

## 《人工智能通识》（科技素养）

# 第2讲 知识推理与问题求解

主讲：丛润民

# 章节知识点概览



知识点1：AI理解之基——知识表示

知识点2：寻根溯源之法——知识推理

知识点3：上下求索之刃——搜索求解

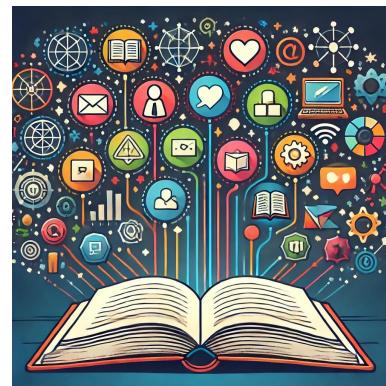
知识点4：田忌赛马的制胜启示——博弈论

知识点5：信任与背叛的无声较量——囚徒困境



# 知识点1：AI理解之基——知识表示

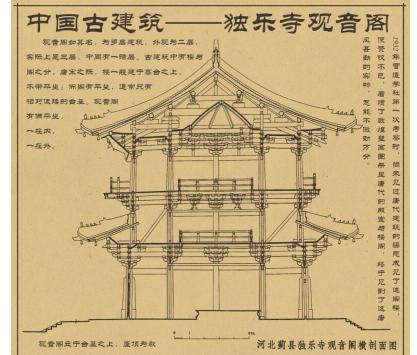
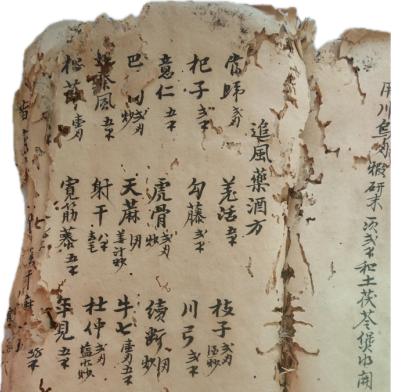
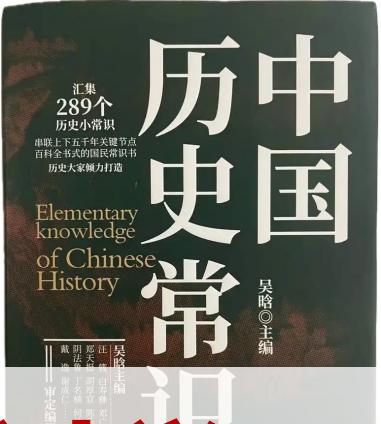
学  
习  
有  
进  
境



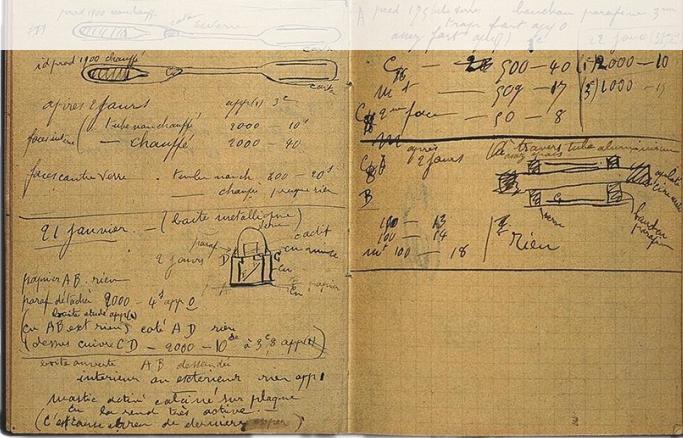
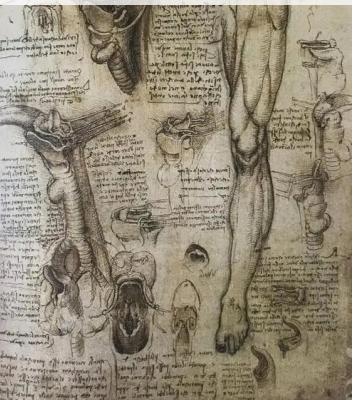
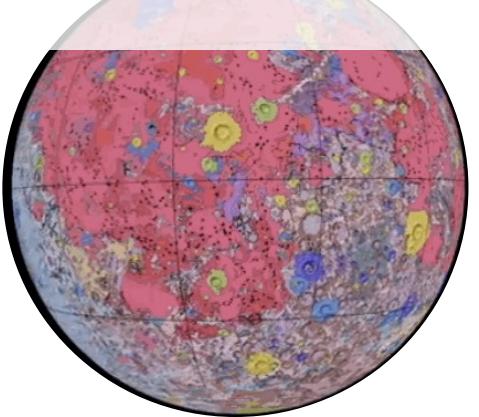
01 知识表示方法

02 逻辑关系与规则

# 什么是知识?



知识是通过学习、经验或推理获得的对事实、信息、原理或技能的理解和掌握。



- 知识表示是指**将现实世界中的知识转化为计算机能够理解和处理的形式**，其目的是使计算机能够储存、检索、理解和操作这些信息，进而实现问题解决和逻辑推理。这一过程**为人工智能系统提供了基础**，使其能够从大量的数据中提取有意义的信息，并支持复杂的推理任务和精准的决策。



## ➤ 符号表示法—逻辑符号

- 符号表示法通过**使用符号和规则来表示知识**，其核心思想是将现实世界中的实体、属性和关系用符号来表示，并通过逻辑规则进行操作，从而实现对知识的表达和推理。
- 符号表示法的基础工具是**逻辑符号**。通过逻辑符号可以清晰地表达命题之间的关系。
  - 与 (AND) 表示两个命题同时为真。例如， $P \wedge Q$  表示 “P和Q都为真” 。
  - 或 (OR) 表示两个命题至少有一个为真。例如， $P \vee Q$  表示 “P或Q为真” 。
  - 非 (NOT) 表示命题的否定。例如， $\neg P$  表示 “P不为真” 。
  - 蕴含 (IMPLIES) 表示一个命题蕴含另一个命题。例如， $P \rightarrow Q$  表示 “如果P为真，则Q为真” 。
  - 等价 (EQUIVALENT) 表示两个命题在逻辑上等价，即它们在所有可能的情况下具有相同的真值。例如， $P \leftrightarrow Q$  表示 “P和Q同时为真或同时为假” 。

## ➤ 符号表示法—命题逻辑和谓词逻辑

- 命题逻辑是一种用来表示和推理关于命题的逻辑系统，它使用命题和逻辑连接词来构建复杂的逻辑表达式。



在命题逻辑中，可以通过“今天下雨”和“我带伞”两个命题，通过逻辑符号“ $\wedge$ ”结合成“今天下雨且我带伞”。

$P$ : “今天下雨”	$Q$ : “我带伞”	$P \wedge Q$ : “今天下雨且我带伞”
0	0	0
0	1	0
1	0	0
1	1	1

命题 $P \wedge Q$ 的真值表

## ➤ 符号表示法—命题逻辑和谓词逻辑

- 谓词逻辑则是一种扩展了命题逻辑的表示方法，它不仅能够表示命题本身，还能够通过引入变量、量词等元素，描述对象之间更复杂的关系和属性。
- 变量代表对象，可以取不同的值。谓词表示对象的属性或对象之间的关系。量词用于描述对象集合的性质。个体域表示谓词中变量的取值范围。



“武汉是个美丽的城市”用谓词逻辑表示为“**美丽的城市(武汉)**”。



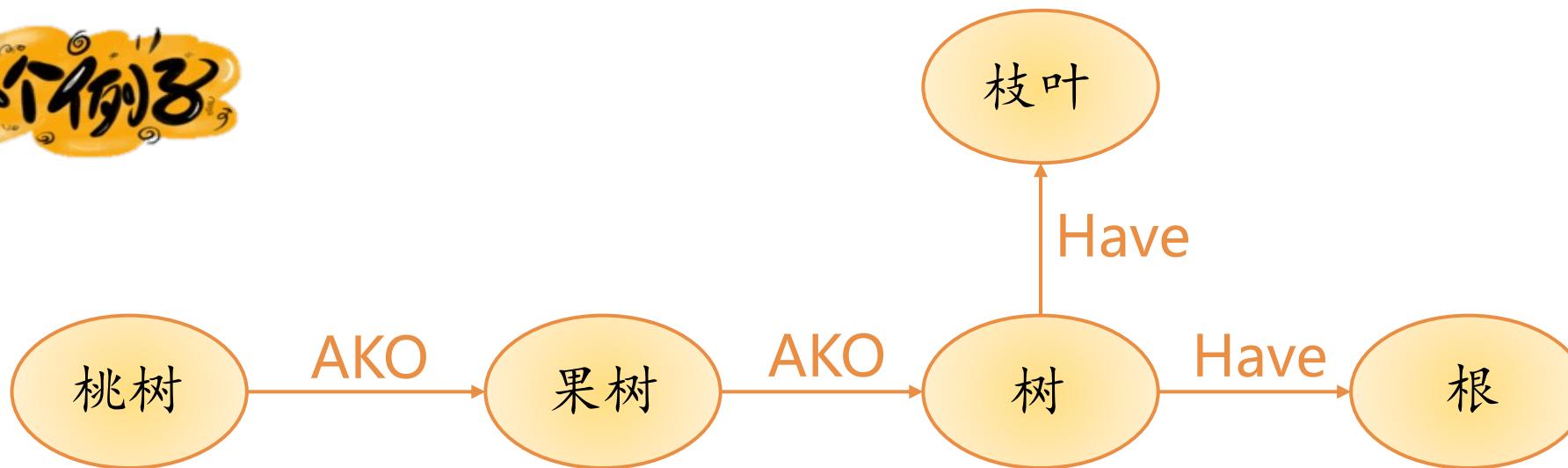
“机器人站在墙边，手里没有拿东西，桌子上放着积木”  
用谓词逻辑表示为

“**在旁边站着(机器人,墙)  $\wedge$  手空着(机器人)  $\wedge$  在上面(积木,桌子)**”。

## ➤ 语义网络表示法

- 语义网络是一种基于图的知识表示方法，用于表示概念及其相互关系。
- 语义网络通过**节点表示概念或实体**，通过**弧表示概念之间的关系**。
- 节点可以表示对象、属性或事件，弧则表示这些节点之间的各种关系，如“是一种”(AKO)、“是一个”(ISA)、“有”(Have)、“是...一部分”(Part-Of)等。

举个例子



## ➤ 框架表示法

- 框架是一种描述某类对象的典型特征的**结构化数据格式**。一个框架由多个“槽”组成，每个槽表示对象的一个属性，槽值则是该属性的具体值。
- 框架表示法的主要优势在于它能够通过**继承机制**，简化和扩展知识的表示。框架表示法特别适用于表示现实世界中的复杂事物，能够处理不同实例之间的共同特征。



在描述教师的框架时，可以定义“姓名”、“年龄”、“性别”等槽，并为每个槽指定具体的值，如“张三”、“30岁”、“男”。

框架名：〈教师〉

姓名：姓、名

年龄：xxx岁

性别：范围(男、女)

住址：〈住址框架〉

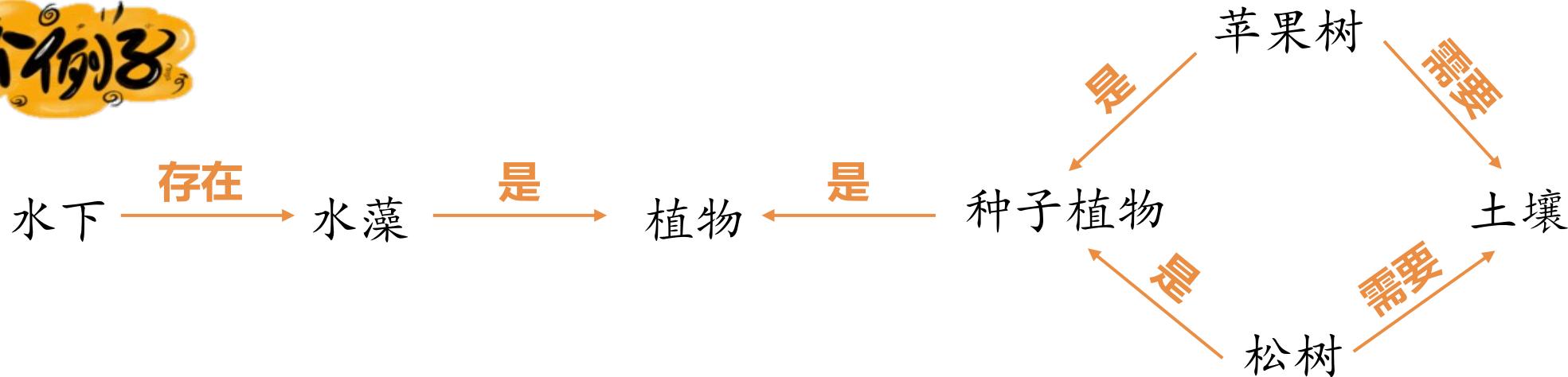
默认值：男

框架表示法还允许为属性添加约束条件，进一步增强其表达能力。

## ➤ 本体论表示法

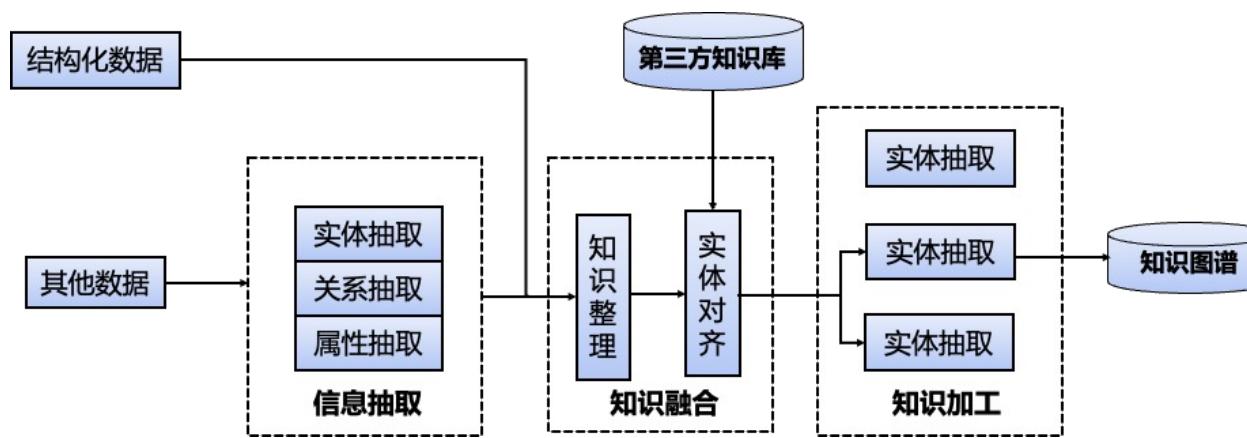
作为对特定领域内**概念及其相互关系**的精确形式化表达，其核心使命在于定义、分类与组织这些概念，旨在促进跨系统间知识的交换与共享，同时实现数据的深度互联与高效互通。一个本体可以由类（class）、关系（relations）、函数（function）、公理（axioms）和实例（instances）五种元素组成。

举个例子

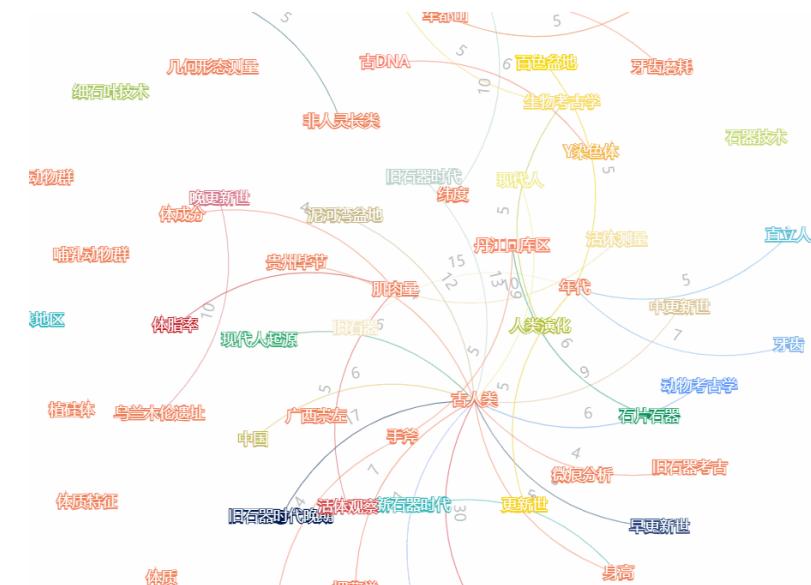


# ➤ 知识图谱

口 知识图谱是一种**基于语义网络发展而来**的新兴知识表示方法，采用图论思想，通过“**三元组**”（**实体-关系-实体**）的形式构建结构化知识库，将知识以节点和弧的方式组织为庞大的图形网络。自2012年Google提出以来，知识图谱广泛应用于搜索引擎、智能问答、个性化推荐、语义理解等领域。



## 知识图谱的构建与更新过程



问题1：露营的有多少人？

推理 (Reasoning) 是“使用理智从某些前提 (Premises) 产生结论 (Conclusion) ”的行动，是由一个或几个已知的判断 (前提) 推出新判断 (结论) 的过程。

问题4：附近是否有城镇？

进一步来说，逻辑推理是从已知事实和规则中推导出新知识的过程。

在人工智能领域，逻辑关系与规则是知识表示的核心部分。通过逻辑推理，人工智能系统能够从已有的知识中得出新的结论。这种能力对于问题求解、决策支持以及知识发现等应用至关重要。

4. 没有小镇，但一个村子在不远处，因为有一只鸡在那儿闲逛。

## ➤ 逻辑推理基础——演绎推理

□ 演绎推理以普遍规则为基础，通过三段论结构推导出**必然正确**的具体结论，适用于数学证明、逻辑分析和计算机科学等领域。尽管它**依赖前提的真实性且对不确定性处理有限**，演绎推理在AI决策支持与规则系统中仍具重要价值。



---

**大前提（Major Premise）：**一个普遍性的声明或规则。

例：所有人都有生日。

**小前提（Minor Premise）：**一个关于特定个体的声明。

例：苏格拉底是人。

**结论（Conclusion）：**根据前提得出的必然结果。

例：苏格拉底有生日。

---

## ➤ 逻辑推理基础——归纳推理

- 归纳推理 (Inductive Reasoning) 是一种从具体实例或观测中推导出一般规律或普遍结论的推理方法。与演绎推理不同，**归纳推理的结论并不是必然真实的**，而是具有一定的概率性和不确定性。
- 例如，我们可以通过观察多次实验中的现象，归纳出一种自然规律，但是这个结论不一定是确定的，它**依赖于所观察到的实例的代表性和数量**。即使所有观察到的实例都符合某一规律，也不能保证所有未观察到的实例也会符合这一规律。
- 尽管可能存在样本不代表性和过度推广的问题，归纳推理通过科学实验和严格设计能显著提高结论的可靠性。

## ➤ 规则推理系统

□ 规则推理系统通过“**如果-那么**”规则实现推理和决策，是一种重要的知识表达与推理技术。它包括规则匹配、冲突解决和执行等步骤，分为前向推理和后向推理两种方向。规则推理系统广泛应用于医疗诊断、决策支持和法律分析等领域，具有透明性、可扩展性和易于维护的优势。



规则：“如果一个人感冒了，那么他会咳嗽”。

事实：“张三感冒了”。

基于上述规则和事实，我们可以推理出“张三会咳嗽”。

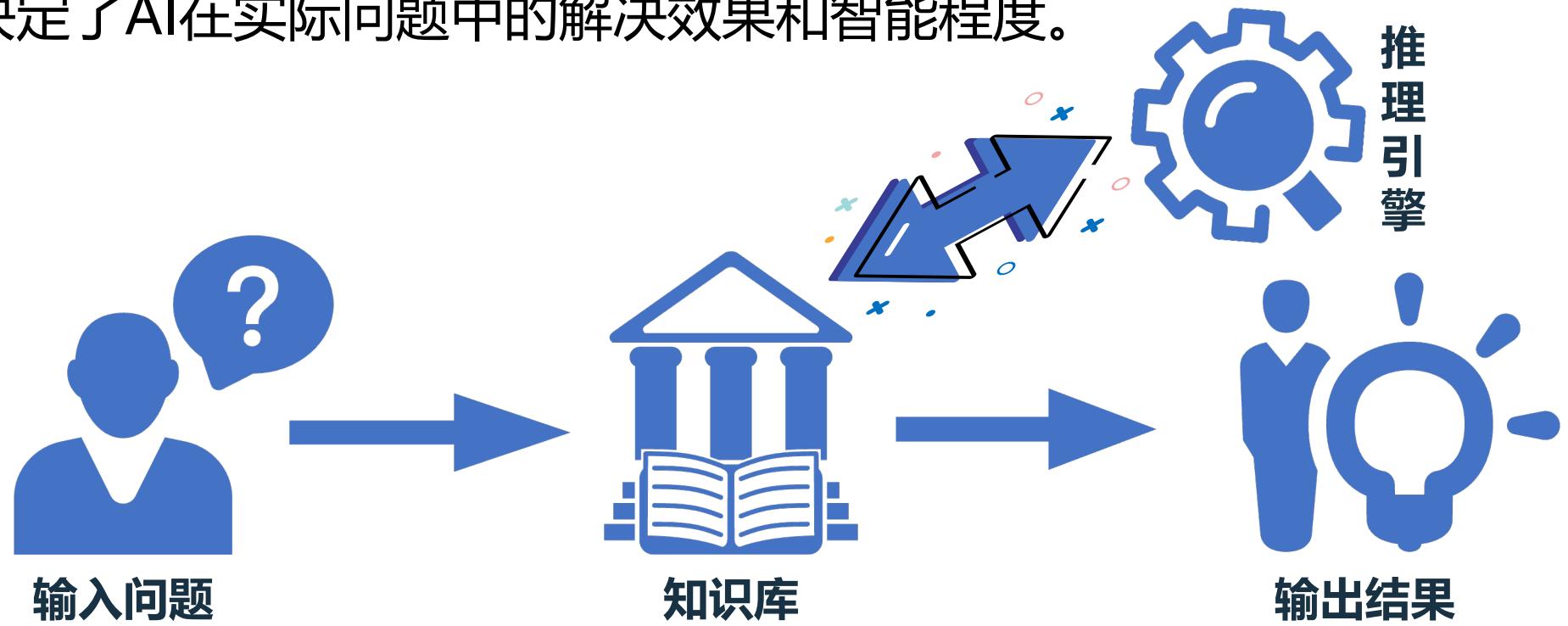
# 知识点2： 寻根溯源之法——知识推理

学  
习  
有  
深  
刻



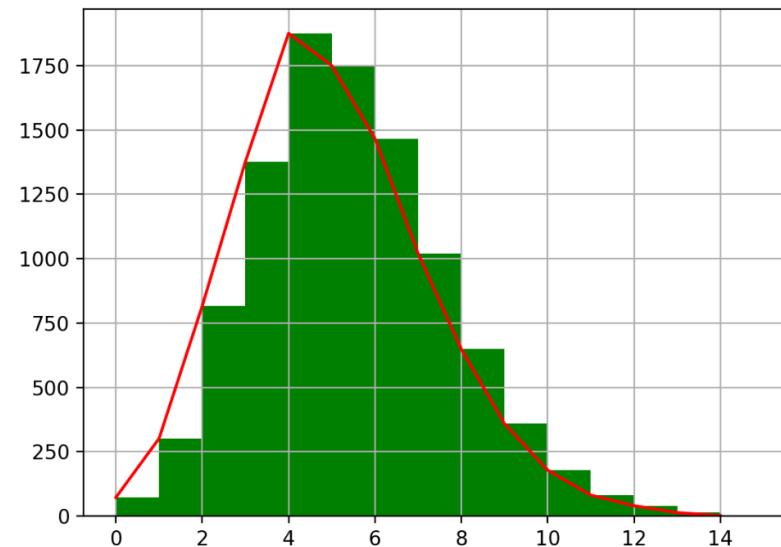
- | 01 知识推理：人工智能的基石
- | 02 核心推理方法与技术

- 知识推理是**基于现有的知识进行推导、判断和决策的过程**，它使人工智能系统能够**从有限的事实中推断出新知识**，帮助系统做出更加准确的决策。
- 知识推理不仅广泛应用于专家系统和自然语言处理等领域，也是智能系统的核心能力，决定了AI在实际问题中的解决效果和智能程度。

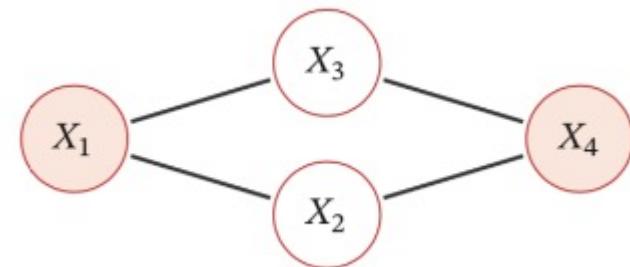


## ➤ 概率推理

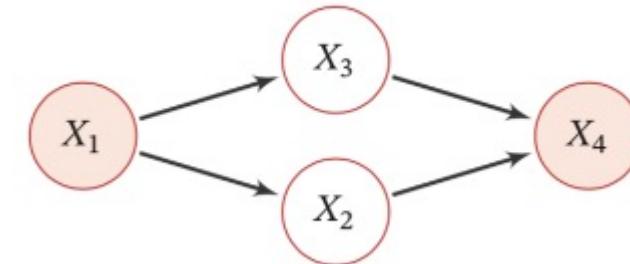
□ 概率推理在人工智能中广泛用于**处理不确定性信息**，尤其在机器学习和深度学习中。通过构建概率模型，AI能够量化不确定性并进行推理。它使得系统能够基于历史数据和当前观测值推断未知的结果。



概率分布曲线图



无向图：马尔可夫链



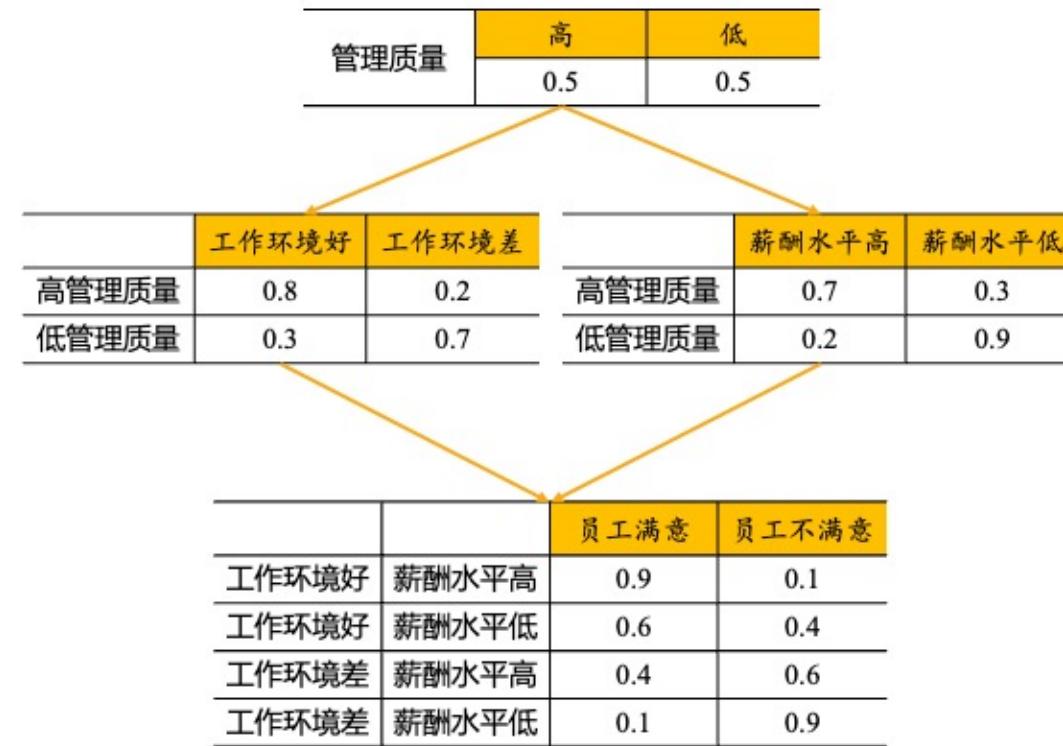
有向图：贝叶斯网络

## ➤ 概率推理——贝叶斯网络

- 贝叶斯网络是一种概率图模型，通过节点表示随机变量，边表示它们的条件依赖关系，从而有效地简化联合概率分布的计算。
- 贝叶斯网络的推理机制包括前向推理（从已知原因推导结果）和后向推理（从观察结果推断可能的原因）。



在公司员工满意度预测的贝叶斯网络中，管理质量、工作环境和薪酬水平决定员工的满意度，通过条件概率计算可得员工满意的概率，从而为决策提供数据支持。



## ➤ 概率推理——马尔可夫模型

- 基于马尔可夫性质，即**未来状态只依赖于当前状态，与过去状态无关**，通过状态转移概率矩阵进行概率推理，常用于预测系统在不同时刻的状态概率。
- 它广泛应用于具有时间依赖性或序列性质的问题，如生态系统状态转移、语音识别、金融市场变化、天气预测和基因序列分析等领域。



假设一个学生每天对学习的态度，分为“积极”和“消极”两种状态。经过一段时间观察发现：如果学生今天学习态度是积极的，那么明天依然积极的概率是 0.7，变为消极的概率是 0.3；如果学生今天学习态度是消极的，那么明天积极的概率是 0.4，继续消极的概率是 0.6。

- **初始状态：**假设第一天学生的学习态度是积极的，所以初始状态向量为积极的概率为 1，消极的概率为 0。
- **预测第二天的状态：**根据状态转移矩阵和初始状态进行计算，那么  
第二天学生学习态度为积极的概率是  $1 \times 0.7 + 0 \times 0.4 = 0.7$ ，为消极的概率是  $1 \times 0.3 + 0 \times 0.6 = 0.3$ 。
- **预测第三天的状态：**若第二天学生积极的概率是 0.7，消极的概率是 0.3，那么  
第三天积极的概率为  $0.7 \times 0.7 + 0.3 \times 0.4 = 0.61$ ，消极的概率为  $0.7 \times 0.3 + 0.3 \times 0.6 = 0.39$ 。

## ➤ 因果推理

- 因果推理是一种用于**确定变量之间因果关系**的方法。因果推理不仅关注变量之间的关联，还关注变量之间的因果关系。
- 通过因果推理，我们可以更深入地理解一个变量是如何影响另一个变量的，从而为决策提供更有力的支持。
- 因果推理中的核心概念：
  - 辛普森悖论
  - 因果干预
  - 因果效应差
  - 反事实推理

## ➤ 因果推理—辛普森悖论

□ 辛普森悖论描述了在分组数据中，一些趋势可能会在合并数据后消失或反转。此悖论的出现主要是由于**混杂变量的存在**，例如在男女药物治疗的例子中，虽然每个性别组内药物有效，但合并后反而显示药物无效。



男性和女性分别服用药物后的康复率

性别	服用药物	康复人数	总人数	康复率
男性	是	192	263	73%
男性	否	55	80	69%
女性	是	81	87	93%
女性	否	234	270	87%

男性和女性的数据合并后的康复率

服用药物	康复人数	总人数	康复率
是	273	350	78%
否	289	350	83%

在分析某药物疗效时，发现男性和女性分别服药的康复率均高于未服药者。但合并数据后，服药的总体康复率（78%）反而低于未服药者（83%）。

## ➤ 因果推理—因果干预

- 因果干预是指通过人为干预来改变某个变量的值，从而观察其他变量的变化。
- 与观察性研究不同，因果干预通过**主动改变变量的值来确定因果关系**。

## ➤ 因果推理—因果干预



假设我们想知道吸烟是否导致肺癌，使用因果图模型表示吸烟、焦油暴露与肺癌的关系。

通过干预吸烟（让某人不吸烟），我们可以使用do算子：

$P(\text{肺癌} | \text{do}(\text{吸烟} = 0))$ 计算不吸烟时患肺癌的概率，从而确定吸烟对肺癌的因果影响。

肺癌样本数据

吸烟	焦油暴露	肺癌	样本数
是	是	是	30
是	是	否	10
是	否	是	5
是	否	否	5
否	是	是	10
否	是	否	20
否	否	是	5
否	否	否	15

不干预条件下的患癌概率：

$$P(\text{肺癌} | \text{吸烟} = 1) = (30 + 5) / (30 + 10 + 5 + 5) = 35 / 50 = 0.7$$
$$P(\text{肺癌} | \text{吸烟} = 0) = (10 + 5) / (10 + 20 + 5 + 15) = 15 / 50 = 0.3$$

干预条件下的患癌概率：

$$P(\text{肺癌} | \text{do}(\text{吸烟} = 0)) = P(\text{肺癌} | \text{焦油暴露} = 1) * P(\text{焦油暴露} = 1 | \text{do}(\text{吸烟} = 0)) + P(\text{肺癌} | \text{焦油暴露} = 0) * P(\text{焦油暴露} = 0 | \text{do}(\text{吸烟} = 0)) = 0.57 * 0.6 + 0.33 * 0.4 \approx 0.45$$

## ➤ 因果推理—因果效应差

- 因果效应差量化了**不同干预条件下因变量期望值的差异**，它有助于评估某种干预对结果的实际影响，即 $\Delta = E[Y \mid do(X = 1)] - E[Y \mid do(X = 0)]$ ，其中， $E[Y \mid do(X = 1)]$ 表示在 $X=1$ 的干预条件下因变量Y的期望值； $E[Y \mid do(X = 0)]$ 表示在 $X=0$ 的干预条件下因变量Y的期望值。



为了更好地理解因果效应差的概念和计算方法，我们来看一个具体的例子。假设我们想评估某种新药对病人康复的效果，如下所示：

某种新药对病人的康复率数据

服用药物	康复人数	总人数	康复率
是	80	100	80%
否	60	100	60%

### 计算步骤

**Step 1:**  $E[\text{康复} \mid do(\text{服用药物} = 1)] = 80/100 = 0.8$

**Step 2:**  $E[\text{康复} \mid do(\text{服用药物} = 0)] = 60/100 = 0.6$

**Step 3:**  $\Delta = E[\text{康复} \mid do(\text{服用药物} = 1)] - E[\text{康复} \mid do(\text{服用药物} = 0)] = 0.8 - 0.6 = 0.2$

## ➤ 因果推理—反事实推理

□ 反事实推理是一种用于推断在不同条件下可能产生的结果的方法，它通过**假设某个变量的值不同来推断其他变量的变化**。反事实推理在因果推理中具有重要意义，它可以帮助我们理解不同条件下的因果关系，并评估干预措施的潜在影响。



假设我们想知道如果某人没有吸烟，他是否会患肺癌。我们可以通过反事实推理来进行推断，即假设该人没有吸烟，然后推断他患肺癌的概率。

肺癌样本数据

吸烟	肺癌	样本数
是	是	30
是	否	70
否	是	10
否	否	90

### 计算步骤

**Step 1:**  $P(\text{肺癌} | \text{吸烟} = 1) = 30/(30 + 70) = 0.3$

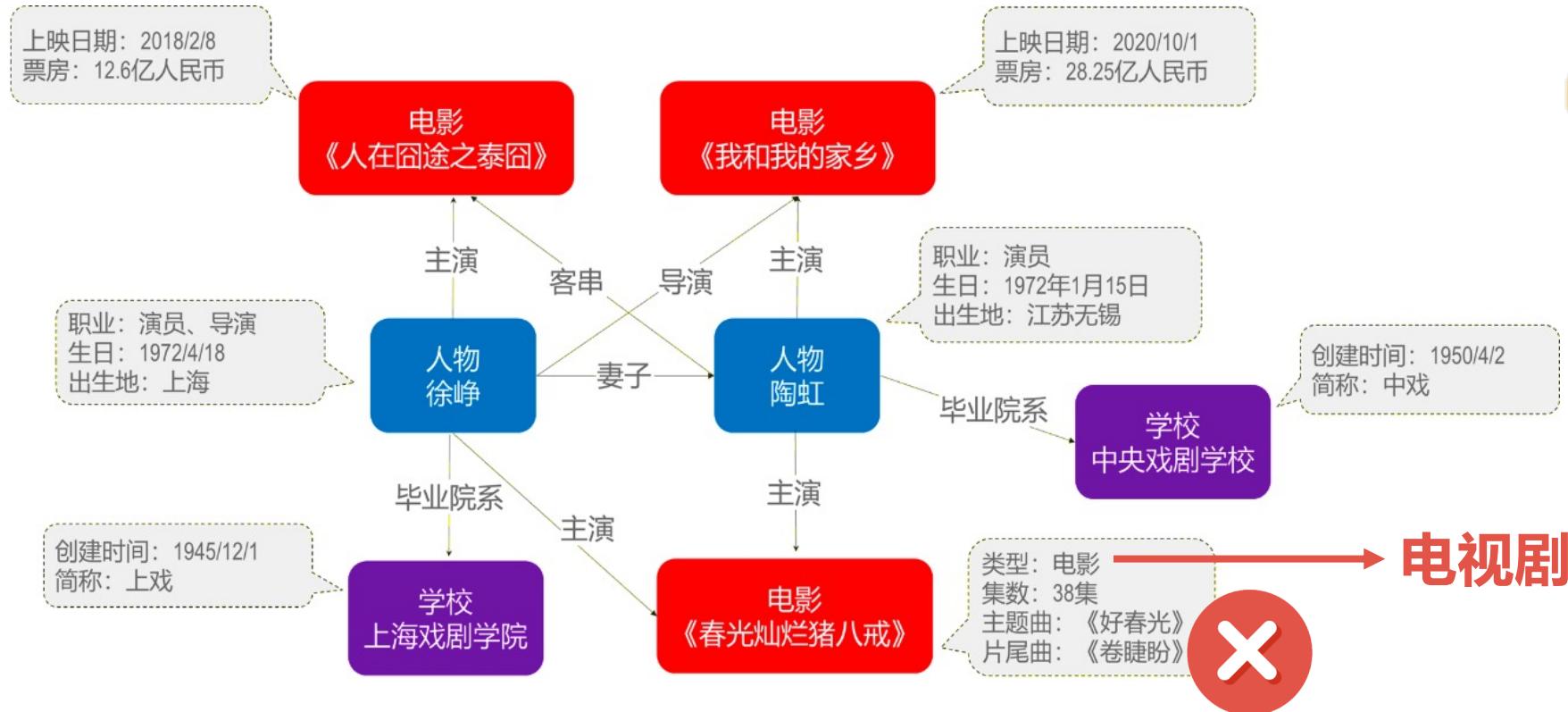
**Step 2:**  $P(\text{肺癌} | \text{吸烟} = 0) = 10/(10 + 90) = 0.1$

**Step 3:** 进行反事实推理，假设某人没有吸烟，然后推断他患肺癌的概率  $P(\text{肺癌} | do(\text{吸烟} = 0)) = 0.1$

## ➤ 知识图谱推理

- 利用图结构中的节点（实体）和弧（关系）进行推导，常见的推理任务包括实体链接、关系预测、路径推理等。
  - **基于嵌入方法的推理**：该方法通过将实体和关系映射到低维向量空间，进而能够捕捉它们之间的相似性与相关性。
  - **基于图神经网络的推理**：图神经网络（GNN）是近年来在知识图谱推理中广受关注的一类模型。GNN通过迭代更新节点的嵌入表示，能够捕捉图中节点间复杂的关系信息。
  - **基于规则学习的推理**：这种方法结合了符号推理和机器学习的优势。通过从图谱数据中自动学习逻辑规则，能够根据这些规则进行推理。
  - **基于大语言模型的推理**：随着大语言模型（LLM）的兴起，它们在自然语言处理领域的强大能力也被引入到知识图谱推理中。

# 核心推理方法与技术——知识图谱推理



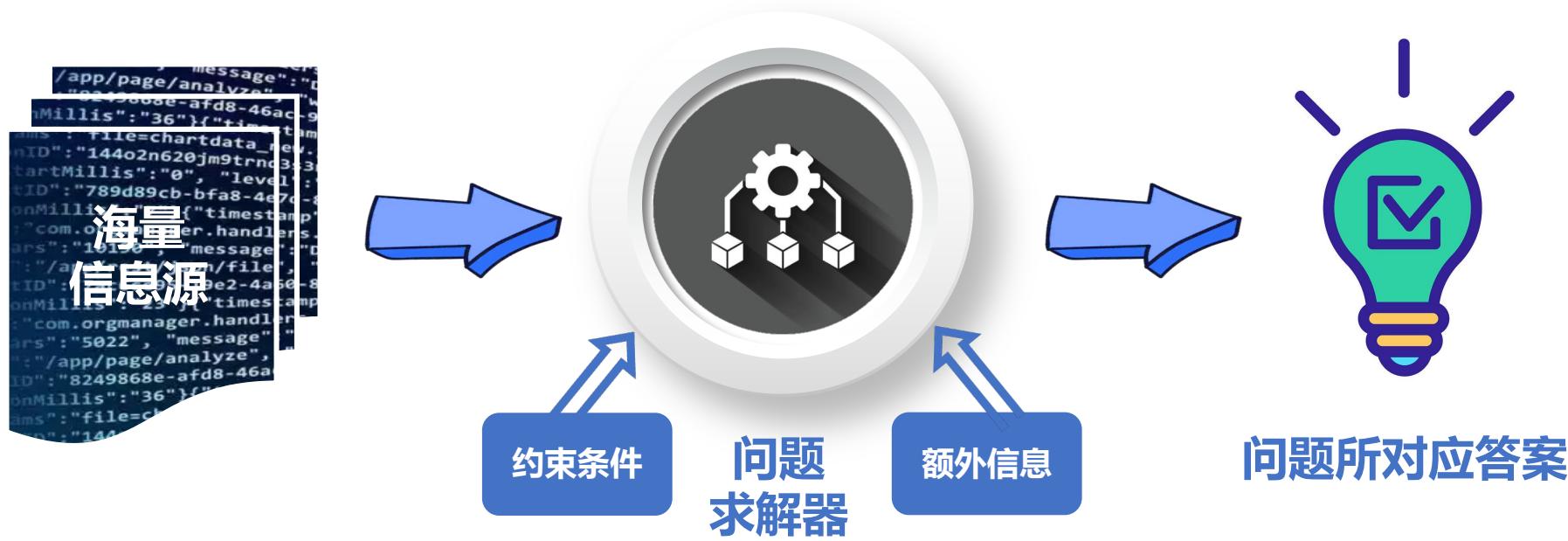
比如，在某个影视知识图谱中，虚线框中的实体《春光灿烂猪八戒》，其类型为“电影”。它的属性有集数，主题曲、片尾曲等。而其他同为“电影”类别的实体，其属性多包含上映时期、票房，且大多没有集数这个属性。则推理可知，《春光灿烂猪八戒》这个实体的类型大概率存在错误，其正确类型应该是电视剧。

# 知识点3：上下求索之刃——搜索求解

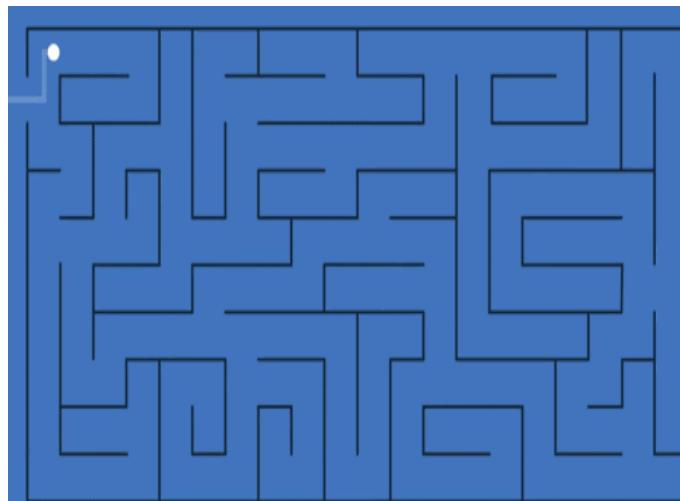


- | 01 探索无尽空间：搜索问题的起点与核心
- | 02 无声之道：盲目搜索与启发式的智慧
- | 03 最优之路：A\*算法和局部最优问题

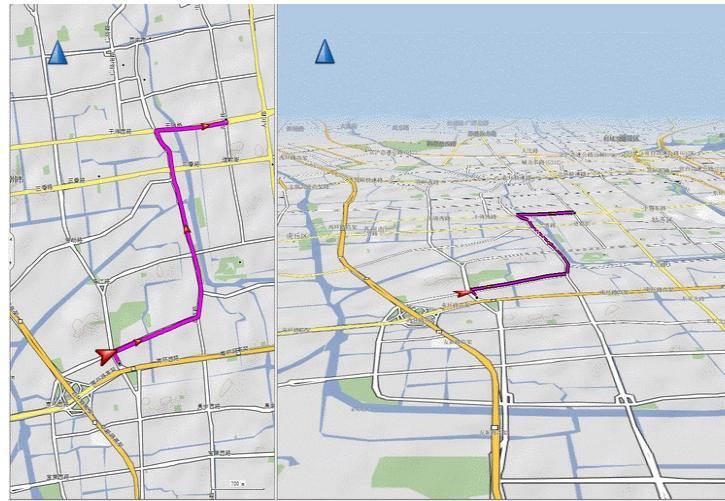
➤ 搜索问题的目标是通过合理的搜索策略，在状态空间中**找到从初始状态到目标状态的最优路径**。搜索问题通常包含初始状态、目标状态、状态空间、操作符和路径成本等要素。状态空间是所有可能的状态集合，操作符定义了状态之间如何转换，路径成本则是评估不同路径优劣的标准。



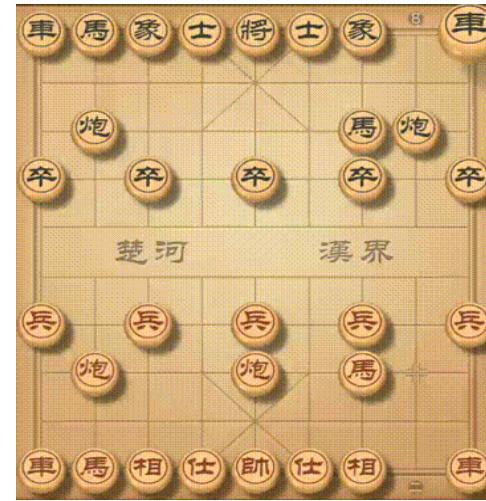
➤ 在面对复杂的挑战时，人类依靠逻辑推理和直觉洞察来应对问题，而计算机则通过精心设计的搜索算法来模拟这一过程。搜索求解的核心是**通过系统地探索潜在的解空间**，寻找最优解或可行解。无论是棋盘游戏中的战略选择，还是交通路线中的最短路径规划，搜索求解都是推动这些任务高效完成的关键技术。



迷宫问题

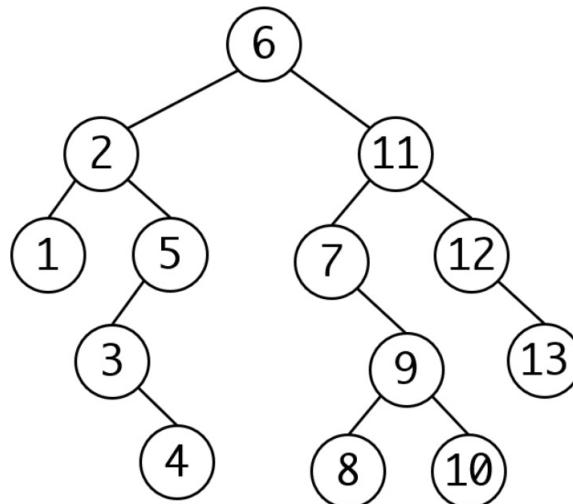


交通路线

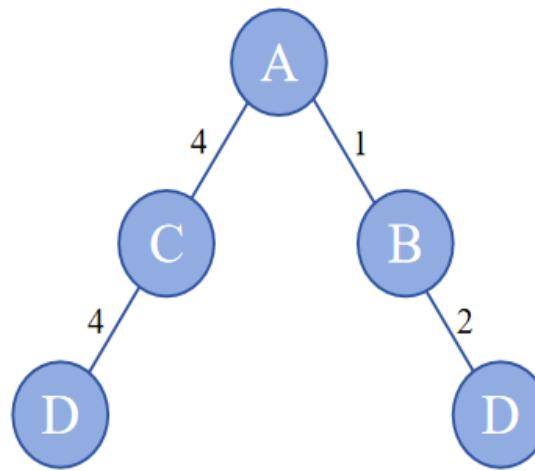


棋盘游戏

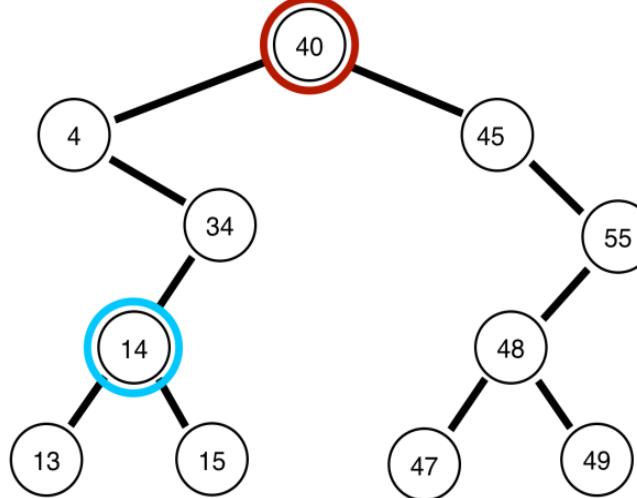
➤ 搜索树是搜索问题中用来**可视化探索过程**的重要工具，树中的每个节点代表一个状态，节点之间的边代表状态转移的动作。通过搜索树，能够清晰地看到从初始状态到目标状态的路径如何被逐步探索，同时也能直观地展示搜索过程中路径的代价及其优化方向。



二叉搜索树



搜索树



搜索树的查找

# 搜索问题的核心——搜索求解（送外卖案例）



你是一个外卖小哥，手握5份订单，配送范围如下：

餐厅位置：中心广场

订单分布：

订单1：学校 (东1km)

订单2：医院 (北2km)

订单3：商场 (西3km)

订单4：公园 (南1km)

订单5：小区 (东4km)

你需要用不同算法规划最短路线，确保30分钟内送完所有订单！

# 搜索问题的核心——搜索求解（送外卖案例）



你是一个外卖小哥，手握5份订单，配送范围如下：

餐厅位置：中心广场

订单分布：

订单1：学校（东1km） 订单2：医院（北2km） 订单3：商场（西3km）

订单4：公园（南1km） 订单5：小区（东4km）

**盲目搜索**

**无头苍蝇法**

方法：随机选择路径，撞墙就回头。

实战：

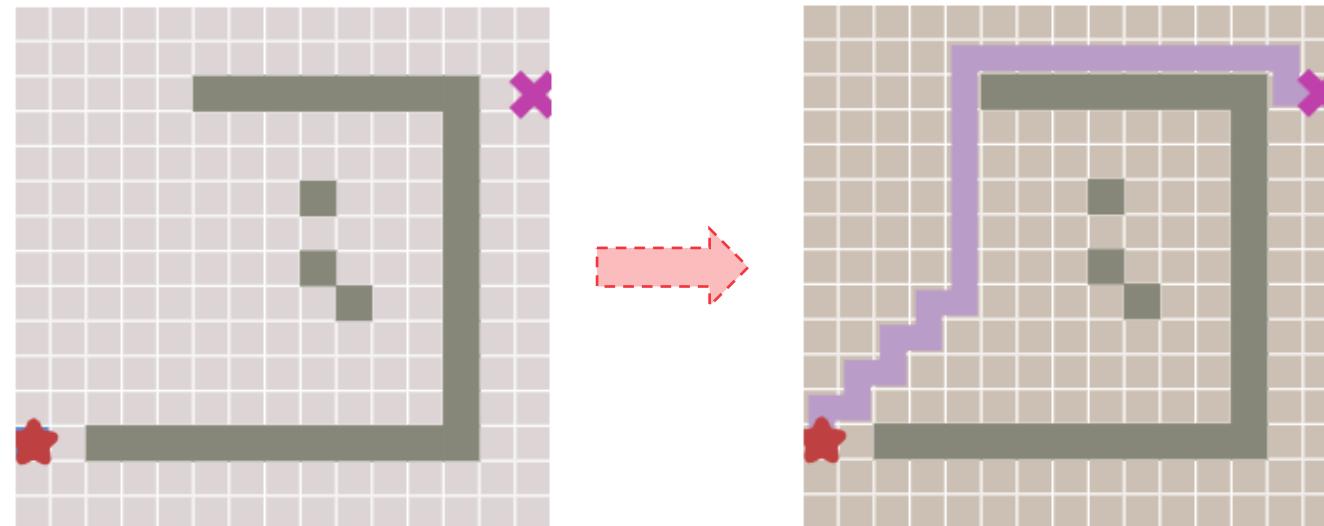
第一次尝试：餐厅 → 订单3（西3km）→ 发现太远，超时 → 放弃。

第二次尝试：餐厅 → 订单1（东1km）→ 订单5（东4km）→ 送完但耗时过长。

缺点：效率极低，像无导航开车。

## ➤ 盲目搜索概述

- 盲目搜索又称为无信息搜索，指的是**没有特定领域知识**的情况下进行的搜索。  
这类算法适用于简单的搜索问题，尤其是当状态空间较小且复杂度不高时。



典型的盲目搜索



你是一个外卖小哥，手握5份订单，配送范围如下：

餐厅位置：中心广场

订单分布：

订单1：学校（东1km） 订单2：医院（北2km） 订单3：商场（西3km）

订单4：公园（南1km） 订单5：小区（东4km）

## 广度优先搜索

## 地毯式排查

方法：按距离餐厅的远近层级送餐。

实战：先送1km内的订单：订单1（东1km）、订单4（南1km）。

再送2km内的订单：订单2（北2km）。

最后送\*\*3km+\*\*的订单：订单3（西3km）、订单5（东4km）。

优点：保证最短总路径。

缺点：可能绕路（比如订单5明明可以顺路送）。

## ➤ 广度优先搜索 (Breadth-First Search, BFS)

□ 基本思想是从初始状态开始，逐层扩展状态空间，直到找到目标状态。BFS 通过一个队列来管理待处理节点，**优先扩展距离起始节点较近的节点**，直到找到最短路径。它**能够保证找到最优解**，但在大规模状态空间下，可能会消耗大量的内存和时间，导致效率较低。搜索过程可分为三个阶段：

- 初始阶段：构建一个空队列存储待处理节点，并将初始状态加入队列。同时，初始化一个空集合‘searched[]’用于记录已访问的节点。
- 遍历阶段：当队列非空时，取出队首状态。如果该状态为目标状态，则返回路径；否则，生成所有可能的后续状态，未访问的状态加入队列，并标记为已访问。
- 终止条件：当队列为空时，表示所有可达节点已检查完。若未找到目标状态，搜索失败。

# 搜索问题的核心——搜索求解（送外卖案例）



你是一个外卖小哥，手握5份订单，配送范围如下：

餐厅位置：中心广场

订单分布：

订单1：学校（东1km） 订单2：医院（北2km） 订单3：商场（西3km）

订单4：公园（南1km） 订单5：小区（东4km）

深度优先搜索

一条道走到黑

方法：选一条路送到底，失败再回溯。

实战：

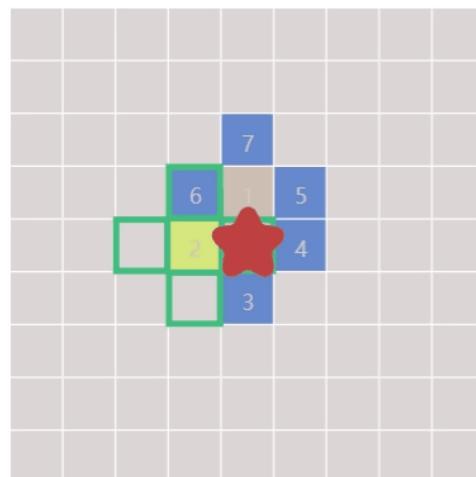
选择路径：餐厅 → 订单1 → 订单5 → 发现订单5太远 → 回溯到订单1 → 改送订单4。

优点：适合订单集中在一侧时（如东区）。

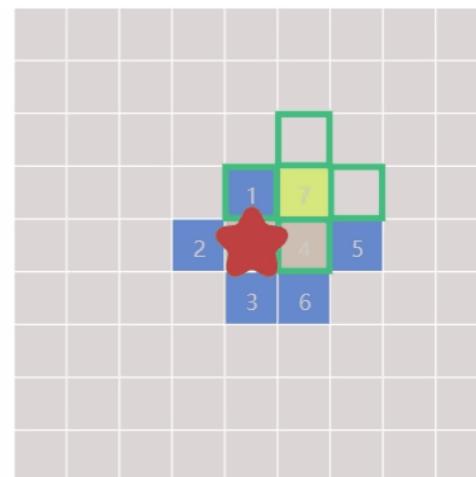
缺点：可能陷入“东边送完才发现西边漏单”。

## ➤ 深度优先搜索 (Depth-First Search, DFS)

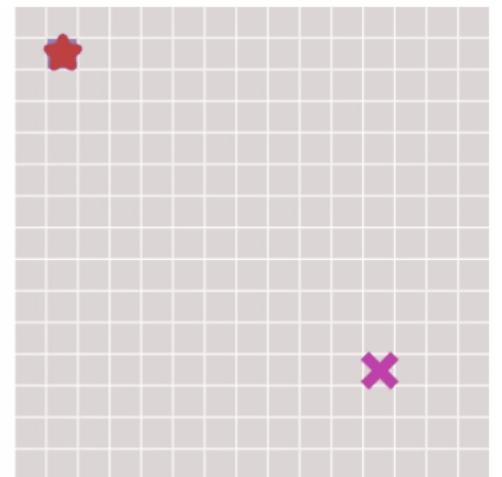
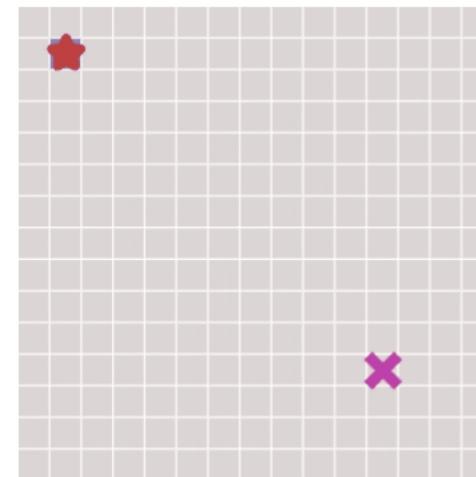
□ 深度优先搜索是一种深入搜索的策略，**优先沿着每一条路径深入**，直到达到叶子节点后回溯继续搜索。DFS使用栈来存储待处理节点，并优先扩展栈顶的节点。它能够快速找到解，但可能陷入死循环，且**无法保证找到最优解**。



广度优先搜索



深度优先搜索



# 搜索问题的核心——搜索求解（送外卖案例）



你是一个外卖小哥，手握5份订单，配送范围如下：

餐厅位置：中心广场

订单分布：

订单1：学校（东1km） 订单2：医院（北2km） 订单3：商场（西3km）

订单4：公园（南1km） 订单5：小区（东4km）

**启发式搜索**

**经验派送餐**

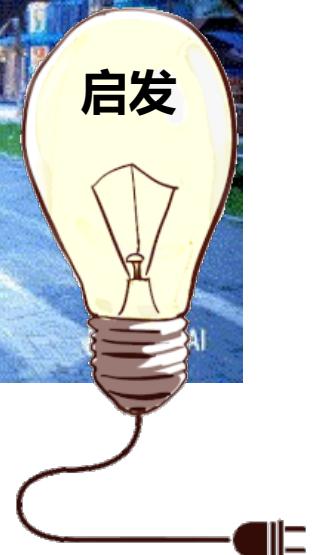
方法：用简单规则优先送“看似最近”的订单。

规则：优先送直线距离最近的订单。

执行：订单4（南1km）→订单1（东1km）→订单2（北2km）  
→订单3（西3km）。

## ➤ 启发式搜索概述

- 口 启发式搜索结合了领域知识，通过启发式函数引导搜索过程，以提高搜索效率。启发式函数通过估算从当前状态到目标状态的成本，帮助算法选择最有希望的路径，从而减少不必要的搜索。
- 口 与盲目搜索相比，启发式搜索能够在更短的时间内找到解，尤其适用于大规模的复杂问题，能够显著提升搜索效率和准确性。



# 搜索问题的核心——搜索求解（送外卖案例）



你是一个外卖小哥，手握5份订单，配送范围如下：

餐厅位置：中心广场

订单分布：

订单1：学校（东1km） 订单2：医院（北2km） 订单3：商场（西3km）

订单4：公园（南1km） 订单5：小区（东4km）

**贪心最佳优先搜索**

**永远选“眼前最优”**

方法：用简单规则优先送“看似最近”的订单。

规则：优先送直线距离最近的订单。

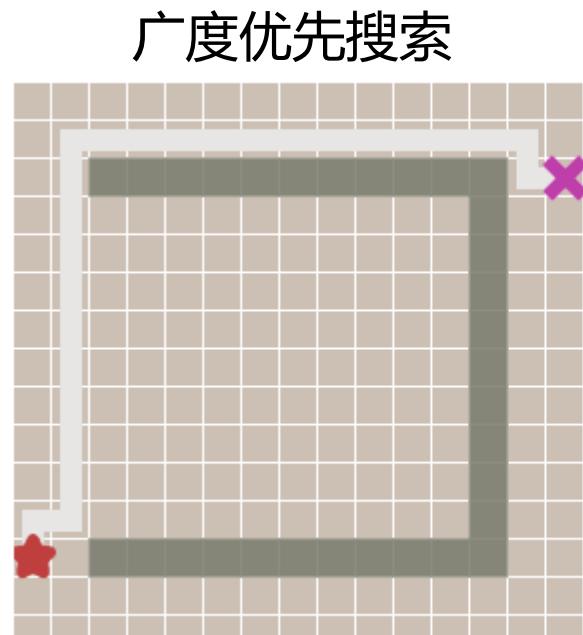
执行：订单4（南1km）→订单1（东1km）→订单2（北2km）→  
订单3（西3km）。

优点：比盲目搜索快。

缺点：可能忽略实际路况（如订单4到订单1需绕路）。

## ➤ 贪心最佳优先搜索 (Greedy Best-First Search, GBFS)

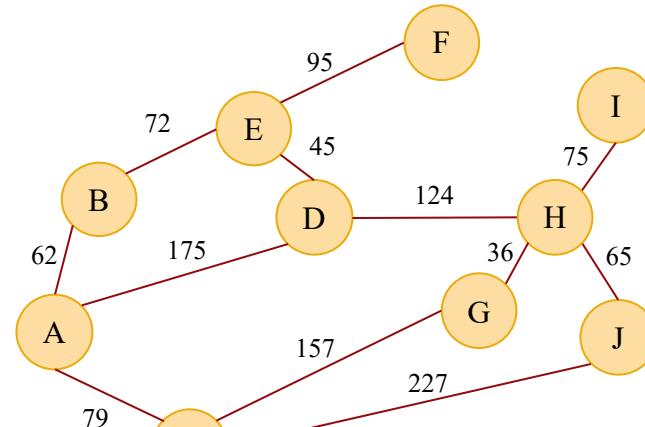
口 贪心最佳优先搜索依赖启发式函数来评估当前节点到目标状态的距离，**每次选择距离目标状态最近的节点进行扩展**。核心思想是根据目标的估算距离来指导搜索方向，能够快速找到解，但它可能陷入局部最优解。



# ➤ 贪心最佳优先搜索 (Greedy Best-First Search, GBFS)

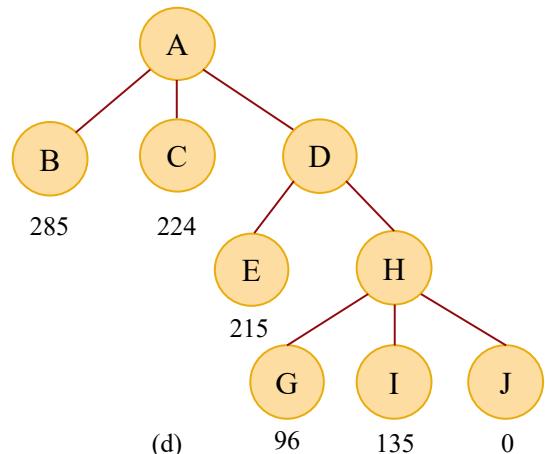
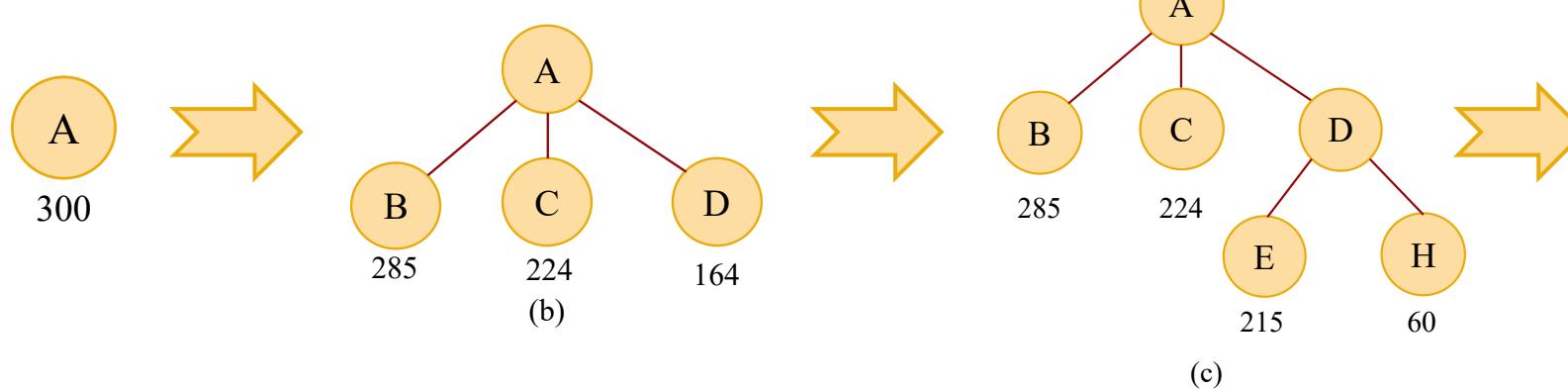


我们要在一个简单的地图上寻找从城市A到城市J的最短路径，每个城市代表一个节点，连接城市的边代表道路。右侧表格为辅助信息，记录了各个城市到J城市的直线距离。启发式函数 $h(n)$ 根据辅助信息选择当前城市到目标城市J的最短直线距离。



辅助信息——各个城市到J城市的直线距离

A	300	B	285
C	224	D	164
E	215	F	153
G	96	H	60
I	135	J	0





你是一个外卖小哥，手握5份订单，配送范围如下：

餐厅位置：中心广场

订单分布：

订单1：学校（东1km） 订单2：医院（北2km） 订单3：商场（西3km）

订单4：公园（南1km） 订单5：小区（东4km）

## A\*搜索

## 学霸级路线规划

方法：综合“已走距离” + “预估剩余距离”，选总代价最小的路径。

关键公式：总代价 = 已走距离 + 预估剩余距离（如直线距离）

选项1：餐厅 → 订单1（东1km）→ 订单5（东4km）→ 总代价 = 1+4

+ 预估（订单5到其他）= 高 ✗

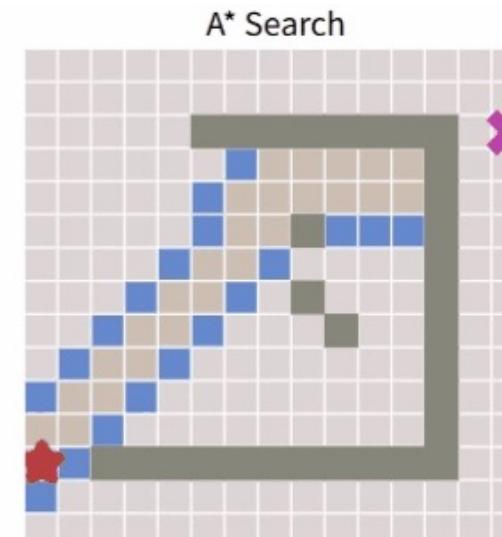
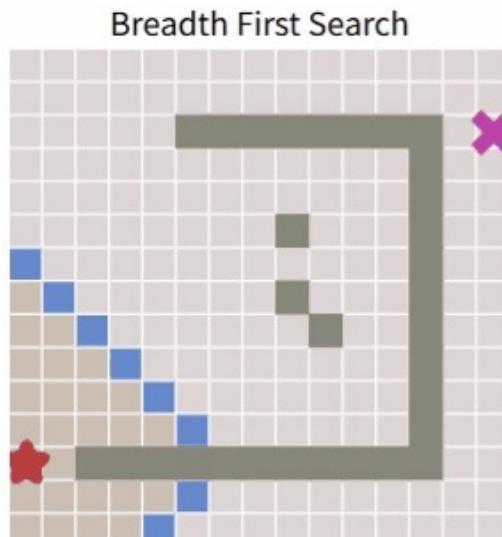
选项2：餐厅 → 订单4（南1km）→ 订单2（北2km）→ 总代价 = 1+2

+ 预估（订单2到其他）= 低 ✓

优点：平衡效率和最优解，像高德地图的智能导航

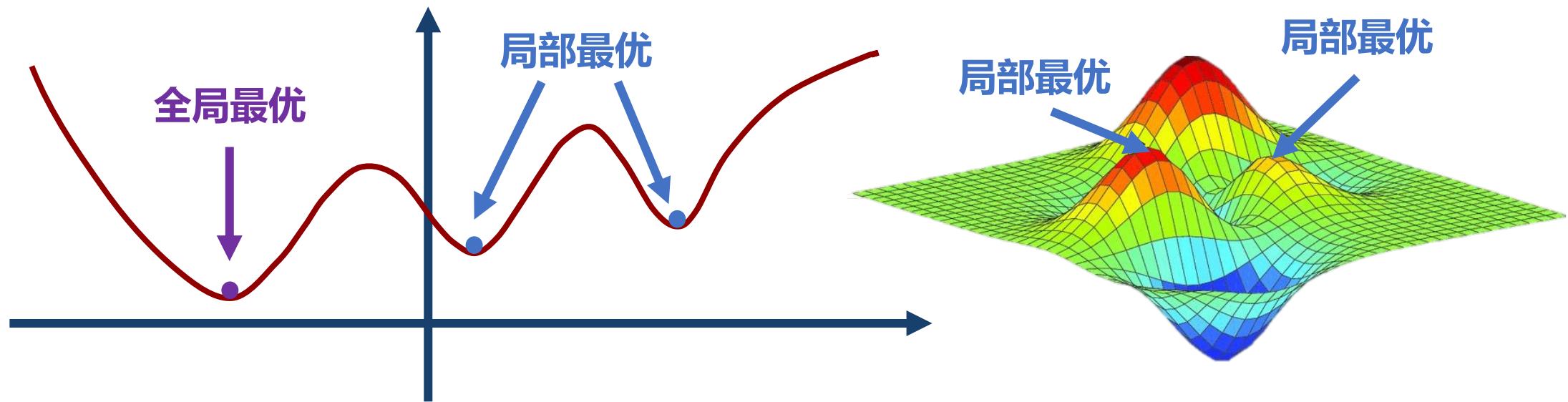
## ➤ A\*搜索算法

□ A\*搜索算法是一种结合了启发式函数和实际路径代价的搜索方法，能够高效地找到从起点到终点的最优路径。A\*算法通过评估每个节点的总成本 ( $f$ 值) 选择路径，该成本由从起始节点到当前节点的实际代价 ( $g$ 值) 和从当前节点到目标节点的启发式估算代价 ( $h$ 值) 之和组成。



## ➤ 局部最优问题

在复杂的搜索空间中，A\*算法也可能会陷入某些局部最优解。这通常是因为启发式函数设计不当，导致搜索路径偏离最优方向。为了解决这一问题，可以采用一些策略，如**调整启发式函数的设计**，或者**结合其他搜索策略**来避免局部最优问题的发生，确保更广泛的搜索探索。



# 不同方法之间的对比

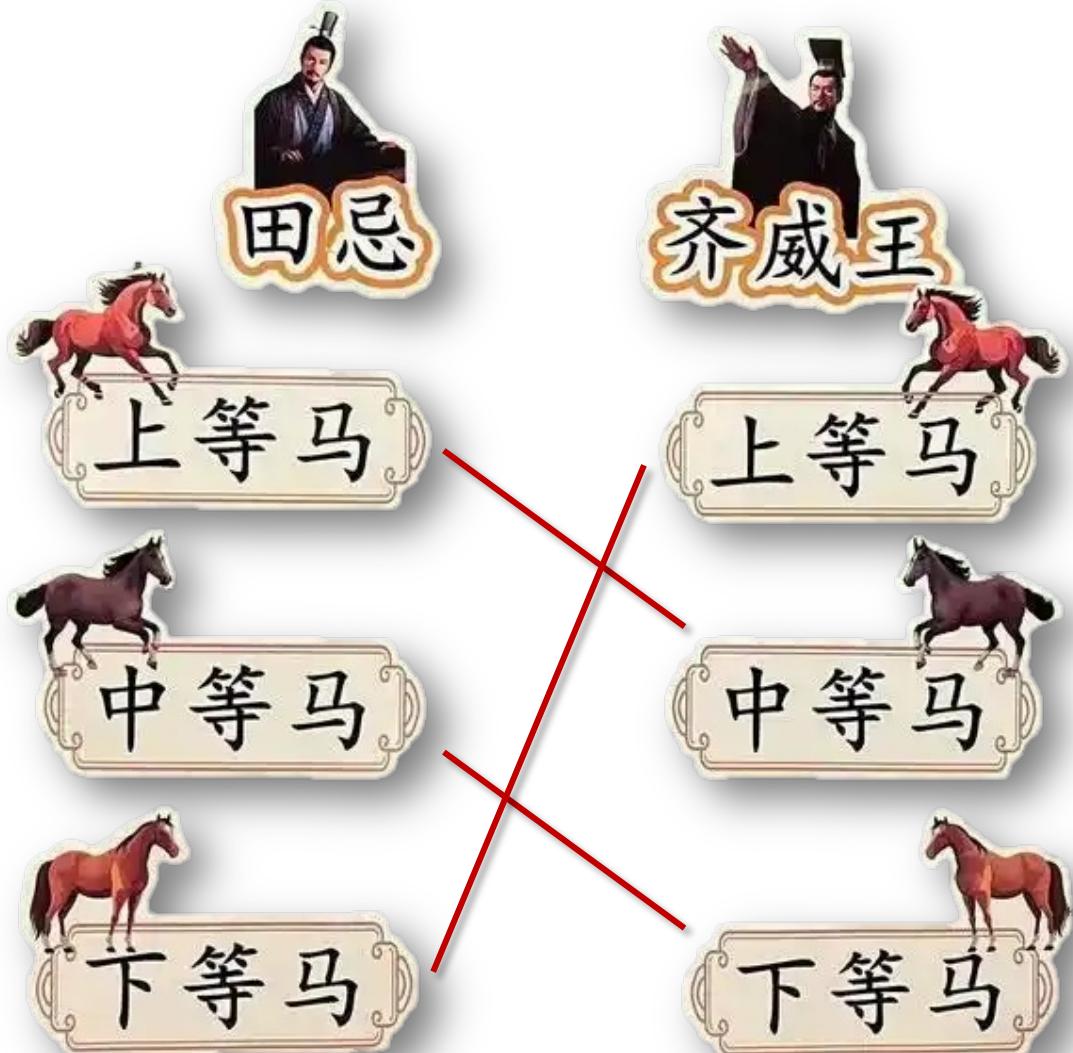
算法		策略	场景表现	适用场景
盲目搜索		随机试错	效率最低，可能超时	无任何信息时
①	广度优先搜索	按距离层级送	路线总距离最短，但可能慢	确保全局最优（如法律审核）
②	深度优先搜索	一条路走到底	可能快，也可能绕远路	解在深层时（如迷宫）
启发式搜索		按经验规则送	比盲目搜索快，但不精确	有部分先验知识时
①	贪心最佳优先搜索	永远选当前最近	开头快，结尾可能崩溃	实时性要求高（如Uber接单）
②	A*搜索	已走距离+预估剩余	又快又省，学霸级方案	路径规划、游戏AI

# 知识点4： 田忌赛马的制胜启示——博弈论



- | 01 博弈之道：智慧与策略的博弈
- | 02 决策的交响：纳什均衡与选择的艺术
- | 03 从囚徒到枪手：博弈中的人性与策略

# 田忌赛马



博

败

胜

胜

第

策

赢



# 博弈



田忌赛马

在一定条件下，遵守一定的规则，一个或几个拥有绝对理性思维的人或团队，从各自允许选择的行为或策略进行选择并加以实施，并从中各自取得相应结果或收益的过程。



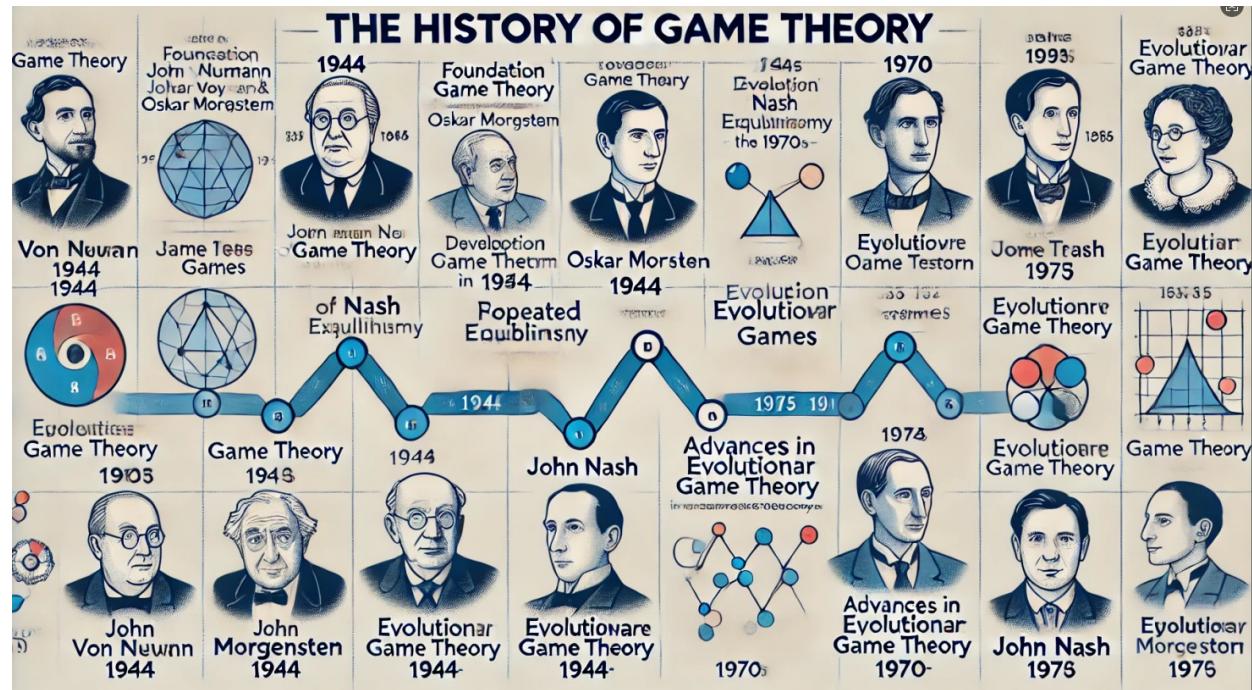
国际象棋



股票

## ► 博弈论的起源与发展

- 博弈论的起源可追溯到中国古代《孙子兵法》，孙武提出了类似博弈论的战略选择框架，如“知己知彼”。
- 博弈论的现代化发展始于策墨洛 (Zermelo)、波雷尔 (Borel) 和 冯·诺伊曼 (von Neumann) 等人，他们提出了零和博弈和极小化-极大化定理，奠定了博弈论的数学基础。
- 约翰·纳什 (John Nash) 提出“纳什均衡”，为博弈论提供了核心分析重要工具，广泛应用于经济学、政治学、社会学及人工智能等领域。

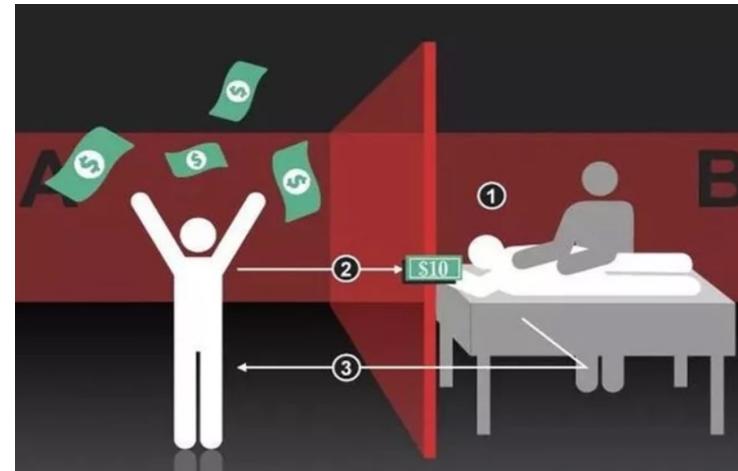


## ➤ 博弈论的定义与基础

- 博弈论描述了在特定规则和环境下，多个决策者（玩家）如何通过**选择策略**进行互动。博弈的核心要素包括局中人（玩家）、策略（行动计划）和收益（结果）。每个玩家根据规则和自身信息做出决策，**旨在最大化自己的收益**。博弈论广泛应用于经济学、政治学、生物学等领域，分析竞争与合作行为。



礼品交换博弈



信任博弈



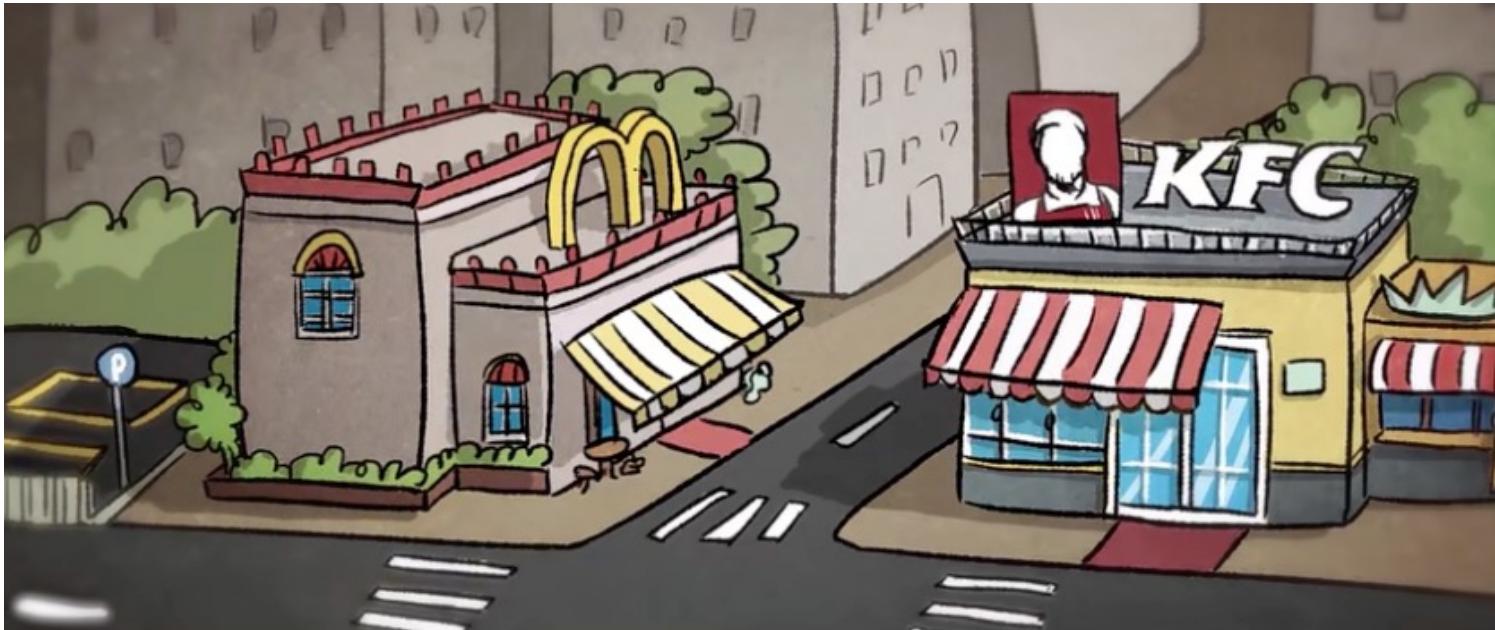
最后通牒博弈

## ➤ 博弈的种类

- 博弈可以按信息完备性、收益总和和合作关系分类。
- 完全信息博弈中所有玩家对策略和结果了如指掌，如国际象棋。
- 不完全信息博弈中，某些信息对玩家是隐藏的，如赌博游戏。
- 零和博弈指一方的损失恰为另一方的收益，典型如围棋。
- 非零和博弈则各方可以共同获益，如团队合作。
- 合作博弈允许玩家形成联盟。
- 非合作博弈则由玩家独立决策。



# 麦当劳肯德基为什么总是开在一起？



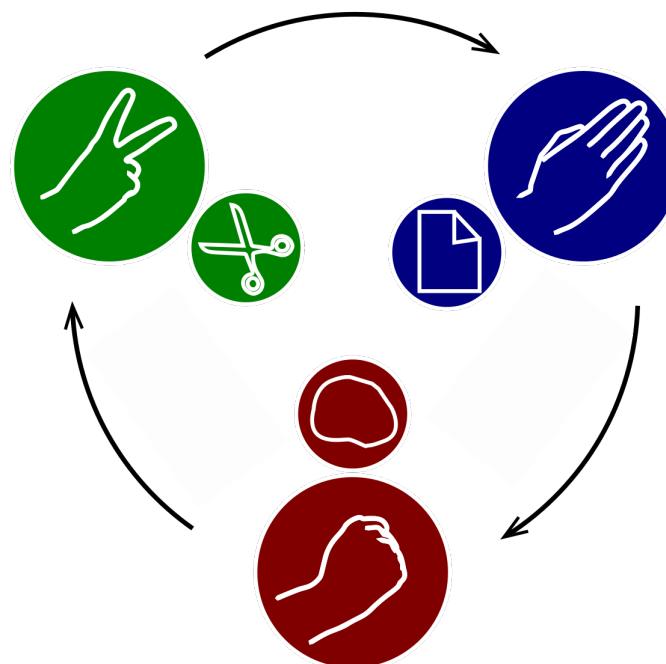
首先，麦当劳和肯德基选址是一场博弈。假设麦当劳和肯德基在城市中选址，双方都希望获得最大客流量。若麦当劳选在 A 区域，肯德基有两个选择：在 A 区域附近开店，或选在其他区域。若肯德基选其他区域，A 区域的大部分顾客就被麦当劳独占；而若肯德基在 A 区域附近开店，虽竞争加剧，但能共同做大市场，双方都有机会获客。最终，双方都意识到在相近位置开店是各自的最优策略，于是形成了一种平衡状态，即都选择开在一起。

## ➤ 纳什均衡

- 纳什均衡 (Nash equilibrium) 由美国数学家、诺贝尔经济奖获得者、美国国家科学院院士约翰·纳什 (John Nash) 提出，描述了博弈中所有玩家选择的策略，使得**没有任何一个玩家能通过单方面改变策略来获得更好的结果。**
- 换句话说，**在纳什均衡下，所有玩家的策略都是最优响应**。该概念广泛应用于非合作博弈中，特别是在经济学和人工智能的决策机制设计中。
- 纳什均衡的存在依赖于几个基本条件：**有限的策略空间，完备的信息和理性假设**。只有在这些条件下，博弈才可能存在稳定的均衡状态。纳什均衡的求解不仅仅是理论上的，也是实际博弈设计中的关键，尤其在计算机科学、网络安全等领域具有重要意义。

## ➤ 经典博弈案例 - 猜拳游戏

- 猜拳游戏是一个典型的二人完全信息零和博弈，其策略空间仅限于“石头、剪刀、布”。该游戏不存在纯策略的纳什均衡，因为任何一个策略总有更优的选择。通过混合策略，玩家可以随机选择不同的手势，以均等的概率选择三种手势，形成均衡状态。



玩家A的收益矩阵	玩家B: 石头	玩家B: 剪刀	玩家B: 布
玩家A: 石头	0	1	-1
玩家A: 剪刀	-1	0	1
玩家A: 布	1	-1	0

玩家B的收益矩阵	玩家B: 石头	玩家B: 剪刀	玩家B: 布
玩家A: 石头	0	1	-1
玩家A: 剪刀	-1	0	1
玩家A: 布	1	-1	0

## ➤ 经典博弈案例 - 囚徒困境

- 囚徒困境是一个经典的非零和博弈，描述了两名囚犯在合作与背叛之间的选择。尽管合作对双方最有利，但由于缺乏信任和沟通，双方倾向于选择背叛。此博弈展示了即使合作是最优解，纳什均衡下仍然会出现背叛的情况，符合理性玩家的选择。



A B	合作	背叛
合作	-1, -1	-10, 0
背叛	0, -10	-8, -8

## ➤ 经典博弈案例 - 枪手博弈

- 枪手博弈是一个动态的三人非合作博弈，分析了多个参与者在竞争环境下的决策。通过博弈树和反向归纳法，玩家可以计算每个行动的最优策略。该博弈展示了在动态博弈中的策略选择与博弈树推理方法。



	命中率	射偏率	最佳策略	存活概率
甲	80%	20%	先瞄准乙	24%
乙	60%	40%	先瞄准甲	20%
丙	40%	60%	先瞄准甲	100%

# 知识点5： 信任与背叛的无声较量——囚徒困境



- | 01 囚徒困境：信任与背叛的博弈与哲学
- | 02 博弈中的抉择：理性、信任与人性
- | 03 囚徒困境与现实决策：从个人到社会的信任挑战

# 什么是囚徒困境?



- “囚徒困境”是1950年由美国兰德公司提出的理论，后来由加拿大数学家阿尔伯特·威廉·塔克 (Albert William Tucker) 以囚徒故事加以阐述，并命名为“囚徒困境”。囚徒困境用于描述在一种博弈情境下，个体之间可能面临的合作与背叛之间的冲突，**反映个人最佳选择并非团体最佳选择**。换句话说，行为主体面临选择的两难境地时，**往往会趋向于考虑相对利己但是不利于集体最大利益的方式**。

■ 两个人因合伙盗窃杀人被捕，警方将他们隔离囚禁，并给他们三个选择：

- ① 如果两个人都抵赖，各判刑一年。
- ② 如果两个人都坦白，各判刑八年。
- ③ 如果两个人中一个坦白而另一个抵赖，坦白的释放，抵赖的判刑十年。

于是，每个囚徒都面临两种选择：坦白或抵赖。

A B	合作	背叛
合作	-1, -1	-10, 0
背叛	0, -10	-8, -8



# 策略分析：合作还是背叛？

- 对于A来说，不论B选择何种策略，A选择“背叛”总能获得更高的收益。因此，对于A来说，“背叛”就是一个严格优势策略。

标记红色的数字代表A的优势策略，标记绿色的数字代表A的劣势策略。

- 对于B来说，不论A选择何种策略，B选择“背叛”的收益相对于“合作”的收益也是更高的。

标记红色的数字代表B的优势策略，标记绿色的数字代表B的劣势策略。

- 综合来看，我们不难发现，“背叛”对于双方来说都是严格优势策略。

A B	合作	背叛
合作	-1, -1	-10, 0
背叛	0, -10	-8, -8

A B	合作	背叛
合作	-1, -1	-10, 0
背叛	0, -10	-8, -8

A B	合作	背叛
合作	-1, -1	-10, 0
背叛	0, -10	-8, -8

- **从个体利益最大化出发，双方都选择背叛是各自的最优策略**，但结果是两人都判 8 年，并非整体的最优结果。
- 若两人合作，整体结果最优，两人每人判 1 年。但在缺乏沟通和信任机制的情况下，个体很难放弃自身的最优策略（背叛）去追求集体的最优结果。这体现了**个体理性与集体理性的冲突**，每个人都为了自己的利益做出看似最优的选择，却导致了整体的非最优结局。
- 在博弈论中的囚徒困境是一个**非零和博弈**，博弈双方的利益并不完全对立，有可能存在合作的空间，使得双方的总收益增加，或者一方收益增加的同时另一方收益并不减少等情况，但也可能个体的获利会导致集体的损失。



- 对于A来说，不论B选择何种策略，A选择“背叛”总能获得更高的收益。因此，对于A来说，“背叛”就是一个严格优势策略。

标记红色的数字代表A的优势策略，标记绿色的数字代表A的劣势策略。

- 而对于B来说，不论A选择何种策略，B选择“背叛”的收益相对于“合作”的收益也是更高的。

标记红色的数字代表B的优势策略，标记绿色的数字代表B的劣势策略。

- 综合来看，我们不难发现，“背叛”对于双方来说都是严格优势策略。

## 纳什均衡



A B	合作	背叛
合作	-1, -1	-10, 0
背叛	0, -10	-8, -8

A B	合作	背叛
合作	-1, -1	-10, 0
背叛	0, -10	-8, -8

A B	合作	背叛
合作	-1, -1	-10, 0
背叛	0, -10	-8, -8

- 纳什均衡是非合作博弈中的核心概念，是指在一个博弈过程中，无论其他参与者选择何种策略，每个参与者都选择了自己的最优策略，从而使得没有人有动机去单方面改变自己的策略的一种策略组合状态。**在纳什均衡点上，任何一个参与者都不会因为单方面改变自己的策略而获得更好的收益。**
- 纳什均衡是一个策略组合，其中每个参与者的策略都是对其他参与者策略的最佳反应。在许多情况下，纳什均衡并不是最理想的结果，特别是当博弈双方缺乏信任或无法沟通时，个体的理性选择可能导致全体的非最优结果。
- 纳什均衡为分析和预测博弈行为提供了重要的工具和理论基础，帮助人们理解在各种相互依赖的决策情境中，个体如何做出决策以及最终可能达到的结果，在经济学、政治学、社会学、生物学等众多领域都有广泛的应用。

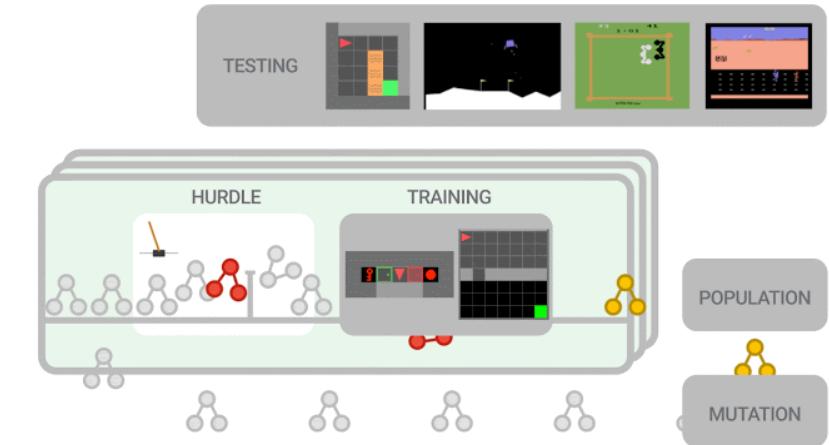
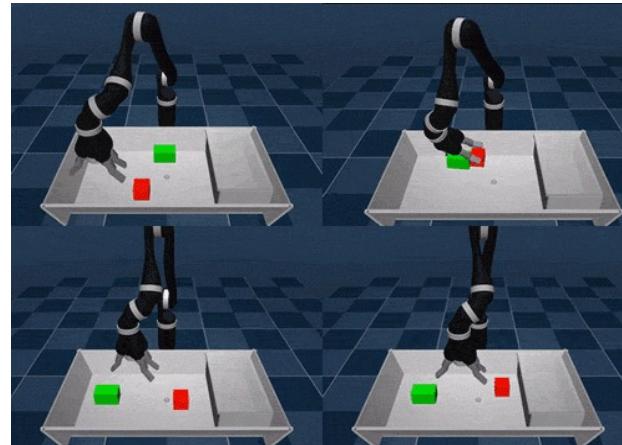
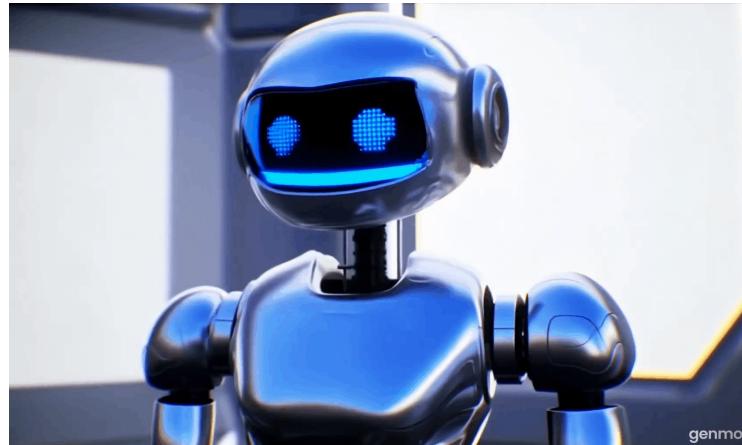
# 现实世界中的囚徒困境——美苏军备竞赛

冷战期间，美苏军备竞赛是一个典型的囚徒困境案例。双方都面临着是否扩充军事力量的选择。如果一方单方面裁军，可能会被对方认为软弱，从而遭受攻击，导致安全威胁增加。因此，理性决策会促使双方都的安全。然而，如果双方内看似能增强各自的实力，但会消耗大量的资源，最终导致政治压力和社会不安等。虽然都希望保持安全，却导致了全球范围内的资源浪费。这面符合囚徒困境的特征——每方内保证自身利益，但从全局来看，却导致了两方的共同损失，未能实现最优的集体利益。



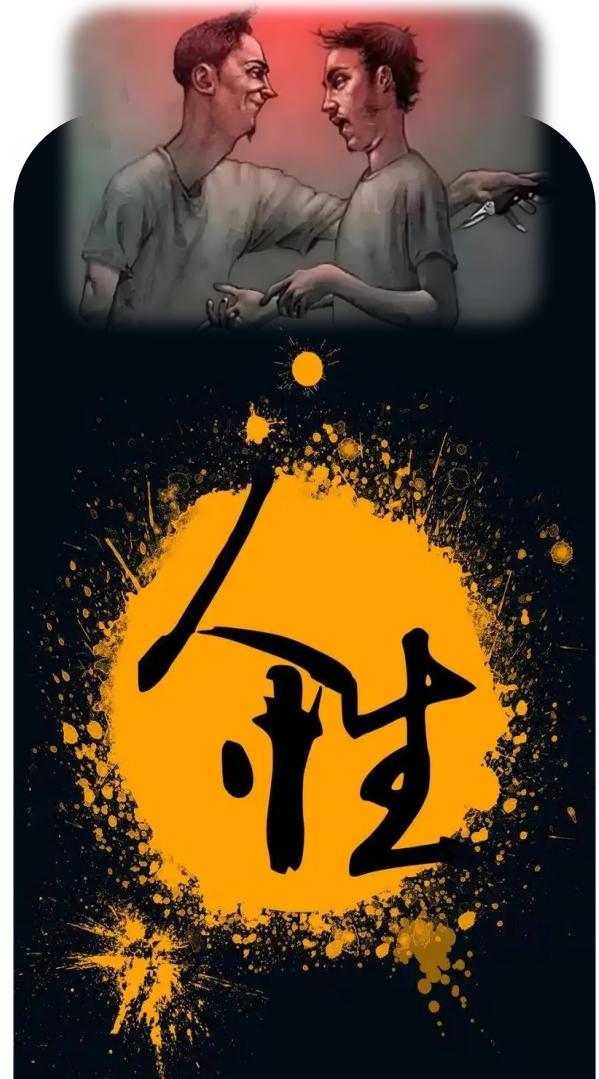
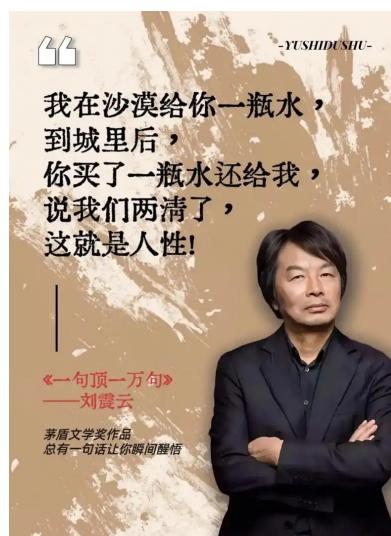
选择继续扩军，以保证自身的安全。然而，如果双方都选择扩军，虽然短期增强了防御能力，实际上却最终形成军事过剩、财政紧张等一系列问题。双方虽然都希望保持安全，却陷入了恶性循环，浪费和紧张局势。这种局的理性选择（扩军）虽然能在短期内保证自身利益，但从全局来看，却导致了两方的共同损失，未能实现最优的集体利益。

- 在囚徒困境中，AI可以利用强化学习（Reinforcement Learning）通过与环境互动来做出最优决策。通过不断试探和反馈，AI能够学习到不同策略的回报，并逐渐优化决策，以**寻找长期合作的最优策略**。尽管短期内背叛可能带来即时利益，AI通过模拟学习能够识别合作策略在多回合博弈中的优势，可以实现整体收益的最大化。这种策略使得AI在复杂博弈环境中做出理性且合作的决策。



# 哲学思考：囚徒困境的人性启示

- 囚徒困境揭示了理性决策与道德选择之间的张力。在追求个人利益的理性决策下，背叛常常成为最优选择，尽管这可能导致集体利益的损失。
- 从道德角度看，合作能够促进集体的长远利益，但短期内却可能牺牲个人利益。这种个人利益与集体利益的矛盾在现实社会中无处不在。要平衡信任与背叛，社会需要通过建立信任机制、完善法律和规范，引导个体在追求个人利益的同时，也能考虑到集体的长远福祉。



# 名堂

John  
Nash



1928年6月13日  
出生于美国弗吉尼亚州

1950年、1951年  
证明了非合作博弈及其均衡解

1952年，24岁，  
任教麻省理工

1958年患上精神分裂症

1994年获得诺贝尔经济学奖

2015年5月23日 因车祸去世

享年86岁

享年86岁

2015年5月23日 因车祸去世

约翰·纳什（John Nash）是博弈论的奠基人之一，他的研究为现代经济学和数学领域带来了深远的影响。他提出的纳什均衡理论，揭示了在多个参与者的博弈中，如何通过理性决策实现稳定的策略组合，成为博弈论的核心概念之一。为了表彰他在博弈论方面的杰出贡献，**1994年，纳什获得了诺贝尔经济学奖**。这个奖项不仅认可了他在理论上的成就，也为他在数学和经济学的交叉领域开辟了新的研究方向。纳什的研究不仅对经济学、政治学和生物学等多个领域产生了深远影响，也为现代决策科学提供了重要的理论基础。



# 天才简史

两个世界的爱  
约翰·纳什



山东大学  
SHANDONG UNIVERSITY

《人工智能通识》AI For Everyone

# 知识推理与问题求解

学无止境 气有浩然

教育部-华为“智能基座”课程