



*Escola Politécnica da USP - Depto. de Enga. Mecatrônica*

# PMR-3510 Inteligência Artificial

## Aula 5- Resolução de problemas por máquinas

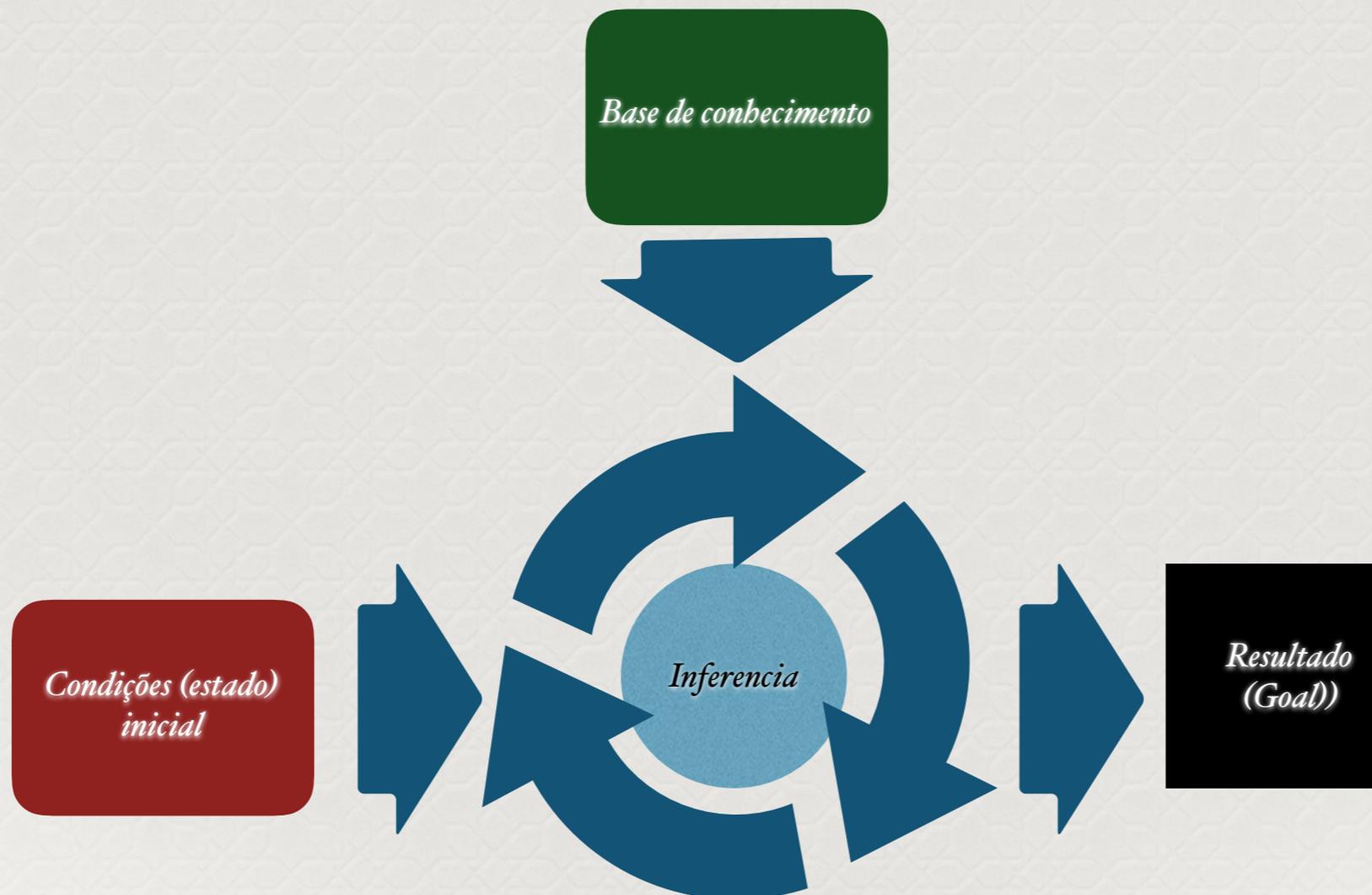
*Prof. José Reinaldo Silva*

*[reinaldo@usp.br](mailto:reinaldo@usp.br)*





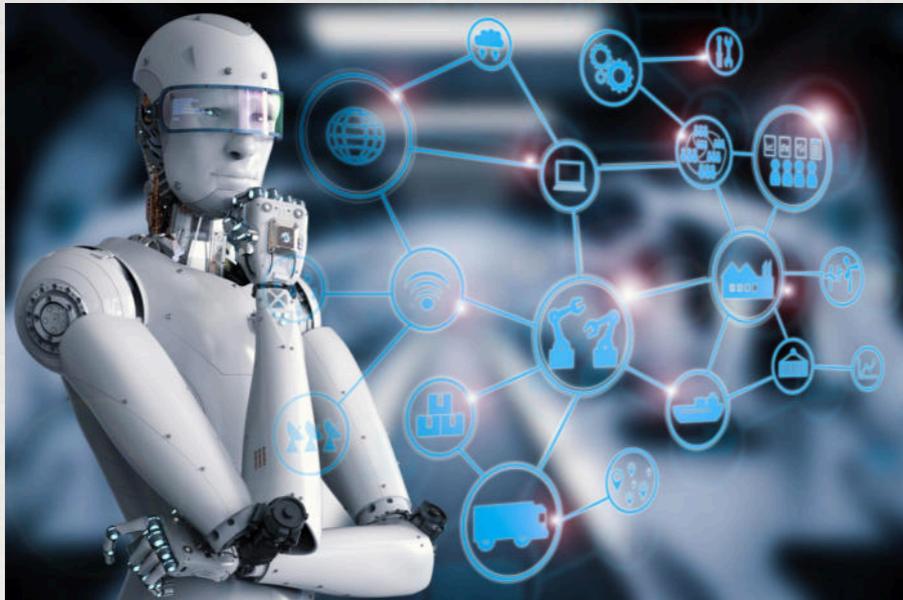
*Em uma primeira abordagem, gostaríamos de ter “agentes inteligentes” capazes de “resolver problemas”. O que significa isso?*





## *Estrutura de um “resolvedor automático de problemas”*

*Para dotar uma máquina da capacidade de resolver problemas (ou uma classe de problemas) é preciso ter uma estrutura com os seguintes atributos:*



1. uma descrição clara do “estado inicial” ou seja das condições iniciais do problema a ser resolvido;
2. uma descrição clara do objetivo ou “estado final”, de modo que seja possível saber quando (e se) o problema foi resolvido;
3. em cada estágio do processo de solução saber quais os próximos estados que podem ser atingidos;
4. poder escolher um (ou o melhor) caminho entre os estados acima;
5. saber que operadores (ou passos) aplicar para fazer a “transição” para um próximo estado;
6. discernir se estamos convergindo para a solução.



*Vamos agora pensar na resolução de problemas  
por uma máquina!*

*Usando ainda o paradigma estado/transição, devemos,  
além de especificar claramente os estados inicial e final,  
saber agora quais são os demais estados e como se  
relacionam, isto é, o espaço de estados. O problema pode ser  
então reduzido a uma busca por um caminho neste espaço  
que leve do estado inicial ao estado final.*



## **STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving<sup>1</sup>**

**Richard E. Fikes**

**Nils J. Nilsson**

*Stanford Research Institute, Menlo Park, California*

Recommended by B. Raphael

Presented at the 2nd IJCAI, Imperial College, London, England, September 1-3, 1971.

---

### **ABSTRACT**

*We describe a new problem solver called STRIPS that attempts to find a sequence of operators in a space of world models to transform a given initial world model into a model in which a given goal formula can be proven to be true. STRIPS represents a world model as an arbitrary collection of first-order predicate calculus formulas and is designed to work with models consisting of large numbers of formulas. It employs a resolution theorem prover to answer questions of particular models and uses means-ends analysis to guide it to the desired goal-satisfying model.*

### **DESCRIPTIVE TERMS**

Problem solving, theorem proving, robot planning, heuristic search.

---

### **1. Introduction**

This paper describes a new problem-solving program called STRIPS (*Stanford Research Institute Problem Solver*). An initial version of the program has been implemented in LISP on a PDP-10 and is being used in conjunction with robot research at SRI. STRIPS is a member of the class of problem solvers that search a space of "world models" to find one in which a given goal is achieved. For any world model, we assume that there exists a set

<sup>1</sup> The research reported herein was sponsored by the Advanced Research Projects Agency and the National Aeronautics and Space Administration under Contract NAS12-2221.

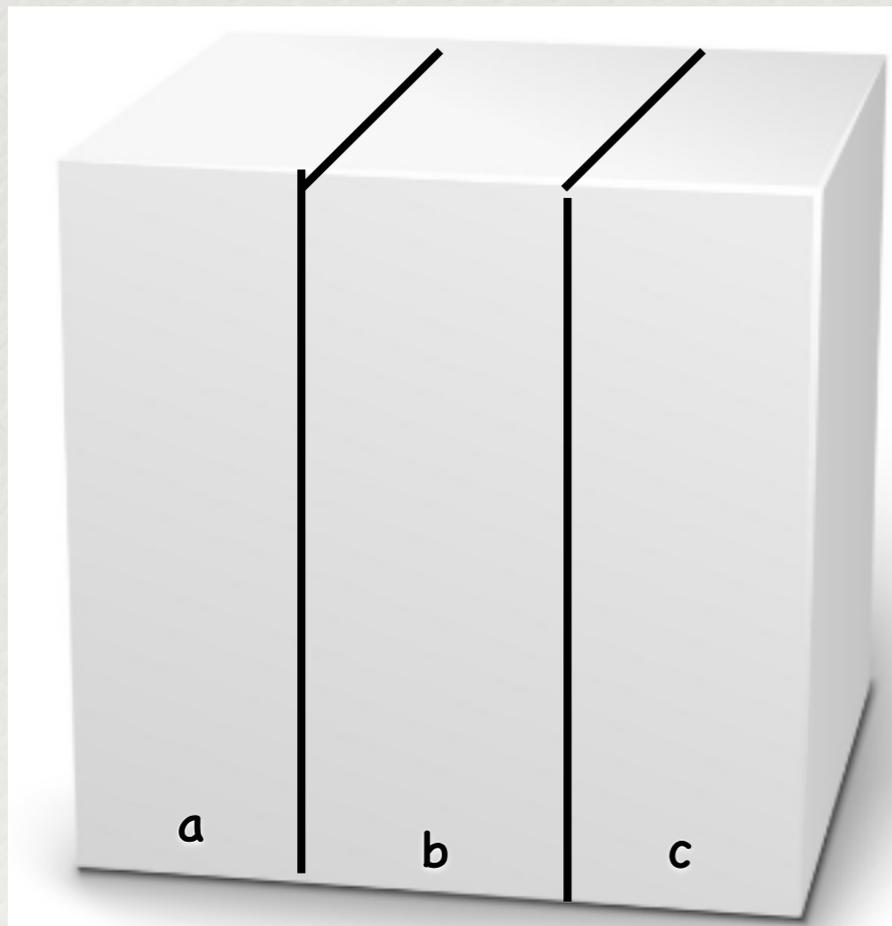
*Artificial Intelligence* 2 (1971), 189-208

Copyright © 1971 by North-Holland Publishing Company

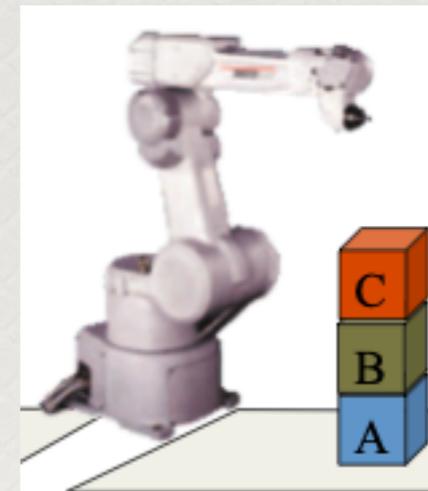


## STRIPS Basic Concepts

O domínio é uma descrição dos elementos que devem ser manipulados e suas restrições: vamos tomar como exemplo o "mundo de blocos"...



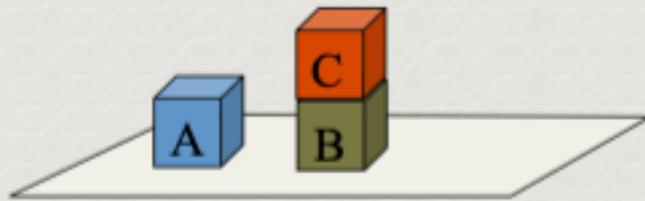
O mundo é composto de três colunas onde é possível acumular até 3 "blocos". Existem no total 3 blocos plenamente identificados. Existe também um robô que pode manipular um bloco de cada vez e só pode pegar o bloco no topo da pilha.





## STRIPS Basic Concepts

O problema de planejamento (planning problem) é dado pela definição de estado do problema e pelas ações e operadores que mudam este estado ...

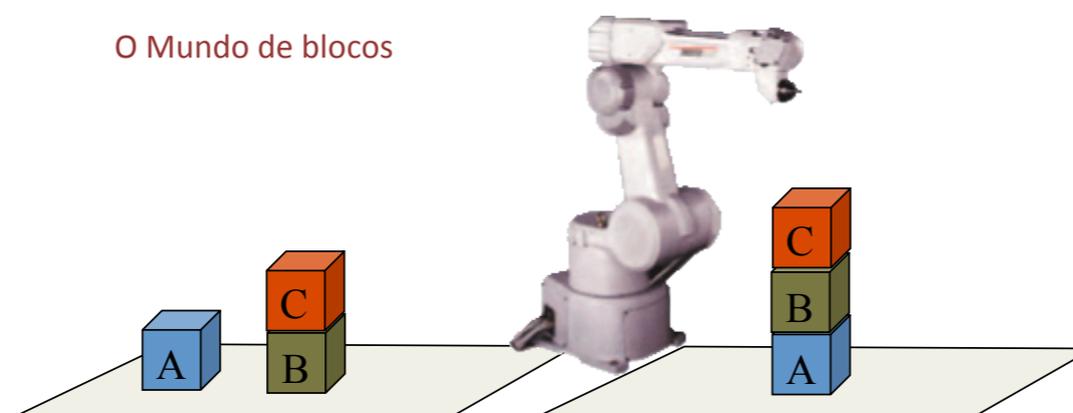


livre(A)  
sobre(A, mesa)  
sobre(C, B)  
sobre(B, mesa)  
livre(C)

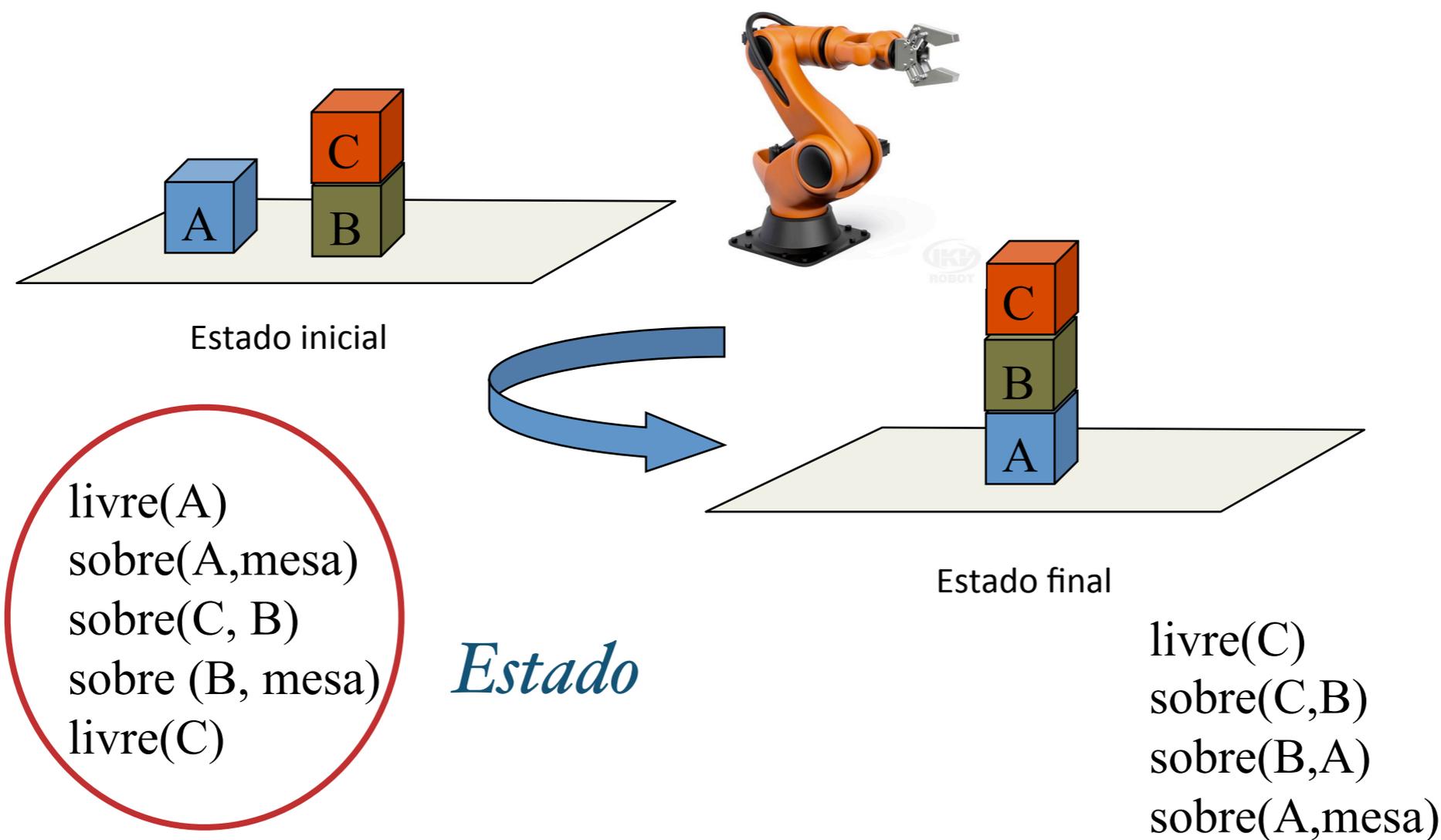
Operadores:  
pick(a), drop(b), move(a, table)...

# IA Planning: o STRIPS

O sistema STRIPS é a estratégia de resolução de problemas mais usada em planning. Note-se que é uma estratégia baseada no método estado-transição e por isso é passível de ser analisada em Redes de Petri. O problema modelo mais conhecido resolvido com o sistema STRIPS é o problema do mundo de blocos.



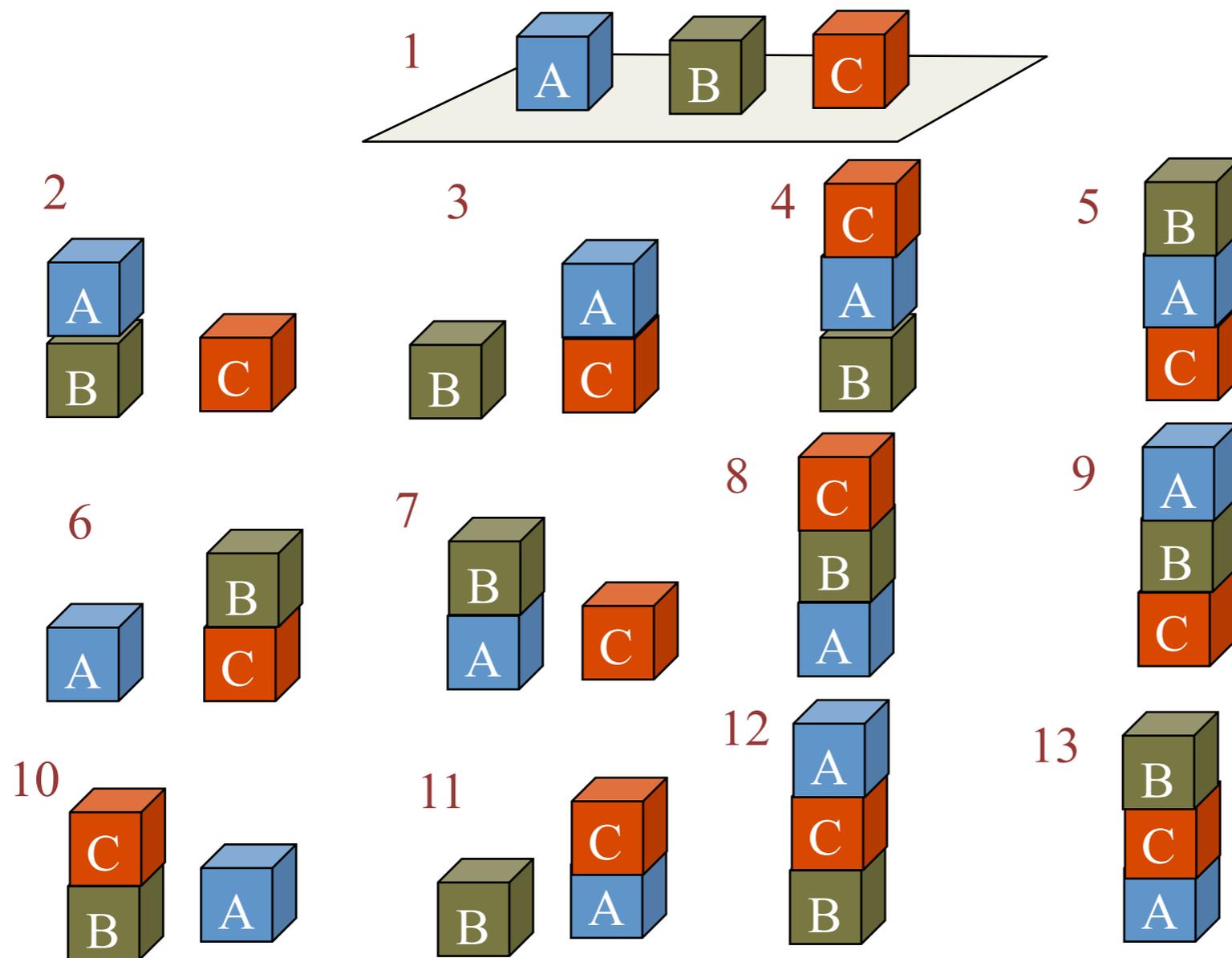
Um plano é a solução de um problema composto de um estado inicial, um estado final, e uma sequência de ações (ou um passo) que transforma o estado inicial no estado final, ou, em outras palavras que os coloca na mesma localidade

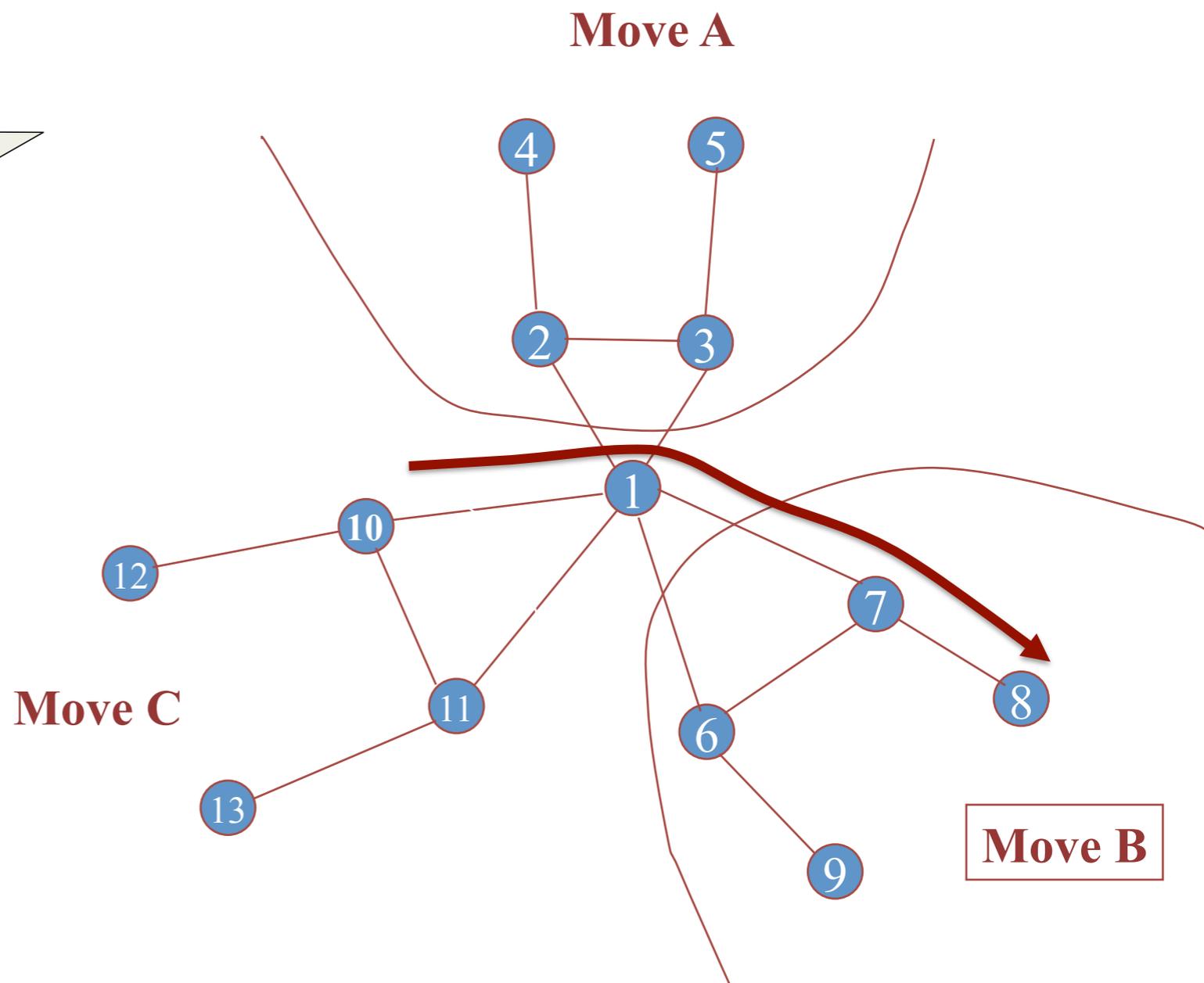
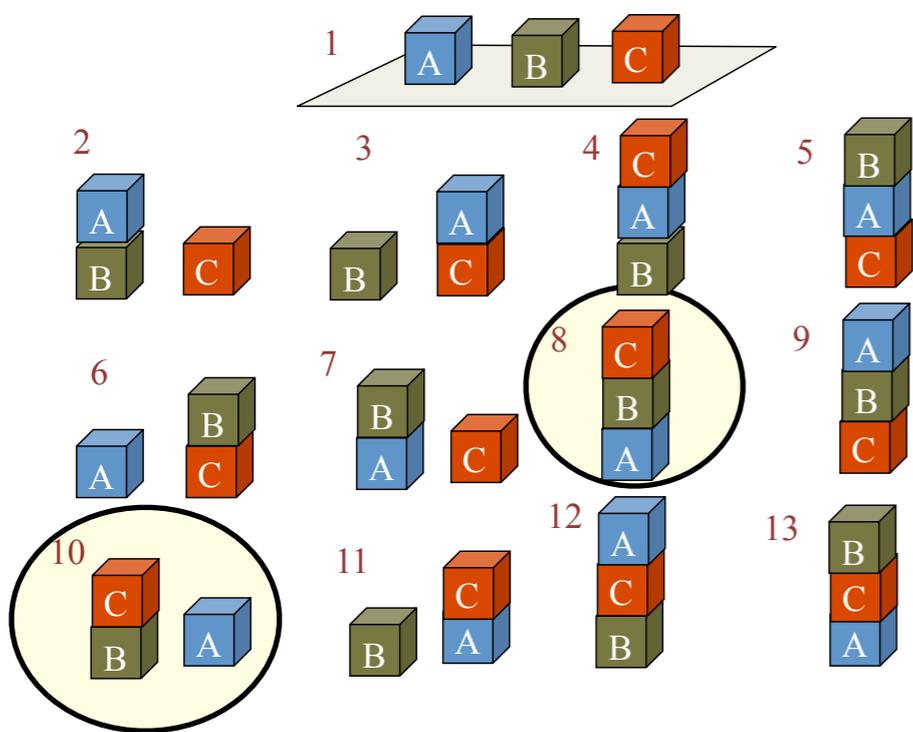
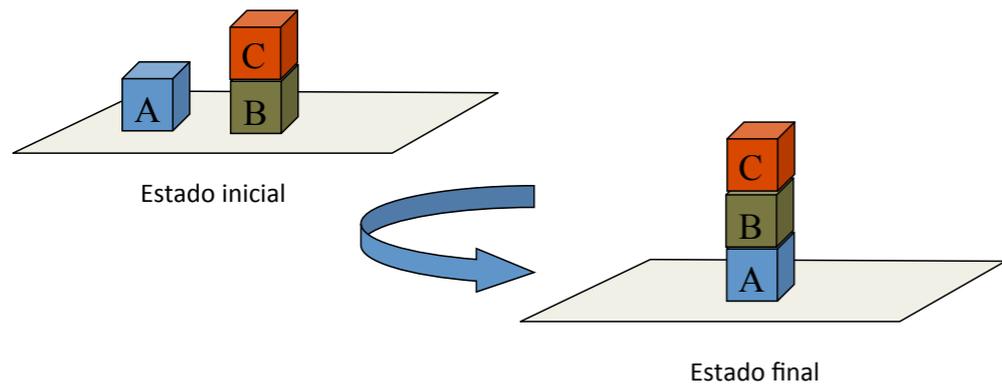


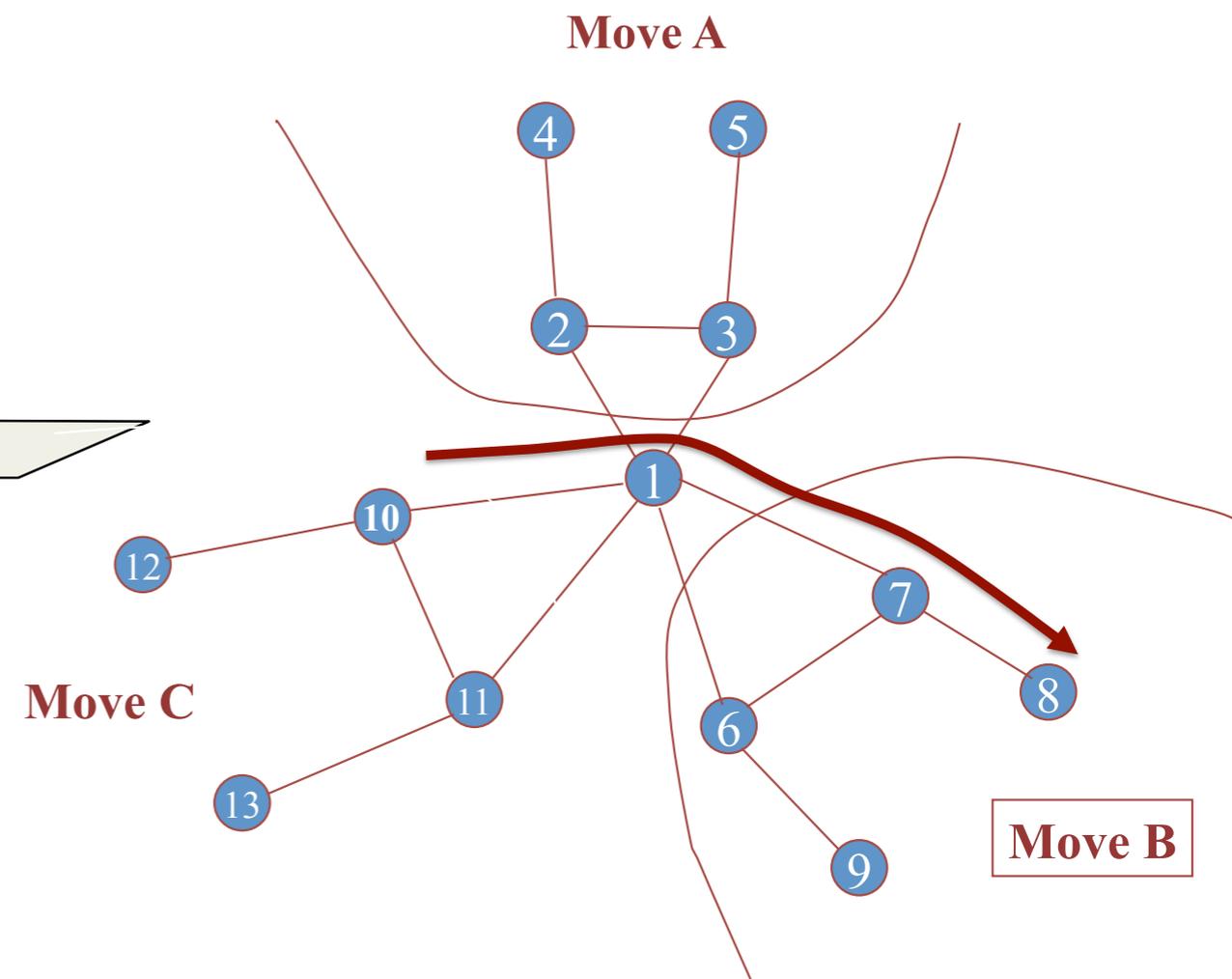
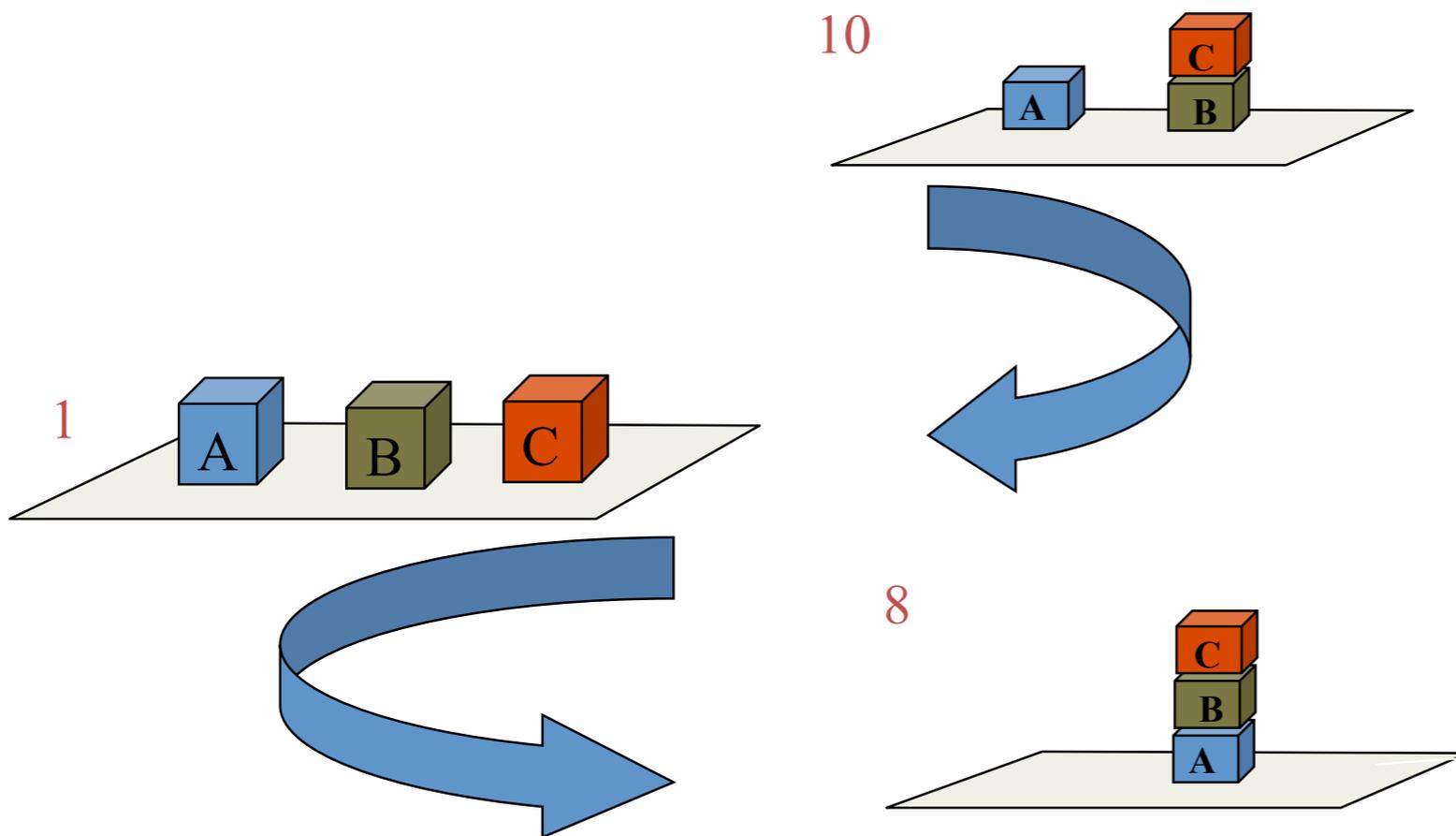


A modelagem estado-transição exigiria a especificação de TODOS os estados possíveis (espaço de estados) e também de todas as transições ou ações possíveis.

O problema é que em quase todos os problemas o espaço de estados é muito grande e é proibitivo computacionalmente, listar os estados. O mundo de blocos é um dos poucos casos em que isso é possível.





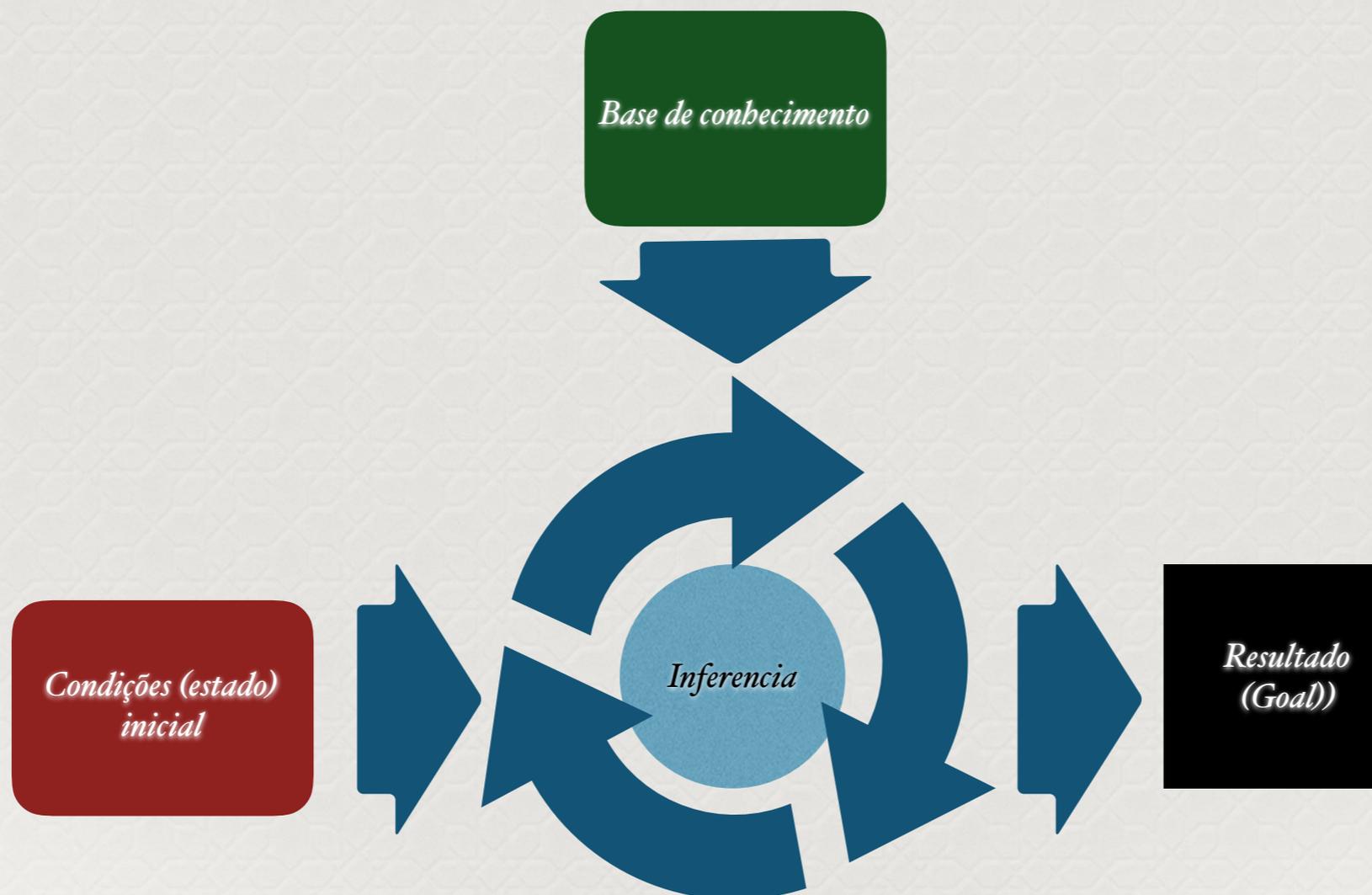




Até aqui definimos os estados, e especificamos de forma direta (listamos) os estados, e definimos também (mais ou menos) as ações e operações que tratam da mudança de estados. Resta agora ver como funcionaria o mecanismo de inferência, isto é, como se faz a efetiva mudança de um estado para outro.



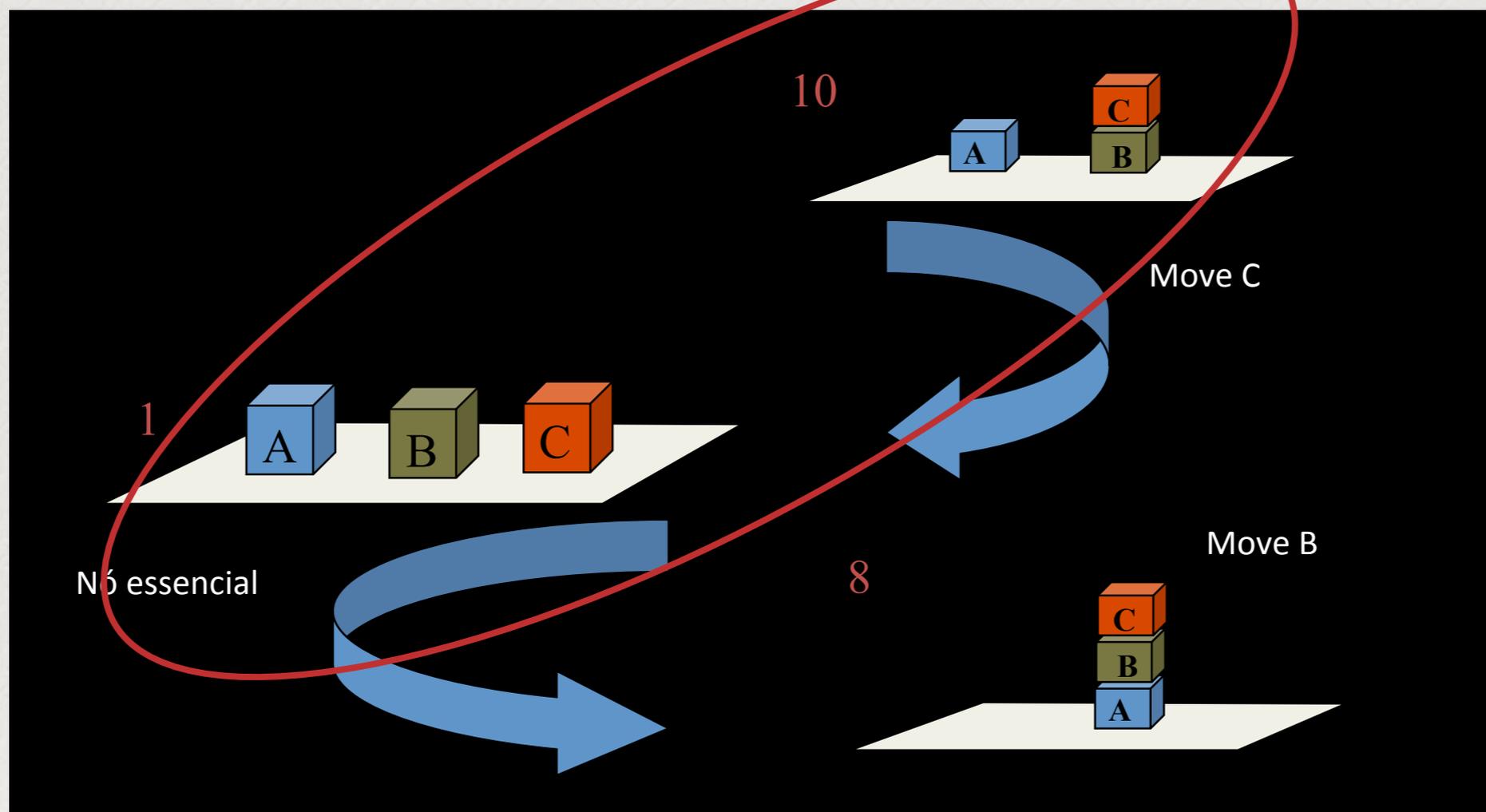
*Em uma primeira abordagem, gostaríamos de ter “agentes inteligentes” capazes de “resolver problemas”. O que significa isso?*

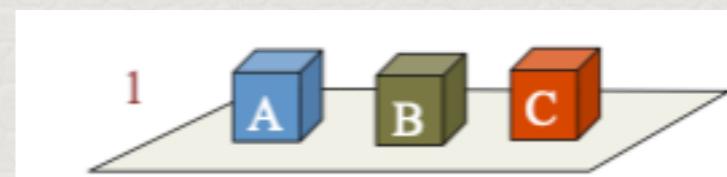
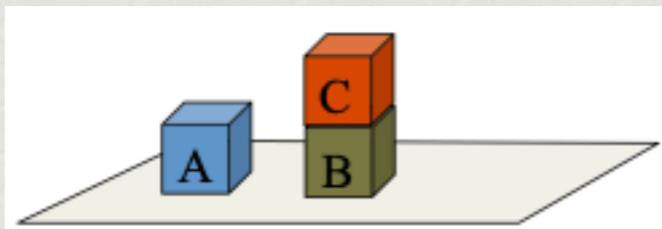




## STRIPS Basic Concepts

Vamos usar novamente o problema modelo do mundo de blocos - que inspirou o STRIPS - para mostrar como opera a "máquina de inferência" ...





livre(A)  
sobre(A, mesa)  
sobre(C, B)  
sobre(B, mesa)  
livre(C)

livre(A)  
sobre(A, mesa)  
livre(B)  
sobre(B, mesa)  
livre(C)  
sobre(C, mesa)

*Pre-condição*

move(C, mesa)

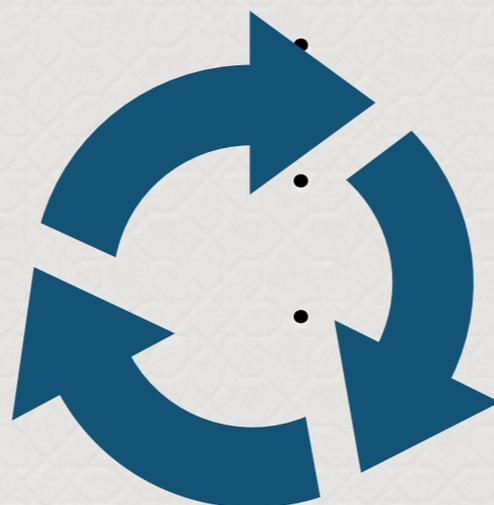
*Pós-condição*

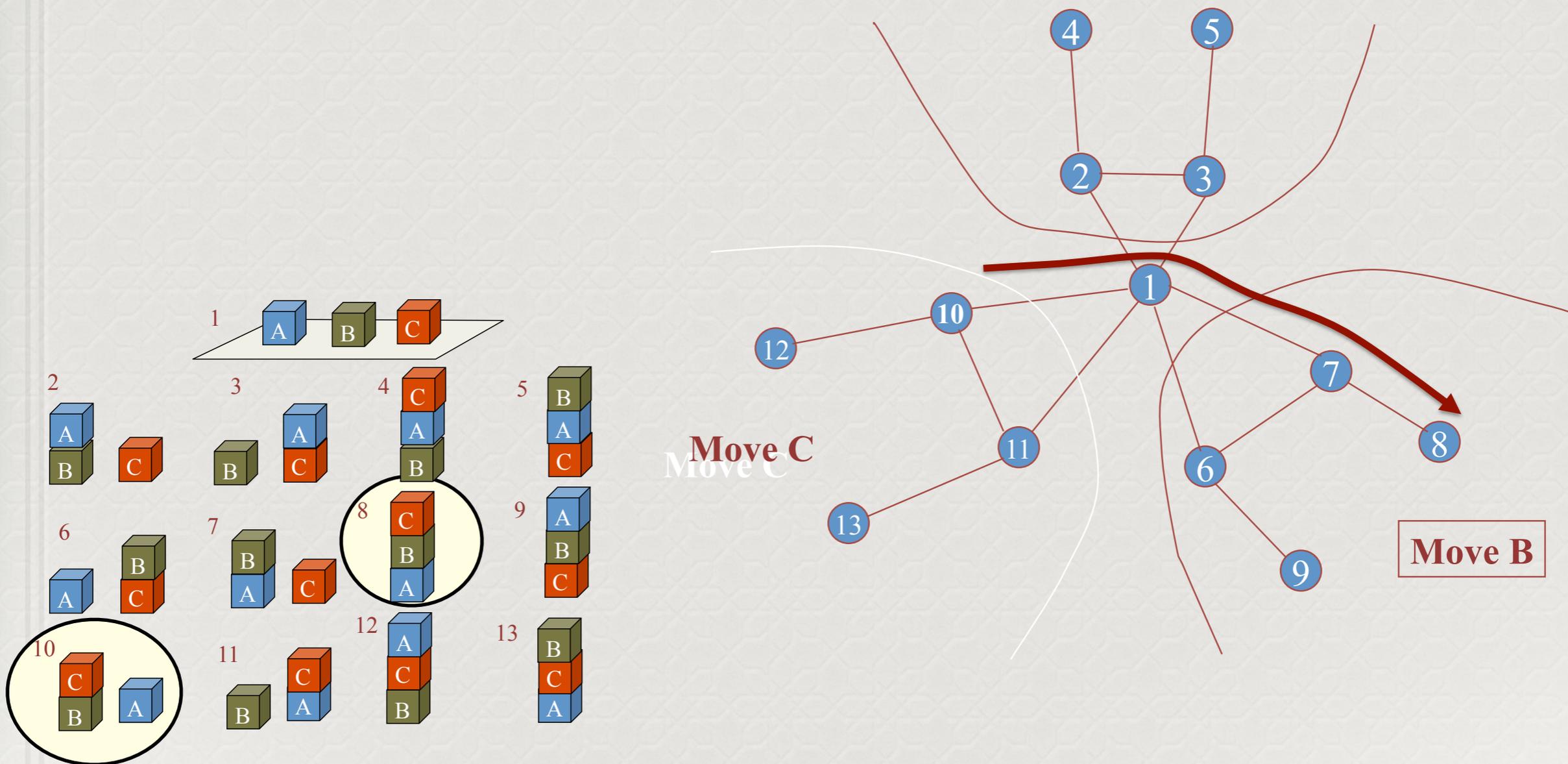
livre(C)  
sobre(C, X)

livre(C)  
sobre(C, mesa)  
livre(B)

*remove*

*add*







Portanto temos agora uma referência de "algoritmo" inteligente (?) de como resolver problemas, baseado nas técnicas de prova de teorema.

Podemos agora aplicar reiteradamente o "mecanismo de inferência" para percorrer os estados em um processo de busca pelo estado final.

**Mas, isso é "inteligente"? Ou somente uma busca cega?**



Temos portanto 3 opções:

- Uma busca cega direta (inteligente?)
- Uma busca orientada por heurísticas!
- Tratar o problema por análise de propriedades para obter maior eficiência

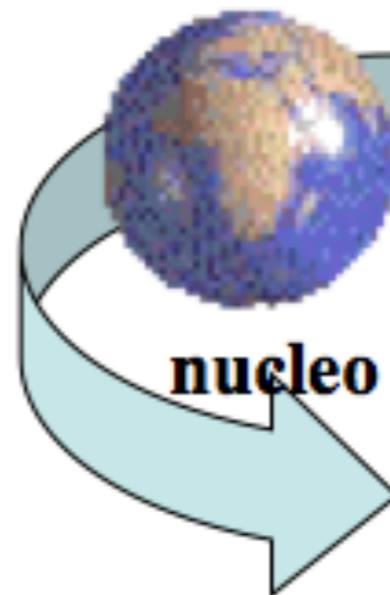


Em qualquer dos casos, podemos dizer que a aplicação do STRIPS resulta no que chamamos de “inteligência de máquina” clássica, dotando estas máquinas da capacidade de resolver problemas... ou pelo menos uma classe de problemas.



## Como se resolve um problema cientificamente ?

### Elementos básicos de uma teoria



Cinturão protetor



Imre Lakatos

heurísticas



## Recapitulando:

- Em 1957 Hebert Simon, J. C. Shaw e Allen Newel criaram a teoria da racionalidade e com ela o General Problem Solver (GPS), um formalismo genérico para resolver problemas, que poderia ser transformado em programa para provar teoremas (Aula 2).
- A pesquisa continua com vários avanços seja por Newel e Simon ou pelo grupo de Stanford reunido no SRI (Stanford Research Institute). Este grupo lança o STanford Research Institute Problem Solver (STRIPS), que desde 1971 é a referência em inteligência de máquina, em um processo formal conhecido como Planejamento Inteligente.



O método do STRIPS é baseado em estado-transição e consiste em:

- Modelar o domínio onde o plano será aplicado: o que implica em definir claramente estados e restrições que devem nortear a aplicação de ações e operadores.
- Definir o problema de planejamento, isto é, as ações admissíveis, pre e pós-condições para a aplicação destas ações;
- Definir um processo de busca, direta, heurística ou definir um modelo de problema para ser resolvido (este último caso está fora do escopo desta disciplina).



Portanto o método, que pode ser aplicado em vários problemas reais, depende do processo de busca e pode ser resolvido por um algoritmo diretamente ligado ao domínio e ao problema de planejamento ou por sistemas mais genéricos (que resolveriam qualquer problema) chamado de planejadores.

No caso do mundo de blocos usaremos a primeira abordagem.



*Será este um caminho possível para qualquer problema?*

Na prática são poucos os problemas onde se pode listar todo o espaço de estados. Exatamente por isso o “mundo de blocos” é um problema-modelo muito importante e um excelente caso de estudo. Portanto embora parece atraente este não pode ser o método adotado para resolver problemas automaticamente por máquinas.



## *Uma outra forma seria gerar o espaço de estados enquanto se busca a solução*

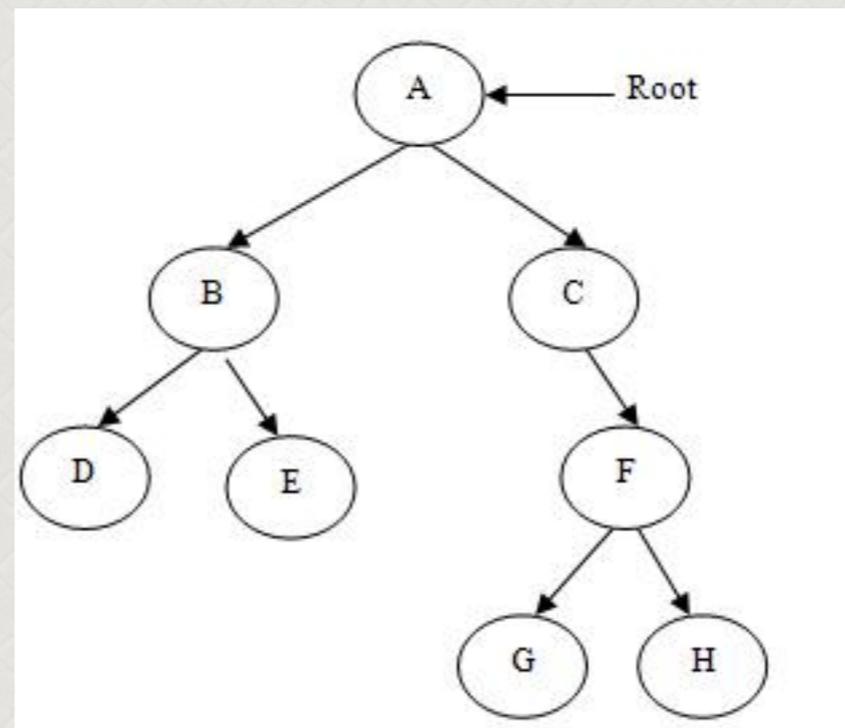
1. uma descrição clara do “estado inicial” ou seja das condições iniciais do problema a ser resolvido;
2. uma descrição clara do objetivo ou “estado final”, de modo que seja possível saber quando (e se) o problema foi resolvido;
3. em cada estágio do processo de solução saber quais os próximos estados que podem ser atingidos;
4. poder escolher um (ou o melhor) caminho entre os estados acima;
5. saber que operadores (ou passos) aplicar para fazer a “transição” para um próximo estado;
6. discernir se estamos convergindo para a solução.

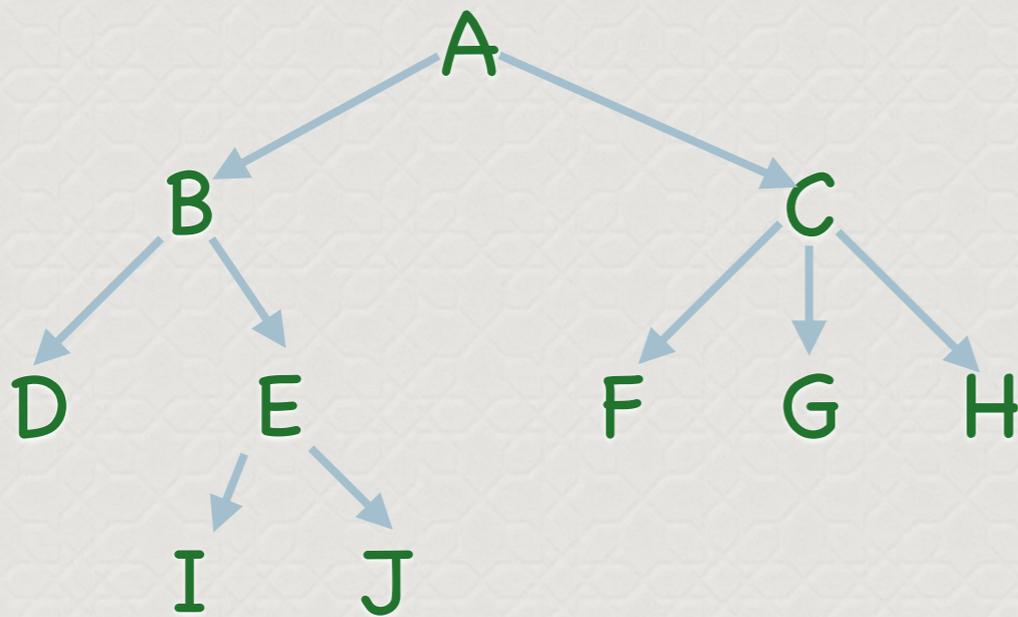


*estrutura*

## *Usando árvores como base para a solução*

*Uma opção muito interessante é modelar o espaço de estados na forma de uma árvore*





[a | [b | [d, [e | [i, j]]]], [c | [f, g, h]]]



## *Algoritmos*

*Existem prós e contras em cada estratégia descrita e nos algoritmos gerados para implementá-las:*

- ✦ *Gerar todo o espaço de estados pode ser proibitivo dependendo do problema, mas, se não for este o caso poderemos usar este espaço para todas as instâncias do problema.*
- ✦ *Gerar o espaço de estados enquanto se faz a busca pode ser interessante, mas temos que levar em conta que teremos que repetir a mesma geração para cada instância do problema.*



## *Algoritmos de busca*

*Busca não informada - quando todos os nós gerados são igualmente promissores, ou não se tem informação sobre o seu potencial: busca em profundidade, busca em largura, busca de custo uniforme*

*Busca informada - quando conhecimento heurístico pode ser levantado que distingue entre os nós gerados em um mesmo nível da árvore.*



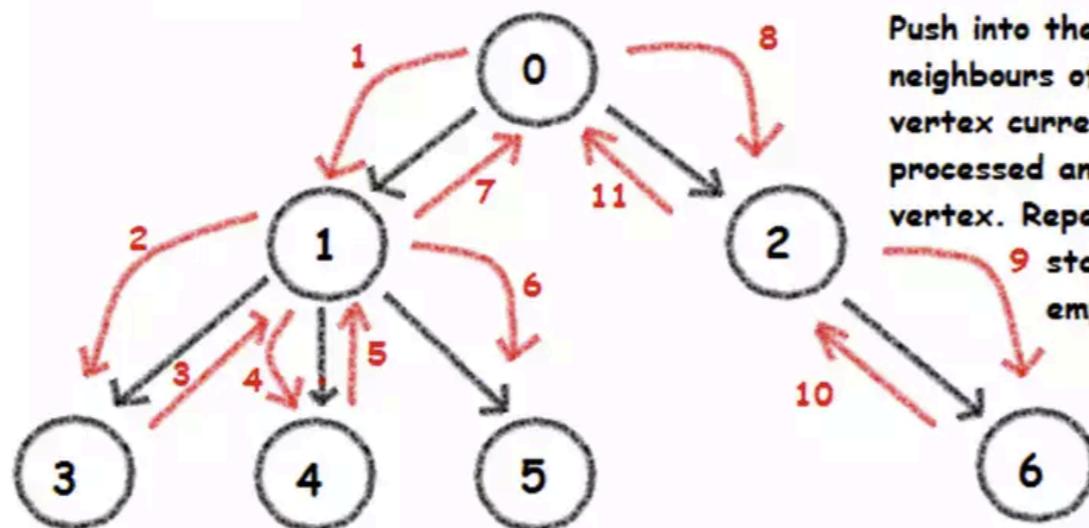
## *Métodos clássicos de busca não-informada:*

- ✦ *Busca em profundidade*
- ✦ *Busca em largura*



## Busca em profundidade

Red arrows indicate the order of search.



Push into the stack the neighbours of the vertex currently being processed and Pop the vertex. Repeat until stack is not empty.

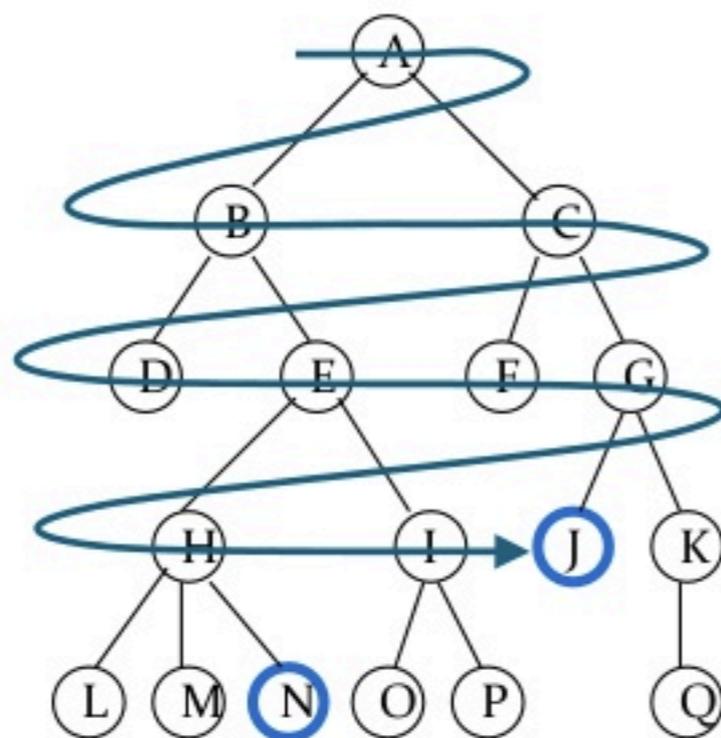
Vertex	Stack
	0
0	1, 2
1	3, 4, 5, 2
3	4, 5, 2
4	5, 2
5	2
2	6
6	

Depth First Search



## Busca em largura

### Breadth-first searching[1]



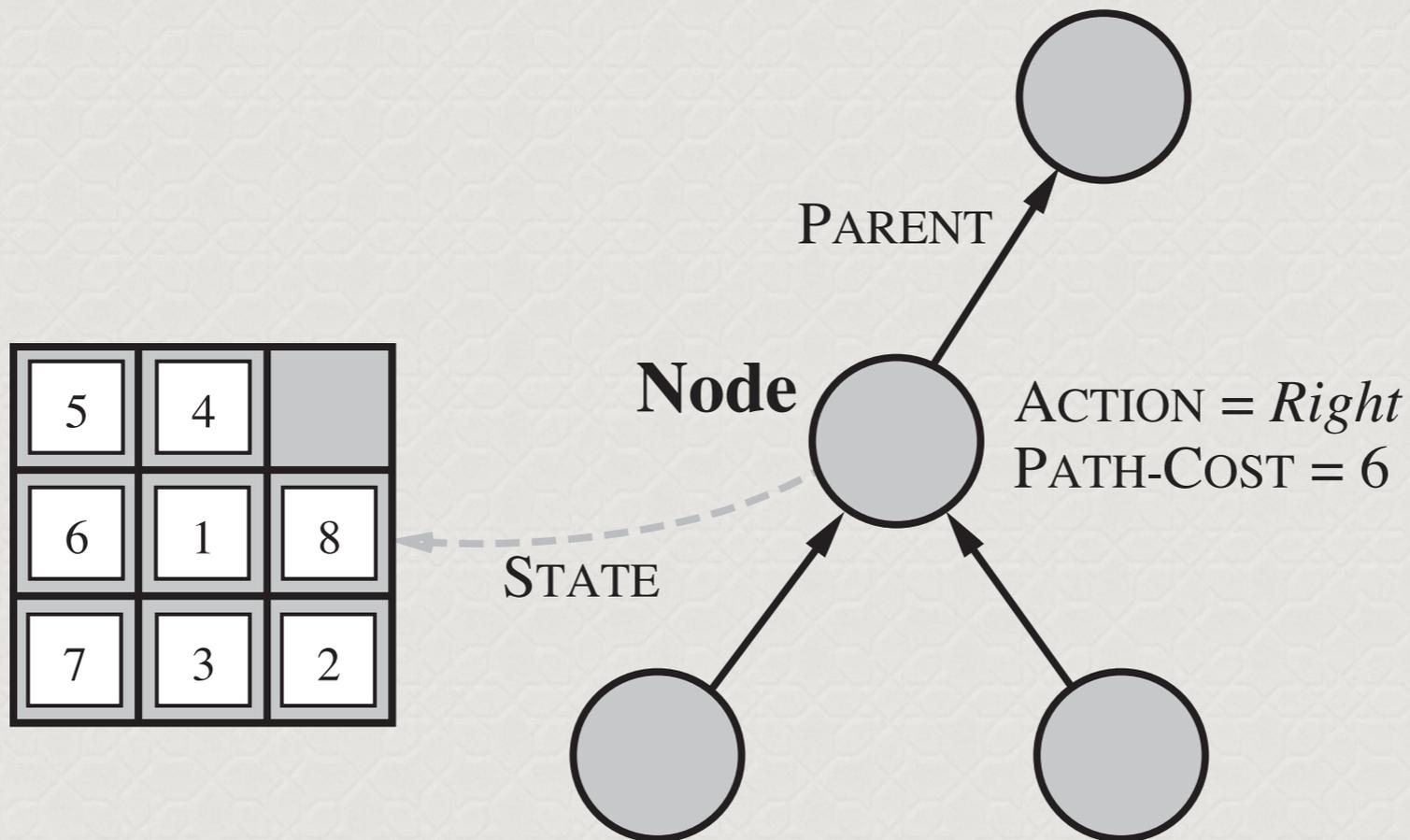
- A breadth-first search (BFS) explores nodes nearest the root before exploring nodes further away
- For example, after searching A, then B, then C, the search proceeds with D, E, F, G
- Node are explored in the order ABCDEFGHIJKLMNOPQ
- J will be found before N



## *Usando os métodos clássicos de busca não-informada*

Comparando os dois métodos podemos concluir que cada um deles pode ser melhor aplicado em situações onde:

- i) a solução está mais próxima da raiz, e a busca em largura será mais eficiente;
- ii) a solução está nos nós de maior profundidade ou nas folhas, a neste caso a busca em profundidade será mais apropriada;





*Na verdade, gerar todo o espaço de estados neste problema é proibitivo, portanto devemos pensar primeiro na geração enquanto se constrói o espaço de estados e em busca informada, que será vista na próxima aula, juntamente com a busca com custo uniforme.*



*Aplicando o STRIPS ao mundo de blocos:*

*Comece modelando o domínio do problema do mundo de blocos. A segunda etapa será a definição das ações, pré e pós-condições para sua aplicação. Lembre que agora precisaremos introduzir o robô como um elemento importante do sistema.*



*Aplicando o STRIPS ao mundo de blocos:*

*Cada equipe deve submeter na página da disciplina até a próxima aula o que conseguir na modelagem do problema. Discutiremos em termos gerais o problema do modelo na próxima aula e as equipes terão a chance de corrigir possíveis erros ou escolhas proibitivas.*



## Machines Who Think

A Personal Inquiry into  
the History and Prospects  
of Artificial Intelligence



Pamela  
McCorduck



A K Peters, Ltd.  
Natick, Massachusetts

## STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving<sup>1</sup>

Richard E. Fikes  
Nils J. Nilsson

Stanford Research Institute, Menlo Park, California

Recommended by B. Raphael

Presented at the 2nd IJCAI, Imperial College, London, England, September  
1-3, 1971.

### ABSTRACT

We describe a new problem solver called STRIPS that attempts to find a sequence of operators in a space of world models to transform a given initial world model into a model in which a given goal formula can be proven to be true. STRIPS represents a world model as an arbitrary collection of first-order predicate calculus formulas and is designed to work with models consisting of large numbers of formulas. It employs a resolution theorem prover to answer questions of particular models and uses means-ends analysis to guide it to the desired goal-satisfying model.

### DESCRIPTIVE TERMS

Problem solving, theorem proving, robot planning, heuristic search.

### 1. Introduction

This paper describes a new problem-solving program called STRIPS (Stanford Research Institute Problem Solver). An initial version of the program has been implemented in LISP on a PDP-10 and is being used in conjunction with robot research at SRI. STRIPS is a member of the class of problem solvers that search a space of "world models" to find one in which a given goal is achieved. For any world model, we assume that there exists a set

<sup>1</sup> The research reported herein was sponsored by the Advanced Research Projects Agency and the National Aeronautics and Space Administration under Contract NAS12-2221.



*Até a próxima aula!*