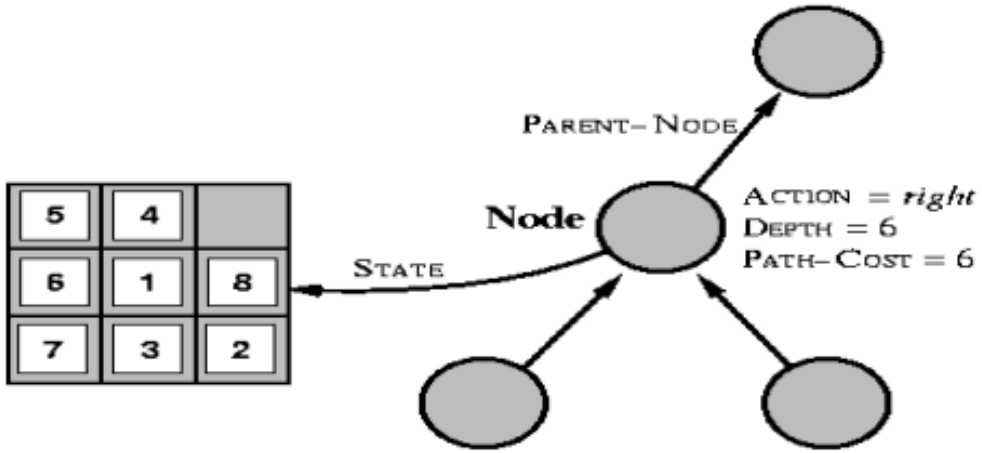


Busca em Espaço de Estados

- Os nós da árvore podem guardar mais informação do que apenas o estado:

→ são uma estrutura de dados com 5 componentes:

1. o estado correspondente
2. o seu nó pai
3. o operador aplicado para gerar o nó (a partir do pai)
4. a profundidade do nó
5. o custo do nó (desde a raiz)



Métodos de Busca

- Busca exaustiva - cega
  - Não sabe qual o **melhor** nó da fronteira a ser expandido. Menor custo de caminho desse nó até um ***nó final*** *(objetivo)*.
  - Estratégias de Busca (ordem de expansão dos nós):
    - Caminhamento em largura
    - Caminhamento em profundidade
  - Direção de Busca:
    - Do estado inicial para o objetivo
    - Do objetivo para o estado inicial

Métodos de Busca

- Busca heurística - informada
  - Estima qual o melhor nó da fronteira a ser expandido com base em ***funções heurísticas*** => conhecimento
  - Estratégia de busca: *best-first search* (melhor escolha); busca tabu; Recozimento Simulado; ...
  - Direção de Busca: idem à busca cega

Critérios de Avaliação das Estratégias de Busca

- **Completude**:
  - a estratégia sempre encontra uma solução quando existe alguma?
- **Custo do tempo**:
  - quanto tempo gasta para encontrar uma solução?
- **Custo de memória**:
  - quanta memória é necessária para realizar a busca?
- **Otimalidade** (*optimality*):
  - a estratégia encontra a melhor solução quando existem diferentes soluções?

Exercício: Modele o problema

- Problema dos Dois Baldes de Água

Você recebe dois baldes de água, um de quatro litros e outro de três litros. Nenhum deles possui qualquer marcação de medida. Há uma torneira que pode ser utilizada para encher os baldes de água. Como colocar exatamente dois litros d'água dentro do balde de quatro litros?

- Defina o agente solucionador de problemas e o ambiente onde o agente está inserido.
- Defina um Sistema de Produção para este cenário.

- Problema dos Dois Baldes de Água
  - **Espaço de estados:** Conjunto de pares ordenados de números naturais (*x,y*) tal que $x \in \{0, 1, 2, 3, 4\}$ e $y \in \{0, 1, 2, 3\}$ onde *x* representa a quantidade de água no balde de 4 litros, e *y* representa a quantidade de água no balde de 3 litros.
  - **Estado inicial:** Estado no qual ambos os baldes estão vazios: (0, 0).
  - **Estados finais**: Todos estados onde a quantidade de água no primeiro balde é 2, ou seja: ***(2,y)***.

- Problema dos Dois Baldes de Água
  - **Conjunto de regras**:

Ação	Regra
Encher o balde de 4 litros	***R*1 = ((x, y) | *x* < 4) → (4, y)**
Encher o balde de 3 litros	***R2* = ((x, y) | y < 3) → (x, 3)**
Esvaziar o balde de 4 litros no chão	***R3* = ((*x, y)* | *x* > 0) → (0, *y*)**
Esvaziar o balde de 3 litros no chão	***R4* = ((*x, y)* | y > 0) → (x, 0)**
Despejar água do balde de 4 litros dentro do balde de 3 litros	***R5* = ((*x, y)* | y < 3) → (x - min(x, 3-y), min(3, x+y))**
Despejar água do balde de 3 litros dentro do balde de 4 litros	***R6* = ((*x, y)* | x < 4) → (min(4, x+y), y - min(4-x, y))**

  - Defina uma seqüência de regras que represente uma possível solução.

- Problema dos Dois Baldes de Água
  - **Ilustração do Espaço de Estados**: