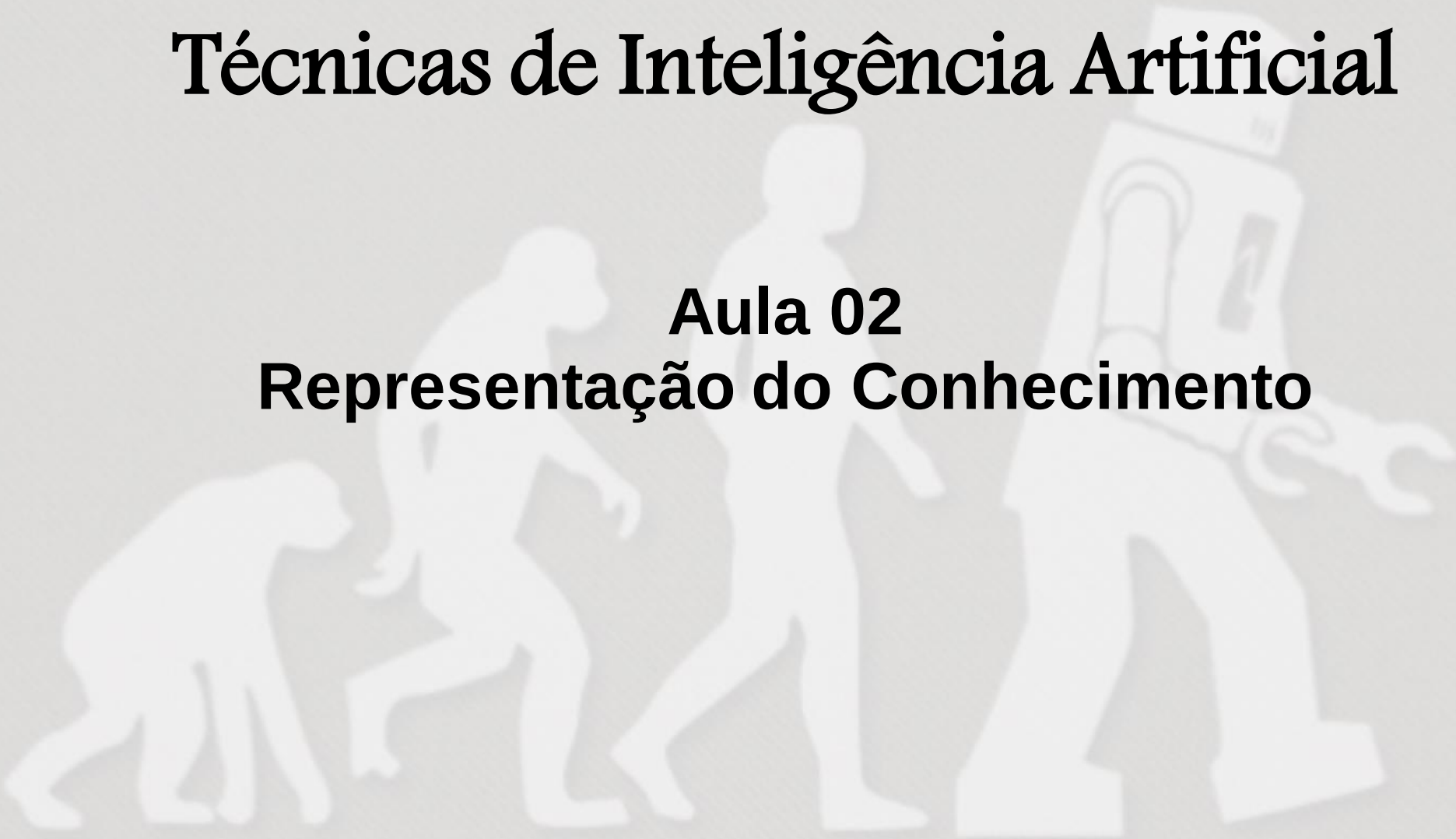
Universidade do Sul de Santa Catarina
Ciência da Computação

Técnicas de Inteligência Artificial



Aula 02
Representação do Conhecimento

Prof. Max Pereira

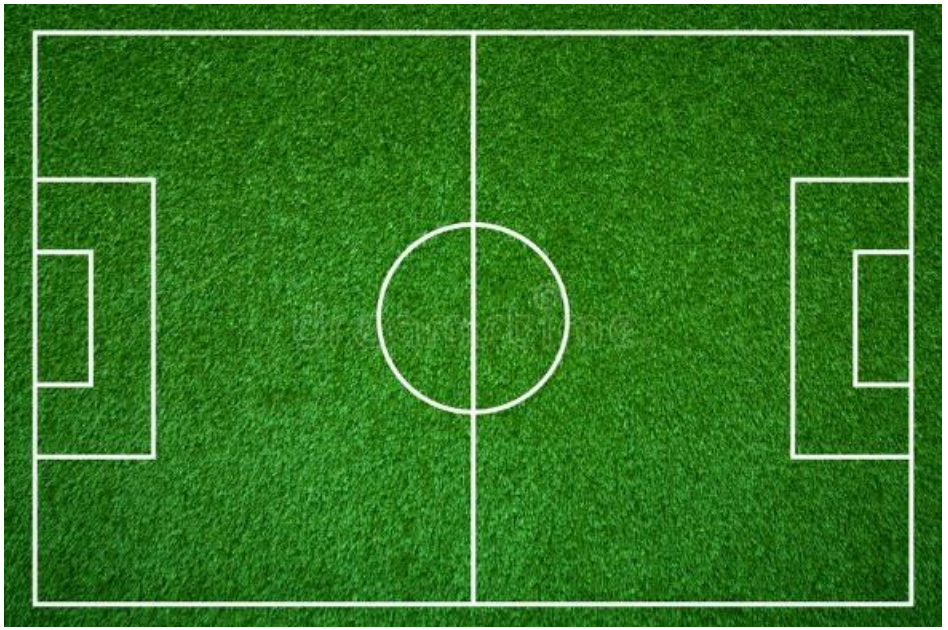
- Para que um computador possa solucionar um problema relacionado ao mundo real, ele primeiro precisa de um **meio para representar** o mundo real internamente.

- O modo pelo qual o computador representa um problema, **as variáveis que ele usa e os operadores que aplica a essas variáveis** podem fazer a diferença entre um algoritmo eficiente e um algoritmo que não funciona.

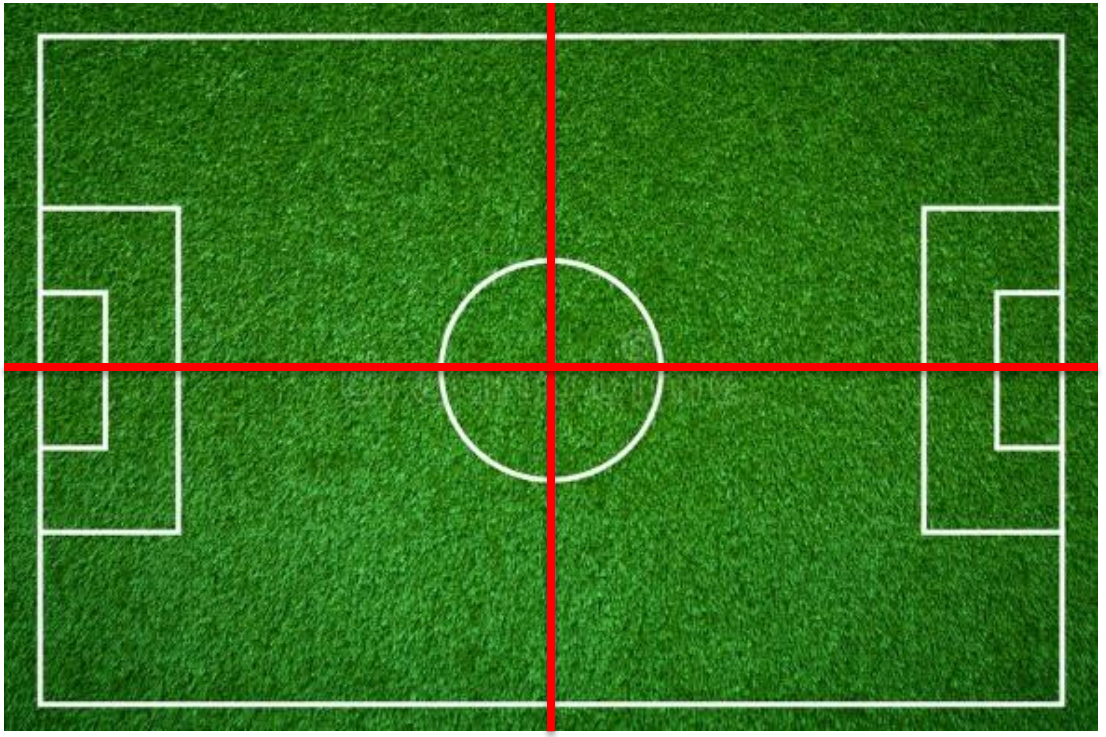
- A questão mais difícil é **determinar a estrutura de dados** que será utilizada para representar o problema que estamos explorando.

Imagine que você esteja procurando uma lente de contato que caiu em um campo de futebol. Provavelmente utilizaria algum **conhecimento** sobre aonde você estava no campo para ajudar a procurá-la. Se você gastar tempo apenas em uma metade do campo, não precisará gastar tempo procurando na outra metade.

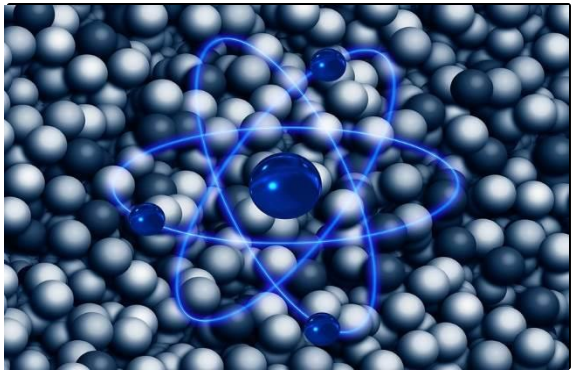
Vamos supor que um computador esteja procurando a lente de contato no campo, baseado em **informações** fornecidas por você. Agora precisamos escolher **uma representação** para o computador utilizar, de modo que ele possa formular as questões corretas para perguntar.



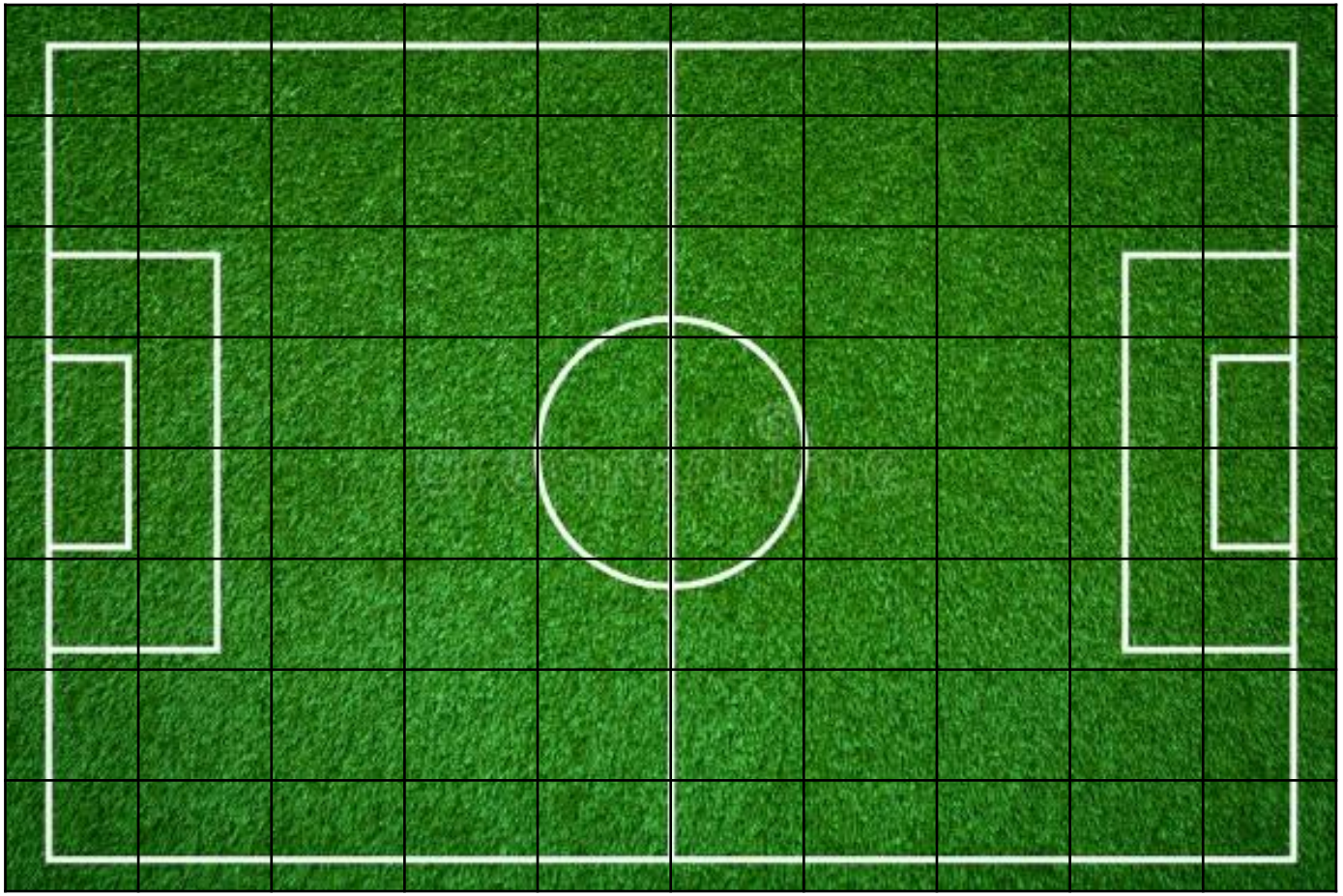
Uma representação seria dividir o campo em quatro quadrados iguais e o computador perguntar em qual **quadrado possivelmente a lente foi perdida**. Isto identificará a posição da lente no campo, mas ainda não será muito útil, pois há ainda uma **grande área para procurar**.



Outra representação seria produzir uma **grade (matriz) contendo uma representação de cada átomo do campo**. Para cada átomo, o computador poderia perguntar, “a lente está associada a esse átomo?”. Essa seria uma resposta muito precisa, mas seria um modo ineficiente de encontrar a lente. Precisaríamos de **muito poder computacional**.



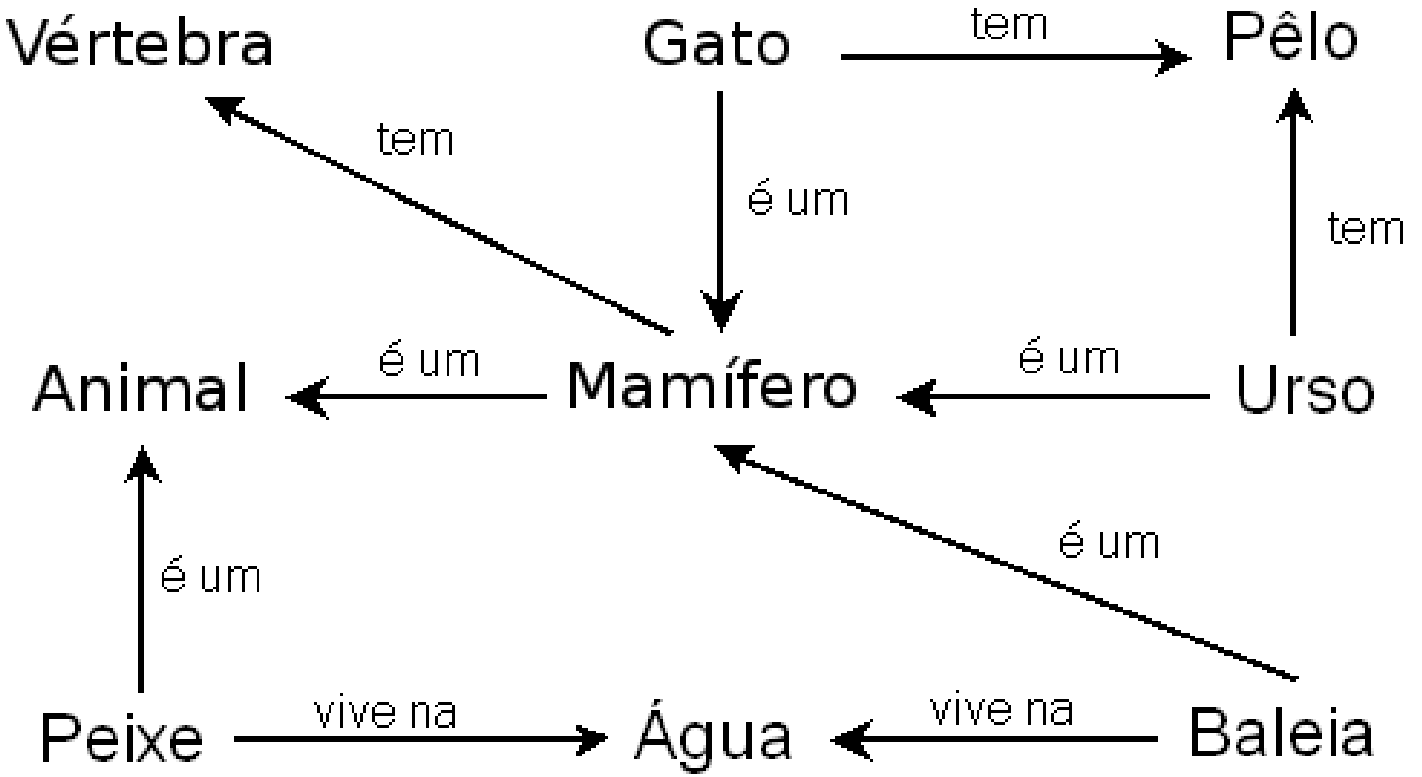
Talvez uma representação melhor seja dividir o campo em uma **grade (matriz) na qual cada quadrado tenha um centímetro por um centímetro e eliminar** todos os quadrados que você sabe que não estão perto de onde você estava quando perdeu a lente.



- ✓ Na verdade todas as representações descritas são idênticas, exceto pelos diferentes **níveis de granularidade**.
- ✓ Ao aplicar Inteligência Artificial a problemas de busca, é essencial uma representação **útil**, **eficiente** e **significativa**.

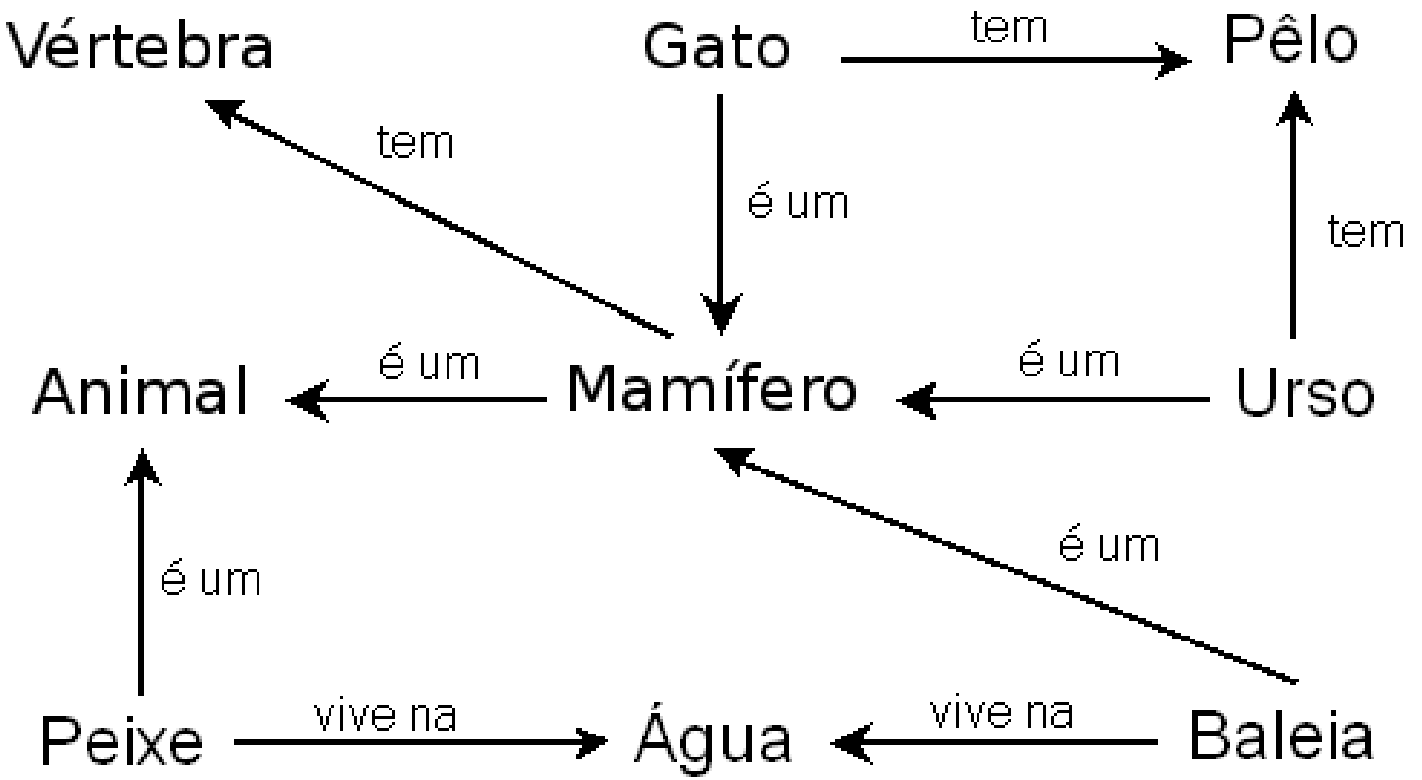
Redes Semânticas

É uma representação comumente utilizada em Inteligência Artificial. Uma rede semântica é uma rede consistindo em **vértices** que são conectados por **arestas**. Os vértices representam objetos e as ligações (arestas) entre os vértices representam relacionamentos entre estes objetos.



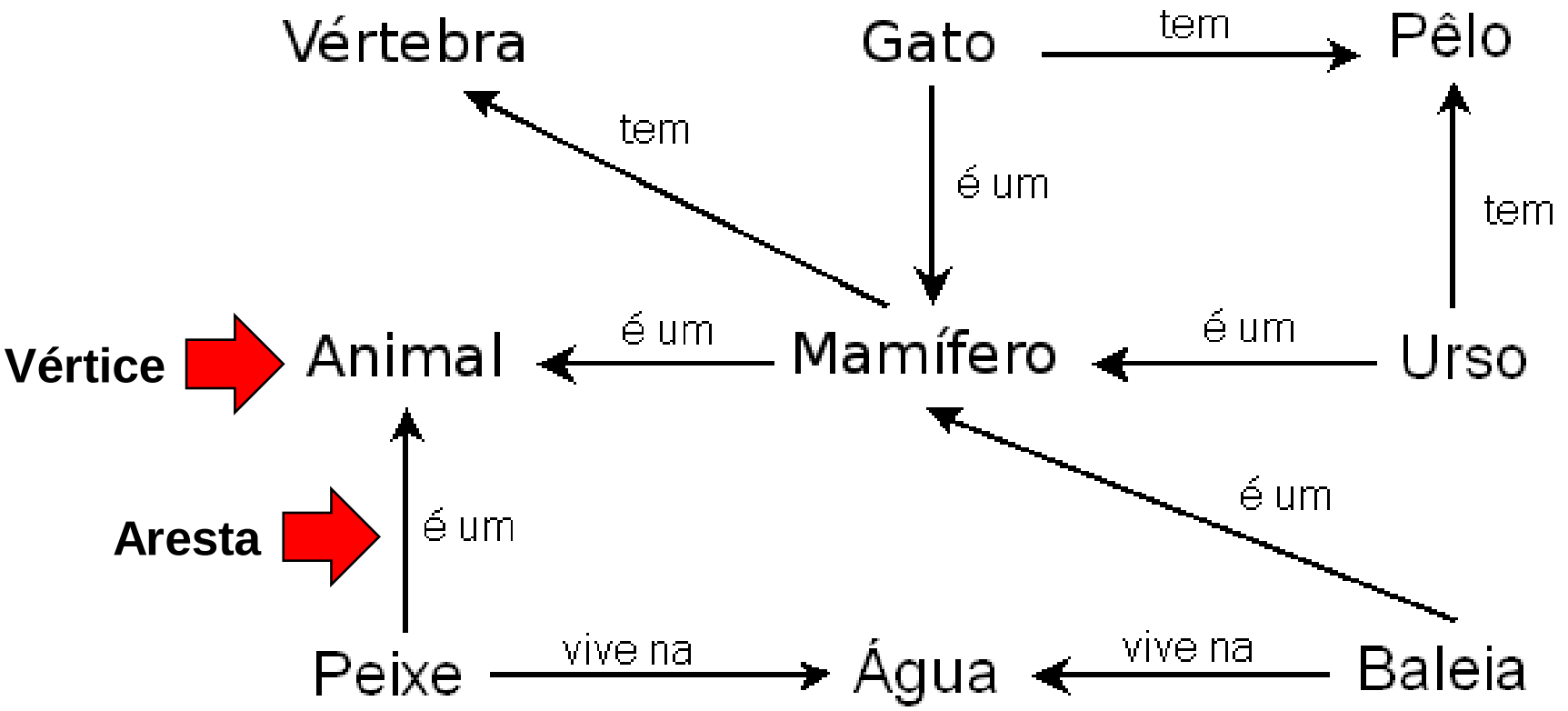
Redes Semânticas

Uma importante característica das redes semânticas é que elas transmitem **significado**. Isto é, relacionamento entre vértices e arestas na rede, transmite **informação** sobre uma situação do mundo real.



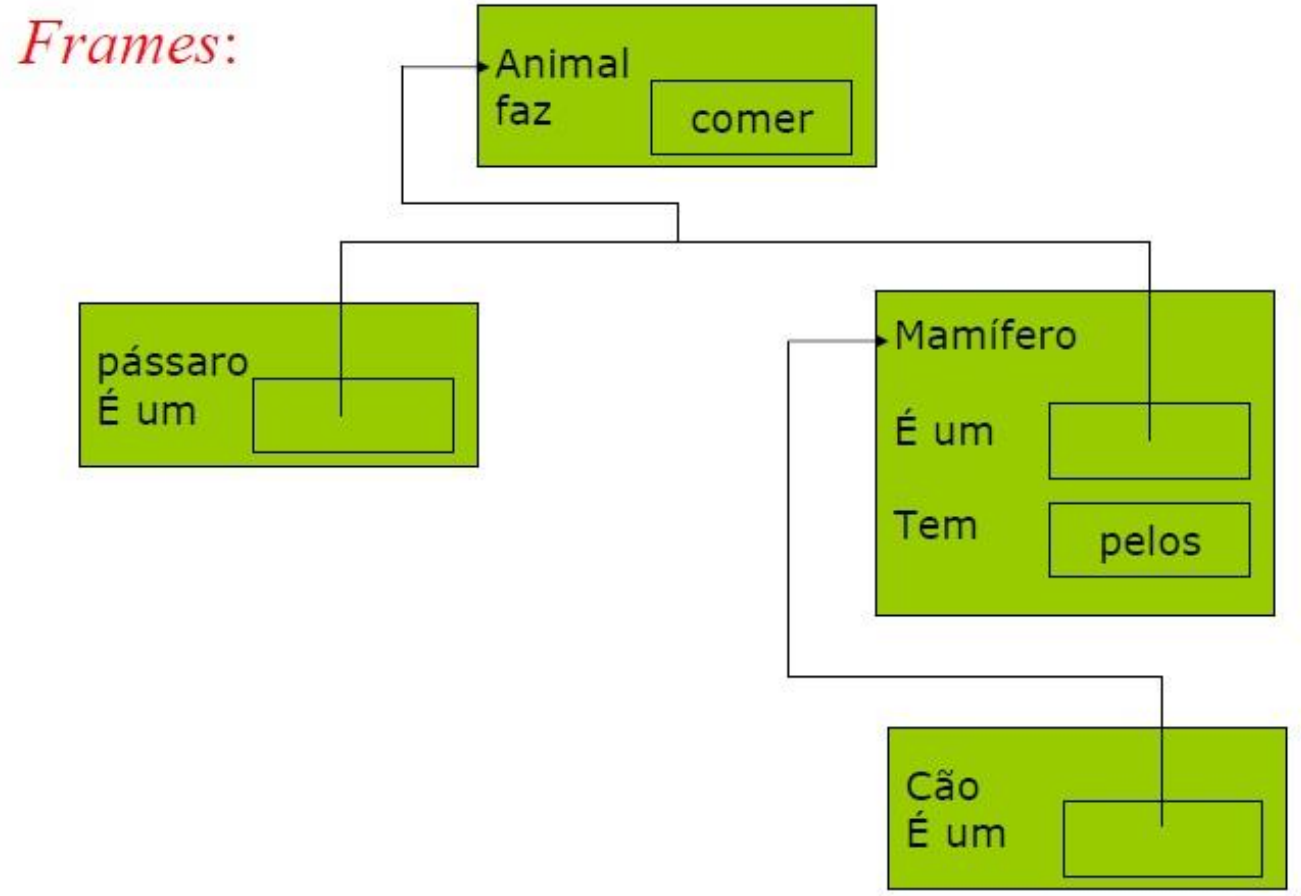
Redes Semânticas

Cada vértice em uma rede semântica tem um rótulo que identifica o que ele representa. Arestas também são rotuladas. Arestas representam conexões ou relacionamentos entre vértices.



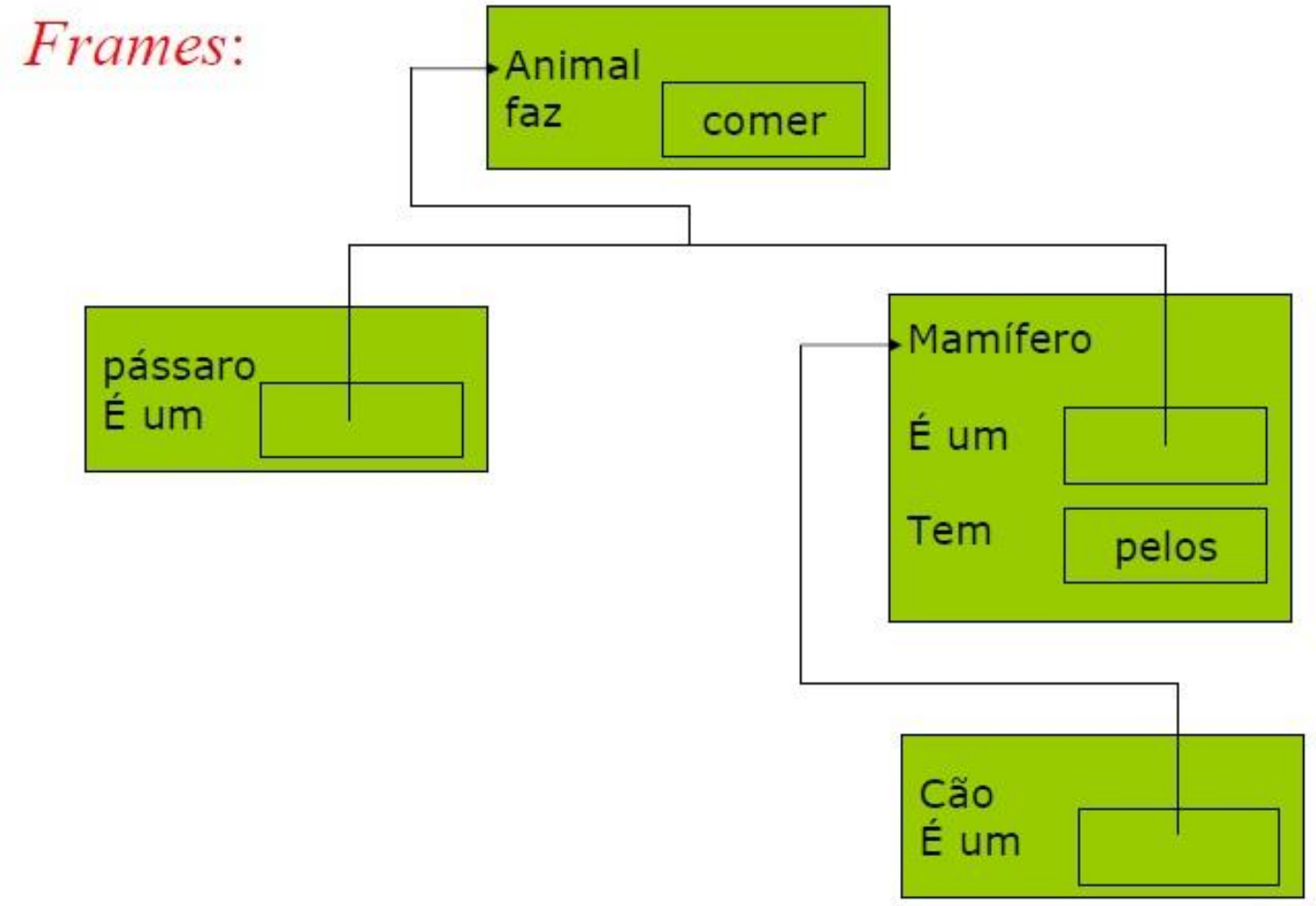
Quadros (*Frames*)

Representação baseada em quadros é um desenvolvimento de redes semânticas e nos permite expressar a ideia de **herança**. Um **sistema de quadros** consiste em um conjunto de **quadros** (ou vértices) que são interligados por relações. Cada quadro descreve uma **instância** ou uma **classe**.



Quadros (*Frames*)

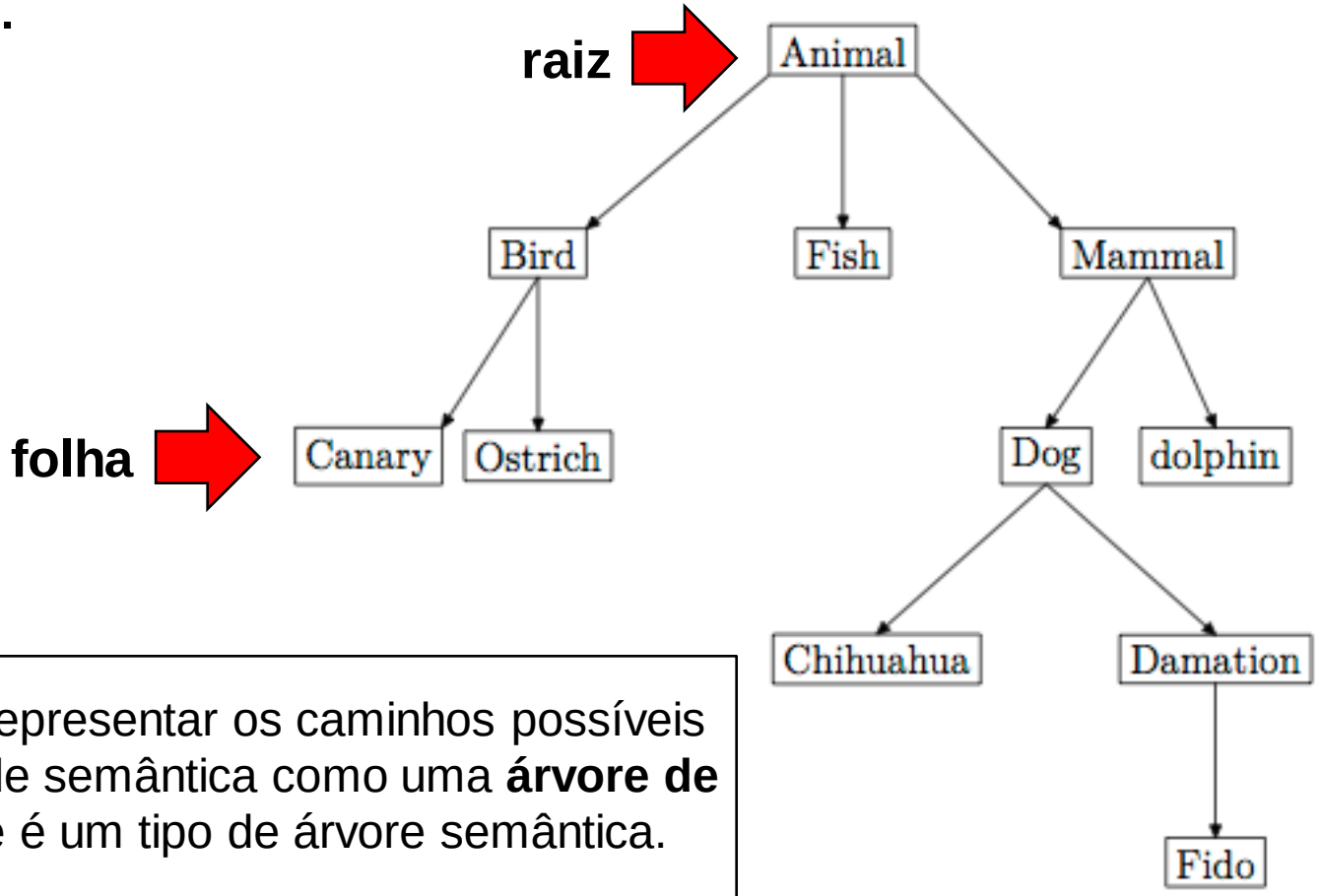
Quando representamos que "*pássaro*" é um "*Animal*" queremos expressar que "**pássaro é uma instância da classe Animal**". Esse relacionamento também e conhecido como **generalização**.



Árvores Semânticas

Uma árvore semântica é um tipo de rede semântica que tem as seguintes propriedades:

- ✓ Cada nó (exceto o nó raiz) tem exatamente um **predecessor** (pai) e um ou mais **sucessores** (filhos).
- ✓ Alguns nós não têm sucessores. Esses nós são chamados de **folhas**.



Podemos representar os caminhos possíveis de uma rede semântica como uma **árvore de busca**, que é um tipo de árvore semântica.

Espaços de Busca

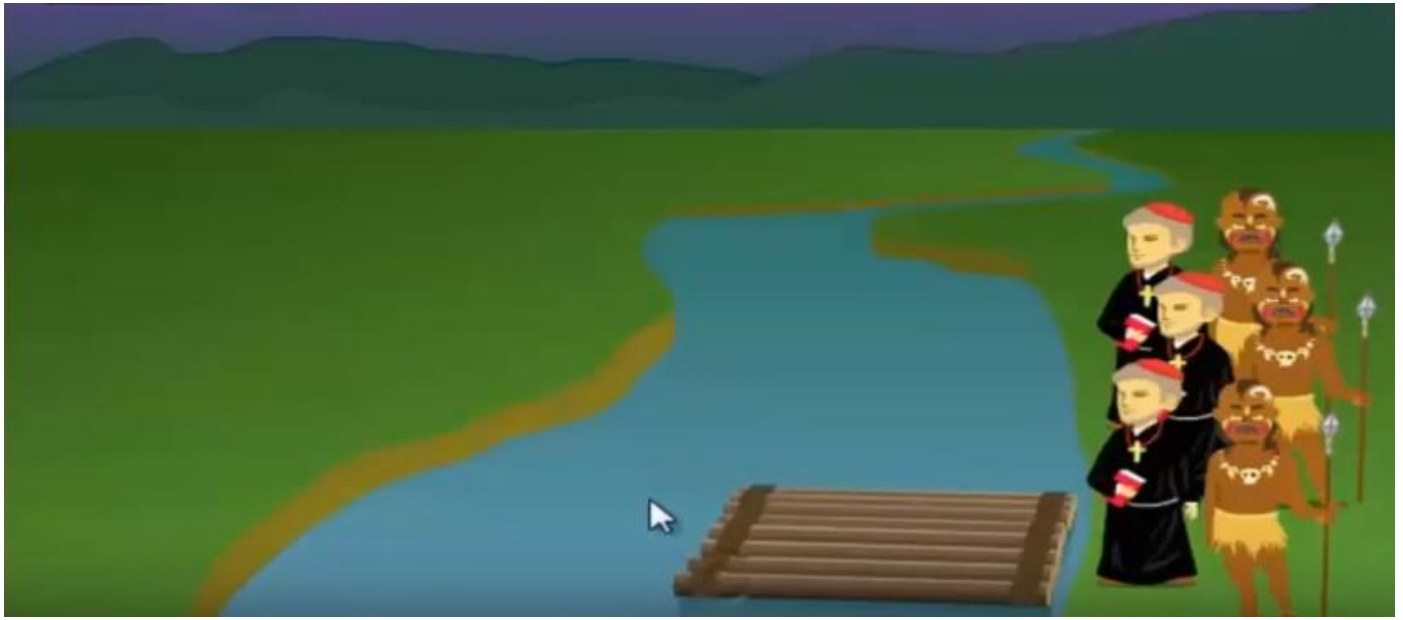
- Um espaço de busca é uma representação do **conjunto de possíveis escolhas** de um dado problema, uma ou mais das quais é **solução do problema**.
- Por exemplo, ao tentar localizar uma palavra específica em um dicionário com 100 páginas, um espaço de busca consistirá em cada uma das 100 páginas. A página que estiver sendo buscada é chamada de **alvo** e pode ser identificada verificando-se se a palavra que estamos procurando está ou não nela.
- O objetivo da maioria dos procedimentos de busca é identificar um ou mais alvos e, geralmente, identificar um ou mais caminhos até estes alvos (frequentemente, o **caminho mais curto** ou o **caminho de menor custo**).
- Devido a um espaço de busca consistir em um conjunto de estados conectados por caminhos que representam ações, eles também são chamados de **espaço de estados**.

Exemplo 1: Missionários e Canibais

Um problema bem conhecido que é frequentemente utilizado para ilustrar técnicas de IA.

Três missionários e três canibais estão em uma das margens de um rio, com uma canoa. Todos querem ir para a outra margem do rio. A canoa somente pode transportar uma ou duas pessoas de cada vez. Não poderá haver, em qualquer momento, mais canibais que missionários em ambas as margens do rio, pois isto poderia resultar em os missionários serem devorados.

Para resolver esse problema precisamos utilizar uma representação adequada.



Em primeiro lugar, para a solução do problema, podemos considerar **um estado** como consistindo no número exato de canibais e no número exato de missionários em cada margem do rio, com a canoa em uma margem ou na outra. Poderíamos representar isto, por exemplo, como:

3,3,1 0,0,0

O conjunto de números à esquerda representa o número de canibais, missionários e canoa em uma das margens do rio e o conjunto à direita, o que está na outra margem. Como os números em uma margem são completamente dependentes dos números que estão na outra margem, podemos utilizar **apenas um conjunto de números**, e mostrar quantos de cada estão na margem de destino, ou seja, o **estado inicial** do problema é representado por

0,0,0

e o **estado objetivo** é

3,3,1

Um exemplo de estado que deve ser evitado é

2,1,1

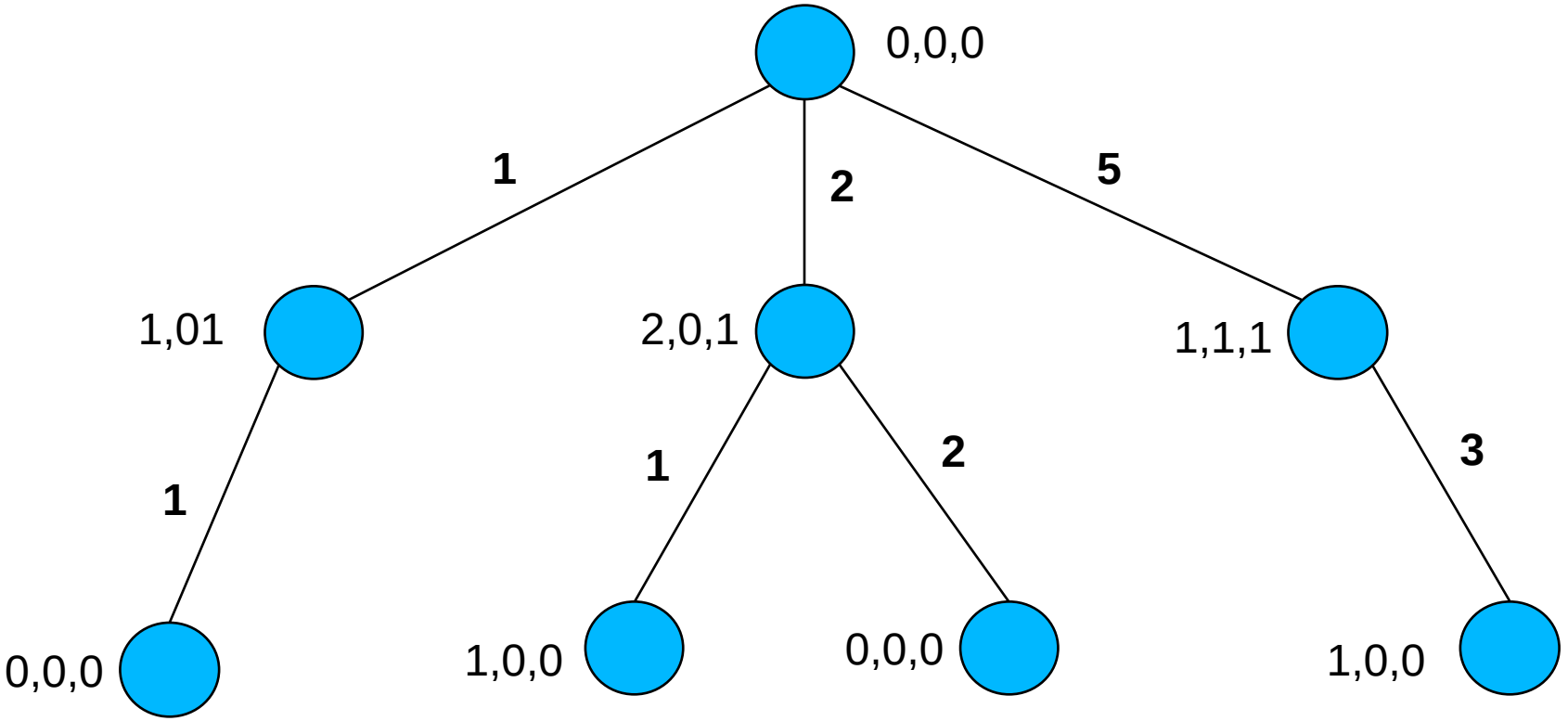
Aqui, são dois canibais, uma canoa e apenas um missionário em uma das margens do rio.

Para ir de um estado a outro, devemos **aplicar um operador**. Os operadores que temos disponíveis são os seguintes:

1. Levar um canibal para a outra margem
2. Levar dois canibais para a outra margem
3. Levar um missionário para a outra margem
4. Levar dois missionários para a outra margem
5. Levar um canibal e um missionário para a outra margem

Então se aplicássemos o operador 5 ao estado representado por 1,1,0 resultaria no estado 2,2,1. Um canibal, um missionário e a canoa deslocaram-se agora para a outra margem. Aplicar o operador 3 a este estado levaria a um estado ilegal: 2,1,0.

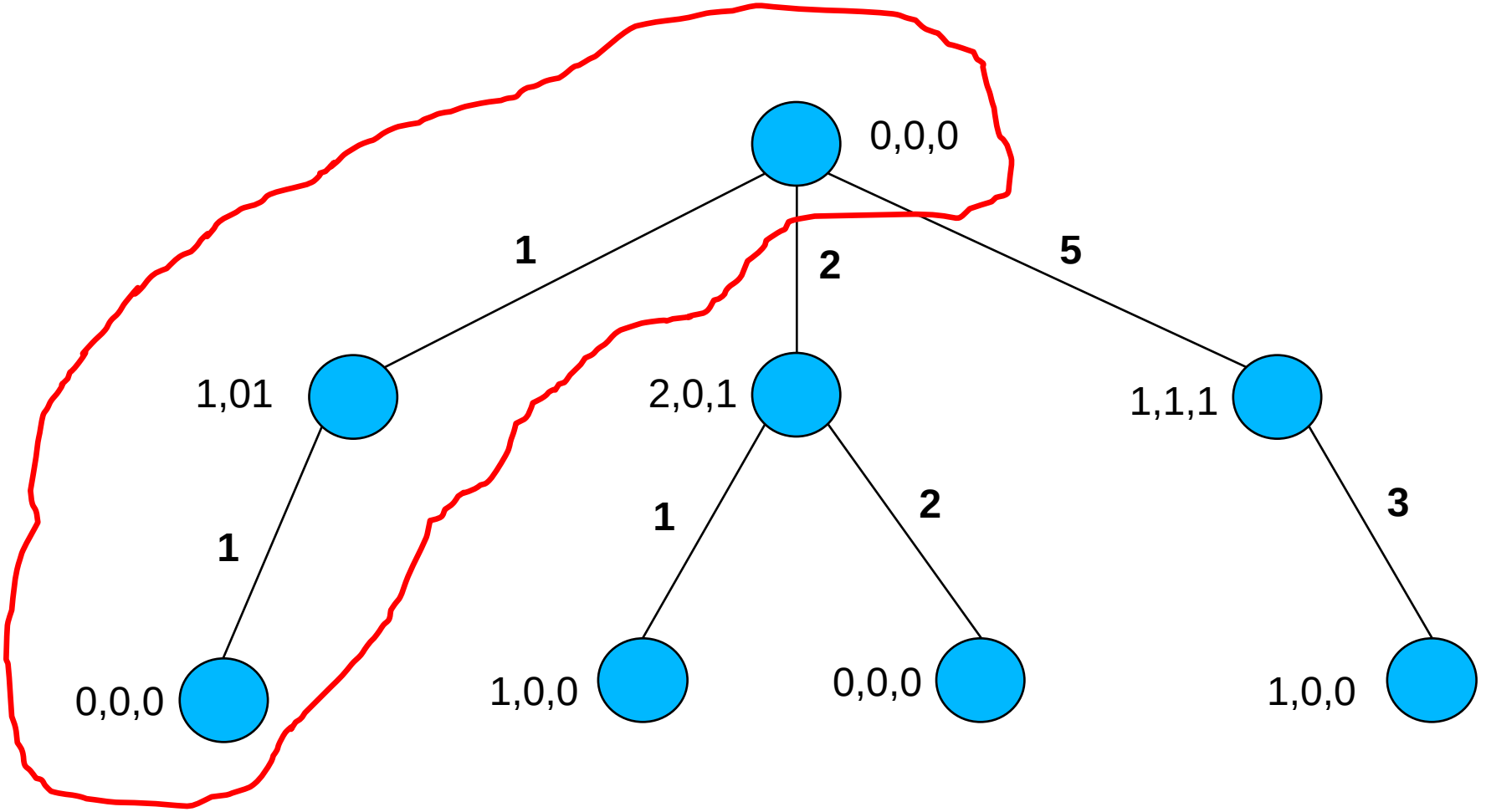
Consideraremos o custo do caminho escolhido como sendo o número de passos que foram dados ou o número de vezes que um operador foi aplicado.



Os três primeiros níveis da **árvore de busca** para o problema dos missionários e canibais (as ligações estão marcadas com o operador que foi aplicado).

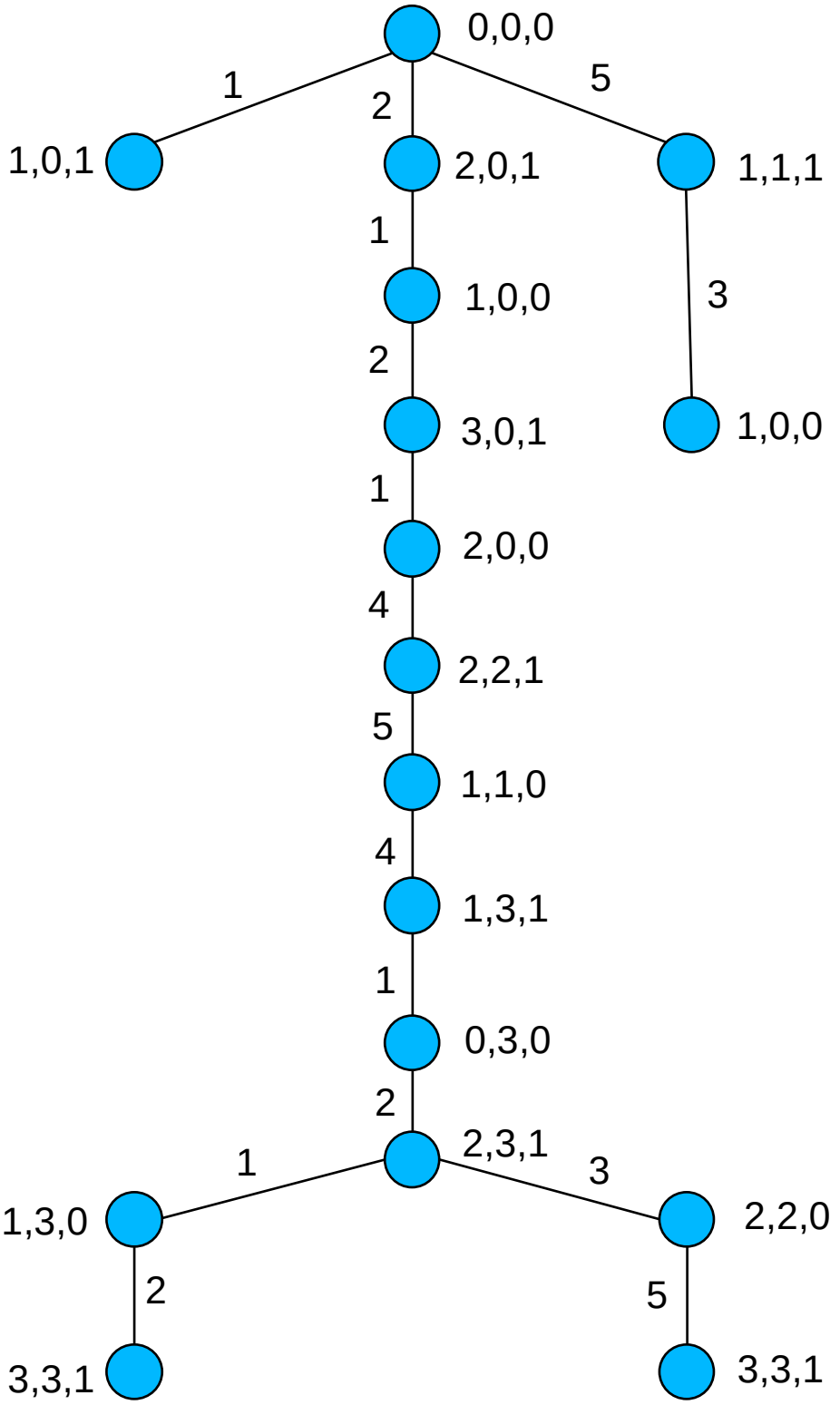
Agora, estendendo essa árvore para incluir todos os caminhos possíveis e os estados por eles atingidos, pode-se encontrar uma solução. Uma solução para o problema seria representada como um caminho da raiz até o alvo.

Esta árvore representa a presença de **ciclos** no espaço de estados. Aplicar o operador 1 (levando um canibal para a outra margem) como primeira ação e, em seguida, aplicá-lo novamente, retornará ao **estado inicial**. Este é um modo perfeitamente válido de tentar resolver o problema, mas não muito eficiente. Uma representação mais efetiva para o problema não deveria incluir ciclos.



Agora temos uma **versão estendida da árvore de busca** para o problema, que omite ciclos e inclui alvos. Observe que foi omitido a maioria dos estados repetidos. Além de evitar ciclos, **caminhos subótimos** também foram removidos da árvore. Se um caminho de comprimento 2 atinge um estado específico e outro caminho de comprimento 3 também atinge aquele estado, não há interesse em seguir o caminho mais longo.
Assim, os dois caminhos que podem ser seguidos até o alvo são rotas mais curtas (os **caminhos de menor custo**), mas não são, de forma alguma, os únicos caminhos. Existem muito caminhos mais longos.

Ao escolher uma representação adequada, somos capazes de melhorar a **eficiência do método de busca**.

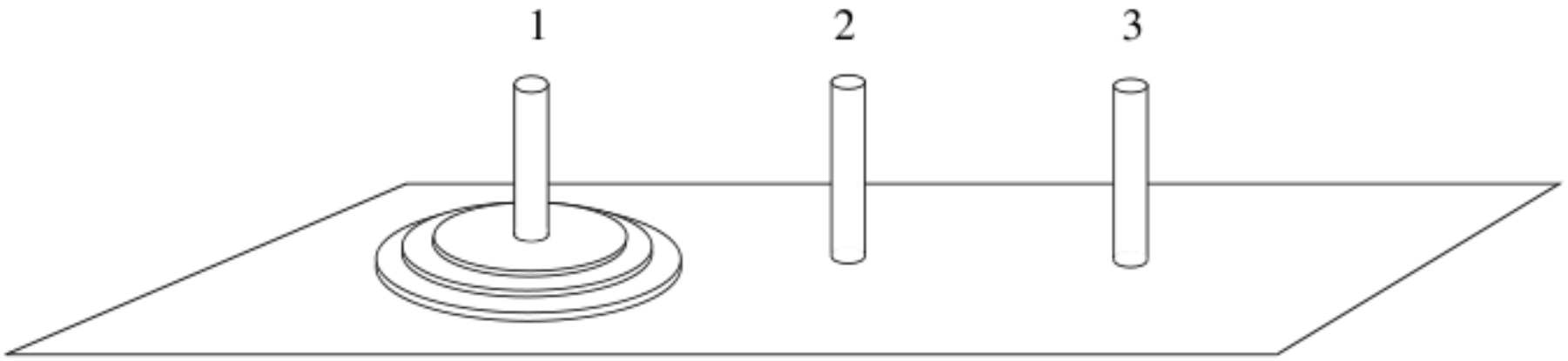


Resolver o problema dos Missionários e Canibais envolve **buscar** na árvore de busca. **Busca** é um método extremamente útil para solucionar problemas e é amplamente utilizado em Inteligência Artificial.

Exemplo 2: As Torres de Hanói

Eis a definição do problema das Torres de Hanói:

Temos três pinos e diversos discos de tamanhos diferentes. O objetivo é partir do estado inicial, no qual todos os discos estão no primeiro pino, por ordem de tamanho (o menor em cima) e chegar ao estado objetivo, no qual todos os discos estão no terceiro pino, também por ordem de tamanho. Podemos mover um disco de cada vez, desde que não haja discos sobre ele e desde que ele não seja movido para cima de um disco que seja menor que ele.



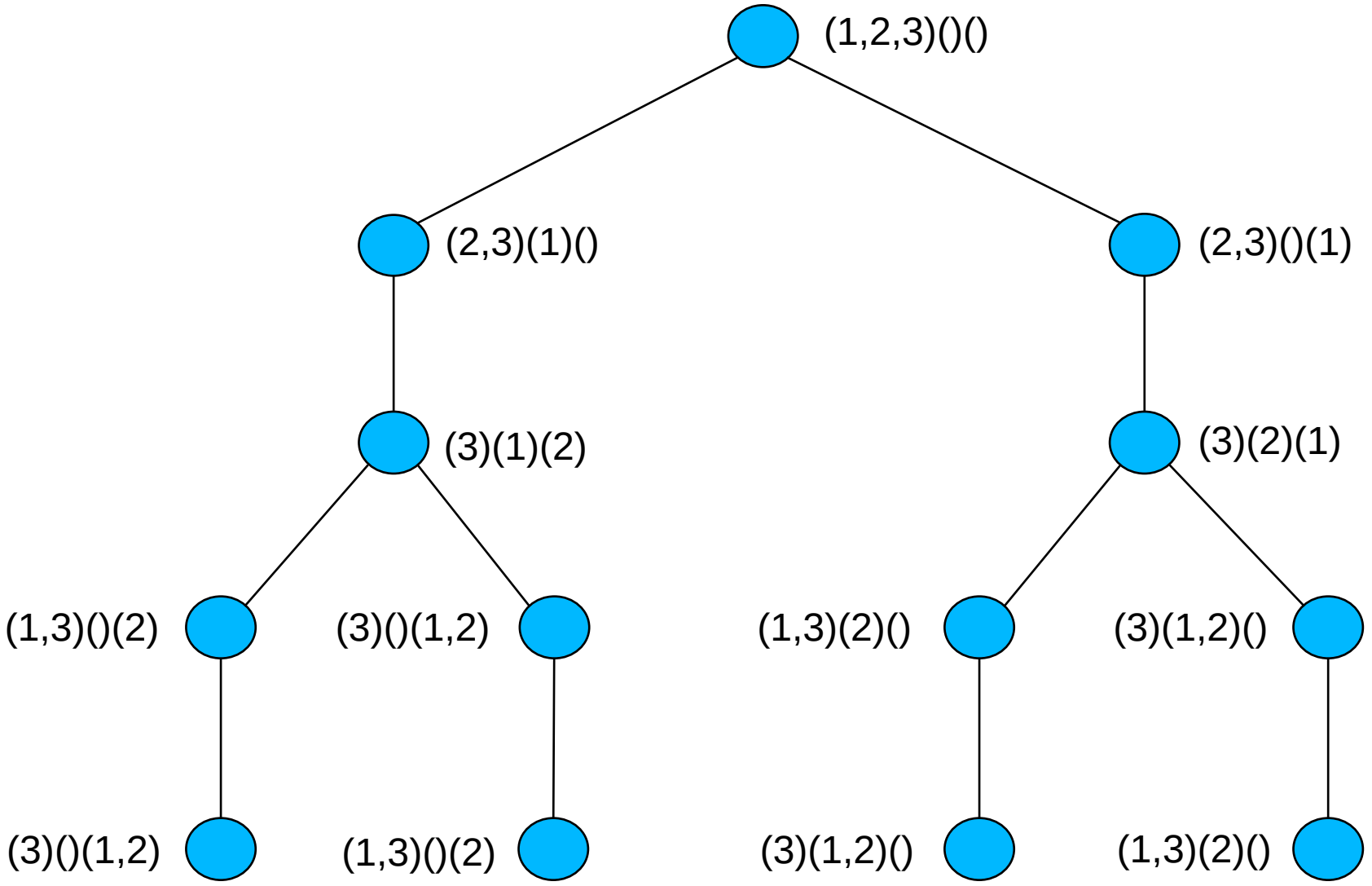
Tendo conhecimento dos estados **inicial** e **objetivo**, precisamos apresentar um conjunto de operadores:

1. Mover disco do pino 1 para o pino 2
2. Mover disco do pino 1 para o pino 3
3. Mover disco do pino 2 para o pino 1
4. Mover disco do pino 2 para o pino 3
5. Mover disco do pino 3 para o pino 1
6. Mover disco do pino 3 para o pino 2

Também precisamos de um modo para **representar cada estado**. Para esse exemplo, vamos utilizar **vetores** de números, nos quais 1 representa o menor disco e 3, o maior disco. O primeiro vetor representa o primeiro pino e assim por diante. Assim o estado inicial é representado por **(1,2,3)()()**.

Por exemplo, o estado mostrado na figura é representado por (2,3)()(1), aplicando o operador 2. O **estado objetivo** é ()()(1,2,3)

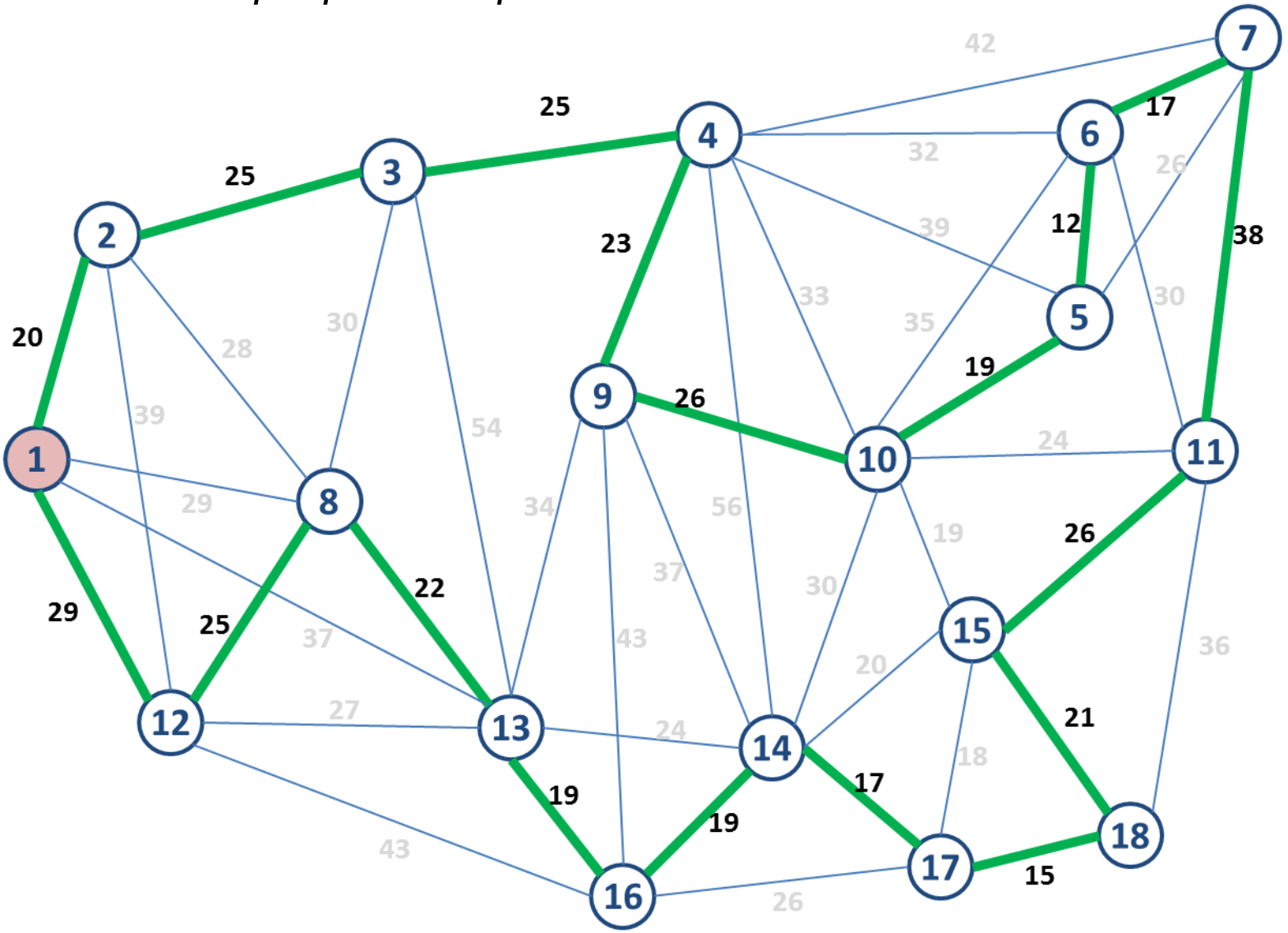
Os primeiros cinco níveis da árvore de busca do problema das Torres de Hanói com três discos. Mais uma vez, ignoramos **caminhos cíclicos**.



Exemplo 3: O Caixeiro Viajante

É outro caso clássico da Inteligência Artificial.

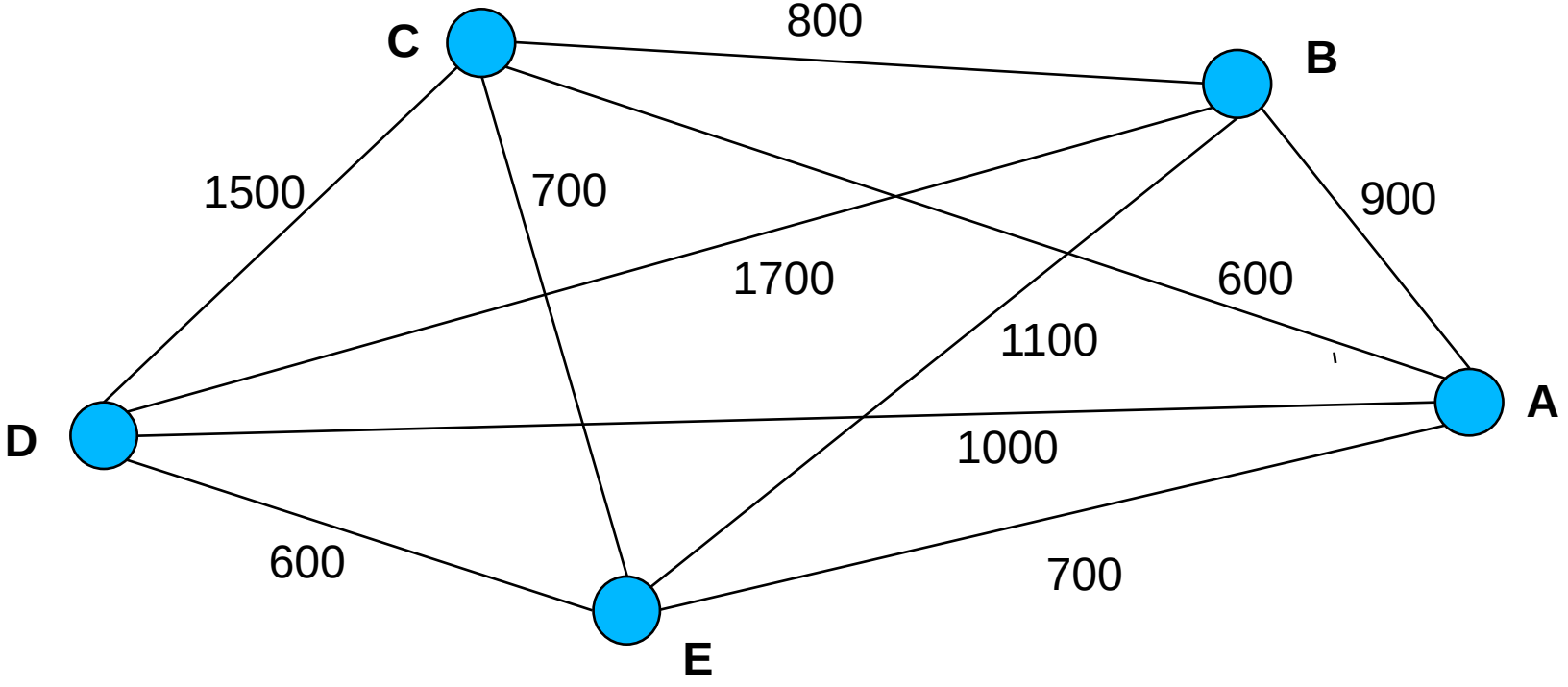
Um caixeiro viajante deve visitar cada uma das cidades de um conjunto de cidades e retornar a cidade de partida. O objetivo do problema é encontrar o caminho mais curto que permita que ele visite cada uma das cidades.

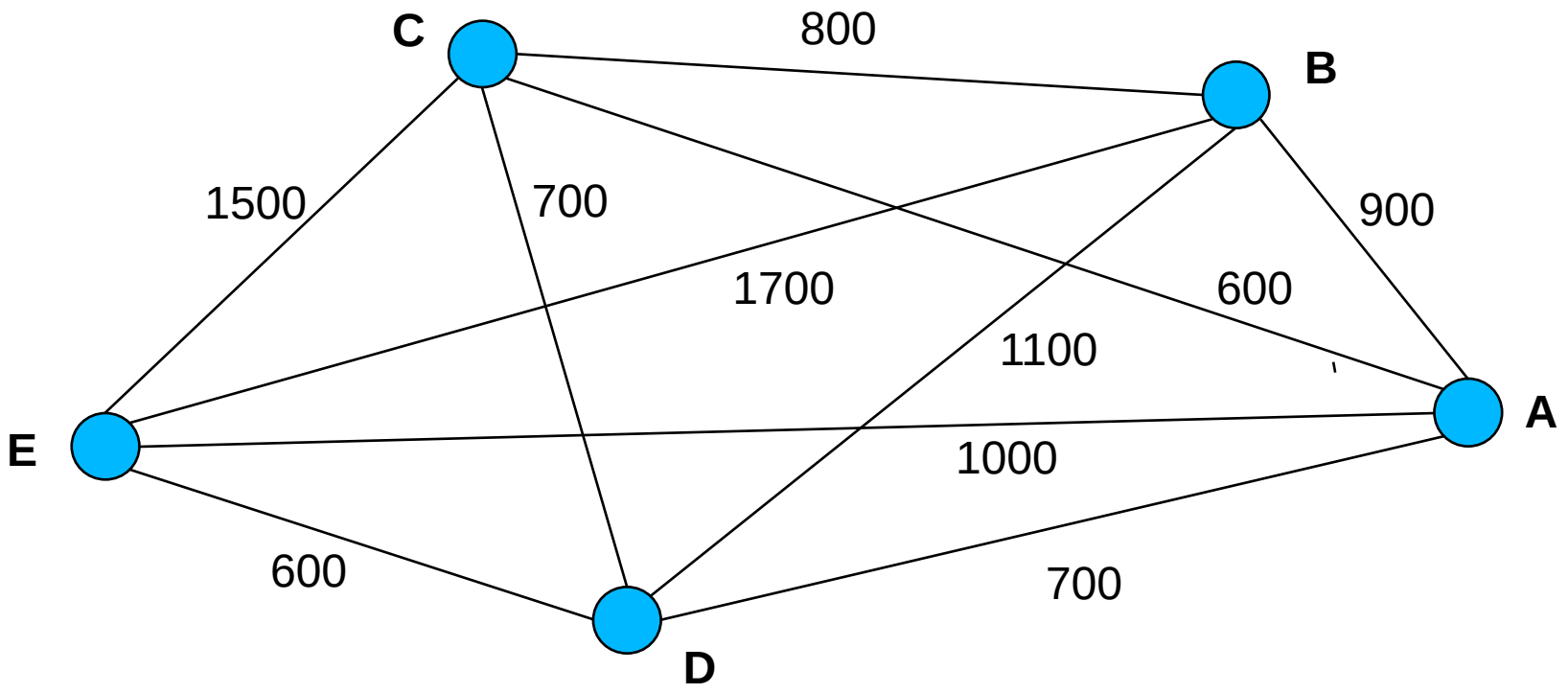


Vamos supor que o caixeiro viajante esteja percorrendo as seguintes cidades:

A Florianópolis
B Joinville
C São Paulo
D Rio de Janeiro
E Belo Horizonte

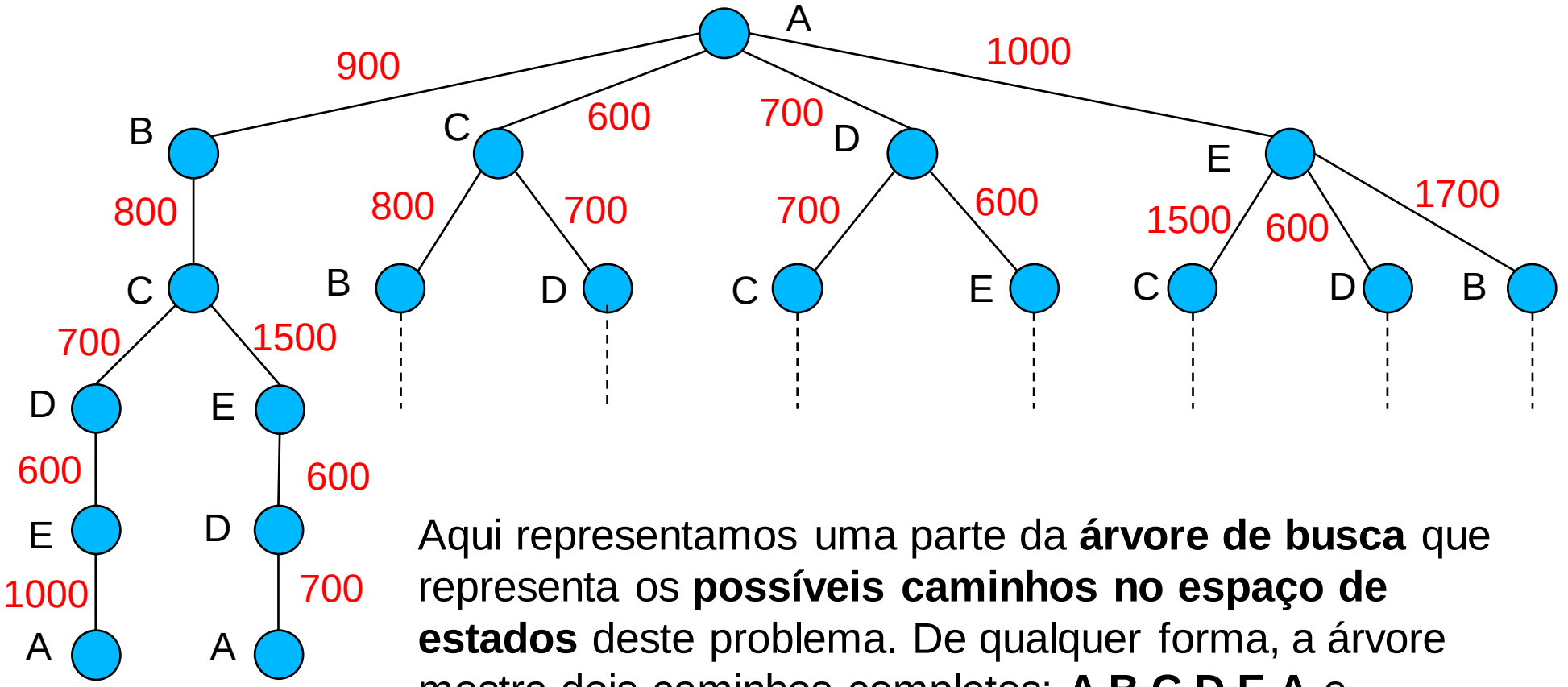
Ele mora em Florianópolis e deve visitar todas as outras quatro cidades antes de retornar para casa. Vamos imaginar que esse caixeiro viajante esteja viajando de avião e que o custo de cada voo seja diretamente proporcional à distância percorrida e que haja voos diretos entre qualquer par de cidades.





Podemos utilizar o mapa para tentar solucionar o problema. Certamente podemos utilizá-lo para encontrar **possíveis caminhos**: um possível caminho é **A,B,C,E,D,A** que tem comprimento de 4.500 quilômetros.

Para solucionar o problema uma representação será necessária.



Aqui representamos uma parte da **árvore de busca** que representa os **possíveis caminhos no espaço de estados** deste problema. De qualquer forma, a árvore mostra dois caminhos completos: **A,B,C,D,E,A** e **A,B,C,E,D,A**. O gasto total desses dois caminhos é de 4.000 quilômetros e 4.500 quilômetros, respectivamente.

Ao todo existirão $(n - 1)!$ caminhos possíveis para o problema do caixeiro viajante com n cidades. Para problemas com um pequeno número de cidades, como 5 ou mesmo 10, isto significa que a árvore de busca completa pode ser avaliada por um programa sem maiores dificuldades, mas se o problema consistir em 40 cidades, seriam cerca de 10^{48} caminhos possíveis. Um número absurdamente grande.