



先修基础知识

Topic 1 for AI

Problem Solution

Yang Muyun

MOE-MS Joint Key Lab of NLP&Speech

School of Computer Science and Technology, HIT

A problem

$$\begin{array}{cccc} & & A & B & C \\ & & & & & & \\ + & & C & D & C & & \\ \hline & A & B & C & D & & \end{array}$$

How would solve this problem?

How would you code to solve this problem?

How would you teach the computer to solve this kind of problem



Contents

- 1. State, Operation and Target**
- 2. State Graph**
- 3. State Graph Search**

State, Operation & Target

◆ Three Characteristics in problems

- Eg: Three cannibals and three missionaries want to cross a river by a boat for only two passengers. On either bank, the priests will be killed if there are more savages. How to cross the river?

◆ State

- Number of priest at river banks;
- Number of savages at river banks;
- Which bank the boat is available at;

State, Operation & Target

◆ Initial State

	左 岸	右 岸
传教士(M)	3	0
野 人(C)	3	0
船 (B)	有	无

◆ A Simpler Graph

	左 岸
传教士(M)	3
野 人(C)	3
船 (B)	有

State, Operation & Target

◆ State Space

- Missioners on L-bank: 0, 1, 2, 3
- Cannibals of L-bank: 0, 1, 2, 3
- Boat: Left or Right *bank*
- total: $4*4*2=32$ situations

◆ Illegal states: when C kill M

- $(1+2+3)*2=12$ (think why)

◆ Impractical states: 4 (why)

- $(3,3,0)/(0,0,1)/(3,0,1)/(0,3,0)$ (for the left bank)

State, Operation & Target

All acceptable states: 16

可能达到的合法状态

左 岸			右 岸		
M	C	B	M	C	B
0	1	有	3	2	无
0	2	有	3	1	无
0	3	有	3	0	无
1	1	有	2	2	无
2	2	有	1	1	无
3	1	有	0	2	无
3	2	有	0	1	无
3	3	有	0	0	无
0	0	无	3	3	有
0	1	无	3	2	有
0	2	无	3	1	有
1	1	无	2	2	有
2	2	无	1	1	有
3	0	无	0	3	有
3	1	无	0	2	有
3	2	无	0	1	有

State, Operation & Target

◆ Operation: what cause the state to change

– Operators and Descriptions:

1. 将 1 个传教士从左岸运到右岸
2. 将 1 个野人从左岸运到右岸
3. 将 1 个传教士和 1 个野人从左岸运到右岸
4. 将 2 个传教士从左岸运到右岸
5. 将 2 个野人从左岸运到右岸
6. 将 1 个传教士从右岸运到左岸
7. 将 1 个野人从右岸运到左岸
8. 将 1 个传教士和 1 个野人从右岸运到左岸
9. 将 2 个传教士从右岸运到左岸
10. 将 2 个野人从右岸运到左岸

State, Operation & Target

- ◆ Target: the desired state by applying operations

	左岸	右岸
传教士(M)	0	3
野人(C)	0	3
船(B)	无	无

State, Operation & Target

A Solution to MC problem

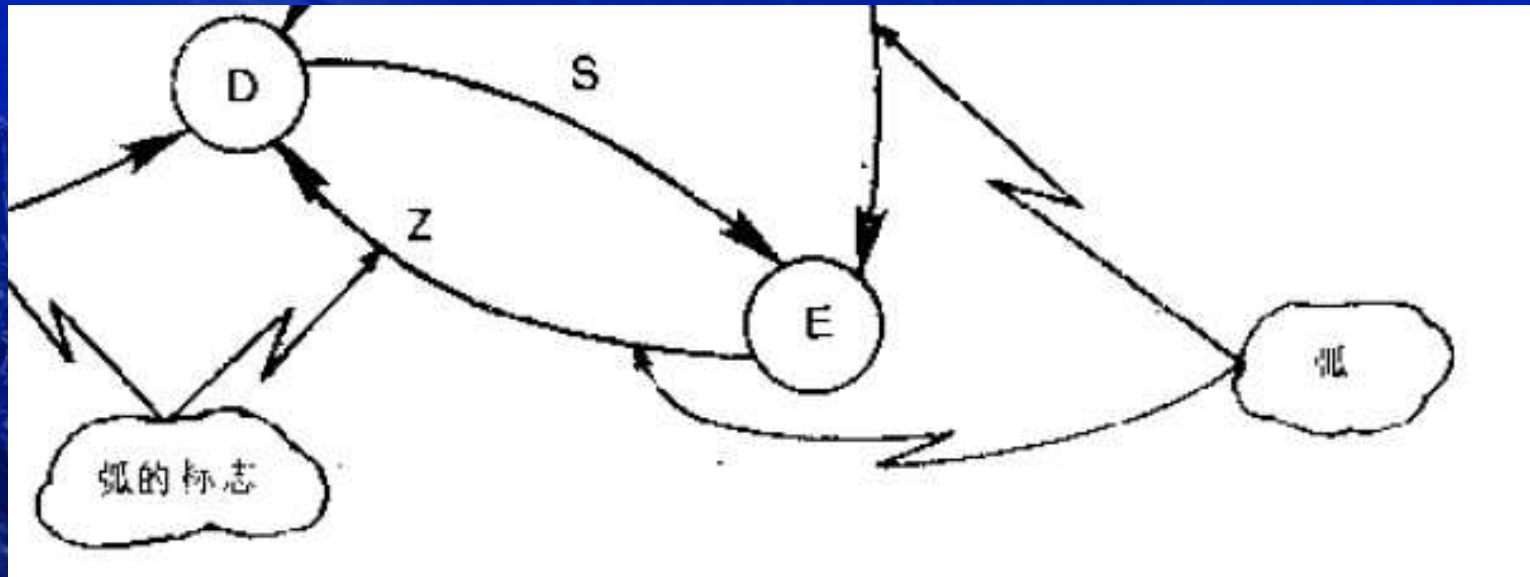
种解答

解 答	状 态					
	左 岸			右 岸		
	M	C	B	M	C	B
初始位置	3	3	有	0	0	无
将 1 个传教士和 1 个野人从左岸运到右岸	2	2	无	1	1	有
将 1 个传教士从右岸运到左岸	3	2	有	0	1	无
将 2 个野人从左岸运到右岸	3	0	无	0	3	有
将 1 个野人从右岸运到左岸	3	1	有	0	2	无
将 2 个传教士从左岸运到右岸	1	1	无	2	2	有
将 1 个传教士和 1 个野人从右岸运到左岸	2	2	无	1	1	无
将 2 个传教士从左岸运到右岸	0	2	无	3	1	有
将 1 个野人从右岸运到左岸	0	3	有	3	0	无
将 2 个野人从左岸运到右岸	0	1	无	3	2	有
将 1 个传教士从右岸运到左岸	1	1	有	2	2	无
将 1 个传教士和 1 个野人从左岸运到右岸	0	0	无	3	3	有

注：M 传教士 C 野人 B 船

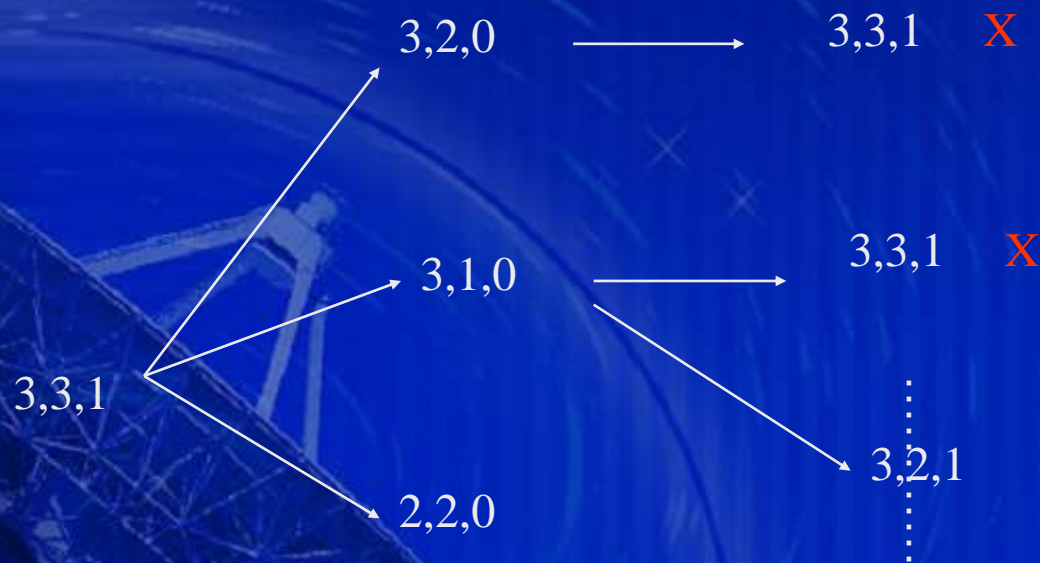
State Graph

- ◆ Definition for Graph
 - Composed by *dot* and *arc*
 - Not function graph or data graph
- ◆ Directed & Undirected Graph
 - *arc representing operation*



State Graph

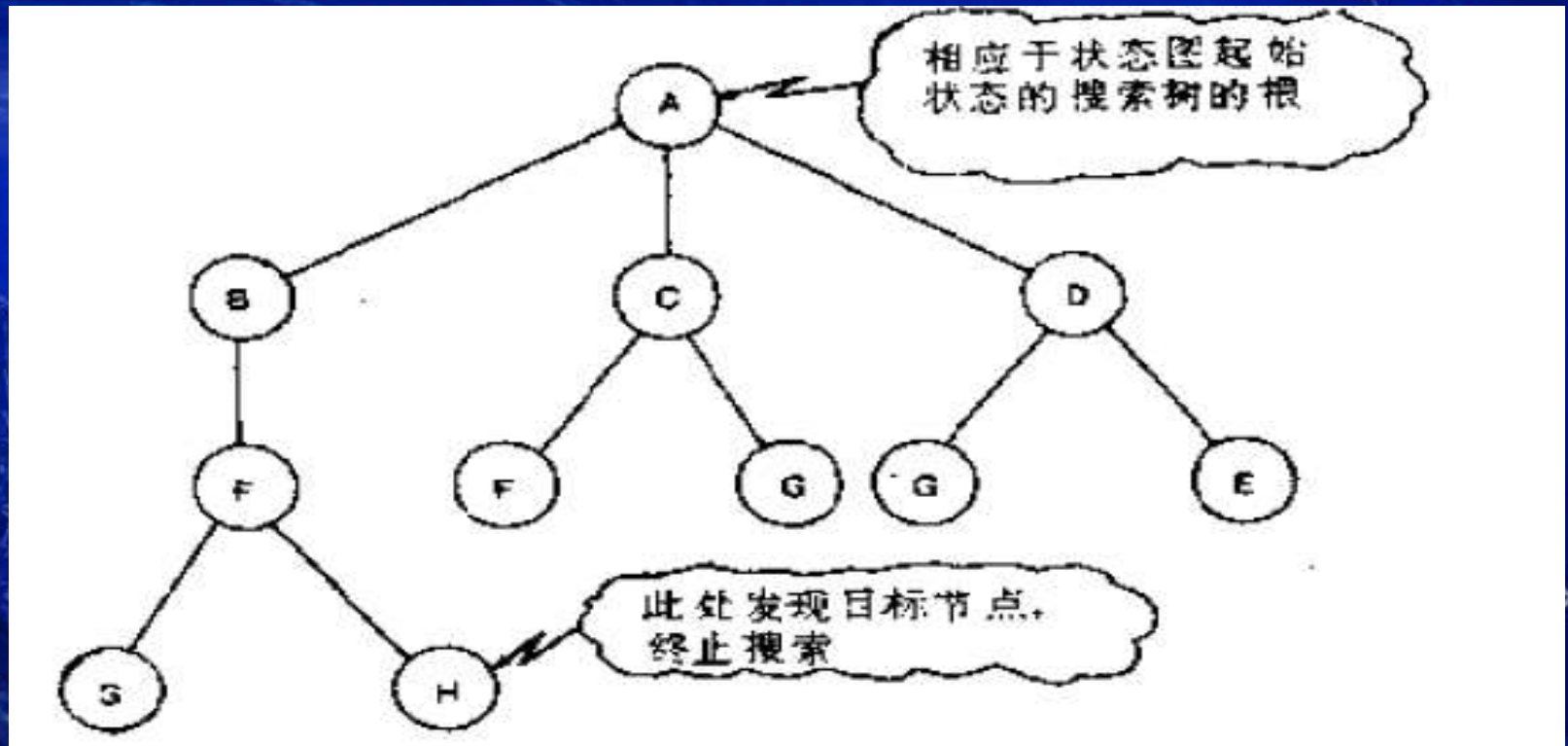
Directed graph represented by *Tree*



State Graph Search

◆ Search Tree

- Root node: initial state
- Leaf node: where targets exist



State Graph Search

- ◆ Problem solution by tree searching
 - Each tree node represents a state;
 - Root node: initial state
 - Search: expand children nodes from the father node according to conditions, i.e. node expansion corresponds to state search;
 - Node expansion sequence represents search strategy;
 - When the target node is expanded, the solution is found;

State Graph Search

◆ Width-first search

- 首先扩展根节点
- 接着扩展根节点的所有后继节点
- 然后再扩展后继节点的后继，依此类推
- 在下一层任何节点扩展之前搜索树上的本层深度的所有节点都已经被扩展

◆ Performance

- 总能找到一个解
- 如果每步扩展的耗散相同时，广度优先搜索能找到最优解
- 内存消耗是比执行时间消耗更大的问题
- 指数级的时间消耗

State Graph Search

◆ Depth-first search

- 总是扩展搜索树的当前扩展分支(边缘)中最深的节点
- 搜索直接伸展到搜索树的最深层，直到那里的节点没有后继节点
- 那些没有后继节点的节点扩展完毕就从边缘中去掉
- 然后搜索算法回退下一个还有未扩展后继节点的上层节点继续扩展

◆ Performance

- 内存需求少—如分支因子= b /最大深度= m 的状态空间深度优先搜索只需要存储 b^{m+1} 个节点(比较广度优先 $O(b^{d+1})$)
- 不是完备的 / 不是最优的
- 最坏情况下时间复杂性也很高 $O(b^m)$

Implementation in Computer

◆ State: Variable-value

例如，我们可以令 M 代表传教士的数目，用 C 代表野人的数目，如果船在左岸，我们可以令 B 的值为 1(或 Yes)，如果船在右岸则 B 值为 0(或 NO)，那么下表所列的值：

变 量	值
M	3
C	1
B	1

Implementation in Computer

◆ Operation: operation/calculatn

1. 将 1 个传教士从左岸运到右岸

$M = M - 1, B = 0$

2. 将 1 个野人从左岸运到右岸

$S = S - 1, B = 0$

3. 将 1 个传教士和 1 个野人从左岸运到右岸

$M = M - 1, S = S - 1, B = 0$

4. 将 2 个传教士从左岸运到右岸

$M = M - 2, B = 0$

5. 将 2 个野人从左岸运到右岸

$S = S - 2, B = 0$

6. 将 1 个传教士从右岸运到左岸

$M = M + 1, B = 1$

7. 将 1 个野人从右岸运到左岸

$S = S + 1, B = 1$

8. 将 1 个传教士和 1 个野人从右岸运到左岸

$M = M + 1, S = S + 1, B = 1$

9. 将 2 个传教士从右岸运到左岸

$M = M + 2, B = 1$

10. 将 2 个野人从右岸运到左岸

$S = S + 2, B = 1$

Implementation in Computer

- ◆ Get target: search among operations

<p>4. $B=1$</p> <p> $AND((M=3 \text{ AND } C=1)$</p> <p> $OR(M=2 \text{ AND } C=2))$</p>	<p>LET $M=M-2$</p> <p>LET $B=0$</p>
<p>5. $B=1$</p> <p> $AND C \geq 2$</p> <p> $AND(M=0 \text{ OR } M=3$</p>	<p>LET $C=C-2$</p> <p>LET $B=0$</p>

Summary

- ◆ Problem solution is the process to search for the target.
- ◆ Problem solution is often referred as search technology.
- ◆ This chapter introduces the problem solution via state space search.

Assignment

$$\begin{array}{cccc} & A & B & C \\ + & C & D & C \\ \hline A & B & C & D \end{array}$$

109

989

1098

写出伪代码，用时不超过算出这个结果！

本章只描述：状态空间求解

- ◆ 是搜索问题中的一种
- ◆ 是问题求解的一种
- ◆ 是人工智能最初提出的技术，要掌握这种思维方式。
- ◆ 但是，后续不断出现不同的方法，新的技术，似乎更有效
 - 有效：可观察到速度、性能等维度的提升，也可以是更好理解、更方便应用
 - 为什么？是这些方法真的更高级？最高级何在？

问题没有那么简单

- ◆ 体会了前文的精妙，经常有人会以为掌握了人工智能的真髓，天下无不可去之处...
- ◆ 但是，从状态空间搜索技术诞生，到今天的机器翻译，人工智能经历了很多：
 - 方法论变化了
 - 方法模型更新了
 - 技术发展了
- ◆ 这一切很多都会体现在自然语言处理课上，请同学们多多体会。

课下作业1：汉字的计算机表示

- ◆ 请回溯汉字当初为什么无法在计算机内表示；
- ◆ 请梳理支持汉字的字符编码方式。
- ◆ 2001年，中国工程院颁发了“二十世纪我国重大工程技术成就”评选结果，“汉字信息处理与印刷革命”当选第二项，比第一项“两弹一星”仅差一票。

请谈谈你对这件事的看法