



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

# PMR-3510 Inteligência Artificial

## Aula 13- Algoritmos para Planejamento

*Prof. José Reinaldo Silva*

*reinaldo@usp.br*





## *Tipos de problema resolvidos por IA*

Podemos classificar os problemas aos quais aplicamos (com sucesso IA em:

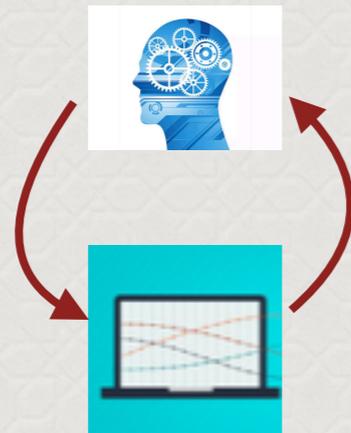
- i) completamente reativos
- ii) interativos (jogos, sistemas especialistas, apoio a decisão)
- iii) preditivos (planning, scheduling, etc.)
- iv) evolutivos (aprendizado de máquina)



## *Tipos de problema resolvidos por IA*

Podemos classificar os problemas aos quais aplicamos (com sucesso IA em:

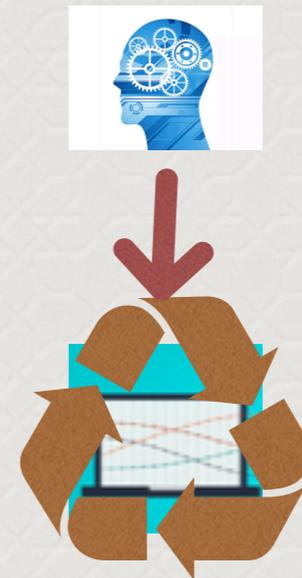
- ✓ i) completamente reativos
- ✓ ii) interativos (jogos, sistemas especialistas, apoio a decisão)
- iii) preditivos (planning, scheduling, etc.)
- iv) evolutivos (aprendizado de máquina)



Completamente reativo



Interativo



Preditivo



## *Base conceitual para o problema de planning (III)*

Um sistema estado-transição é uma 4-upla  $\Sigma = (S, A, E, \gamma)$  onde:

- $S$  é um conjunto de finito (ou recursivamente enumerável) de estados;
- $A$  é um conjunto finito (ou recursivamente enumerável) de ações;
- $E$  é um conjunto finito (ou recursivamente enumerável) de eventos;
- $\gamma = (S \times A \times E) \rightarrow 2^S$  é a função de transição.

Ghallab, M., Nau, D., Traverso, P.; Automated Planning: Theory and Practice, Morgan Kaufmann, 2004



*Domain Specific Planning (DSP)*

*X*



*Domain Independent Planning (DIP)*

*... e buscar um algoritmo viável par o DIP*

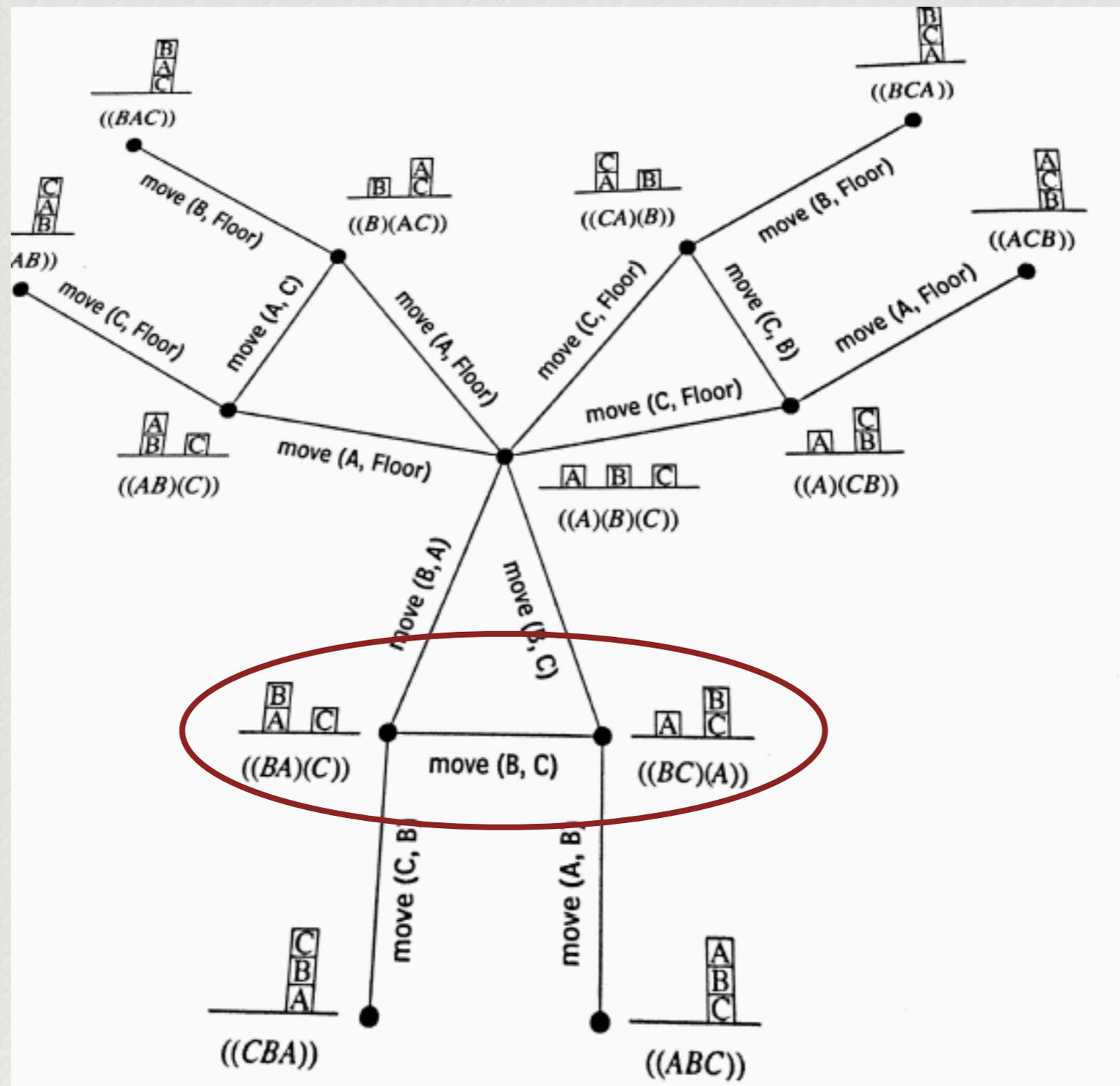


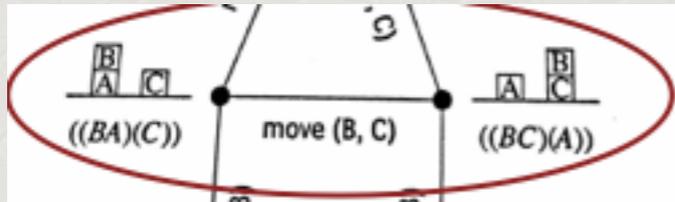
## *STRIPS: Stanford Research Institute Problem Solver*

Em 1971, Richard Fikes e Nils Nilsson, propuseram um algoritmo geral para resolução de problemas em IA que ficou conhecido como STRIPS, em uma alusão ao SRI, onde a pesquisa foi realizada. Mais tarde STRIPS passou a significar também uma linguagem para tratar problemas de planejamento. Foi substituída por outras mas todas inspiradas no algoritmo e na linguagem original.



Nilsson, N., Artificial Intelligence: a new synthesis, Morgan Kaufmann, 1998.





on(b, a), on(a, table)  
on(c, table), free(c),  
free(b)



move(b,c)

on(b, c), on(a, table)  
on(c, table), free(a),  
free(b)

free(b), free(c)  
on(b, X)

remove[free(b), free(c)]  
add[free(b), free(a), on(b, c)]



on(b, a), on(a, table)  
on(c, table), free(c),  
free(b)



move(b,c)

on(b, c), on(a, table)  
on(c, table), free(a),  
free(b)

free(b), free(c)  
on(b, X)

remove[free(b), free(c), on(b, X)]  
add[free(b), free(a), on(b, c)]

A lista das sentenças relacionadas às pré-condições passa a ser uma lista de remoção de predicados, enquanto a lista das sentenças relacionadas às pós-condições é acrescentada à base.



## General Problem Solver (A. Newell & H. Simon)

The General Problem Solver (GPS) was a theory of human problem solving stated in the form of a simulation program (Ernst & Newell, 1969; Newell & Simon, 1972). This program and the associated theoretical framework had a significant impact on the subsequent direction of cognitive psychology. It also introduced the use of productions as a method for specifying cognitive models.



*Allen Newell dedicou parte da sua vida à pesquisa sobre ciência cognitiva e ao que se chama hoje Inteligência Artificial. O foco do seu trabalho foi buscar uma “teoria unificada” da racionalidade e dos processos cognitivos (humanos). Inclui-se aí o General Problem Solver (GPS), um sistema artificial capaz de reproduzir - pelo menos parcialmente - o processo cognitivo humano.*



Artificial Intelligence 59 (1993) 227–232  
Elsevier

227

ARTINT 980

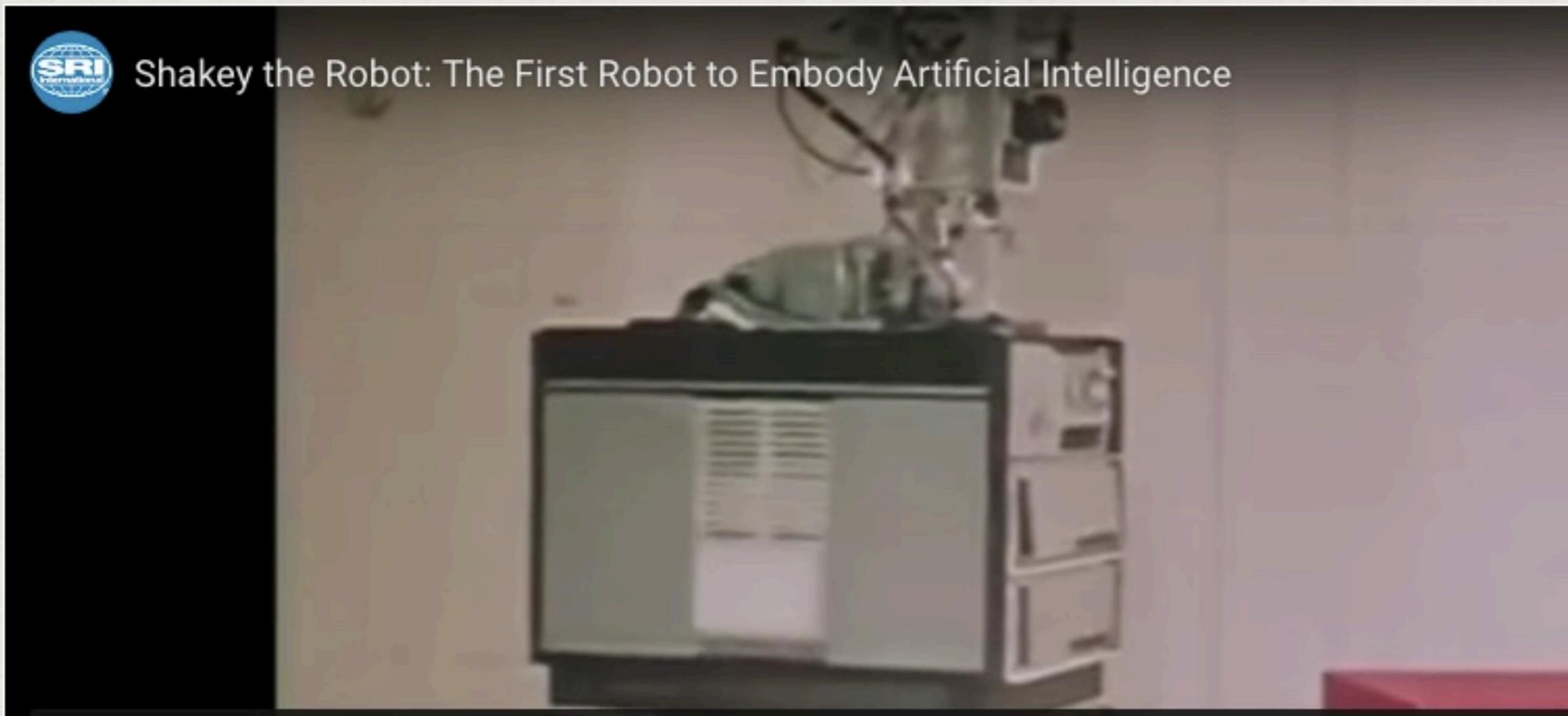
## STRIPS, a retrospective

Richard E. Fikes and Nils J. Nilsson

*Computer Science Department, Stanford University, Stanford, CA 94305, USA*

### Introduction

During the late 1960s and early 1970s, an enthusiastic group of researchers at the SRI AI Laboratory focused their energies on a single experimental project in which a mobile robot was being developed that could navigate and push objects around in a multi-room environment (Nilsson [11]). The project team consisted of many people over the years, including Steve Coles, Richard Duda, Richard Fikes, Tom Garvey, Cordell Green, Peter Hart, John Munson, Nils Nilsson, Bert Raphael, Charlie Rosen, and Earl Sacerdoti. The hardware consisted of a mobile cart, about the size of a small refrigerator, with touch-sensitive “feelers”, a television camera, and an optical range-finder. The cart was capable of rolling around an environment consisting of large boxes in rooms separated by walls and doorways; it could push the boxes from one place to another in its world. Its suite of programs consisted of those needed for visual scene analysis (it could recognize boxes, doorways, and room corners), for planning (it could plan sequences of actions to achieve goals), and for converting its plans into intermediate-level and low-level actions in its world. When the robot moved, its television camera shook so much that it became affectionately known as “Shakey the Robot”.





## A\*

P.E. Hart, N.J. Nilsson and B. Raphael, A formal basis for the heuristic determination of minimum cost paths, IEEE Trans. Syst. Sci. Cybern. 4 (2) (1968) 100-107.

## ABSTRIPS

E. Sacerdoti, Planning in a hierarchy of abstraction spaces, Artif. Intell. 5 (2) (1975) 115-135.



## STRIPS

R.E. Fikes and N.J. Nilsson, STRIPS: a new approach to the application of theorem proving to problem solving, Artif Intell. 2 (1981) 189-208.

## MACROPS

R.E. Fikes, P.E. Hart and N.J. Nilsson, Learning and executing generalized robot plans, Artif Intell. 3 (4) (1972) 251-288.



Using GPS as our paradigmatic problem solving architecture, we needed to define meaningful "differences" between a situation described by a set of predicate calculus sentences and a goal situation in which a given predicate calculus sentence is true. Once differences were defined, we needed to specify what it meant for an operator to be "relevant" to "reducing" the difference.



A regra geral seguida pelos operadores era alterar a descrição do domínio do seguinte modo:

```
if ( preconditions ) then ( retract ( deletions ) )  
( assert ( additions ) )
```



In retrospect, STRIPS was extremely limited in both the scope of planning issues it addressed and the complexity of problems it could solve. Whereas current planning research is concerned with multiple agents operating in dynamic environments, STRIPS assumed that only one action could occur at any time, that nothing changed except as a result of the planned actions, and that actions were effectively instantaneous. Also, the STRIPS "solution" to the frame problem was vague and flawed.

R.E. Fikes, N. J. Nilsson, STRIPS, A Retrospective, *Artificial Intelligence*, 59, pp. 227-232, Elsevier, 1993.



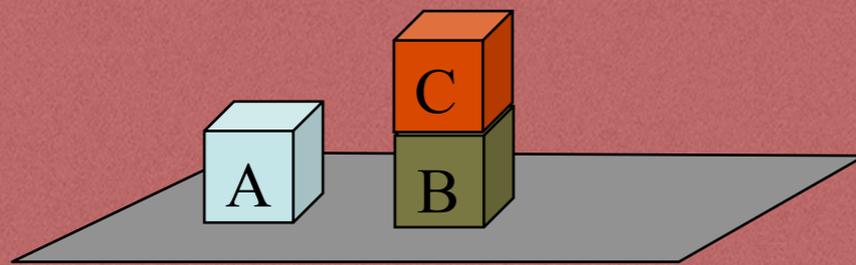
Algumas hipótese simplificadoras do STRIPS:

- i) só é possível realizar uma ação de cada vez;
- ii) a única possibilidade de mudança (de estado) do mundo é através da realização de uma destas ações;
- iii) as ações são determinísticas

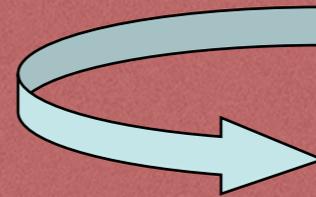


Gerald J. Susan  
CSAIL - MIT

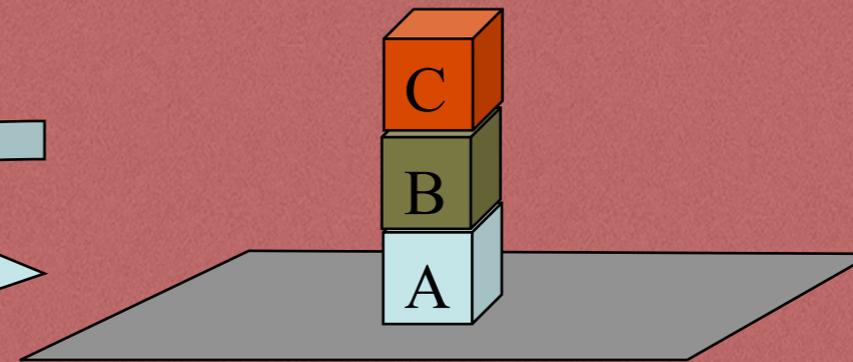
Na sua tese de doutorado em 1973, Gerald Sussman apontou falhas na abordagem do STRIPS, ligadas a estas limitações, especialmente à abordagem ligada ao método means-ends e à ordenação dos objetivos intermediários no processo de planning. Vamos usar o mundo de blocos para dar um exemplo desta "anomalia".



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



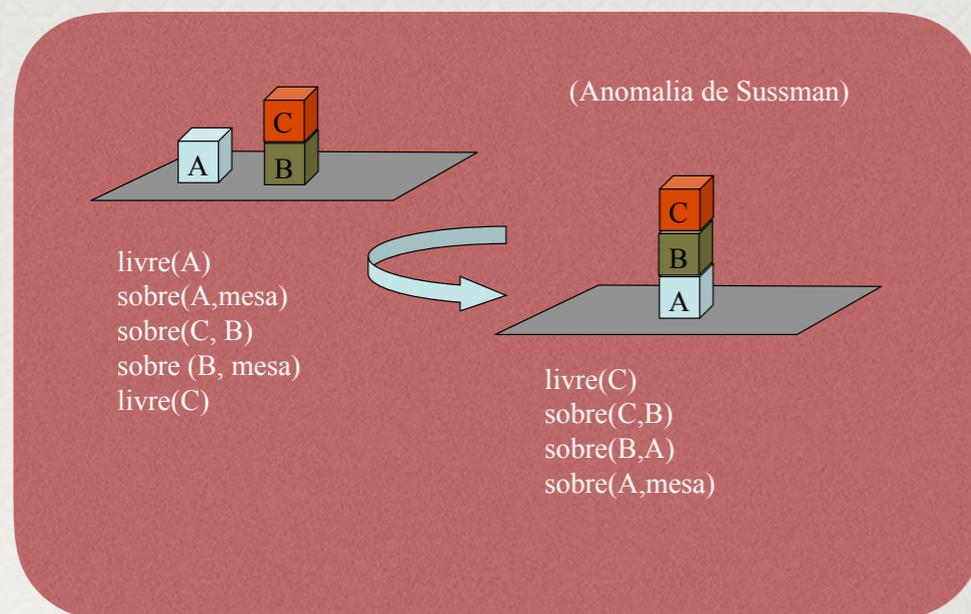
(Anomalia de Sussman)



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

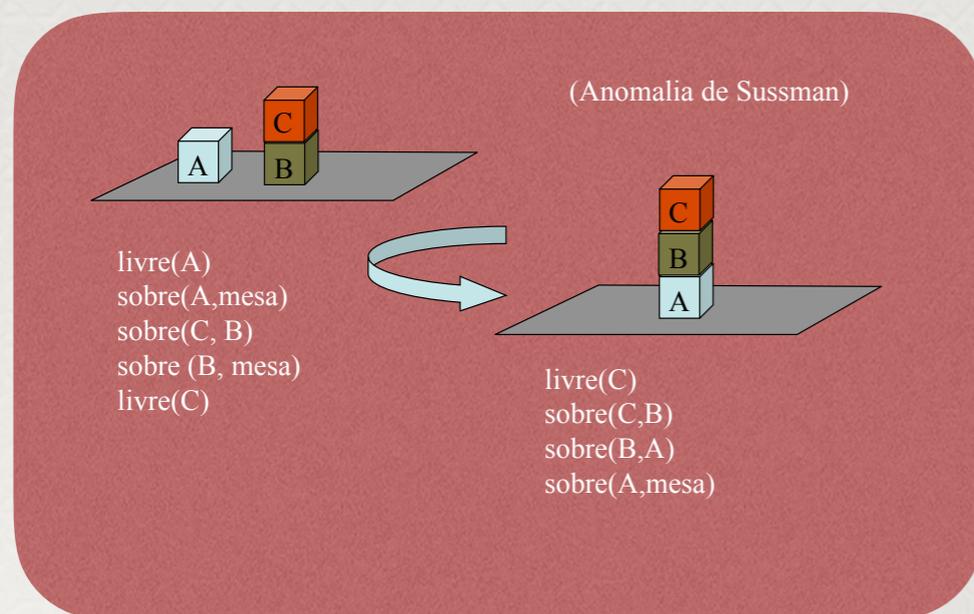


O caminho até o estado final pode ser dividido em dois "goals": colocar o bloco C sobre o bloco B e colocar o bloco B sobre o bloco A, em outras palavras, ter o predicado  $on(C, B)$  e  $on(B, A)$  na lista de adição, em sequência.





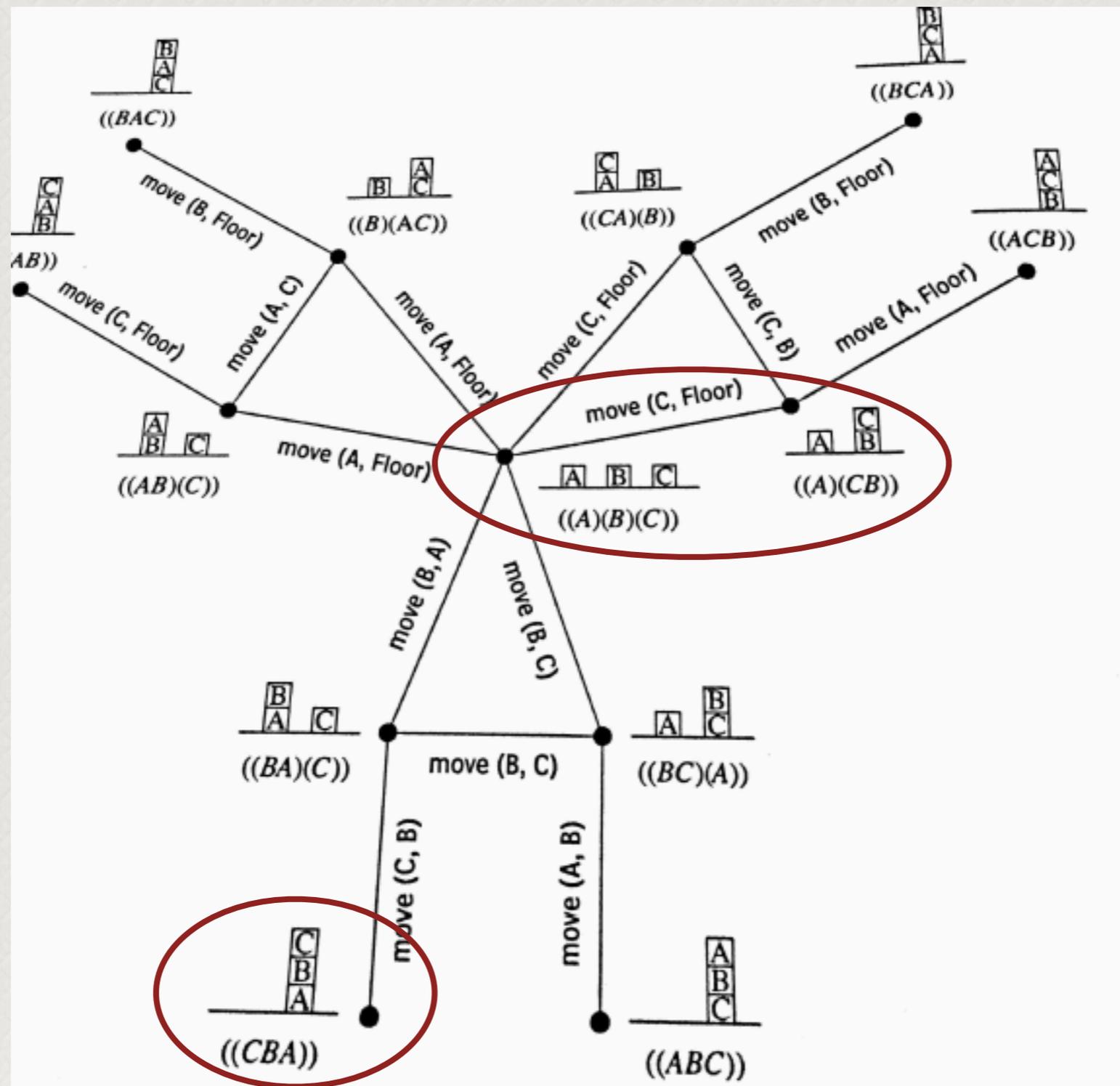
Pelas limitações do STRIPS chegar a  $on(C, B)$  está já no estado inicial, mas para chegar no segundo objetivo,  $on(B, A)$  é preciso desfazer este objetivo já conseguido, voltando ao estado onde todos os blocos estão sobre a mesa.



sub-goals:  
 $on(C, B)$   
 $on(B, A)$

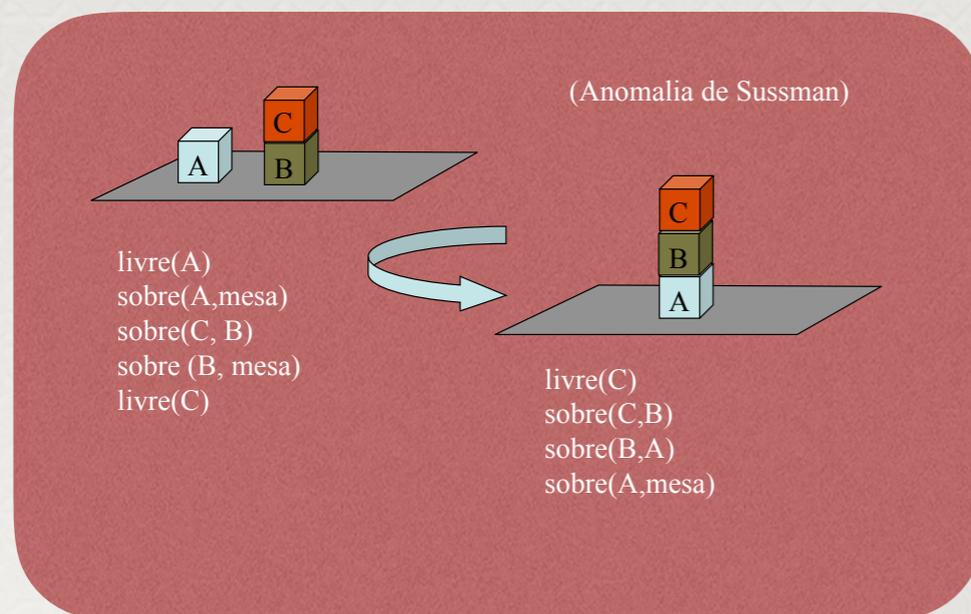


Nilsson, N., Artificial Intelligence: a new synthesis, Morgan Kaufmann, 1998.





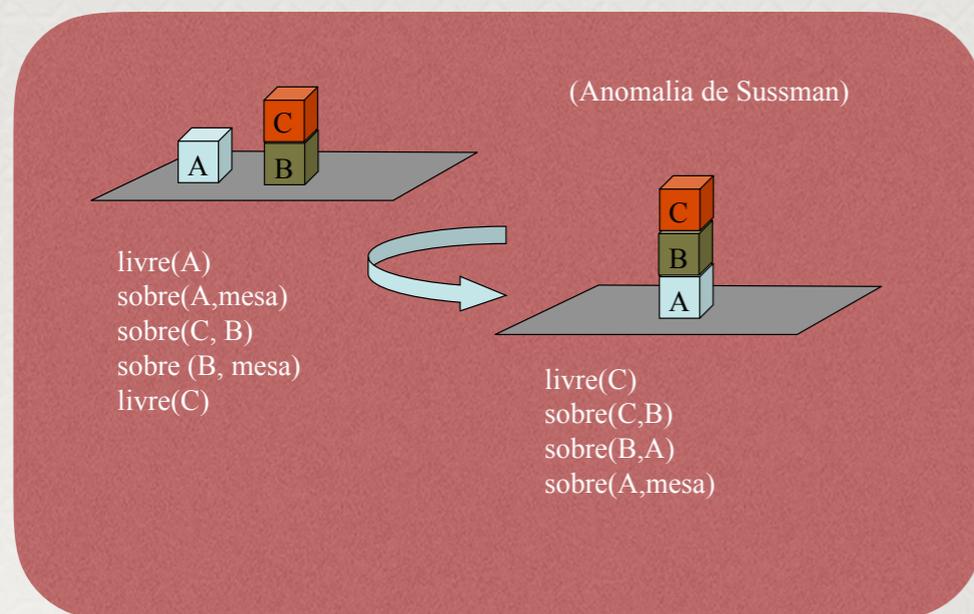
O movimento mais indicado para atingir o objetivo final seria  $\text{move}(B, A)$ , mas, olhando para a os objetivos intermediários o que vem primeiro seria "inserir  $\text{on}(C, B)$ " e isto só pode atingido por  $\text{move}(C, B)$ , o que leva novamente ao estado inicial.



sub-goals:  
 $\text{on}(C, B)$   
 $\text{on}(B, A)$



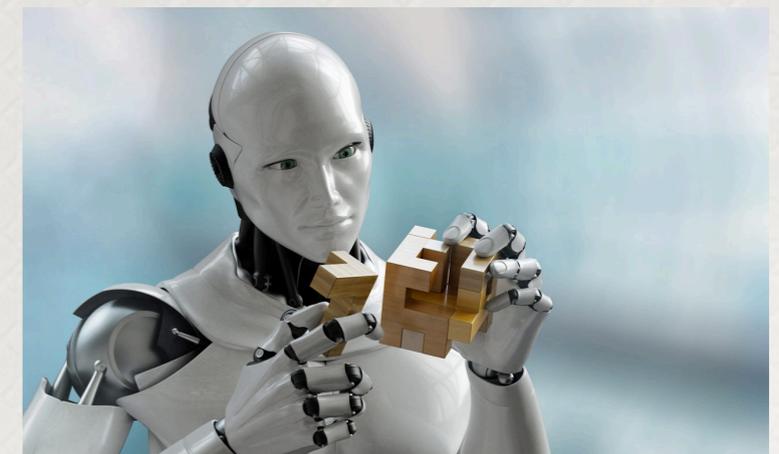
Portanto o sistema entra em loop e não atinge o estado final.



sub-goals:  
on(C, B)  
on(B, A)

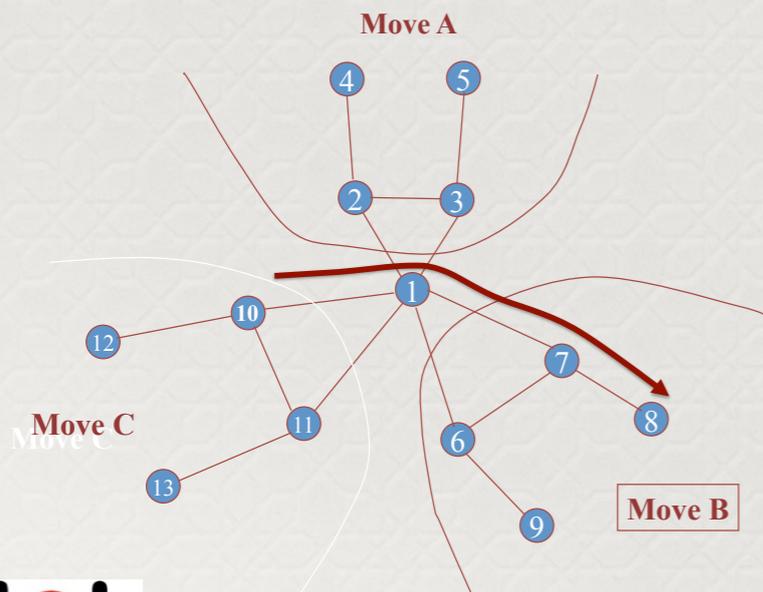
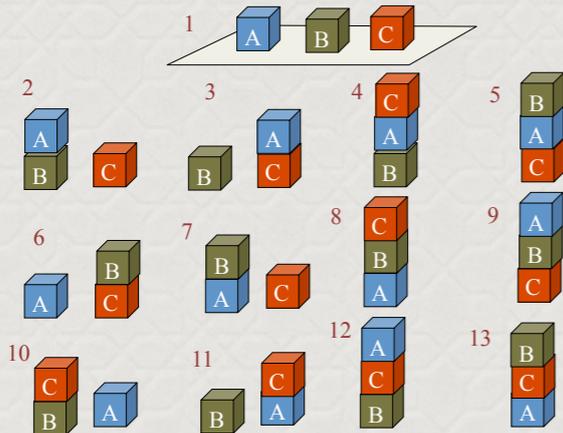
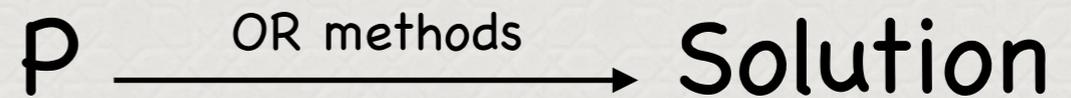
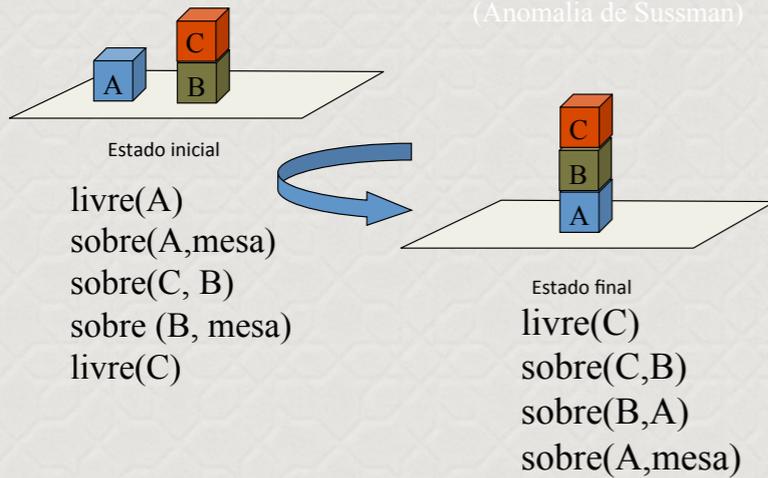


Os planejadores modernos podem evitar esta anomalia sem problema, mas é preciso levar em conta as limitações do método usado no STRIPS. Ainda assim, o STRIPS continua até hoje como uma referência para os planejadores e para os métodos em geral de resolução de problemas.





(Anomalia de Sussman)





# ICKEPS - International Competition on Knowledge Engineering for Planning Systems



itSIMPLE Performance

2005: 2nd. place

2007: honor winner

2009: winner





*Até a próxima aula!*