

人工智能及其应用（第 4 版）  
习题解析



# 第一章 绪论

- 1.1 什么是人工智能？试从学科和能力两方面加以说明。
- 1.2 在人工智能的发展过程中，有哪些思想和思潮起了重要作用？
- 1.3 在过去的 20 年中，人工智能发生了什么变化？
- 1.4 为什么能够用机器（计算机）模仿人的智能？
- 1.5 现在人工智能有哪些学派？它们的认知观是什么？现在这些学派的关系如何？
- 1.6 你认为应从哪些层次对认知行为进行研究？
- 1.7 你是如何理解人工智能的研究目标的？

- 1.8 人工智能研究包括哪些内容？这些内容的重要性如何？
- 1.9 人工智能的基本研究方法有哪几类？它们与人工智能学派的关系如何？
- 1.10 人工智能的主要研究和应用领域是什么？其中，哪些是新的研究热点？
- 1.11 你对人工智能课程教学有何意见和建议？

## 第二章 知识表示方法

- 2.1 状态空间法、问题归约法、谓词逻辑法和语义网络法的要点是什么？它们有何本质上的联系及异同点？

状态空间法：基于解答空间的问题表示和求解方法，它是以状态和算符为基础来表示和求解问题的。一般用状态空间法来表示下述方法：从某个初始状态开始，每次加一个操作符，递增的建立起操作符的试验序列，直到达到目标状态为止。

问题规约法：已知问题的描述，通过一系列变换把此问题最终变成一个子问题集合：这些子问题的解可以直接得到，从而解决了初始问题。问题规约的实质：从目标（要解决的问题）出发逆向推理，建立子问题以及子问题的子问题，直至最后把出示问题规约为一个平凡的本原问题集合。

谓词逻辑法：采用谓词合式公式和一阶谓词算法。要解决的问题变为一个有待证明的问题，然后采用消解定理和消解反演来证明一个新语句是从已知的正确语句导出的，从而证明这个新语句也是正确的。

语义网络法：是一种结构化表示方法，它由节点和弧线或链组成。节点用于表示物体、概念和状态，弧线用于表示节点间的关系。语义网络的解答是一个经过推理和匹配而得到的具有明确结果的新的语义网络。语义网络可用于表示多元关系，扩展后可以表示更复杂的问题。

- 2.2 设有 3 个传教士和 3 个野人来到河边，打算成一条船从右岸渡到左岸去。该船的负载能力为两人。在任何时候，如果野人人数超过传教士人数，那么野人就会把传教士吃掉。怎样才能用这条船安全地把所有人都渡过河去？

2.3 利用  $\times\times$ ，用状态空间法规划一个最短的旅行路程：此旅程从城市 A 开始，访问其他城市不多于一次，并返回 A。选择一个状态表示，表示出所求得的状态空间的节点及弧线，标出适当的代价，并指明图中从起始节点到目标节点的最佳路径。

2.4 试说明怎样把一棵与或解树用来表达  $\times\times$  所示的电网络阻抗的计算。单独的  $R$ 、 $L$  或  $C$  可分别用  $R$ 、 $j\omega L$  或  $1/j\omega C$  来计算，这个事实用作本原问题。后继算符应以复合并联和串联阻抗的规则为基础。

2.5 试用四元数列结构表示四圆盘梵塔问题，并画出求解该问题的与或图。

2.6 把下列句子变换成子句形式：

- (1)  $(\forall x) \{P(x) \rightarrow P(x)\}$
- (2)  $\forall x \forall y (\text{On}(x, y) \rightarrow \text{Above}(x, y))$
- (3)  $\forall x \forall y \forall z (\text{Above}(x, y) \wedge \text{Above}(y, z) \rightarrow \text{Above}(x, z))$
- (4)  $\sim \{(\forall x) \{(\forall y) [p(y) \rightarrow p(f(x, y))]\} \wedge (\forall y) [Q(x, y) \rightarrow P(y)]\}$

2.7 用谓词演算公式表示下列英文句子（多用而不是省用不同谓词和项。例如不要用单一的谓词字母来表示每个句子）。

A computer system is intelligent if it can perform a task which, if performed by a human, requires intelligence.

2.8 把下列语句表示称语义网络描述：

- (1) All men are moral.

- (2) Every cloud has a silver lining.
- (3) All branch manager of DEC participate in a profit-sharing plan.

2.9 试构造一个描述你的寝室或办公室的框架系统。

2.10 框架和本体有何关系与区别?





## 第三章 确定性推理

- 3.1 什么是图搜索过程？其中，重排 OPEN 表意味着什么？重排的原则是什么？

图搜索的一般过程如下：

- (1) 建立一个只含有起始节点  $S$  的搜索图  $G$ ，把  $S$  放到一个叫做 OPEN 的未扩展节点表中；
- (2) 建立一个叫 CLOSED 的已扩展节点表，其初始为空表；
- (3) LOOP：检查 OPEN 表是否为空，若为空，则失败退出；
- (4) 选择 OPEN 表上的第一个节点，把它从 OPEN 表移出放进 CLOSED 表。并记该节点为节点  $n$ ；
- (5) 考察节点  $n$  是否为目标节点。若  $n$  为一目标节点，则有解并成功退出。此解是追踪图  $G$  中沿着指针从  $n$  到  $S$  这条路径而得到的（指针将在第 7 步中设置）；
- (6) 扩展节点  $n$ ，生成一组子节点。把这些子节点中不是  $n$  的祖先的那些后继节点记入集合  $M$ 。把  $M$  的这些成员作为  $n$  的后继节点添入图  $G$  中；
- (7) 针对  $M$  中子节点的不同情况，分别作如下处理：
  - 未曾在  $G$  中出现过的  $M$  成员设置一个通向其父节点（即节点  $n$ ）的指针，并把  $M$  的这些成员加进 OPEN 表；（新生成的）
  - 对已经在 OPEN 或 CLOSED 表上的每一个  $M$  成员，确定是否需要更改通向其父节点  $n$  的指针方向；（原生成但未扩展的）
  - 对已在 CLOSED 表上的每个  $M$  成员，确定是否需要更改图  $G$  中通向它的每个后裔节点的指针方向；（原生成也扩

展过的)

(8) 按某一任意方式或按某个试探值, 重排 OPEN 表;

(9) GO LOOP。

重排 OPEN 表意味着什么, 在第 (6) 步中, 将优先扩展哪个节点, 不同的排序标准对应着不同的搜索策略。是否重新安排 OPEN 表, 即是否按照某个试探值 (或准则、启发信息等) 重新对未扩展节点进行排序, 将决定该图搜索过程是无信息搜索或启发式搜索。各种搜索策略的主要区别在于对 OPEN 表中节点的排列顺序不同。例如, 广度优先搜索把先生成的子节点排在前面, 而深度优先搜索则把后生成的子节点排在前面。

重排的原则当视具体需求而定, 不同的原则对应着不同的搜索策略, 如果想尽快地找到一个解, 则应当将最有可能达到目标节点的那些节点排在 OPEN 表的前面部分, 如果想找到代价最小的解, 则应当按代价从小到大的顺序重排 OPEN 表。

### 3.2 试举例比较各种搜索方法的效率。

**宽度优先搜索** 只要问题有解, 用宽度优先搜索总可以得到解, 而且得到的是路径最短的解。宽度优先搜索盲目性较大, 当目标节点距初始节点较远时将会产生许多无用节点, 搜索效率低。

**深度优先搜索** 搜索一旦进入某个分支, 就将沿着该分支一直向下搜索。如果目标节点恰好在此分支上, 则可较快地得到解。但是, 如果目标节点不在此分支上, 而该分支又是一个无穷分支, 则就不可能得到解。所以深度优先搜索是不完备的, 即使问题有解, 它也不一定能求得解。所求得的答案路径不一定是最短路径。

**有界深度优先搜索** 如果问题有解, 且其路径长度  $\leq dm$ , 则搜索过程一定能求得解。但是, 若解的路径长度  $> dm$ , 则搜索过程就得不到解。这说明在有界深度优先搜索中, 深度界限的选择是很重要的。要恰当地给出  $dm$  的值是比较困难的。即使能求出解, 它也不一定是最优解。

**等代价搜索** 是宽度优先搜索的一种推广, 不是沿着等长度路径断层进行扩展, 而是沿着等代价路径断层进行扩展。搜索树中每条

连接弧线上的有关代价，表示时间、距离等花费。若所有连接弧线具有相等代价，则简化为宽度优先搜索算法。

**盲目搜索** 具有较大的盲目性，产生的无用节点较多，效率不高，耗费过多的计算空间与时间。

**有序搜索** 利用启发信息，决定哪个是下一步要扩展的节点。选“最有希望”的节点作为下一个被扩展的节点。选择 OPEN 表上具有最小  $f$  值的节点作为下一个要扩展的节点，又称为最佳优先搜索。正确选择估价函数对搜索结果具有决定性的作用。使用不能识别某些节点真实希望的估价函数会形成非最小代价路径；使用一个过多地估计了全部节点希望的估价函数又会扩展过多的节点。

**A 算法** 虽提高了算法效率，但不能保证找到最优解。

**A\* 算法** 搜索效率很大程度上取决于估价函数  $h(n)$ 。一般来说，在满足  $h(n) \leq h^*(n)$  的前提下， $h(n)$  的值越大越好。 $h(n)$  的值越大，说明它携带的启发性信息越多，A\* 算法搜索时扩展的节点就越少，搜索效率就越高。

### 3.3 化为子句型有哪些步骤？请结合例子说明。

任一谓词演算公式可以化成一个子句集。其变换过程由下列九个步骤组成：

- (1) 消去蕴涵符号，将蕴涵符号化为析取和否定符号；
- (2) 减少否定符号的辖域，每个否定符号最多只用到一个谓词符号上，并反复应用狄·摩根定律；
- (3) 对变量标准化，对哑元改名，以保证每个量词有其自己唯一的哑元；
- (4) 消去存在量词，引入 Skolem 函数，消去存在量词。如果要消去的存在量词不在任何一个全称量词的辖域内，那么我们就用不含变量的 Skolem 函数即常量；
- (5) 化为前束形，把所有全称量词移到公式的左边，并使每个量词的辖域包括这个量词后面公式的整个部分；
- (6) 把母式化为合取范式，反复应用分配律，将母式写成许多合取项的合取的形式，而每一个合取项是一些谓词公式和 (或) 谓词公式的否定的析取；

- (7) 消去全称量词，消去前缀，即消去明显出现的全称量词；
- (8) 消去连词符号（合取），用合取项 1, 合取项 2 替换明显出现的合取符号；
- (9) 更换变量名称，更换变量符号的名称，使一个变量符号不出现在一个以上的子句中；

### 3.4 如何通过消解反演求取问题的答案。

给出一个公式集  $S$  和目标公式  $L$ ，通过反证或反演来求证目标公式  $L$ ，其证明步骤如下：

- (1) 否定  $L$ ，得  $\sim L$ ；
- (2) 把  $\sim L$  添加到  $S$  中去；
- (3) 把新产生的集合  $\{\sim L, S\}$  化成子句集；
- (4) 应用消解原理，力图推导出一个表示矛盾的空子句。

### 3.5 什么叫合式公式？合式公式有哪些等价关系？

合式公式的递归定义如下：

- (1) 原子谓词公式是合式公式。
- (2) 若  $A$  为合式公式，则  $\sim A$  也是一个合式公式。
- (3) 若  $A, B$  是合式公式，则  $A \vee B, A \wedge B, A \rightarrow B, A \leftrightarrow B$  也都是合式公式。
- (4) 若  $A$  是合式公式， $x$  为  $A$  中的自由变量，则  $(\forall x) A$  和  $(\exists x) A$  都是合式公式。
- (5) 运用有限步上述规则 (1) 至 (4) 求得的那些公式是合式公式。

等价关系有：

- (1) 否定之否定： $\sim(\sim P) = P$
- (2) 蕴含与与或形式的等价：
$$\begin{cases} (P \rightarrow Q) = (\sim P \vee Q) \\ (\sim P \rightarrow Q) = (P \vee Q) \end{cases}$$

### 3.6 用宽度优先搜索求 $\times\times$ 所示迷宫的出路。

- 3.7 用有界深度优先搜索方法求解  $\times\times$  所示八数码难题。
- 3.8 应用最新的方法表达传教士和野人问题，编写一个计算机程序，以求得安全渡过全部 6 个人的解答。（提示：在应用状态空间表示和搜索方法时，可用  $(N_m, N_c)$  来表示状态描述，其中  $N_m$  和  $N_c$  分别为传教士和野人的人数。初始状态为 (3,3)，而可能的中间状态为 (0,1),(0,2),(0,3),(1,1),(2,1),(2,2),(3,0),(3,1) 和 (3,2) 等。
- 3.9 试比较宽度优先搜索、有界深度优先搜索及有序搜索的搜索效率，并以实例数据加以说明。
- 3.10 一个机器人驾驶卡车，携带包裹（编号分别为 #1、#2 和 #3）分别投递到林 (LIN)、吴 (WU) 和胡 (HU) 3 家住宅处。规定了某些简单的操作符，如表示驾驶方位的  $\text{drive}(x,y)$  和表示卸下包裹的  $\text{unload}(z)$ ；对于每个操作符，都有一定的先决条件和结果。试说明状态空间问题求解系统如何能够应用谓词演算求得一个操作符序列，该序列能够生成一个满足  $\text{AT}(\#1, \text{LIN}) \wedge \text{AT}(\#2, \text{WU}) \wedge \text{AT}(\#3, \text{HU})$  的目标状态。
- 3.11 规则演绎系统和产生式系统有哪几种推理方式？各自的特点为什么？
- 3.12 单调推理有何局限性？什么叫缺省推理？非单调推理系统如何证实一个节点的有效性？

	正向推理	逆向推理
推理方向	从 if 部分向 then 部分推理的过程,它是从事实或状况向目标或动作进行操作的	从 then 部分向 if 部分推理的过程,它是从目标或动作向事实或状况进行操作的
目标表达式	文字的析取	任意形式
事实表达式	任意形式	文字的合取

单调系统不能很好地处理常常出现在现实问题领域中的 3 类情况,即不完全的信息、不断变化的情况、以及求解复杂问题过程中生成的假设。有两种方法可以证实节点的有效性:

- (1) 支持表。(SL (IN-节点表) (OUT-节点表))  
如果某节点的 IN 节点表中提到的节点当前都是 IN, 且 OUT 节点表中提到的节点当前都是 OUT, 则它是有效的。
- (2) 条件证明。(CP(结论) (IN-假设) (OUT-假设))  
条件证明 (CP) 的证实表示有前提的论点, 无论何时, 只要在 IN 假设中的节点为 IN, OUT 假设中的节点为 OUT, 则结论节点往往为 IN, 于是条件证明的证实有效。

3.13 在什么情况下需要采用不确定推理或非单调推理?

不完全的信息、不断变化的情况、以及求解复杂问题过程中生成的假设。

3.14 下列语句是一些几何定理, 把这些语句表示为基于规则的几何证明系统的产生式规则:

- (1) 两个全等三角形的各对应角相等。
- (2) 两个全等三角形的各对应边相等。
- (3) 各对应边相等的三角形是全等三角形。
- (4) 等腰三角形的两底角相等。

	正向推理	逆向推理
驱动方式	数据驱动	目标驱动
推理方法	从一组数据出发 向前推到结论	从可能的解答出发, 向后推理验证 解答
启动方法	从一个事件启动	由询问关于目标 状态的一个问题 而启动
透明程度	不能解释其推理 过程	可解释其推理过 程
推理方向	由底向上推理	由顶向下推理
优点	算法简单、容易实 现, 允许用户一开 始就把有关的事 实数据存入数据 库, 在执行过程中 系统能很快获得 这些数据, 而不必 等到系统需要数 据时才向用户询 问	搜索目的性强, 推 理效率高
缺点	盲目搜索, 可能会 求解许多与总目 标无关的子目标, 每当总数据库内 容更新后都要遍 历整个规则库, 推 理效率较低	目标的选择具有 盲目性, 可能会求 解许多假的目标; 当可能的结论数 目很多, 即目标空 间很大时, 推理效 率不高; 当规则的 右部是执行某种 动作不是结论时, 逆向推理不便使 用
适用场合	主要用于已知初 始数据, 而无法提 供推理目标, 或解 空间很大的一类 问题, 如监控、预 测、规划、设计等 问题的求解	主要用于结论单 一或者已知目标 结论, 而要求验证 的系统, 如选择、 分类、故障诊断等 问题的求解
典型系统	CLIPS, OPS	PROLOG

- (1) IF 两个三角形全等 THEN 各对应角相等
- (2) IF 两个三角形全等 THEN 各对应边相等
- (3) IF 两个三角形各对应边相等 THEN 两三角形全等
- (4) IF 它是等腰三角形 THEN 它的两底角相等