



Universidade Federal de Santa Catarina  
Disciplina de Inteligência Artificial  
Campus Araranguá

# Teoria de problemas e sua resolução

Prof<sup>a</sup>. Eliane Pozzebon  
*[epozzebon@gmail.com](mailto:epozzebon@gmail.com)*

# Plano de aula

## **Unidade 2 - Teoria de problemas e sua resolução.**

- Características de problemas
- Complexidade de algoritmos na solução de problemas
- Estratégias para resolver problemas
- Exemplo de problemas clássico de IA

# Características de problemas

A IA se ocupa da resolução de problemas, para tal é necessário conhecimento sobre o problema e técnicas de manipular este conhecimento para obter a solução.

## O que é um PROBLEMA?

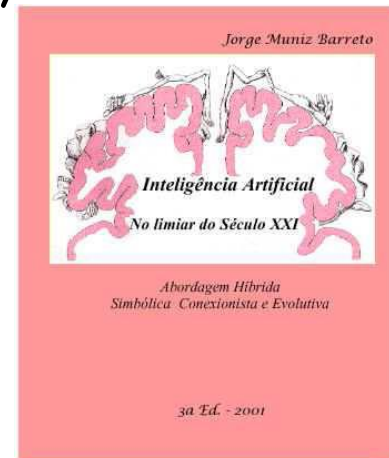
Resolver um problema é diferente de ter um método para resolvê-lo.

Antes de tentar buscar a solução de um problema, deve-se responder as seguintes perguntas:

Quais são os dados?

Quais são as soluções possíveis?

O que caracteriza uma solução satisfatória?



Exemplo de problema:

## O tabuleiro de xadrez mutilado

- Suponha que se tem um tabuleiro de xadrez do qual dois cantos diagonalmente opostos são retirados, restando portanto, no tabuleiro, sessenta e dois quadrados. Agora suponha que se tem trinta e um dominós, cada qual cobrindo exatamente dois quadrados do tabuleiro.
- A pergunta do problema é:  
“Pode-se encontrar algum modo de arranjar estes trinta e um dominós no tabuleiro de modo a cobrir todos os sessenta e dois quadrados?” Se isto pode ser feito, explicar como. Se não puder, prove porque.

## A resposta correta é:

que **o tabuleiro não pode ser coberto pelos dominós** e somente se chega a esta solução se incluir na representação do problema o fato de que **cada dominó precisa cobrir um quadrado preto e um branco**, e não qualquer dois quadrados.

A partir disto tem-se que não existe outro modo de colocar um dominó em dois quadrados do tabuleiro sem ter que cobrir exatamente um quadrado branco e quadrado preto. **Isto significa que com trinta e um dominós se pode cobrir trinta e um quadrados pretos e trinta e um quadrados brancos.**

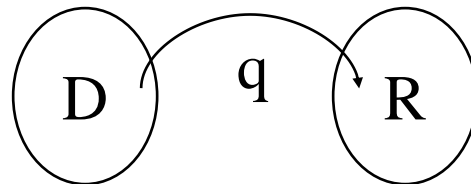
Mas a mutilação removeu dois quadrados pretos e portanto, restam no tabuleiro trinta quadrados pretos e trinta e dois quadrados brancos. Isto segue que o tabuleiro não pode ser coberto por trinta e um dominós.

Pode-se perceber como a solução fica fácil quando se usa a representação adequada, ou seja, **quando representa cada dominó como cobrindo um quadrado preto e um quadrado branco.**

Com esta representação do problema se é encorajado a contar e comparar o número de quadrados brancos e pretos. Portanto o efeito da representação do problema é que ela possibilita o uso de operadores críticos para se chegar à solução.

# Teoria de Problemas

- **Definição:** Um problema é um objeto matemático  $P=\{D,R,q\}$ , consistindo de dois conjuntos não vazios,  $D$  os dados e  $R$  os resultados possíveis, e de uma relação binária  $q \subset D \times R$ , a condição que caracteriza uma solução satisfatória, associando a cada elemento do conjunto de dados a solução desejada.
- **Exemplo:** Um problema de diagnóstico médico
  - O conjunto de dados disponíveis  $d \in D$  (observação da anamnese, sintomas, exames, etc.)
  - $R$  é o conjunto de doenças possíveis
  - Solução satisfatória: encontrar o par  $(d,r)$  onde  $r \in R$  é o diagnóstico correto.
- A definição de um problema permite testar se um certo elemento é ou não solução, mas não guia na busca deste elemento.



# Teoria de Problemas

## Modos de definir uma FUNÇÃO PROBLEMA

1. Por ENUMERAÇÃO EXAUSTIVA
2. DECLARATIVAMENTE
3. Por um PROGRAMA (um algoritmo)
4. Por EXEMPLOS

# Teoria de Problemas

- **Enumeração exaustiva**

Neste caso fornece-se todos os conjuntos de pares, dado, resultado. Evidentemente, este modo de definir uma função, só se aplica no caso que o conjunto de dados é finito.

- **Exemplo:** seja uma agenda de telefones. Ela pode ser considerada como a função que associa a cada nome de pessoa seu telefone.



# Teoria de Problemas

## **Declarativamente:**

Definir declarativamente um problema é dar propriedades que devem ser satisfeitas pela solução do problema.

É baseado em axiomas (verdades universais) e regras de inferência.

**Exemplo :** Dado um número real associa dois números cuja soma de seus quadrados é igual ao número real dado.

# Teoria de Problemas

- **Por um algoritmo:**

Um programa de computador define a correspondência entre dados e resultados sempre que ele para conseguindo chegar a uma solução. Portanto um programa pode ser considerado como um modo de definir um problema.

- **Exemplo:** Formulário de Imposto de Renda em um País com leis mais complicadas que o nosso...

# Teoria de Problemas

- **Por exemplos:**

Pode-se reconhecer que, neste caso, a solução não é única: todas as funções que sejam iguais dentro da região em que o problema é definido são válidas. Trata-se de fazer uma aproximação.

- Procura-se aproximar a solução que satisfaça aos exemplos (ou casos) conhecidos. Tradicionalmente é resolvido por raciocínio baseado em casos mas pode ser usada a IAC

# Teoria de Problemas

Os modos de definir uma função levam ao conceito de

- **COMPUTABILIDADE**

**Definição 1:** Uma função é dita computável se é possível calcular seu valor para todos os elementos de seu domínio de definição.

- Ex1.: Equações Diofantinas
  - $a^n + b^n = c^n \quad n \geq 3 \quad a, b, c \text{ Inteiros}$
- Ex2.: Problema da parada de um programa
  - Dado um programa e um conjunto de dados infinito, é impossível ter um outro programa que decida se o primeiro programa vai conseguir parar para todos os dados.

- **COMPLEXIDADE**

**Definição 2:** A complexidade de um problema, com relação a um conjunto de recursos conhecidos para a solução do problema.

# Teoria de Problemas

- Se a computabilidade diz respeito à existência de solução para um problema, a complexidade se refere a quantidade de recursos necessários para resolvê-los.
- Um mesmo problema pode ter complexidade diferente, dependendo da técnica que se utiliza para resolvê-lo.
- **Definição 3:** Um problema é dito NP-Completo quando não se conhece algoritmo de ordem polinomial capaz de resolvê-lo.
  - Ex.: Problema do caixeiro-viajante resolvido de maneira algorítmica.

# Teoria de Problemas

## HEURÍSTICAS

**Definição 4:** Conjunto de regras e métodos que conduzem à descoberta, à invenção e à resolução de problemas.

O papel das heurísticas - “boa solução”

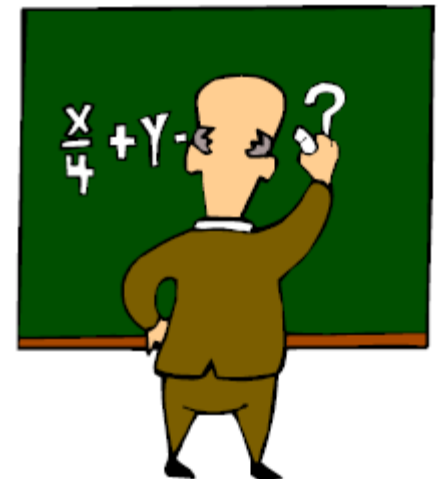
Na IA as heurísticas são as “técnicas” que possibilitam tratar problemas NP-Completo e buscar algoritmos de ordem mínima para problemas polinomiais

# Teoria de Problemas

- **Estratégias Básicas para Resolver Problemas** (Estratégias constituem os modos básicos de raciocínio para resolver problemas)  
Pela definição do problema, o qual se apresenta como uma função, estes modos de raciocínio devem se adaptar ao modo que a função foi definida.
  1. *Por enumeração exhaustiva:* o conhecimento necessário para resolver o problema está na enumeração.
  2. *Declarativamente:* leva frequentemente a problemas de busca. "Utilizar um método de busca em que, por passos sucessivos se aproxima da solução, usando algumas vezes técnicas sem grande justificativa teórica". ESTA É A **ABORDAGEM DA IA SIMBÓLICA!**
  3. *Por exemplos:* Se o problema foi definido por exemplos, se deverá usar um método para aproximar a função. ESTA É A **ABORDAGEM DA IA CONEXIONISTA!**
- **ALGUNS PROBLEMAS CLÁSSICOS:**
  - Missionários e canibais; Torres de Hanói; Baldes de Água; Jogo do Oito; Reconhecimento de Caracteres, Previsão, etc.

# Resolução de Problemas

- Já vimos o que é um problema. Vamos agora buscar mecanismos para representá-lo e resolvê-lo, utilizando as técnicas da IA, ou seja, usando e manipulando CONHECIMENTO
- O Estudo do Conhecimento
- Resolução de Problemas por Busca
- Representação de Conhecimento





# Resolução de Problemas

- O Estudo do Conhecimento
  - Uma teoria em IA consiste na especificação do conhecimento necessário a uma entidade cognitiva.
  - O que é uma ENTIDADE COGNITIVA?
    - É o “mecanismo” inteligente que permite entre outras atividades: solução de problemas, uso de linguagem, tomada de decisões, percepção, etc...
    - Na abordagem da IA Simbólica, a simulação da capacidade cognitiva requer conhecimento declarativo (definição declarativa da função) e algum tipo de raciocínio. Além disso, a evolução dos estados de conhecimento de um agente pode ser descrita em forma de linguagem (lógica ou natural).

# Resolução de Problemas

- O conhecimento é central para a tarefa inteligente e na IAS, para que esta tarefa ocorra são necessários:
  - Uma BASE DE CONHECIMENTOS
  - Um MOTOR DE INFERÊNCIA
- **Base de Conhecimentos:**
  - Contém a informação específica sobre o domínio e será tão complexa quanto for o domínio e a capacidade cognitiva a ser simulada.
- **Motor de Inferência:**
  - Mecanismo que manipula a Base de Conhecimentos e gera novos conhecimentos.

# Métodos de Busca

- A maioria dos problemas interessantes de IA não dispõe de soluções algorítmicas. Porém:
  - São solucionáveis por seres humanos e, neste caso, sua solução está associada à “inteligência”;
  - Formam classes de complexidade variável existindo desde pequenos problemas triviais (jogo da velha) até instâncias extremamente complexas (xadrez);
  - São problemas de conhecimento total, isto é, tudo o que é necessário para solucioná-los é conhecido, o que facilita sua formalização.
  - Suas soluções têm a forma de uma sequência de situações legais e as maneiras de passar de uma situação para outra são em número finito e conhecidas.
- Diante da falta de solução algorítmica viável, o único método de solução possível é a BUSCA.

# Métodos de Busca

- Decidem o que fazer pela busca de ações que levem a estados desejáveis
  - estado inicial
  - operadores
  - teste de meta
  - função de custo de caminho
- Desempenho da Busca
  - Encontra uma solução?
  - É uma boa solução?
    - Custo do caminho
  - Qual o custo da busca?
    - Tempo e memória
  - Custo total = Custo da busca + custo do caminho

# Métodos de Busca

- Exemplo: Jogo do Oito

5	4	
6	1	8
7	3	2

Start State

1	2	3
8		4
7	6	5

Goal State

- **Estados:** local de cada uma das peças e do espaço
- **Operadores:** mover o espaço para cima, para baixo, esquerda ou direita.
- **Teste de meta:** dado na figura
- **Custo do caminho:** 1 para cada movimento

# Métodos de Busca

- Estratégias de Busca
  - Critérios
    - Completude
    - Complexidade de Tempo
    - Complexidade de Espaço
    - Otimização
  - Métodos
    - Busca Cega - Não existe informação
    - Busca Heurística - Faz uso de informação

# Busca Cega

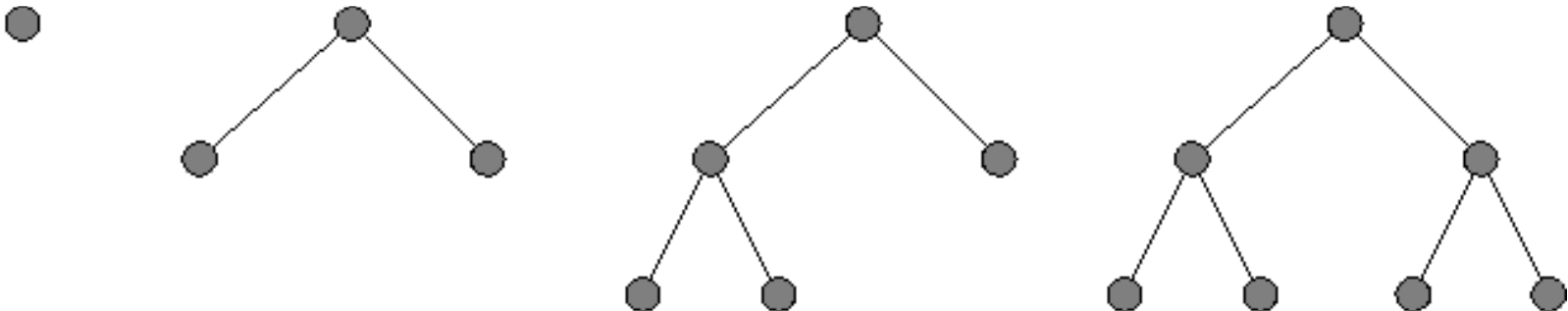
(Blind Search ou Uninformed Search)

- Uma estratégia de busca é dita cega se ela não leva em conta informações específicas sobre o problema a ser resolvido.
- Tipos de Busca Cega
  - Busca em largura
  - Busca pelo custo uniforme
  - Busca em profundidade
  - Busca em profundidade limitada
  - Busca por aprofundamento iterativo
  - Busca bidirecional

## Busca Cega

# Busca em Largura (Amplitude)

- Consiste em construir uma árvore de estados a partir do estado inicial, aplicando a cada momento, todas as regras possíveis aos estados do nível mais baixo, gerando todos os estados sucessores de cada um destes estados. Assim, cada nível da árvore é completamente construído antes de qualquer nodo do próximo nível seja adicionado à árvore





## Busca Cega

# Busca em Largura (Amplitude)

- Características: Completa e Ótima
  - Se existe solução, esta será encontrada;
  - A solução encontrada primeiro será a de menor profundidade.
- Análise de Complexidade - Tempo e Memória
  - Seja um fator de ramificação  $b$ .
  - Nível 0: 1 nó
  - Nível 1:  $b$  nós
  - Nível 2:  $b^2$  nós
  - Nível 3:  $b^3$  nós
  - Nível  $d$  (solução)  $b^d$  nós

## Busca Cega

# Busca em Largura (Amplitude)

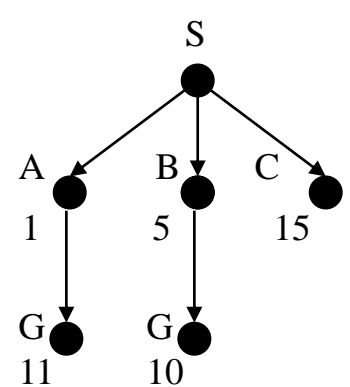
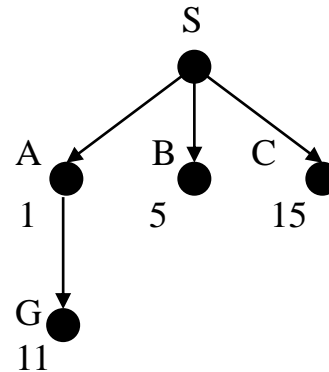
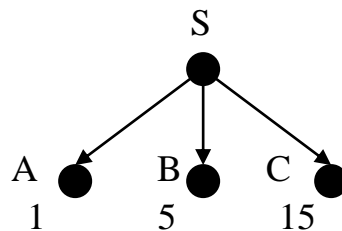
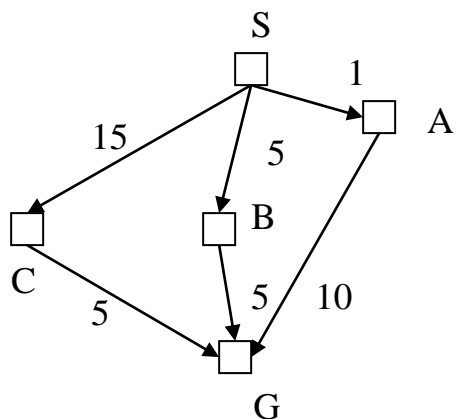
- Análise de Complexidade - Tempo e Memória

Depth	Nodes	Time	Memory
0	1	1 millisecond	100 bytes
2	111	.1 seconds	11 kilobytes
4	11,111	11 seconds	1 megabyte
6	$10^6$	18 minutes	111 megabytes
8	$10^8$	31 hours	11 gigabytes
10	$10^{10}$	128 days	1 terabyte
12	$10^{12}$	35 years	111 terabytes
14	$10^{14}$	3500 years	11,111 terabytes

# Busca Cega

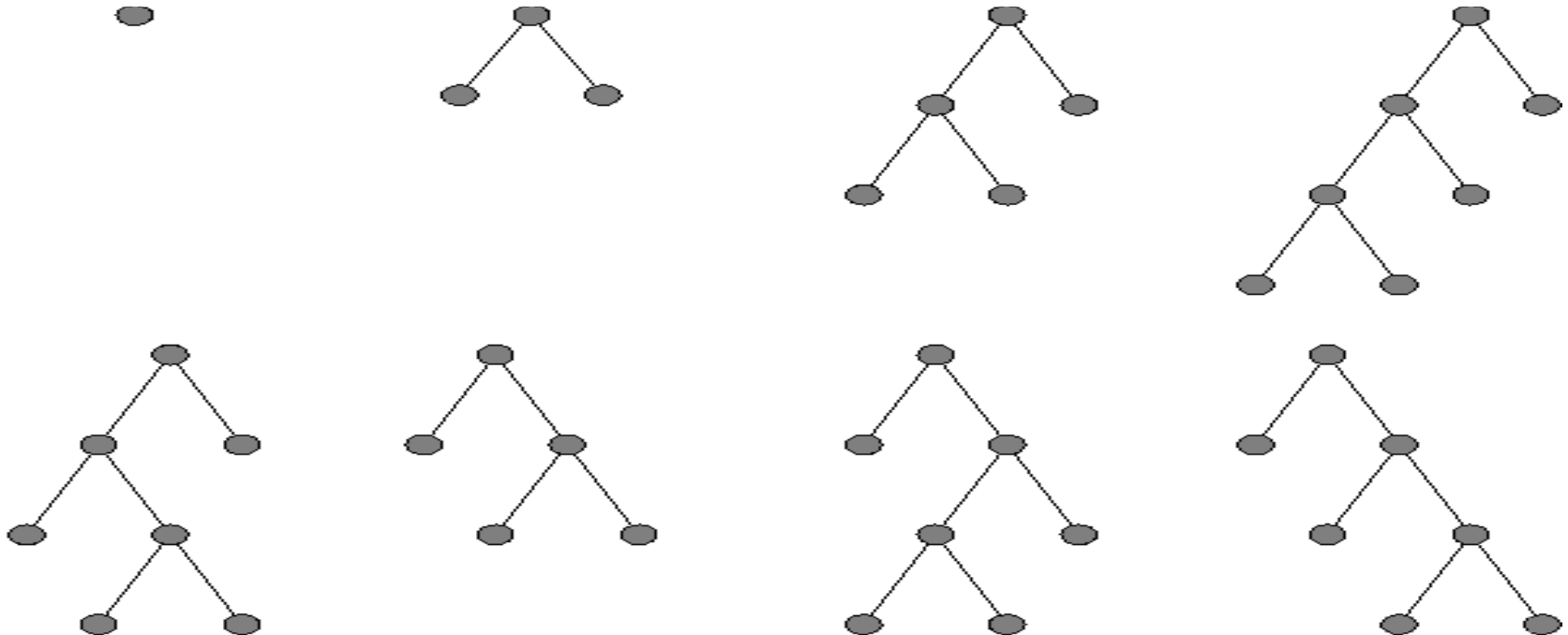
## Método do Custo Uniforme

- Supondo que exista um "custo do caminho" associado a cada nó percorrido e que se deseje achar o caminho de custo mínimo.
- Neste caso, o algoritmo anterior é modificado para expandir primeiro o nó de menor custo.
- Exemplo: Problema de Rota entre S e G



# Busca Cega

- Procurar explorar completamente cada ramo da árvore antes de tentar o ramo vizinho.



## Busca Cega

# Busca em Profundidade

- Exemplo: Um balde de 4 litros e um balde de 3 litros. Inicialmente vazios. Estado Final: um dos baldes com 2 litros de água.
- O que acontece quando nenhuma regra pode ser aplicada, ou a árvore atinge uma profundidade muito grande sem que tenha encontrado uma solução?
  - Neste caso ocorre o BACKTRACKING, ou seja, o algoritmo volta atrás e tenta outro caminho.

# Busca Cega

## Busca em Profundidade

- Características: Não é Completa e Não é Ótima
  - Se admitir estados repetidos ou um nível máximo de profundidade, pode nunca encontrar a solução.
  - A solução encontrada primeiro poderá não ser a de menor profundidade.
  - O algoritmo não encontra necessariamente a solução mais próxima, mas pode ser **MAIS EFICIENTE** se o problema possui um grande número de soluções ou se a maioria dos caminhos pode levar a uma solução.

# Busca Cega

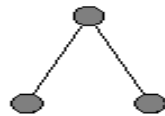
## Busca por Aprofundamento Iterativo

- Teste de todos os possíveis limites com busca por profundidade limitada.
- Em geral é o melhor método quando o espaço de busca é grande e a profundidade é desconhecida.

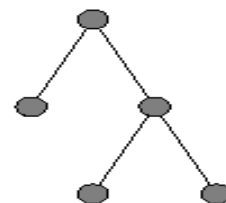
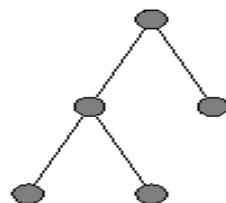
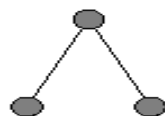
Limit = 0



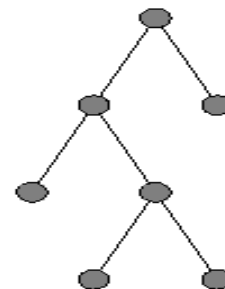
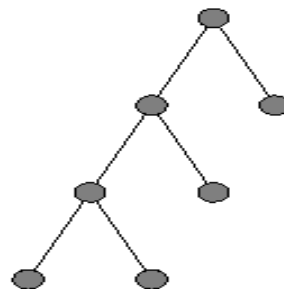
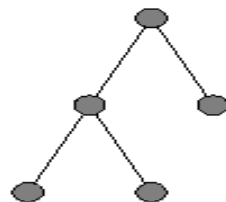
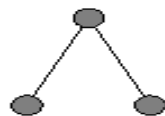
Limit = 1



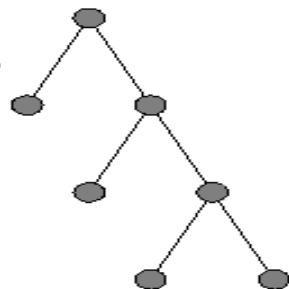
Limit = 2



Limit = 3



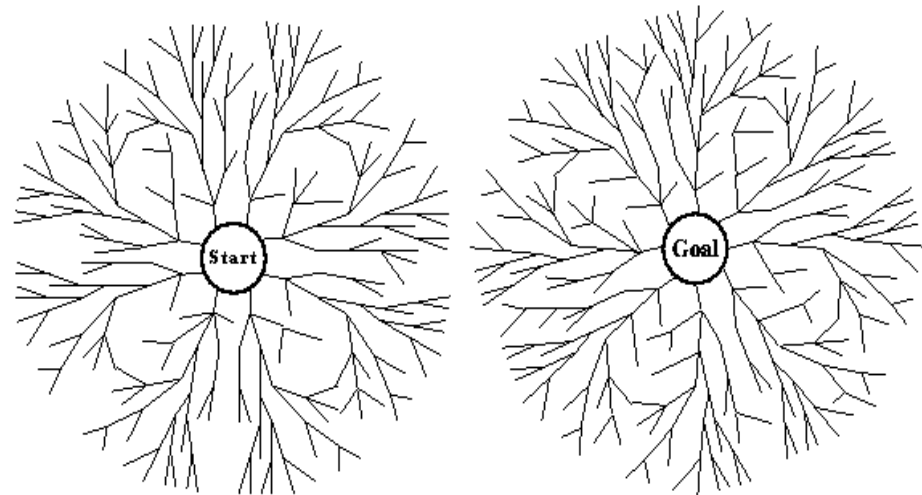
.....



# Busca Cega

## Busca Bidirecional

- A idéia deste método de busca é procurar simultaneamente “para a frente” a partir do estado inicial e “para trás” a partir do estado final, e parar quando as duas buscas se encontrarem no meio.
- Nem sempre isto é possível, para alguns problemas os operadores não são reversíveis, isto é, não existe a função predecessora e portanto não é possível fazer a busca “para trás”.





# Características de Problemas

- Não existe um único método de resolução para todos os problemas.
- São conhecidos os passos para achar a solução?
  - O problema é suficientemente bem definido?
    - Ex: identificação de assinaturas
- O problema é decomponível?
  - Na solução para o problema inicial pode ser obtida pela composição da solução de alguns problemas mais elementares.

# Características de Problemas

- Os passos para as soluções podem ser desfeitos?
  - Na solução de um problema retroceder é voltar a trajetória no espaço de soluções (backtrack). É um dos mecanismos básicos utilizados no Prolog.
- O universo é desprezível?
  - É possível planejar uma sequência de passos e o estado resultante será sempre o mesmo,
    - Existe problemas onde um fator de chance esta envolvido (Exemplo: jogo de cartas)

# Características de Problemas

- Uma boa solução é relativa ou absoluta?
  - Uma boa solução encontrada é absoluta,  
Devemos estar certos de que se começarmos com condições iniciais diferentes, obteremos a mesma solução.
- O conhecimento disponível é consistente?
  - Uma base de conhecimento é dita consistente se não existe incompatibilidade entre peças elementares de conhecimento dentro dela.
    - O estado A é verdadeiro
    - O estado C é verdadeiro
    - Se C então D. Se D então A ou-exclusivo C

# **Problema dos canibais e dos missionários**

## **Método: Busca em Profundidade**

- **Três canibais e três missionários estão viajando juntos e chegam à margem de um rio. Eles desejam atravessar para a outra margem para, desta forma, continuar a viagem. O único meio de transporte disponível é um barco que comporta no máximo duas pessoas. Há uma outra dificuldade: em nenhum momento o número de canibais pode ser superior ao número de missionários pois desta forma os missionários estariam em grande perigo de vida. Como administrar a travessia?**

# Problema dos canibais e dos missionários

- Para resolver este problema, considere as situações (ou estados) no que se refere ao número de canibais e missionários que estão numa das duas margens, sempre considerando que o barco está aportado, ou seja, antes do início de uma travessia ou quando ela foi completada. Podemos representar cada um destes estados por uma tripla de valores **(c,m,b)**, onde **c** e **m** representam o número de canibais e de missionários, respectivamente, que estão na margem **b** onde está o barco.
- Representação:
  - M = Missionário
  - C = Canibal
  - B = Barco
  - () = Margem do rio
  - Estado inicial (MMMCCCB) ()

# Métodos de Busca

- Exemplo: Jogo do Oito

5	4	
6	1	8
7	3	2

Start State

1	2	3
8		4
7	6	5

Goal State

- **Estados:** local de cada uma das peças e do espaço
- **Operadores:** mover o espaço para cima, para baixo, esquerda ou direita.
- **Teste de meta:** dado na figura
- **Custo do caminho:** 1 para cada movimento