



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

# PMR-3510 Inteligência Artificial

## Aula 14- Algoritmos para Planejamento

*Prof. José Reinaldo Silva*

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





## *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.



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

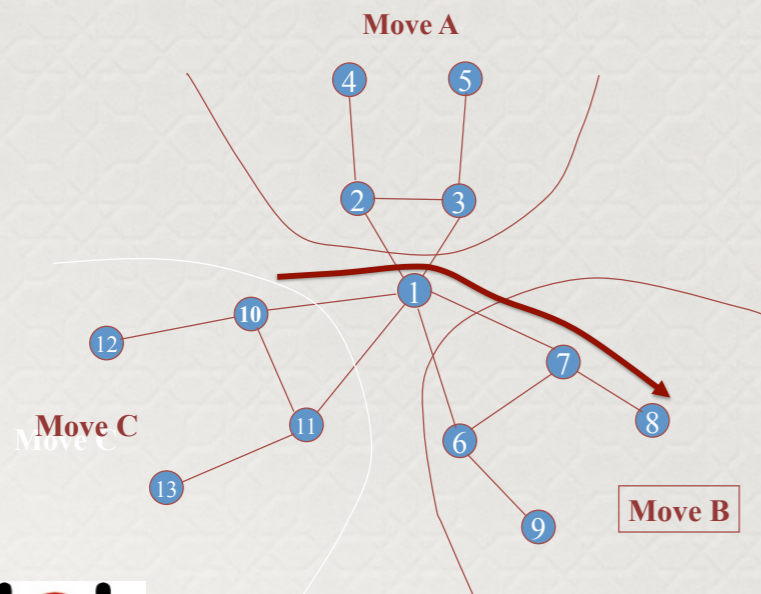
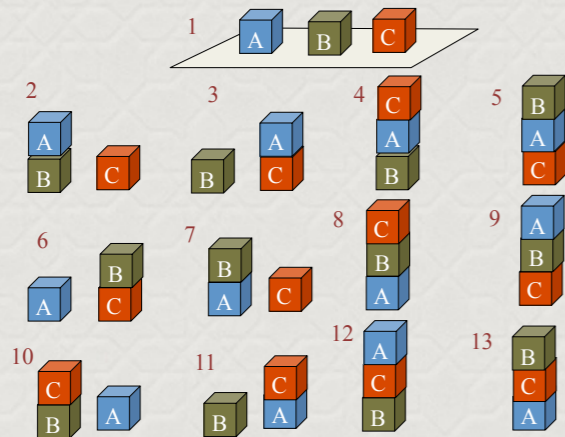
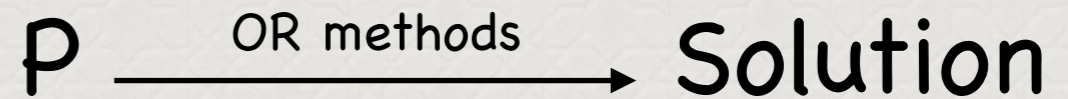
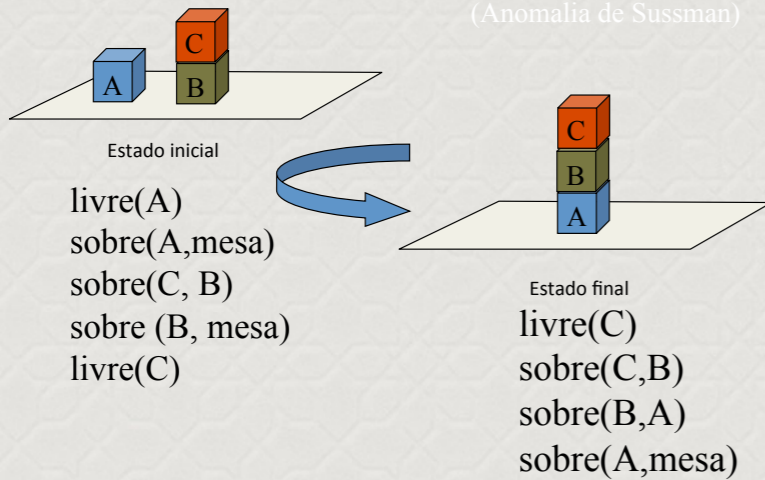


Gerald J. Susan  
CSAIL - MIT

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".



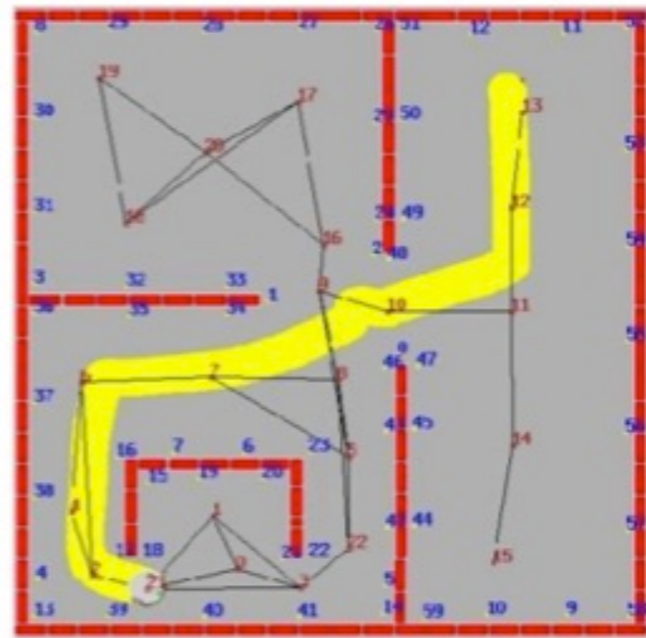
(Anomalia de Sussman)







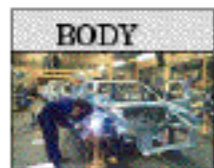
# Escola Politécnica da USP - Depto. de Enga. Mecatrônica



Source: <http://www.ics.forth.gr/cvrl/>



Robô Valkyrie mede 1,8 metro e pesa 125 quilos. (Foto: Nasa)



Laboratório de Design de Sistemas



Escola Politécnica da USP - Depto. de Enga. Mecatrônica



# EURO/ROADEF 2016 Challenge

## Inventory routing for gas distribution

April 2015

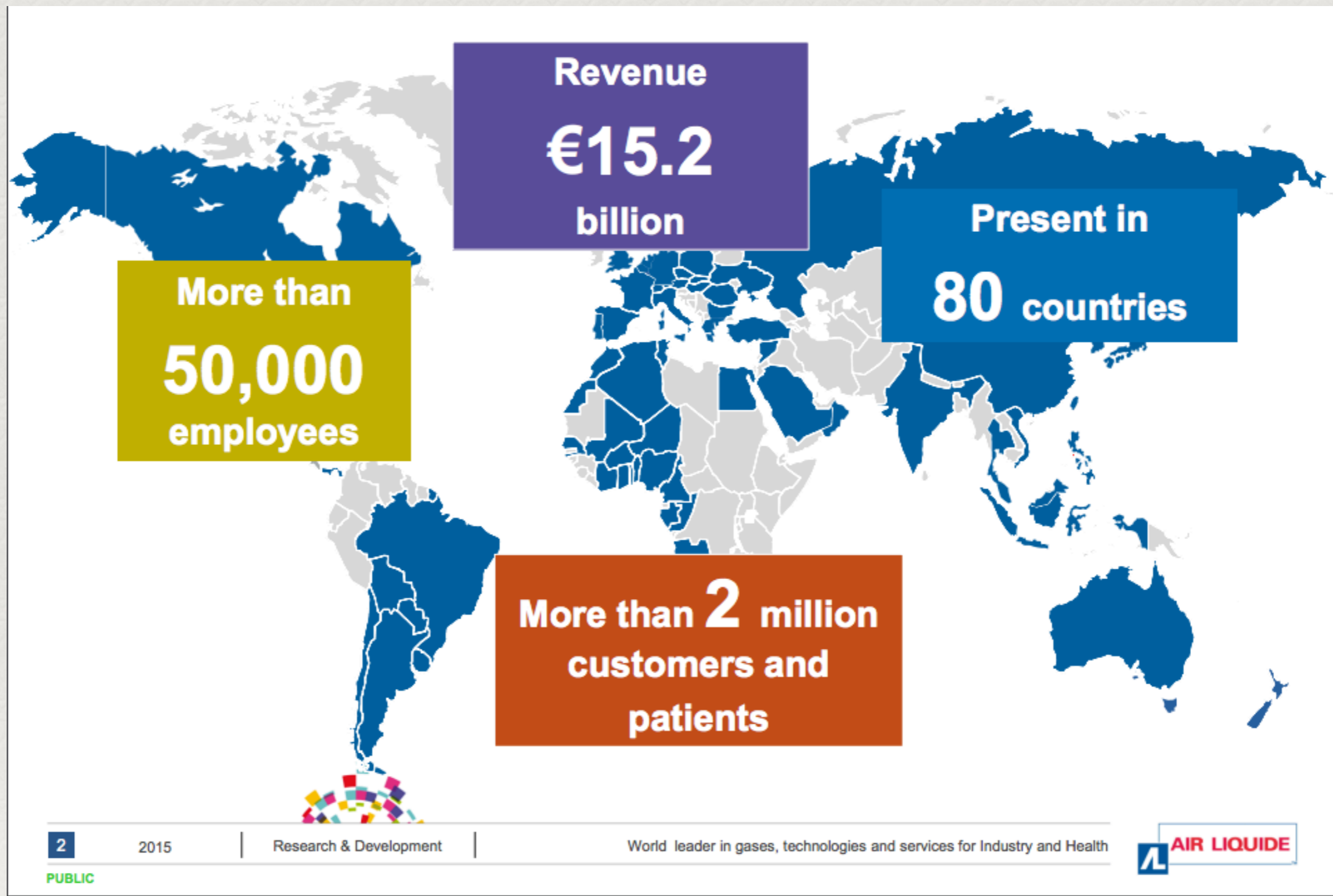
Air Liquide R&D | Applied Mathematics Global Lab | Operations Research Team

PUBLIC



Laboratório de Design de Sistemas

PMR-3510





## Air Liquide distribution challenge

- Generalization of the inventory routing problem (IRP) with vendor managed inventory replenishment
- Several decisions need to be made:
  - **When** to deliver to each customer?
  - **How much** to deliver to each customer?
  - **How** to deliver to each customer?
    - **From where:** depot / source?
    - **With what resources:** tractor / trailer / driver?
    - In **combination with** other customer deliveries (route)?





Portanto o apelo prático é claro. Na verdade AI Planning é um dos campos de maior difusão do uso da inteligência artificial, mesmo tendo algumas limitações.

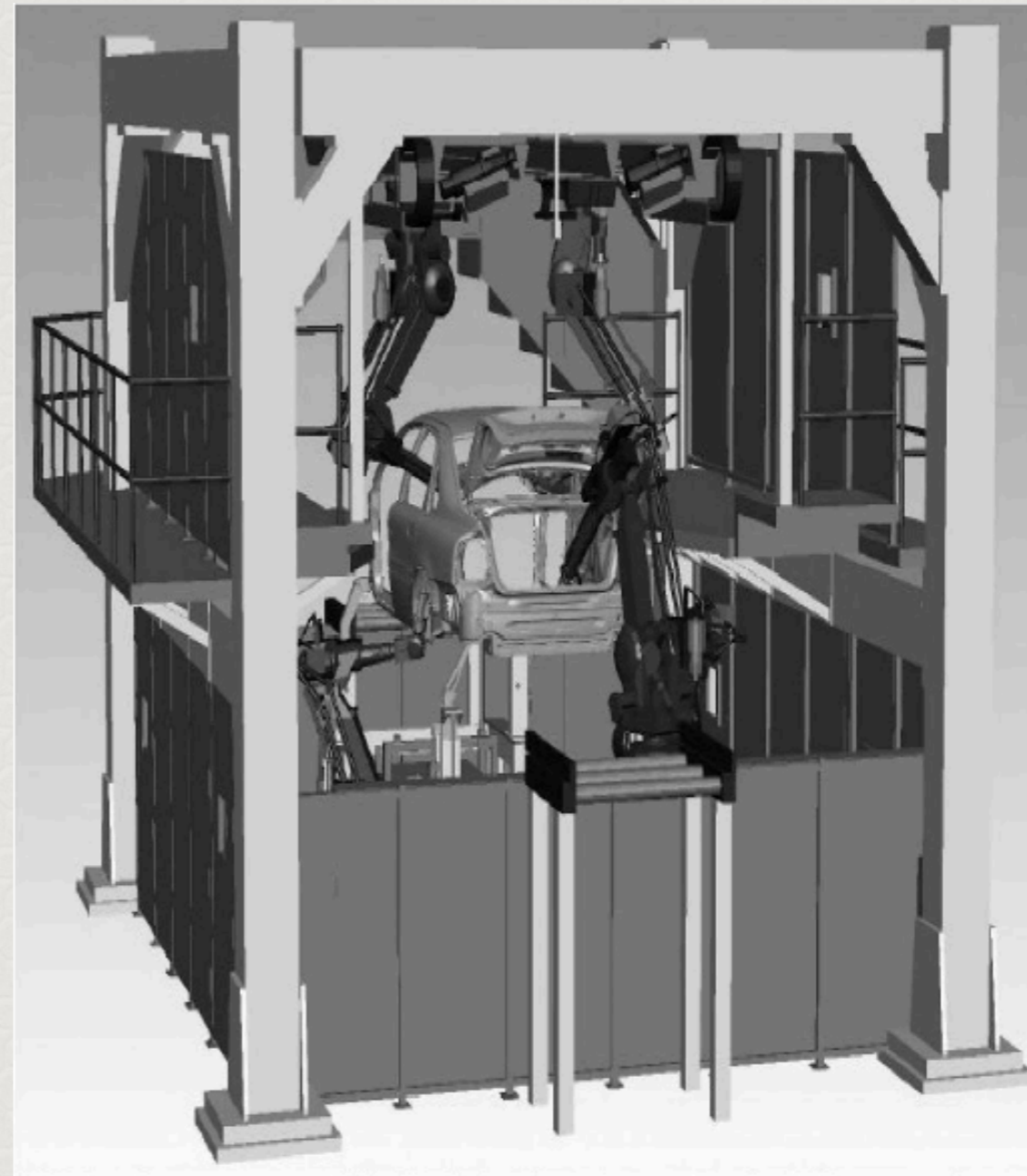


## Algumas limitações:

- a primeira limitação é que vamos supor que temos problemas de planejamento determinísticos: as ações produzem sempre o mesmo efeito, são sempre as mesmas e agem uma de cada vez;
- as ações não causam nenhum efeito acumulativo e portanto a aproximação já feita vale: o problema de planning pode ser formalizado como um sistema dinâmico discreto e abordado com um sistema estado-transição;
- uma vez delimitado o problema de planejamento este não está sujeito a nenhuma influência externa.



## Aplicações práticas





Para chegar nos algoritmos que resolvem problemas reais (e complexos) o primeiro passo é remover por completa o problema da não-linearidade apontada pela anomalia de Sussman.

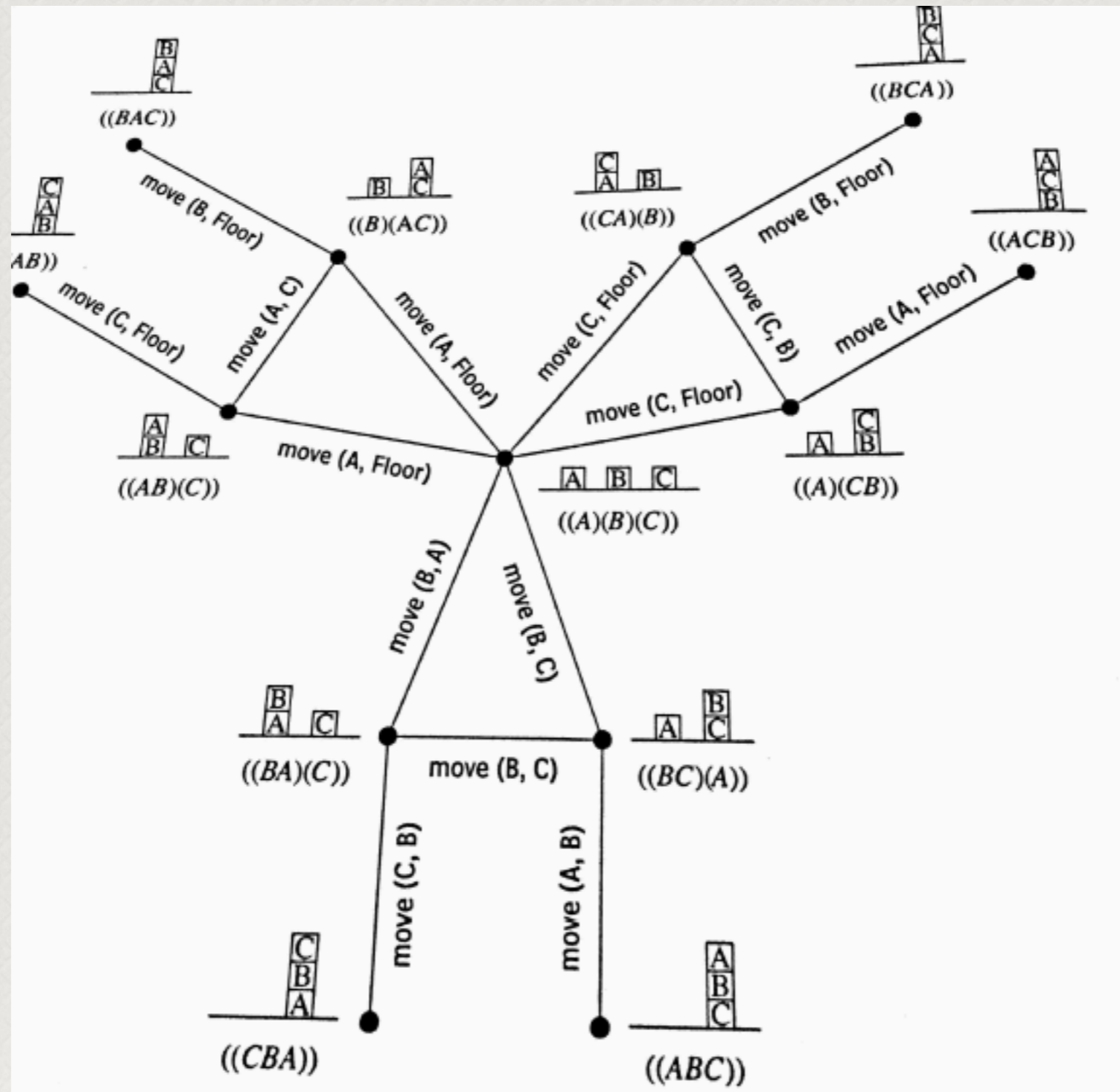


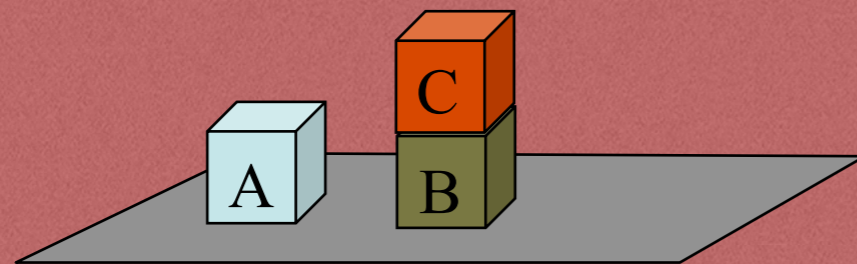


No final dos anos 70 esta é a principal discussão na comunidade, e em 1977 Ashton Tate propôs o algoritmo NONLIN para tratar deste problema

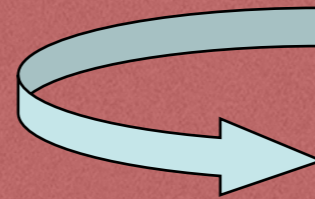


Nilsson, N., Artificial Intelligence: a new synthesis, Morgan Kaufmann, 1998.

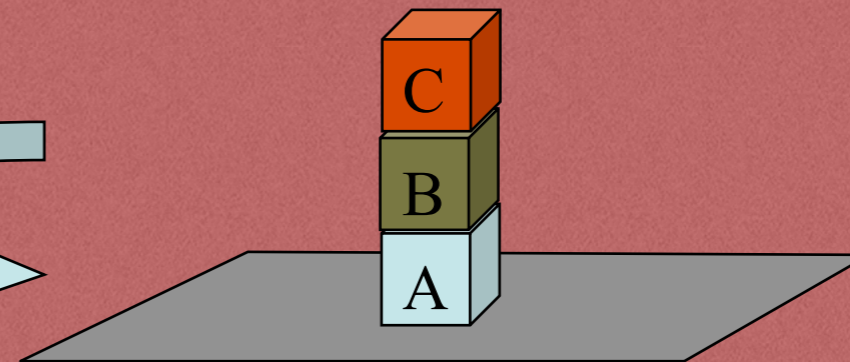




livre(A)  
sobre(A, mesa)  
sobre(C, B)  
sobre (B, mesa)  
livre(C)



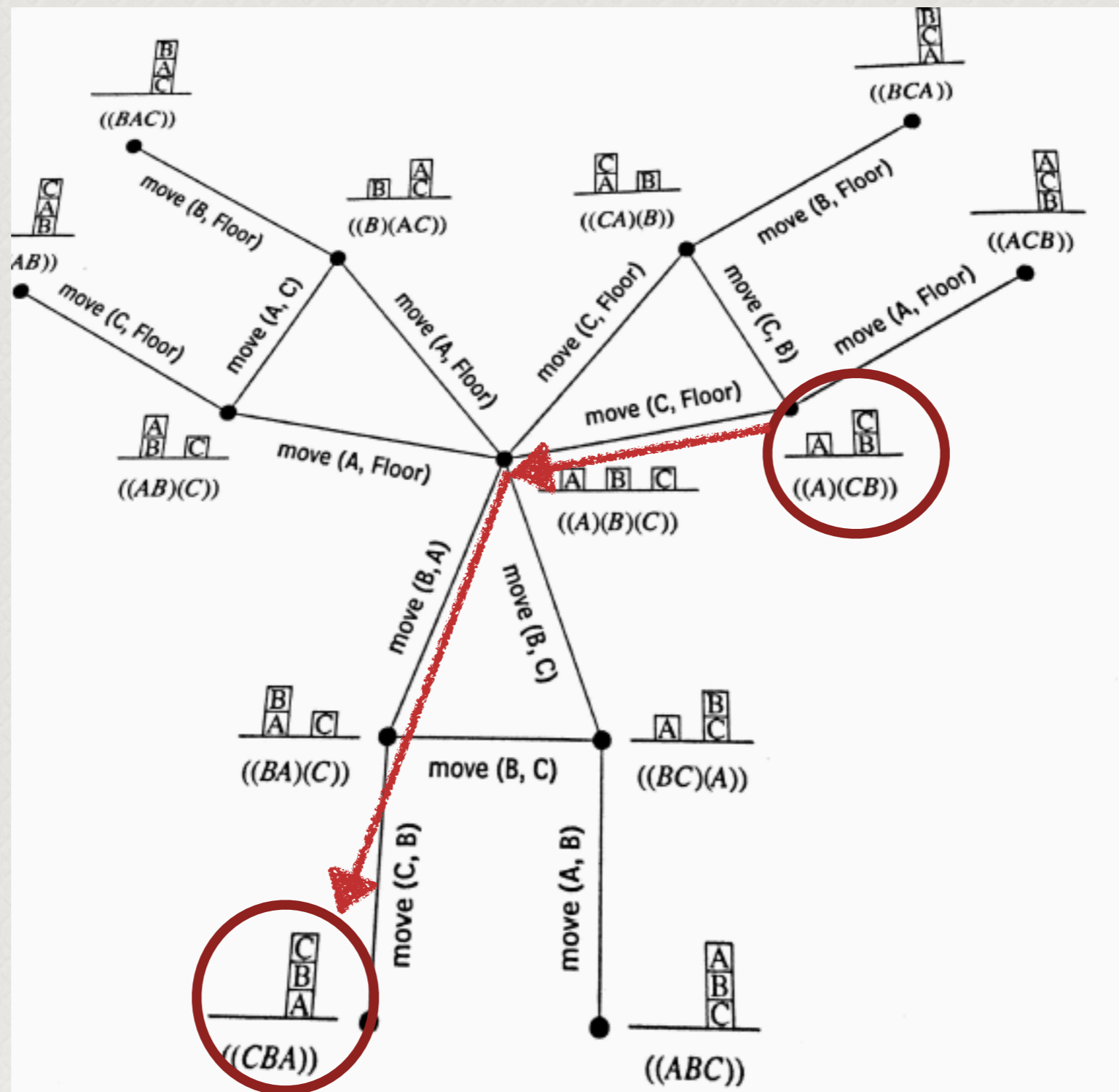
(Anomalia de Sussman)



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

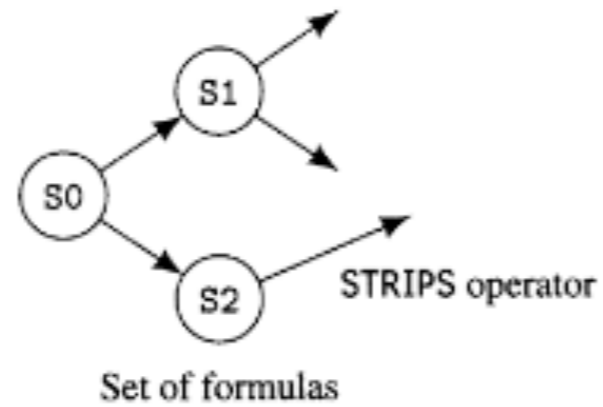


Nilsson, N., Artificial Intelligence: a new synthesis, Morgan Kaufmann, 1998.

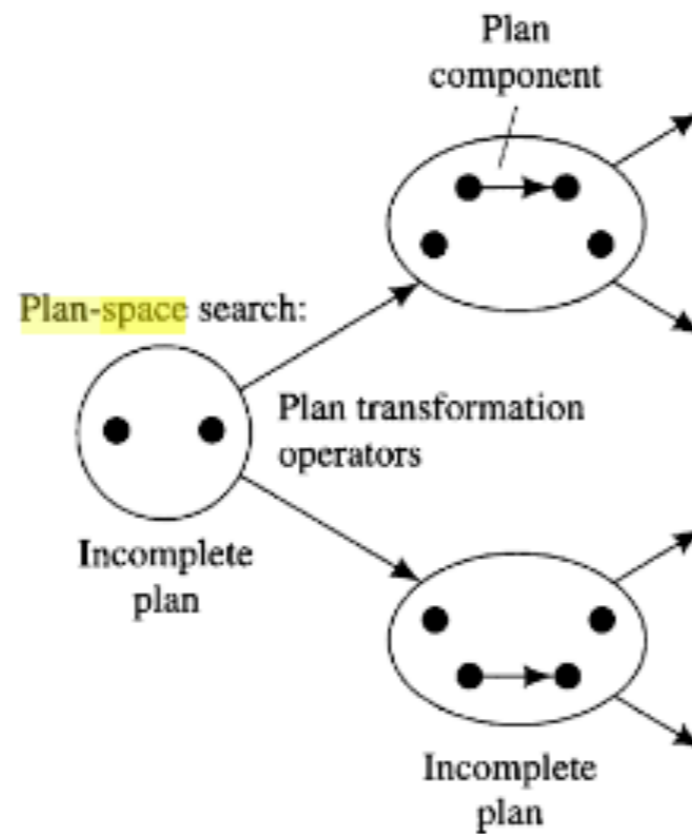


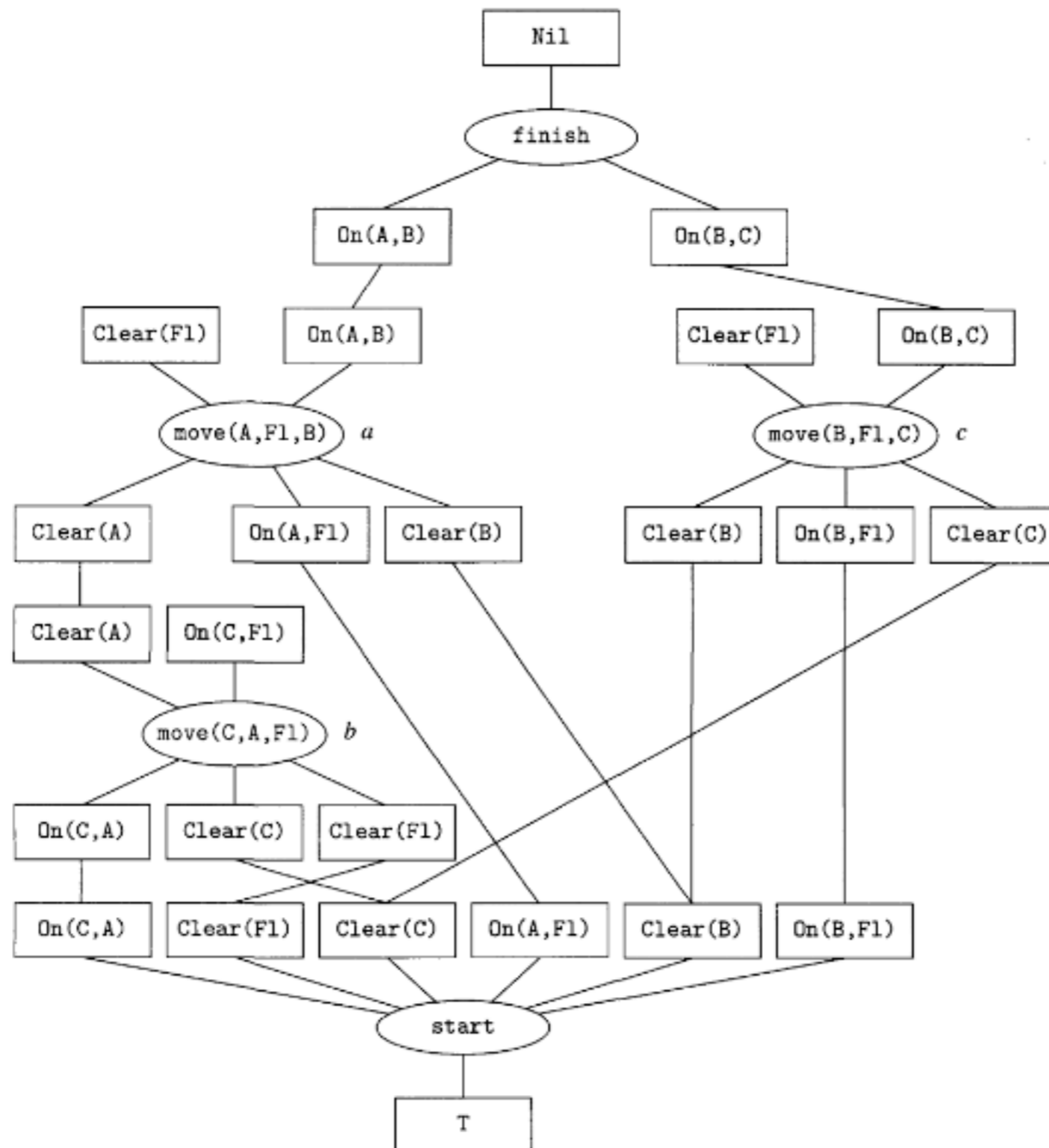


State-space search:



Plan-space search:

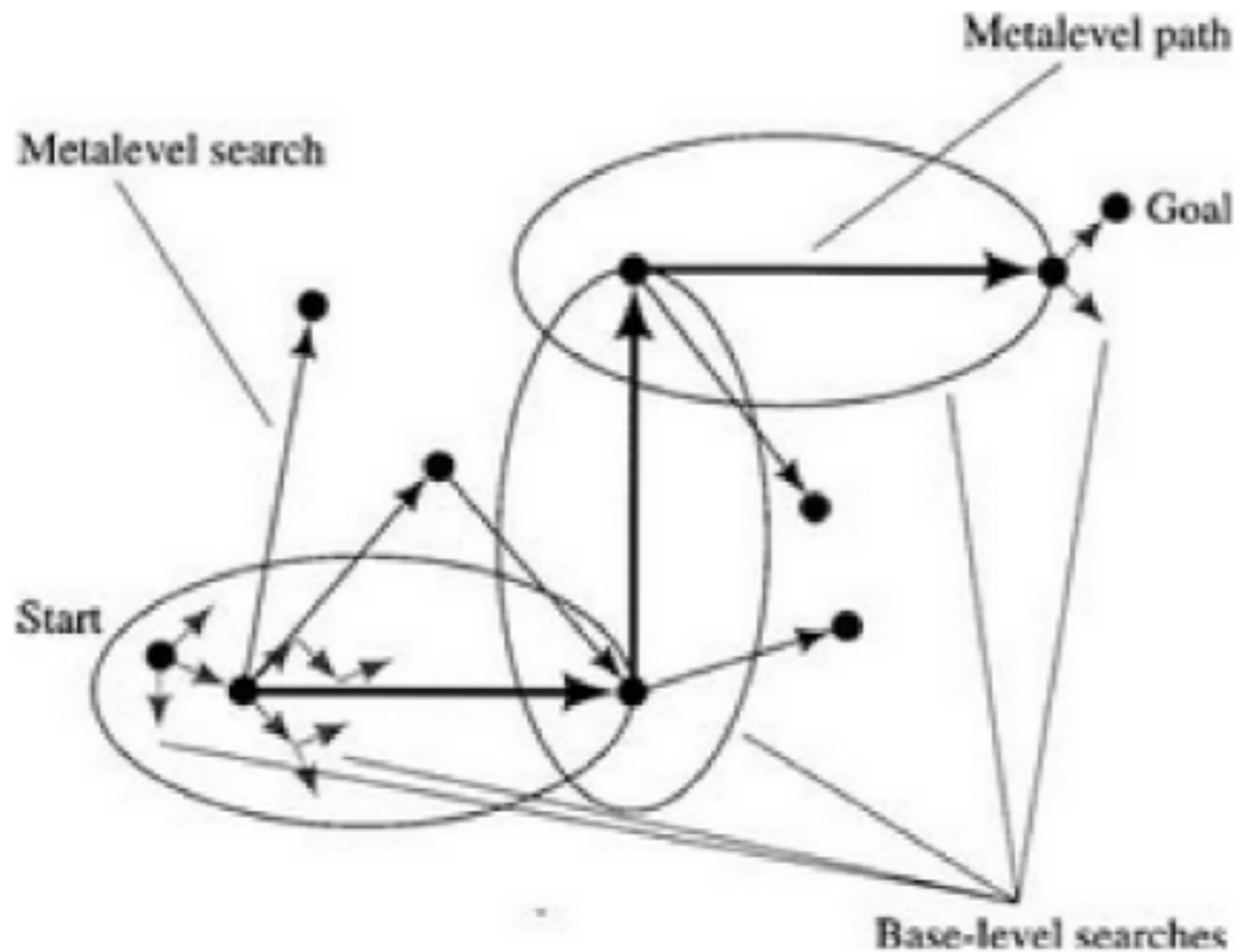






## NOAH: associando não-linearidade, abstração e hierarquia

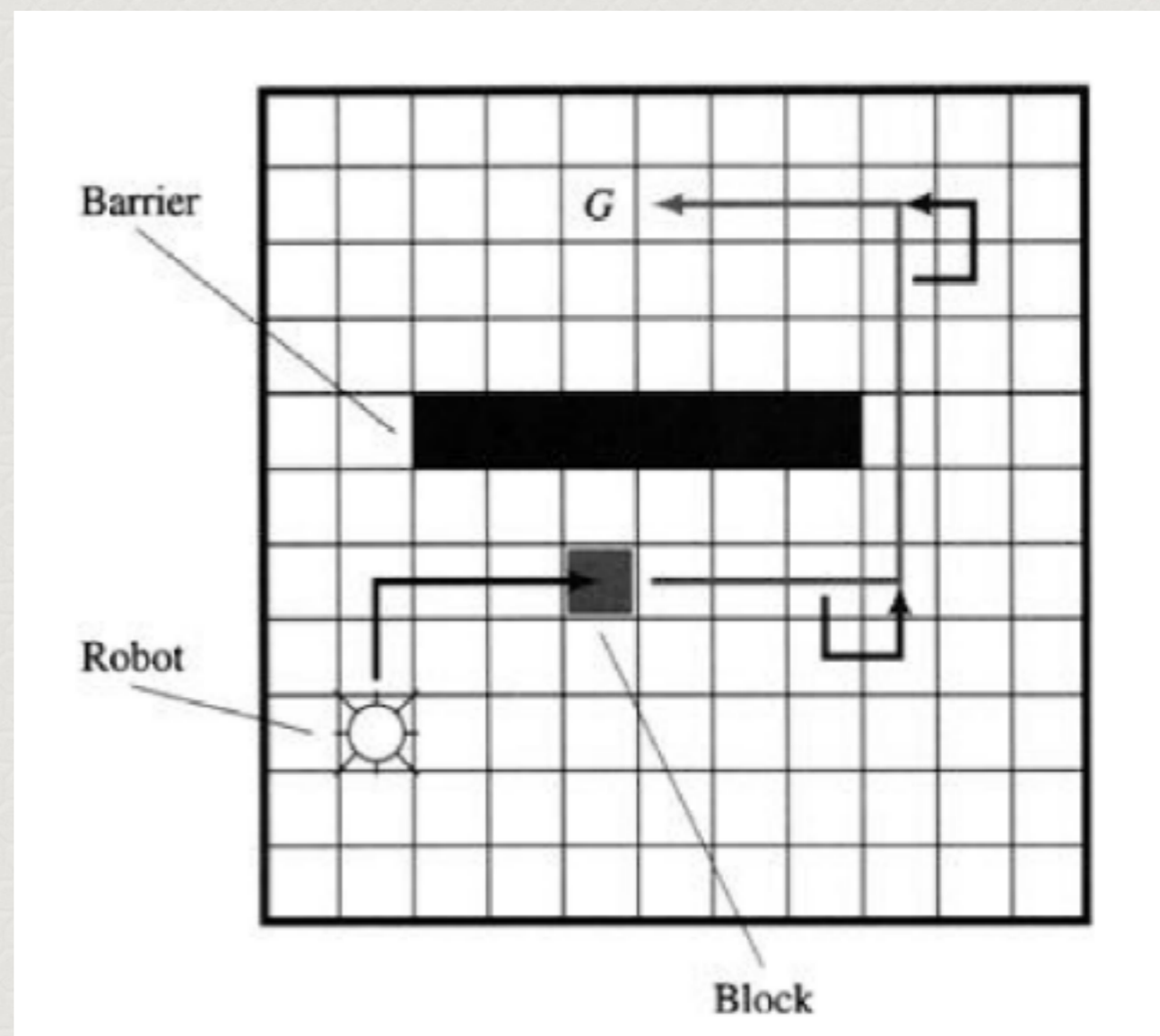
Em 1977 Sacerdoti propôs um algoritmo que ao mesmo tempo dava conta do problema de não-linearidade do planejamento automático, incluindo ao mesmo tempo a abstração do problema e o que na época foi chamado de hierarquia. O algoritmo consiste em analisar o problema de planning e propor um modelo abstrato (uma possível solução) baseado em conhecimento do domínio. Cada trecho deste plano abstrato poderia ser refinada até chegar (ou não) em um plano que possa ser executado com o conjunto de ações admissíveis.







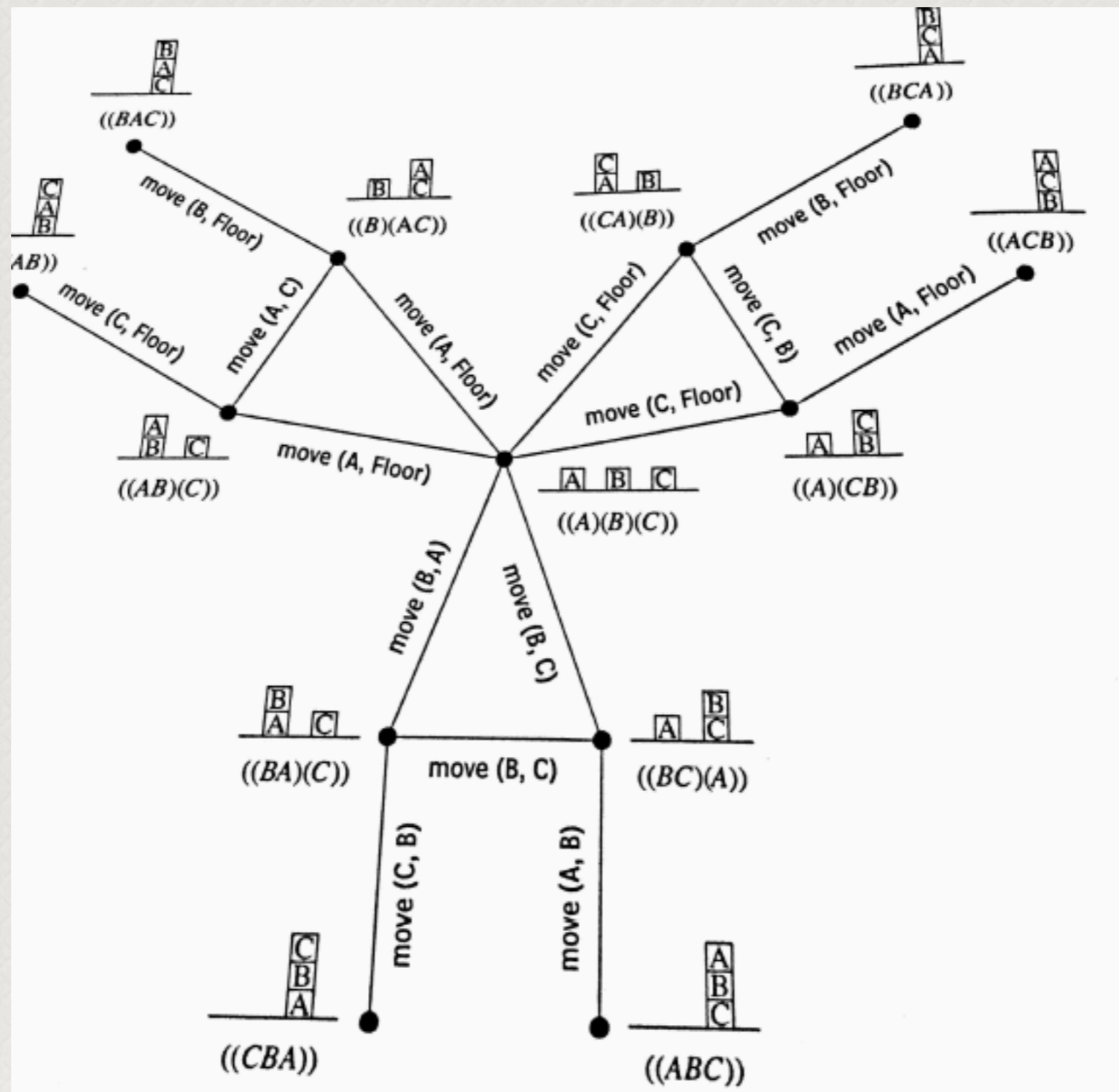
Imagine um robô móvel, que é capaz de "empurrar" caixas em um ambiente fechado evitando obstáculos. Se o robô tem como problema de planning levar a caixa (quadrado preto) até o ponto marcado pela letra G, ele pode dividir este task em vários sub-tasks de forma abstrata (hierárquica) e refinar cada um deles.



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



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





Os algoritmos

NOAH - Sacerdoti, 1977

SIPE - Wilkins, 1988

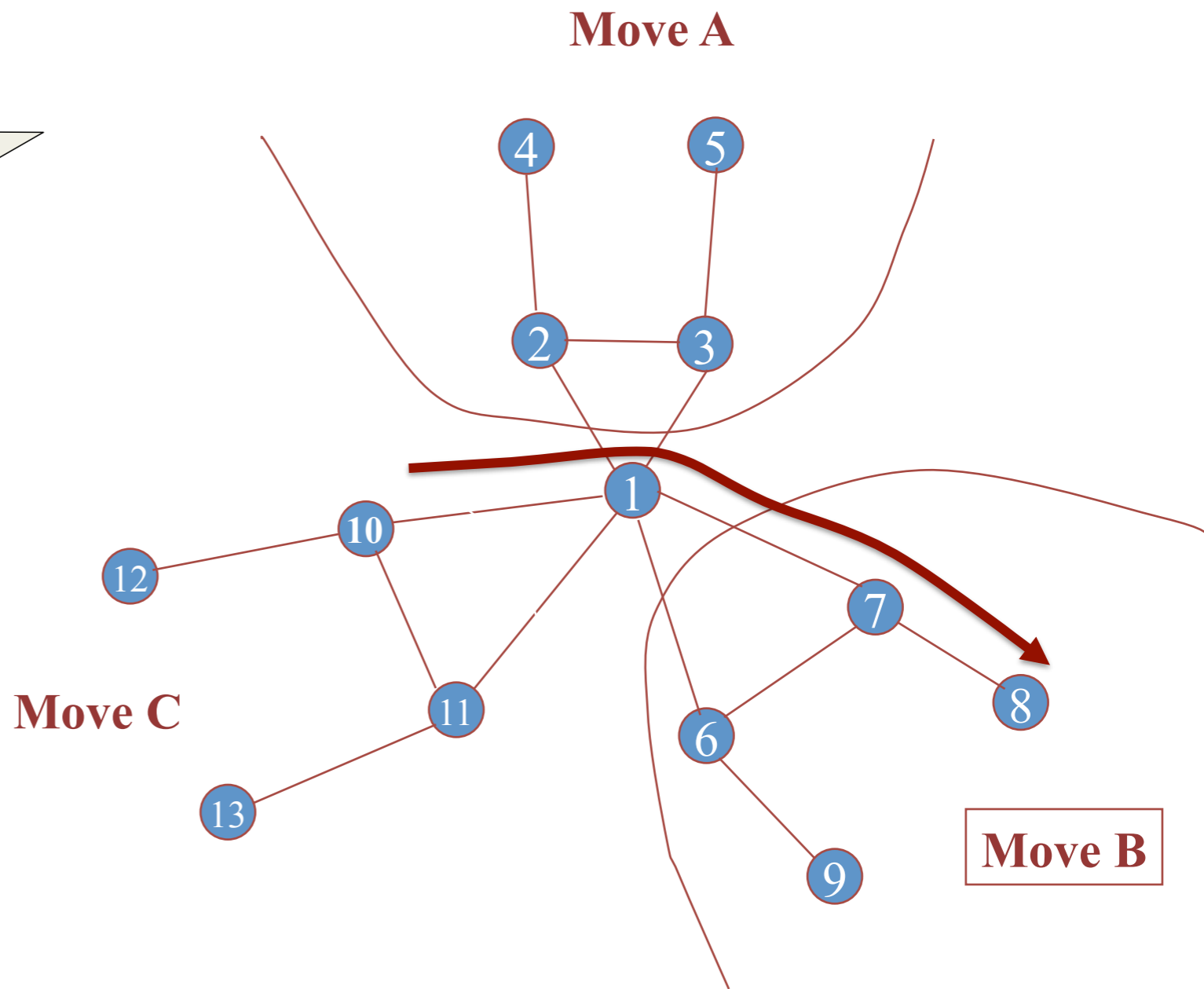
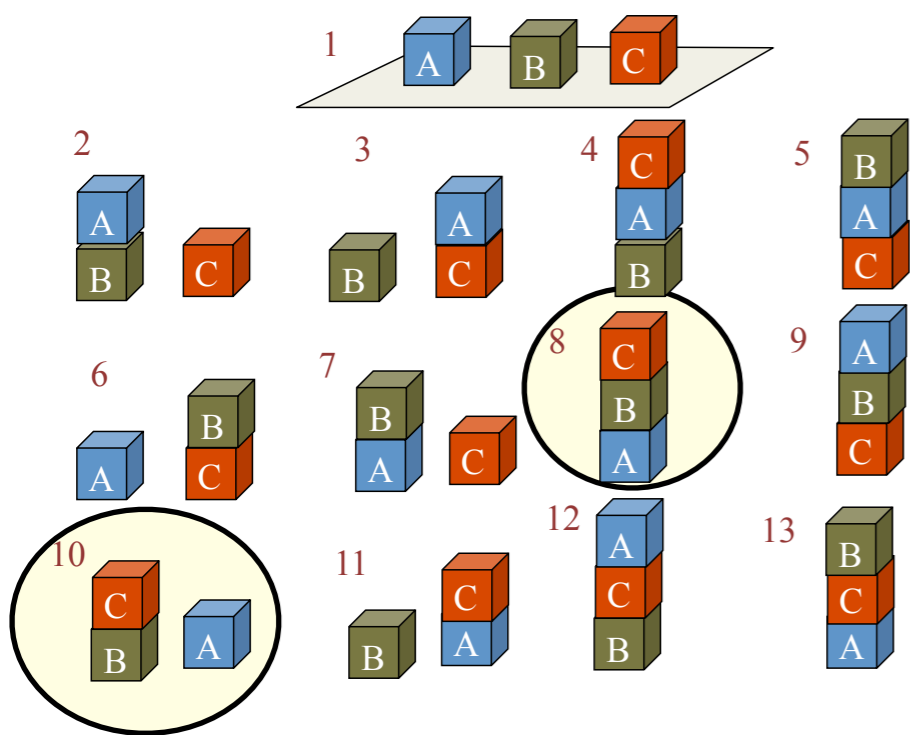
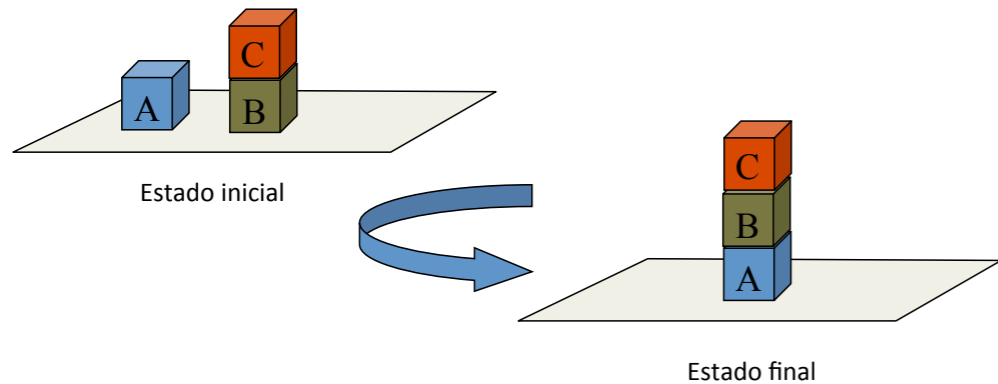
O-Plan - Carrie & Tate, 1991

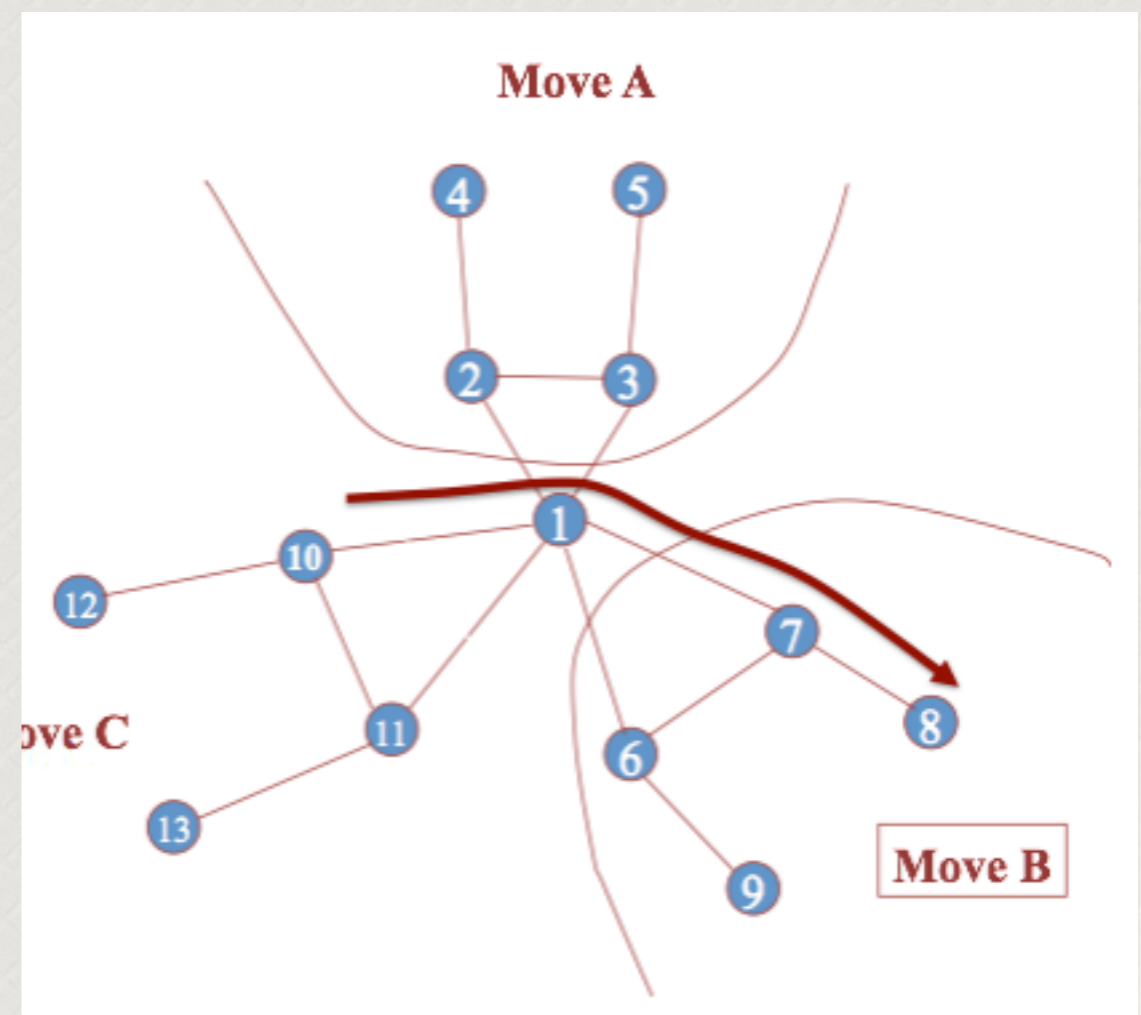
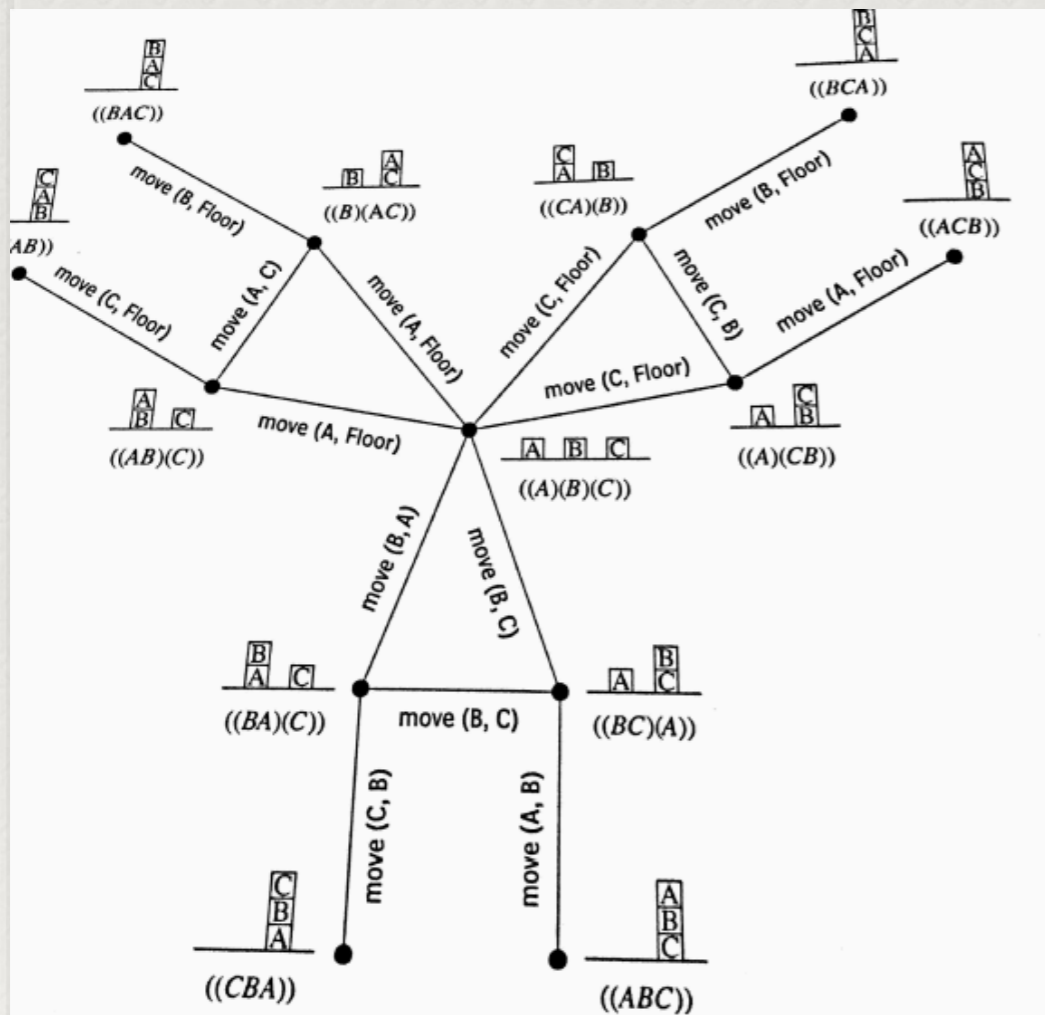
Seguem esta linha, combinando não-linearidade e ordenamento parcial.



## Graphplan

O Graphplan foi proposto por Avrim Blum e Merrick Furst em 1995. Basicamente, o algoritmo consiste em, dado o problema de planning gerar um “planning graph” e mapear o problema sobre ele.







www.cs.cmu.edu/~avrim/graphplan.html

Bookmarks Bar (Chrom... Bookmarks Artificial Intelligence: Notícias Popular Save to Mendeley

## Welcome to the Graphplan Home Page

School of Computer Science, Carnegie Mellon University, Pittsburgh PA 15213-3891  
People: [Avrim Blum](#), Merrick Furst, [John Langford](#)

---

### About Graphplan

**Graphplan** is a general-purpose planner for STRIPS-style domains, based on ideas used in graph algorithms. Given a problem statement, Graphplan explicitly constructs and annotates a compact structure called a Planning Graph, in which a plan is a kind of "flow" of truth-values through the graph. This graph has the property that useful information for constraining search can quickly be propagated through the graph as it is being built. Graphplan then exploits this information in the search for a plan. Graphplan was created by Avrim Blum and Merrick Furst, with subsequent extensions and improvements made by many researchers at many different institutions around the world.

### How to try it out

To try out Graphplan, go to the [Graphplan home directory](#). This directory contains [source code](#), object code for the [DECstation](#) and [Sparcstation](#), a variety of sample domains, and a [README](#) file that describes how to run Graphplan and how to make up your own domains and problems. The program allows you to see an **animation** (in X) of what it's doing. For instance, look at a [simple TSP domain](#). Here is what the animation looks like on a more interesting [Flat Tire World \(fixit\) domain](#) (graph creation omitted). Here is an [explanation of what's going on](#).

### Publications

[A. Blum and M. Furst, "Fast Planning Through Planning Graph Analysis" \[pdf\], \*Artificial Intelligence\*, 90:281--300 \(1997\)](#). This is the original paper that describes the algorithm used.

[A. Blum and J. Langford, "Probabilistic Planning in the Graphplan Framework", in Proceedings of ECP'99. \(c\) Springer-Verlag](#). Describes how the planning graph structure can be used for probabilistic planning. See the [Probabilistic Graphplan](#) page.

### Related Work

Since the initial creation of Graphplan, a number of researchers have pushed these ideas in a variety of exciting directions, including:

- broadening the class of problems for which this style of planning can be applied,
- reducing the running time (via improvements both to the basic strategy and to the code itself), and
- using Graphplan as a preprocessor to other search strategies.

In particular, check out the [BLACKBOX](#) (AT&T/Washington), [IPP](#) (U. Freiburg), [STAN](#) (U. Durham), and [Sensory Graphplan](#) (U. Washington) planners. See also John Langford's [Plan compilation page](#).

---

[Algorithms and Complexity](#) | [Computer Science Department](#) | [School of Computer Science](#)

This page maintained by Avrim Blum (avrim@cs.cmu.edu). Last modified: June 2001

<http://www.cs.cmu.edu/~avrim/graphplan.html>



Até aqui vimos algoritmos clássicos (mais funcionais) que podem resolver alguns problemas de planejamento automático. Este número de problemas é restrito e talvez muito diferente dos problemas atuais. Para os aproximarmos destes problemas teremos que relaxar algumas das restrições assumidas como válidas...



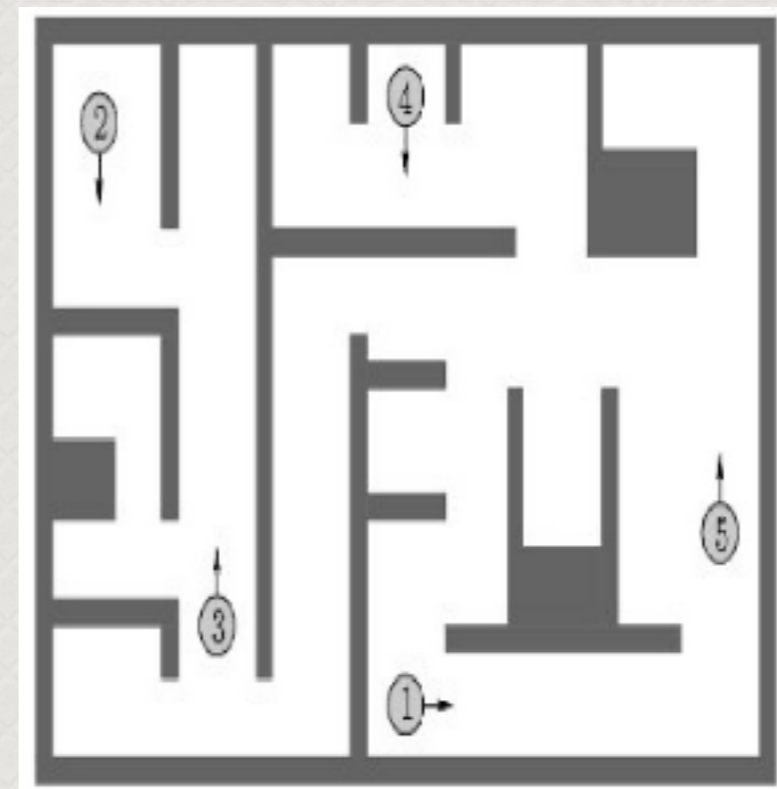


## Algumas limitações:

- a primeira limitação é que vamos supor que temos problemas de planejamento determinísticos: as ações produzem sempre o mesmo efeito, são sempre as mesmas e agem uma de cada vez;
- as ações não causam nenhum efeito acumulativo e portanto a aproximação já feita vale: o problema de planning pode ser formalizado como um sistema dinâmico discreto e abordado com um sistema estado-transição;
- uma vez delimitado o problema de planejamento este não está sujeito a nenhuma influência externa.



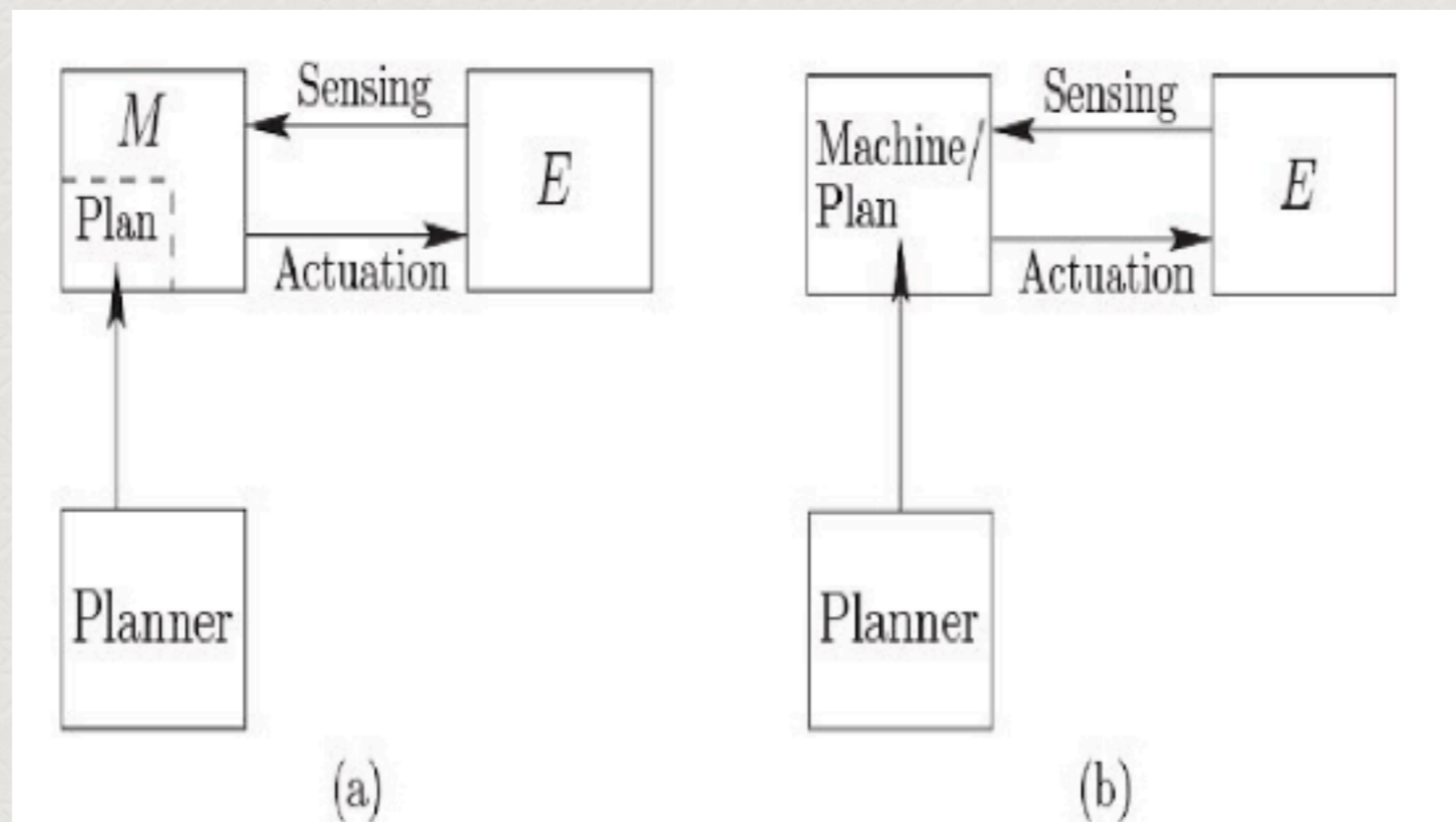
Podemos por exemplo considerar o problema distribuído composto de vários agentes (robôs) que podem por exemplo ter como objetivo encontrar pessoas perdidas (ou objetos). Portanto o estado final não é claramente definido.





Outro problema importante é acoplar planning a detecção e processamento de movimento, um problema essencial para automação.





**Figure 1.18:** (a) A planner produces a plan that may be executed by the machine. The planner may either be a machine itself or even a human. (b) Alternatively, the planner may design the entire machine.

Steven LaValle, Planning Algorithms, Cambridge Associated Press, 2006



A geração de restrições, especialmente associadas equações diferenciais pode elevar em muito a capacidade dos planejadores clássicos, trazendo-os para as aplicações atuais.



*Até a próxima aula!*