

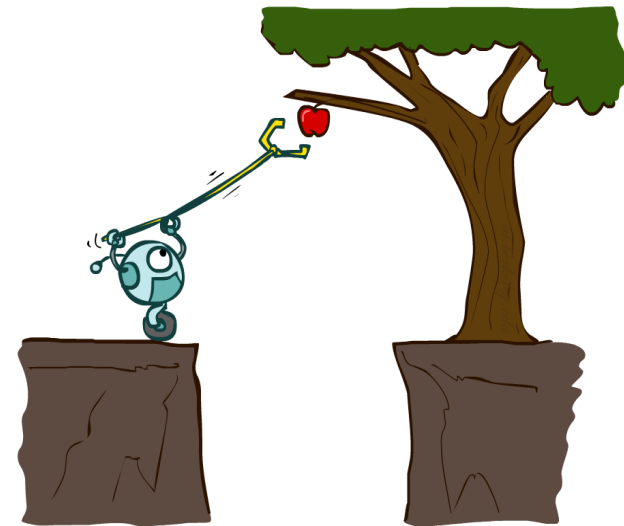
人工智能导论

主讲：王博

人工智能与自动化学院

第1.5章 人工智能与问题求解

人工智能与计算机求解问题的基本范式



为什么是问题求解Problem Solving?

- 北京大学数学学院信息科学系教授，裘宗燕老师：

“计算机工作者关注的最基本问题是各种问题及其计算机求解，以及如何从遇到的问题得到能解决问题的计算机系统。”

Ref

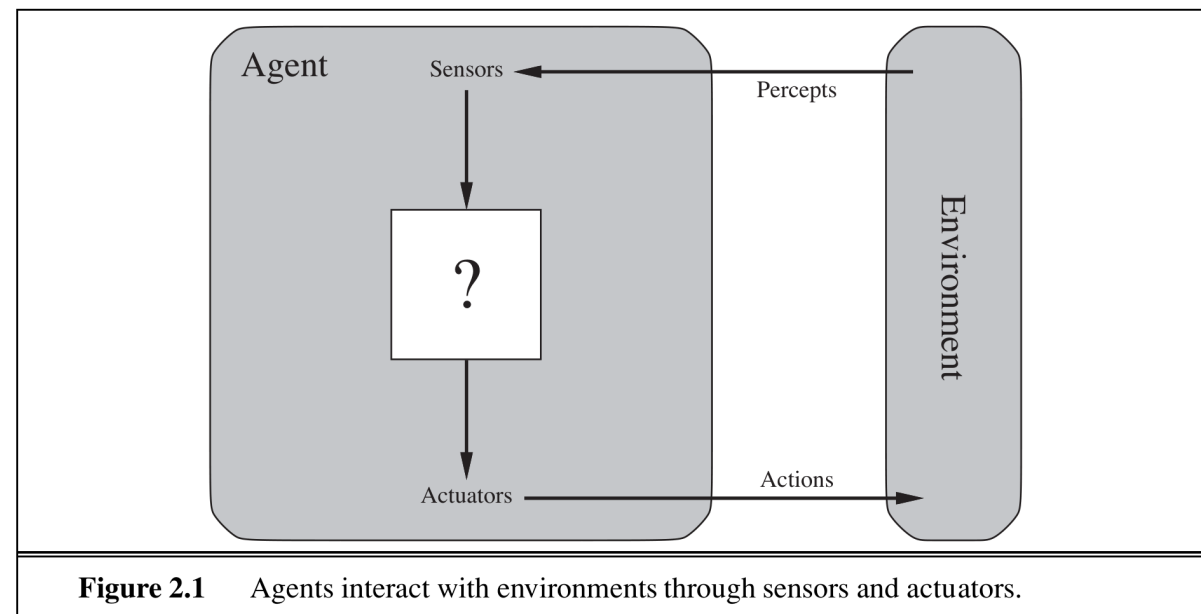
CCCF专栏 | 裘宗燕：计算机问题求解的三类方法



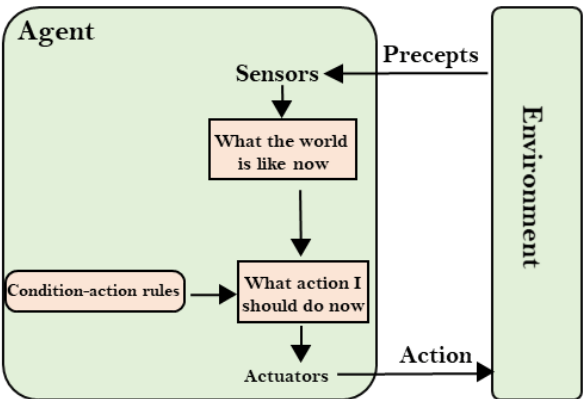
智能体-环境范式

- 按照感知能力和处理方式分成五类：
 - 简单反射式智能体 Simple Reflex Agent
 - 模型反射式智能体 Model-based reflex agent
 - 基于目标智能体 Goal-based agents
 - 基于效用智能体 Utility-based agent
 - 学习智能体 Learning agent

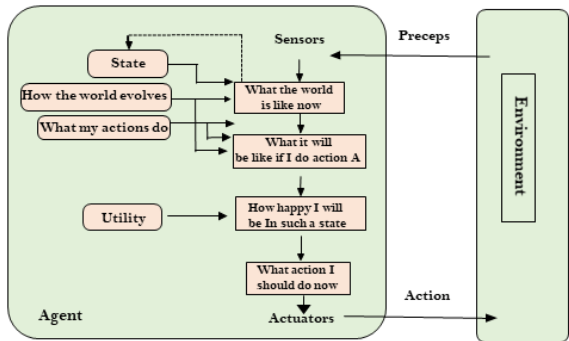
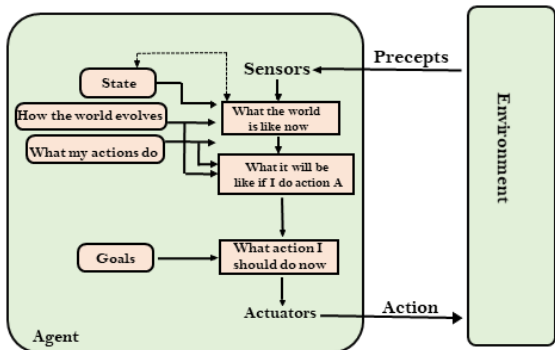
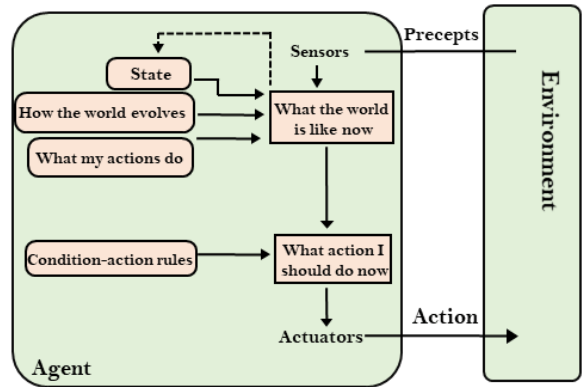
[Types of AI Agents - Javatpoint](#)



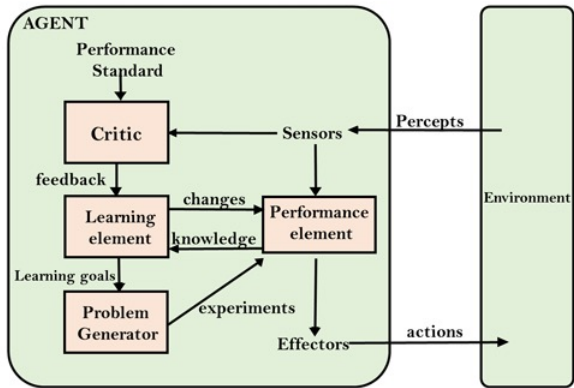
智能体-环境范式



反射式智能体
Reflex Agent



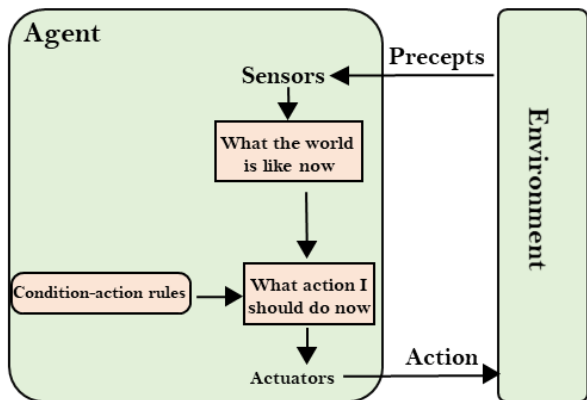
基于目标/效用的智能体
Goal/Utility - based agent



学习智能体
Learning agent

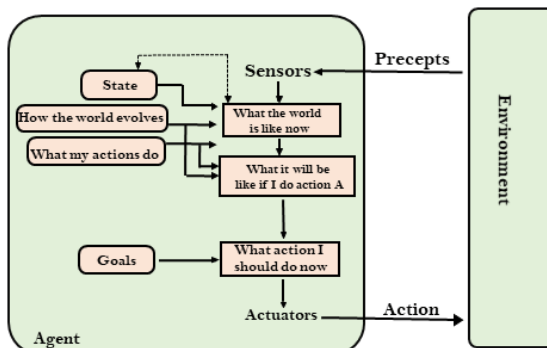
问题求解的三个层次

信息相对问题**充足**
直接求解



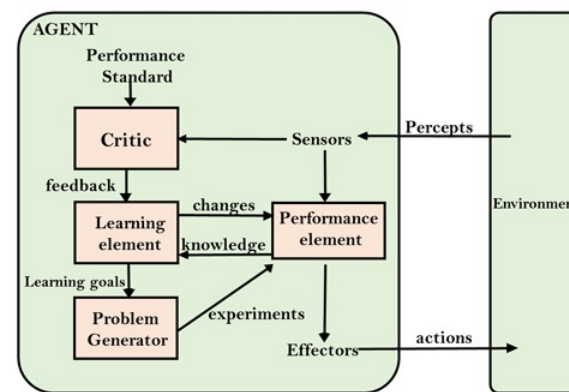
- 反射式智能体：
通过确定的规则(算法)可以解决问题

信息相对问题**较为不足**
需要对问题进行抽象



- 基于目标的智能体：
确定目标后，有时需要针对目标运用探查和回溯(搜索)的方法解决问题

信息相对问题**严重不足**
需要进行机器归纳，建立问题



- 学习智能体：
通过对感知信息的归纳(机器学习)建立对问题的近似，形成问题，解决问题

问题求解的抽象表达

- 考虑一种抽象的统一说法：

- 一个问题就是从一个实例描述集合到解集合的映射：

$$T: I \rightarrow O$$

- 令 $i \in I$ or $(i, o) \in I \times O$ ，则称 i 为问题 T 的实例

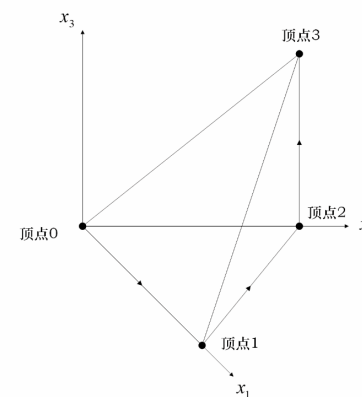
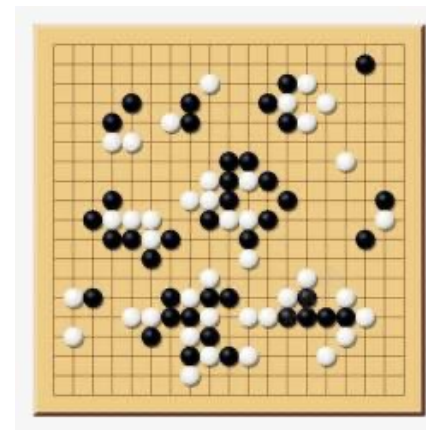
- 如果 I 是有穷集， T 就是有穷问题
 - $o = T(i)$ 为问题 T 对应 i 的解

- 计算机求解问题：

- 写出一个解决这个问题的有穷程序
 - 如果针对问题 T 写了程序 P ，把 T 的任意实例 i 作为 P 的输入， P 的输出总是 $o = T(i)$ ，我们就可以说 P 解决了问题 T 。

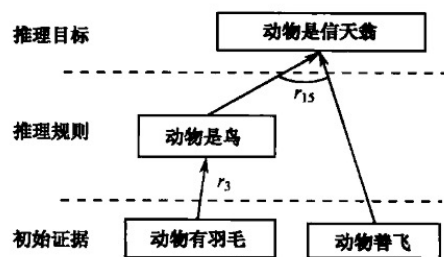
基于算法的系统 (Algorithm-Based Systems, ABS)

- 算法要求对问题的**全面认识**
- 任何有穷问题都有算法。但，以围棋为例：
 - 对任何棋局，确定下一个棋子的最佳位置。
 - 有穷问题。理论上存在最佳算法；
 - 可能局面太多（约为 $3^{361} \approx 10^{172}$ ）
 - 分析每个局面需要的时间太长；
 - 程序太长，没有足够的存储器存放；
- 能用算法解决的问题，必须：
 - 建模后得到的抽象问题是可计算的
 - 算法存在有穷长度的描述，而且描述不“过于长”
 - 每个实例的求解都能在合理时间内完成



基于搜索的系统(Searching-Based Systems, SBS)

- 算法的局限
 - 找不到算法，或者虽然有算法但不适用
 - 缺乏全面认识，但有部分清晰的知识片段
- 另一类方法：**搜索**，通过探查和回溯的方法寻找解。
 - 优势：可能利用部分知识解决问题
 - 缺陷：搜索状态爆炸，规则集不完全
 - 特点：算法不直接针对问题，而是针对规则的使用。规则可独立描述，也可以嵌入搜索算法中。



- 人工智能领域研究过许多搜索方法，开发了一些通用框架（如产生式系统、黑板系统）；
- 提出了许多技术（如各种启发式搜索算法），实现了一些应用（如基于规则的专家系统）；
- 提出了一些编程泛型（如逻辑程序设计、约束程序设计）。还有自动推理、定理证明等方面的大量研究和成果。
- 赫伯特·西蒙（司马贺）等人称搜索为**通用问题求解**(general problem solving)方法。

基于实例的系统 (Case-Based Systems, CBS)

实际情况中

可能对问题的了解非常少，或基本上没有有关求解的知识

- 给定一组输入和结果

$$\{(i_1, o_1), (i_2, o_2), \dots, (i_n, o_n)\}$$

- 是一个问题的**实例**，则有一个简单的求解程序：

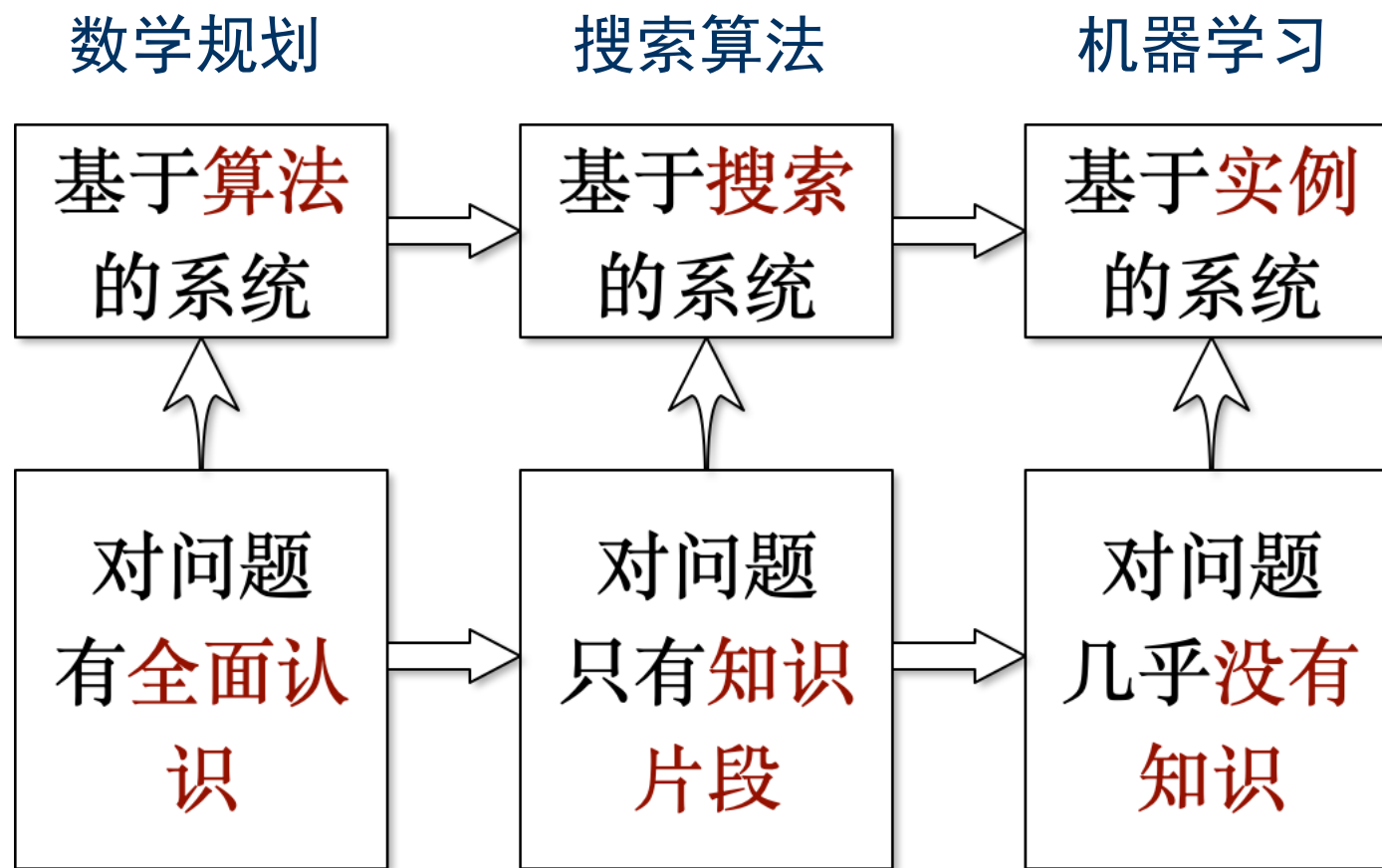
(x 是输入)

```
if x is i1 then o1
elif x is i2 then o2
...
elif x is in then on
else "I can't handle it"
```

- 这里的实例 → 所有可能实例的一部分
- 为了更好“归纳”出规律，需要计算机进行自动推广
- 也就是“**机器学习**”

第2章
第2节

计算机求解问题的三种思路

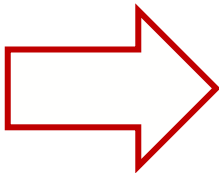


计算机求解问题的三种思路

信息相对问题**充足**
问题完备，可计算，有穷

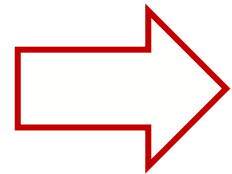
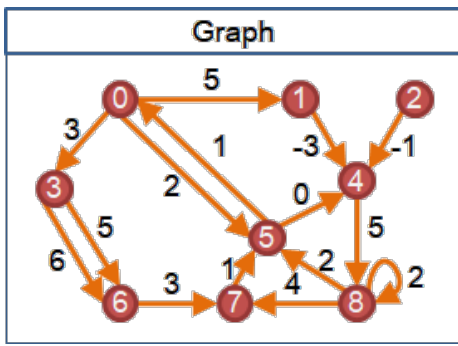
- 寻找最短路径：
直接给出各路径长度

路径	长度
A	21
B	19
...	...
T	35



信息相对问题**较为不足**
对问题认识不完整，有清晰知识片段

- 寻找最短路径：
有地图，有边的长度



信息相对问题**严重不足**
没有直接求解所需的知识，通过实例总结规律

- 寻找最短路径：
环境不确定 + 缺少信息

