

人工智能重点总结

第一章：开展简史 [此处为简答题]

1. 人工智能的萌芽 [1956 年以前]

1936 年，图灵创立了自动机理论 [后人称为图灵机]，提出一个理论计算机模型，为电子计算机设计奠定了根底，促进了人工智能，特别是思维机器的研究。

麦克洛克和皮茨于 1943 年提出“拟脑模型”是世界上第一个神经网络模型 [MP 模型]，开创了从结构上研究人类大脑的途径。

1948 年维纳发表?控制论—关于动物与机器中的控制与通信的科学?，不但开创了近代控制论，而且为人工智能的控制学派树立了里程碑。

- 1、古希腊伟大的哲学家思想家亚里士多德的主要奉献是为形式逻辑奠定了根底。形式逻辑是一切推理活动的最根本的出发点。在他的代表作?工具论?中，就给出了形式逻辑的一些根本规律，如矛盾律、排中律，并且实际上已经提到了同一律和充足理由律。此外亚里士多得还研究了概念、判断问题，以及概念的分类和概念之间的关系判断问题的分类和它们之间的关系。其最著名的创造就是提出人人熟知的三段论。
- 2、英国的哲学家、自然科学家 Bacon [培根] [1561–1626]，他的主要奉献是系统地给出了归纳法，成为和 Aristotle 的演绎法相辅相成的思维法那么。Bacon 另一个功绩是强调了知识的作用。Bacon 的著名警句是“知识就是力量”。
- 3、德国数学家、哲学家 Leibnitz [莱布尼茨] [1646–1716]，他提出了关于数理逻辑的思想，把形式逻辑符号化，从而能对人的思维进行运算和推理。他曾经做出了能进行四那么运算的手摇计算机
- 4、英国数学家、逻辑学家 Boole [布尔] [1815–1864]，他初步实现了布莱尼茨的思维符号化和数学化的思想，提出了一种崭新的代数系统 — 布尔代数。
- 5、美籍奥地利数理逻辑学家 Godel [哥德尔] [1906–1978]，他证明了一阶谓词的完备性定理；任何包含初等数论的形式系统，如果它是无矛盾的，那么一定是不完备的。此定理的意义在于，人的思维形式化和机械化的某种极限，在理论上证明了有些事是做不到的。

- 6、英国数学家 Turing(图灵) [1912–1954]，1936 年提出了一种理想计算机的数学模型 [图灵机]，1950 年提出了图灵试验，发表了“计算机与智能”的论文。当今世界上计算机科学最高荣誉奖励为“图灵奖”。名词解释：名词解释：图灵试验。当一个人与一个封闭房间里的人或者机器交谈时，如果他不能分辨自己问题的答复是计算机还是人给出时，那么称该机器是具有智能的。以往该试验几乎是衡量机器人工智能的唯一标准，但是从九十年代开始，现代人工智能领域的科学家开始对此试验提出异议：反对封闭式的，机器完全自主的智能；提出与外界交流的，人机交互的智能。
- 7、美国数学家 Mauchly，1946 创造了电子数字计算机 ENIAC
- 8、美国神经生理学家 McCulloch，建立了第一个神经网络数学模型。从某种意义上可以说近代人工智能的开展，首先是从人工神经网络研究开始的。但是由于某种原因，神经网络的研究一度进入低潮。详细内容参见第六章?人工神经元网络？
- 9、美国数学家 Shannon [香农]，1948 年发表了?通讯的数学理论?，标志着“信息论”的诞生。
- 10、美国数学家、计算机科学家 McCarthy，人工智能的早期研究者。1956 年，他和其他一些学者联合发起召开了世界上第一次人工智能学术大会，在他的提议下，会上正式决定使用人工智能这个词来概括这个研究方向。参加大会的有 Minsky, Rochester, Shannon, Moore, Samuel, Selfridge, Solomonoff, Simon, Newell 等数学家、心理学家、神经生理学家、计算机科学家。McCarthy 也被尊为“人工智能之父”。

2. 人工智能的形成 [1956–1969 年]

费根鲍姆于 1968 年研究成功第一个专家系统 DENDRAL，用于质谱仪分析有机化合物的分子结构。

1969 年召开了第一届国际人工智能会议，标志着人工智能作为一门独立学科登上国际学术舞台。

1970 年?人工智能国际杂志?创刊。

◆ 50 年代初开始有了符号处理，搜索法产生。

人工智能的根本方法是逻辑法和搜索法。最初的搜索应用于机器翻译、机器定理证明、跳棋程序等。

- ◆ 60 年代 Simon 由试验得到结论：人类问题的求解是一个搜索的过程，效果与启发式函数有关。表达了智能系统的特点：智能表示、智能推理、智能搜索。
- ◆ Nilson发表了 A* 算法〔搜索方法〕
- ◆ McCarthy 建立了人工智能程序设计语言 Lisp
- ◆ 1965 年 Robinson提出了归结原理。
- ◆ 1968 年 Quillia 提出了语义网络的知识表示方法
- ◆ 1969 年 Minsky 出了一本书“感知机”，给当时的神经网络研究结果判了死刑

3. 人工智能的开展〔1970 年以后〕

费根鲍姆 1972—1976 年成功开发 MYCIN 医疗专家系统，用于抗生素药物治疗
1987 年在美国召开第一届神经网络国际会议，并发起成立国际神经网络学会〔INNS〕

- 1989 年首次召开了中国人工控制联合会议〔CJCAI〕
- ◆ 70 年代，人工智能开始从理论走向实践，解决一些实际问题。同时很快就发现问题：归结法费时、下棋赢不了全国冠军、机器翻译一团糟。此时，以 Feigenbaum 为首的一批年轻科学家改变了战略思想，1977 年提出了知识工程的概念，开展了以知识为根底的专家咨询系统研究与应用。

著名的专家系统有：

- DENDRAL 化学分析专家系统〔斯坦福大学 1968〕；
MACSYMA 符号数学专家系统〔麻省理工 1971〕；
MYCIN 诊断和治疗细菌感染性血液病的专家咨询系统〔斯坦福大学 1973〕；
CASNET (Causal ASsociation Network) 诊断和治疗青光眼的专家咨询系统
〔拉特格尔斯〔Rutgers〕大学 70 年代中〕；
CADUCEUS (原名 INTERNIST) 医疗咨询系统〔匹兹堡大学〕；
HEARSAY I 和 II 语音理解系统〔卡内基梅隆大学〕；
PROSPECTOR 地质勘探专家系统〔斯坦福大学 1976〕；
XCON 计算机配置专家系统〔卡内基梅隆大学 1978〕。

应该说，知识工程和专家系统是近十余年来人工智能研究中最有成就的分支

之一。

◆ 80 年代，人工智能开展到达阶段性的顶峰。87, 89年世界大会有 6-7 千人参加。硬件公司有上千个。Lisp 硬件、Lisp 机形成产品。同时，在专家系统及其工具越来越商品化的过程中，国际软件市场上形成了一门旨在生产和加工知识的新产业—知识产业。

◆ 同年代，1986 年 Rumhart 领导的并行分布处理研究小组提出了神经元网络的反向传播学习算法，解决了神经网络分类能力有限这一根本问题。从此，神经网络的研究进入新的高潮。

◆ 90 年代，计算机开展趋势为小型化、并行化、网络化、智能化。人工智能技术逐渐与数据库、多媒体等主流技术相结合，并融合在主流技术之中，旨在使计算机更聪明、更有效、与人更接近。

二、三大学派：

1、**符号主义**(Symbolicism)，又称为逻辑主义(Logicism)、心理学派(Psychlogism)或计算机学派(Computerism)，其原理主要为物理符号系统(即符号操作系统)假设和有限合理性原理。

符号主义学派认为： 人工智能源于数学逻辑。

代表性成果： 是启发式程序 LT 逻辑理论家，证明了 38 条数学定理，说明我们可以应用计算机研究人的思维过程，模拟人类智能活动。

代表人物：纽厄尔、肖 • 西蒙和尼尔逊。

2、**联结主义**(Connectionism)，又称为仿生学派(Bionicsism)或生理学派(Physiologism)，其原理主要为神经网络及神经网络间的连接机制与学习算法。

这一学派认为： 人工智能源于仿生学，特别是人脑模型的研究

代表性成果： 1943 年由麦克洛奇和皮兹提出的形式化神经元模型，即 M-P 模型

代表人物： 麦克洛奇、皮兹、霍普菲尔特、鲁梅尔哈特

3、行为主义(Actionism)，又称进化主义(Evolutionism)或控制论学派(Cyberneticsism)，其原理为控制论。

这一学派认为： 人工智能源于控制论

代表性成果： 布鲁克斯的六足机器人，它被看做新一代的“控制论动物”，是一个基于感知——动作模式的模拟昆虫行为的控制系统。

代表人物： 布鲁克斯

第二章 知识表示

1. 状态空间 [在搜索那里考一个大题]

了解个三元状态 $[S, F, G]$ ，其中 S : 初始状态集， F 操作符集合

G ：目标状态集合 [这里只用了解个大概就可以了，详细在搜索局部介绍]

2. 问题归约 [只考一个名词解释]

解树：由可解节点构成，并且由这些可解节点可推出初始节点 [对应初始问题] 为可解节点的子树称为解树

3. 谓词表示法 [会在第二道大题中考 4-5 个应用]

用谓词公式表示知识时，需要首先定义谓词，然后再用连接词把有关的谓词连接起来，形成一个谓词公式表达一个完整的意义。

例题 设有以下知识：

① 刘欢比他父亲知名。

② 高扬是计算机系的一名学生，但他不喜欢编程。

为了用谓词公式表示上述知识，首先需要定义谓词：

BIGGER(x, y) : x 比 y 知名

COMPUTER(x) : 是计算机系的

LIKE(x, y) : 喜欢 x y

解答：此时可用谓词公式把上述知识表示为：

① BIGGER(liuhuan, father(liuhuan)) 觉得那个 father

函数最好也定义下，保险一点]

② COMPUTER(gaoyang) \wedge LIKE(gaoyang, programing)

总结：〔上面的例题应该就是考试的形式〕

A)首先必须知道什么是合取、析取、蕴含、否认以及两种量词的用法

B)全称量词后面跟蕴含，存在量词后面跟合取

C)必须先定义〔切记〕，再表示。一般步骤为

1>提取谓词，使用类似于 $P[x, y]$ ：谓词内容 的格式定义谓词

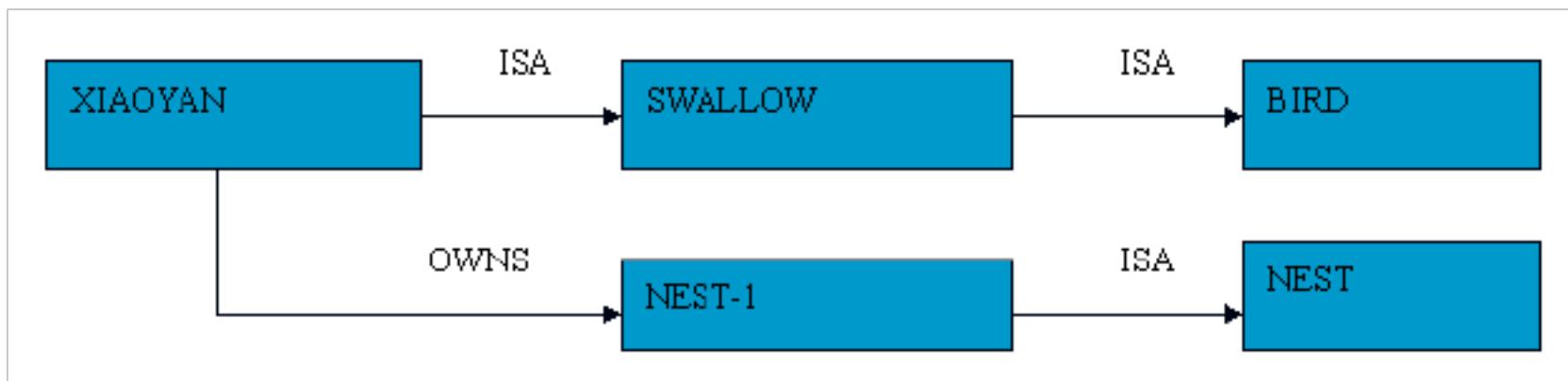
2>用连接词和量词加以表示

D)置换和合一那会用就 OK 了。只会在归结演绎推理那块最后的证明时用一下，
不理解的话看那个“黄书”P81 中那个反演树里用到的置换。

4.语义网络〔会考画图题〕

只考二元关系网络

例题 小燕是一只燕子，燕子是鸟；巢-1是小燕的巢，巢-1是巢中的一个。”



注意：

A) 语义网络中不会考量词、继承、匹配

B) 就根据题目所描述的写，不要笨的写什么小明 ISA 人 ISA 动物 ISA 生物…… 题目上怎么说怎么写就可以〔老师原话〕

5.框架表示〔只有概念题〕

1>框架：我们无法把过去的经验一一都存在脑子里，而只能以一个通用的数据结构的形式存储以往的经验。这样的数据结构称为框架

2>框架的构成：

框架通常由描述事物的各个方面的槽组成，每个槽可以拥有假设干个侧面，而每个侧面又可拥有假设干个值。一个框架的一般结构如下：

〈框架名〉

 〈槽 1〉〈侧面 11〉〈值 111〉…〈侧面 12〉〈值 121〉… …

 〈槽 2〉〈侧面 21〉〈值 211〉…

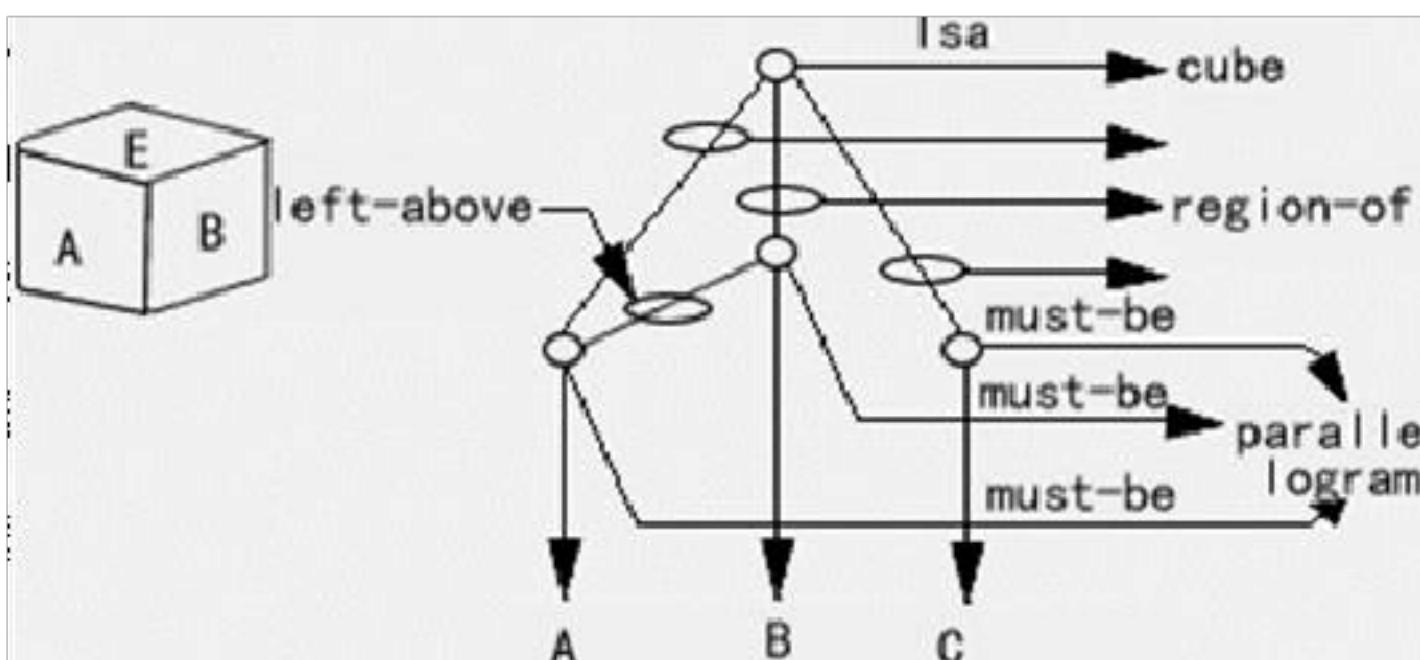
...
...

<槽 n><侧面 n1><值 n11>...

...

<侧面 nm>< 值 nm1> ...

3>一个框架系统 [我觉得应该不会考这个，保险起见所以放上来了] 以下图所示为表示立方体的一个视图的框架。图中，最高层的框架，用 isa槽说明它是一个立方体，并由 region槽指示出它所拥有的 3 个可见面 A、B、E。而 A、B、E 又分别用 3 个框架来具体描述。用 must-be 槽指示出它们必须是一个平行四边形。为了能从各个不同的角度来描述物体，可以对不同角度的视图分别建立框架，然后再把它们联系起来组成一个框架系统。以下图所示的就是从 3 个不同的角度来研究一个立方体的例子



6.过程、剧本表示不考

第三章 经典逻辑推理

3. 1 归结演绎推理 [问题求解&证明]

定理证明即证明 $P \rightarrow Q (P \vee Q)$ 的永真性。根据反证法，只要证明其否认 ($P \wedge Q$) 不可满足性即可。

海伯伦(Herbrand)定理为自动定理证明奠定了理论根底；鲁宾逊(Robinson)提出的归结原理使机器定理证明成为现实。

在谓词逻辑中，把原子谓词公式及其否认统称为文字。如： $P(x)$, $P(x, f(x))$, $Q(x, g(x))$ 任何文字的析取式称为子句，不包含任何文字的子句称为空子句。

(1) 合取范式： $C_1 \wedge C_2 \wedge C_3 \dots \wedge C_n$

(2) 子句集： $S = \{C_1, C_2, C_3 \dots, C_n\}$

(3) 任何谓词公式 F 都可通过等价关系及推理规则化为相应的子句集 S 。

子句集的性质：

(1) 子句集中子句之间是合取关系。

(2) 子句集中的变元受全称量词的约束。

把谓词公式化成子句集的步骤：

1) 利用等价关系消去 “ \rightarrow ” 和 “ \neg ”

例如公式 $(\exists x)((\forall y)P(x, y) \neg (\forall y)Q(x, y) \neg R(x, y))$

可等价转换成 $(\exists x)((\forall y)P(x, y) (\forall y)(Q(x, y) \neg R(x, y)))$

2) 利用等价关系把 “ \neg ” 移到紧靠谓词的位置上

上式经等价转换后 $(\exists x)((\forall y)P(x, y) (\forall y)Q(x, y) \neg R(x, y))$

3) 重新命名变元，使不同量词约束的变元有不同的名字

上式经转换后 $(\exists x)((\forall y)P(x, y) (\forall z)Q(x, z) \neg R(x, z))$

4) 消去存在量词

a. 存在量词不出现在全称量词的辖域内，那么只要用一个新的个体常量替换受该量词约束的变元。

b. 存在量词位于一个或者多个全称量词的辖域内，此时要用

Skolem 函数 $f(x_1, x_2, \dots, x_n)$ 替换受该存在量词约束的变元。

上式中存在量词 $(\exists y)$ 及 $(\exists z)$ 都位于 $(\exists x)$ 的辖域内，所以需要用

Skolem 函数替换，设替换 y 和 z 的 Skolem 函数分别是 $f(x)$ 和 $g(x)$ ，

那么替换后得到 $(\exists x)(P(x, f(x)) Q(x, g(x)) \neg R(x, g(x)))$

- 5) 把全称量词全部移到公式的左边
 6) 利用等价关系把公式化为 Skolem 标准形

$$P \quad (Q \quad R) \quad (P \quad Q) \quad (P \quad R)$$

Skolem 标准形的一般形式是
 $(\underset{1}{x})(\underset{2}{x})\dots(\underset{n}{x})M$

其中， M 是子句的合取式，称为 Skolem 标准形的母式。上式化为 Skolem 标准形后得到：

$$(\underset{}{x})((P(x, f(x)) \quad Q(x, g(x)))) \quad (P(x, f(x)) \quad R(x, g(x)))$$

- 7) 消去全称量词

- 8) 对变元更名，使不同子句中的变元不同名。上式化为

$$(P(x, f(x)) \quad Q(x, g(x))) \quad (P(y, f(y)) \quad R(y, g(y)))$$

- 9) 消去合取词，就得到子句集

$$\begin{aligned} & P(x, f(x)) \quad Q(x, g(x)) \\ & P(y, f(y)) \quad R(y, g(y)) \end{aligned}$$

推论 1 设 C_1 与 C_2 是子句集 S 中的两个子句， C_{12} 是它们的归结式。假设用 C_{12} 代替 C_1 和 C_2 后得到新子句集 S_1 ，那么由 S_1 的不可满足性可推出原子句集 S 的不可满足性，即：

$$S_1 \text{ 的不可满足性 } \rightarrow S \text{ 的不可满足性}$$

推论 2 设 C_1 与 C_2 是子句集 S 中的两个子句， C_{12} 是它们的归结式。假设把 C_{12} 参加 S 中得到新子句集 S_2 ，那么 S 与 S_2 在不可满足的意义上是等价的，即：

$$S_2 \text{ 的不可满足性 } \leftrightarrow S \text{ 的不可满足性}$$

推论 1 及推论 2 保证了我们可以用归结的方法来证明子句集 S 的不可满足性。

为了要证明子句集 S 的不可满足性，只要对其中可进行归结的子句进行归结，并把归结式参加子句集 S ，或者用归结式替换它的亲本子句，

然后对新子句集 (S_1 或者 S_2) 证明不可满足性就可以了。如果经过归结能得到空子句，那么立即可得原子句集 S 是不可满足的结论。

在命题逻辑中，对不可满足的子句集 S ，归结原理是完备的。即，假设子句集不可满足，那么必然存在一个从 S 到空子句的归结演绎；假设存在一个从 S 到空子句的归结演绎，那么 S 一定是不可满足的。

命题逻辑中的归结原理

定义 4.9 假设 P 是原子谓词公式，那么称 P 与 $\neg P$ 为互补文字。在命题逻辑中， P 为命题。

定义 4.10 设 C_1 与 C_2 是子句集中的任意两个子句。如果 C_1 中的文字 L_1 与 C_2 中文字 L_2 互补，那么从 C_1 和 C_2 中分别消去 L_1 和 L_2 ，并将两个子句中余下的局部析取，构成一个新子句 C_{12} ，那么称这一过程为归结。称 C_{12} 为 C_1 和 C_2 的归结式， C_1 和 C_2 为 C_{12} 的亲本子句。

例 4.9 设 $C_1 = P \vee Q$, $C_2 = Q \vee R$, $C_3 = P$

C_1 与 C_2 归结得到: $C_{12} = P \vee R$

C_{12} 与 C_3 归结得到: $C_{123} = R$

谓词逻辑中的归结原理

在谓词逻辑中，由于子句中含有变元，所以不能像命题逻辑那样直接消去互补文字，而需要先用最一般合一对变元进行代换，然后才能进行归结。

例如，设有两个子句

$$C_1 = P(x) \vee Q(x), \quad C_2 = P(a) \vee R(y)$$

由于 $P(x)$ 与 $P(a)$ 不同，所以 C_1 与 C_2 不能直接进行归结。但是假设用最一般合一

$$\sigma = \{a/x\}$$

对两个子句分别进行代换：

$$C_1 \sigma = P(a) \vee Q(a) \quad C_2 \sigma = P(a) \vee R(y)$$

就可对它们进行归结，得到归结式：

$$Q(a) \vee R(y)$$

归结反演及其例如*

如欲证明 Q 为 P_1, P_2, \dots, P_n 的逻辑结论，只需证

$$(P_1 \wedge P_2 \wedge \dots \wedge P_n) \wedge Q$$

是不可满足的，或证明其子句集是不可满足的。而子句集的不可满足性可用归结原理来证明。

应用归结原理证明定理的过程称为归结反演。

设 F 为前提的公式集， Q 为目标公式(结论)，用归结反演证明 Q 为真的步骤是：

- a) 否认 Q ，得到 $\neg Q$ ；
- b) 把 $\neg Q$ 并入到公式集 F 中，得到 $\{F, \neg Q\}$ ；
- c) 把公式集 $\{F, \neg Q\}$ 化为子句集 S ；
- d) 应用归结原理对子句集 S 中的子句进行归结，并把每次归结得到的归结式都并入 S 中。如此反复进行，假设出现了空子句，那么停止归结，此时就证明了 Q 为真。

例 4.12

$$\begin{aligned} F : & (\exists x) ((\forall y) (A(x, y) \wedge B(y)) \quad (\forall y) C(y) \wedge D(x, y))) \\ G : & (\exists x) C(x) \quad (\exists x) (\forall y) (A(x, y) \wedge B(y)) \end{aligned}$$

求证： G 是 F 的逻辑结论。

证明：首先把 F 和 G 化为子句集：

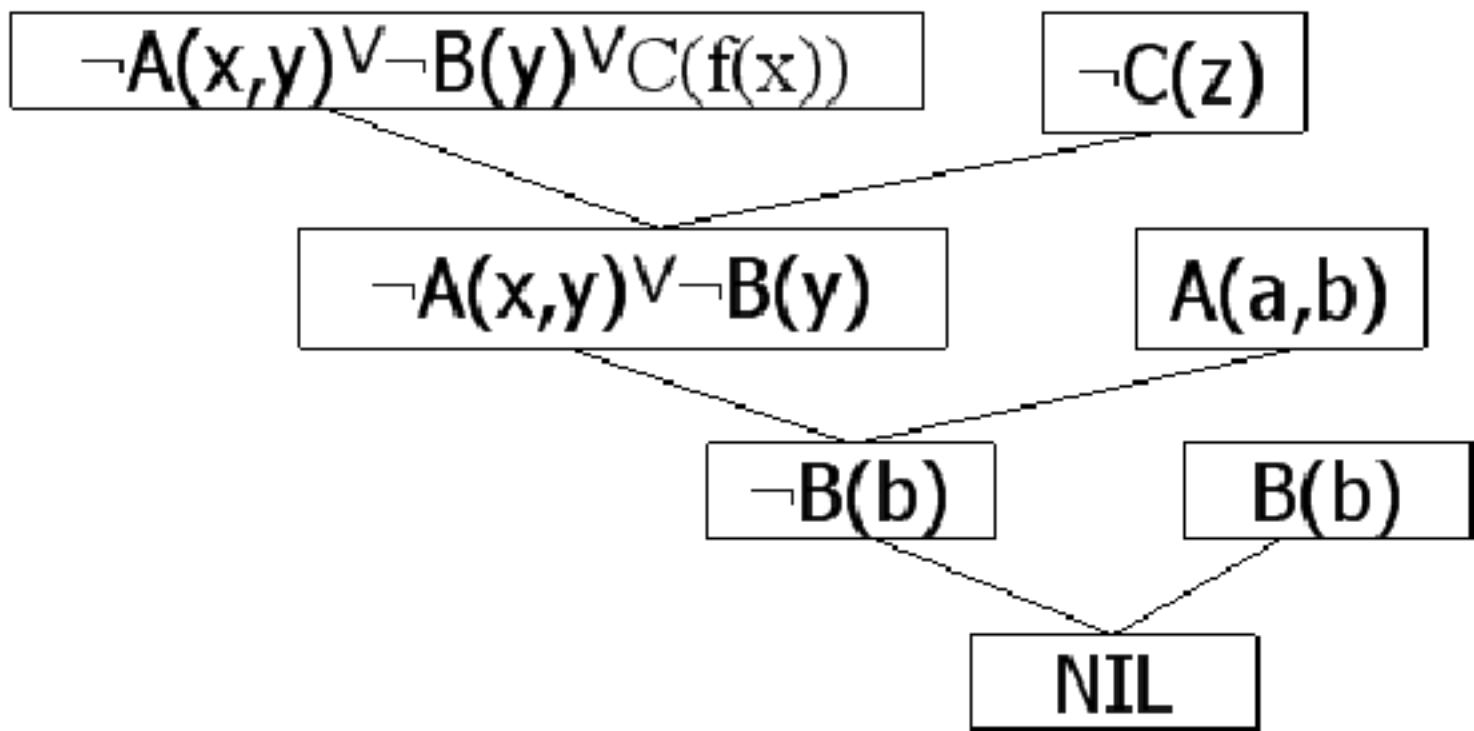
$$\begin{aligned} F & \{ A(x, y) \wedge B(y) \wedge C(f(x)), A(x, y) \wedge B(y) \wedge D(x, f(x)) \} \\ G & \{ C(z), A(a, b), B(b) \} \end{aligned}$$

然后进行归结：

- (6) $A(x, y) \wedge B(y)$ 由(1)与(3)归结， $\{f(x)/z\}$
- (7) $B(b)$ 由(4)与(6)归结， $\{a/x, b/y\}$
- (8) NIL 由(5)与(7)归结

所以 G 是 F 的逻辑结论。

上述归结过程如以下图归结树所示。



应用归结原理求取问题的答案及其例如

归结时，并不要求把子句集中所有的子句都用到。

在归结过程中，一个子句可以屡次被用来进行归结。

求解的步骤：

1. 把前提用谓词公式表示出来，并且化为相应的子句集。设该子句集的名字为 S 。
2. 把待求解的问题也用谓词公式表示出来，然后把它否认并与谓词 Answer 构成析取式。 Answer 是一个为了求解问题而专设的谓词。
3. 把此析取式化为子句集，并且把该子句集并入到子句集 S 中，得到子句集 S' 。
4. 对 S' 应用归结原理进行归结。
5. 假设得到归结式 Answer ，那么答案就在 Answer 中。

例 设 A, B, C 三人中有人从不说真话，也有人从不说假话。某人向这三人分别提出同一个问题：谁是说谎者？ A 答：“ A 和 C 都是说谎者”； B 答：“ A 和 C 都是说谎者”； C 答：“ A 和 B 中至少有一个是说谎者”。求谁是老实人，谁是说谎者？

解：设用 $T(x)$ 表示 x 说真话。

$$T(C) \vee T(A) \vee T(B)$$

$$T(C) \vee T(A) \vee T(B)$$

$$T(A) \rightarrow T(B) \wedge T(C)$$

$$T(A) \rightarrow T(B) \vee T(C)$$

$$T(B) \rightarrow T(A) \wedge T(C)$$

$$T(B) \rightarrow T(A) \vee T(C)$$

$$T(C) \rightarrow T(A) \vee T(B)$$

$$T(C) \rightarrow T(A) \wedge T(B)$$

把上述公式化成子句集，得到 S ：

$$(1) T(A) \vee T(B)$$

$$(2) T(A) \vee T(C)$$

$$(3) T(C) \vee T(A) \vee T(B)$$

$$(4) T(B) \vee T(C)$$

$$(5) T(C) \vee T(A) \vee T(B)$$

$$(6) T(A) \wedge T(C)$$

$$(7) T(B) \wedge T(C)$$

下面先求谁是老实人。把 $T(x) \vee \text{Ansewer}(x)$ 并入 S 得到 S_1 。即多一个子句：

$$(8) T(x) \vee \text{Ansewer}(x)$$

应用归结原理对 S_1 进行归结：

$$(9) T(A) \vee T(C) \quad (1) \text{和} (7) \text{归结}$$

$$(10) T(C) \quad (6) \text{和} (9) \text{归结}$$

$$(11) \text{Ansewer}(C) \quad (8) \text{和} (10) \text{归结}$$

所以 C 是老实人，即 C 从不说假话。

下面证明 A 不是老实人，即证明 $\neg T(A)$ 。

对 $T(A)$ 进行否认，并入 S 中，得到子句集 S_2 ，即 S_2 比 S 多如下子句：

$$(8) (\neg T(A)), \text{ 即 } T(A)$$

应用归结原理对 S_2 进行归结：

$$(9) T(A) \vee T(C) \quad (1) \text{和} (7) \text{归结}$$

$$(10) T(A) \quad (2) \text{和} (9) \text{归结}$$

$$(11) \text{NIL} \quad (8) \text{和} (10) \text{归结}$$

所以 A 不是老实人。同样可以证明 B 也不是老实人。

应用归结原理的练习

1. 设：

[1] 如果 x 是 y 的父亲， y 是 z 的父亲，那么 x 是 z 的祖父；

[2] 每个人都有一个父亲。

试用归结演绎推理证明：对于某人 u ，一定存在一个人 v ， v 是 u 的祖父。

2. 张某被盗，公安局派出五个侦察员去调查。研究案情时，侦察员 A 说“赵与钱中至少有一人作案”；侦察员 B 说“钱与孙中至少有一人作案”；侦察员 C 说“孙与李中至少有一人作案”；侦察员 D 说“赵与孙中至少有一人与此案无关”；侦察员 E 说“钱与李中至少有一人与此案无关”。如果这五个侦察员的话都是可信的，试用归结演绎推理求出谁是盗窃犯。

不确定性——只考模糊理论

简单模糊推理

知识中只含有简单条件，且不带可信度因子的模糊推理称为简单模糊推理。

合成推理规则：对于知识

IF x is A THEN y is B

首先构造出 A 与 B 之间的模糊关系 R，然后通过 R 与证据的合成求出结论。

如果证据是

x is A'

且 A 与 A' 可以模糊匹配，那么通过下述合成运算求取 B'：

$B' = A \cdot R$

如果证据是

y is B'

且 B 与 B' 可以模糊匹配，那么通过下述合成运算求出 A'：

$A' = R \cdot B'$

模糊集的运算

模糊集上的运算主要有：包含、交、并、补等等。

1. 包含运算

定义 2.14 设 $A, B \in F(U)$, 假设对任意 $u \in U$, 都有

$$\mu_B(u) < \mu_A(u)$$

成立, 那么称 A 包含 B, 记为 $A \supset B$ 。

2. 交、并、补运算

定义 设 $A, B \in F(U)$, 以下为扎德算子

$$\begin{aligned} A \cup B : \quad & \max_{u \in U} \{ \mu_A(u), \mu_B(u) \} & A(u) & B(u) \\ A \cap B : \quad & \min_{u \in U} \{ \mu_A(u), \mu_B(u) \} & A(u) & B(u) \\ A : \quad & 1 - \mu_A(u) & A(u) & \end{aligned}$$

例 设 $U = \{u_1, u_2, u_3\}$,

$$u_1 u_2 u_3$$

$$u_1 u_2 u_3$$

那么:

$$\wedge 0.6) \mu_1 \wedge 0.4) \mu_2 \wedge 0.7) \mu_3$$

$$u_1 u_2 u_3$$

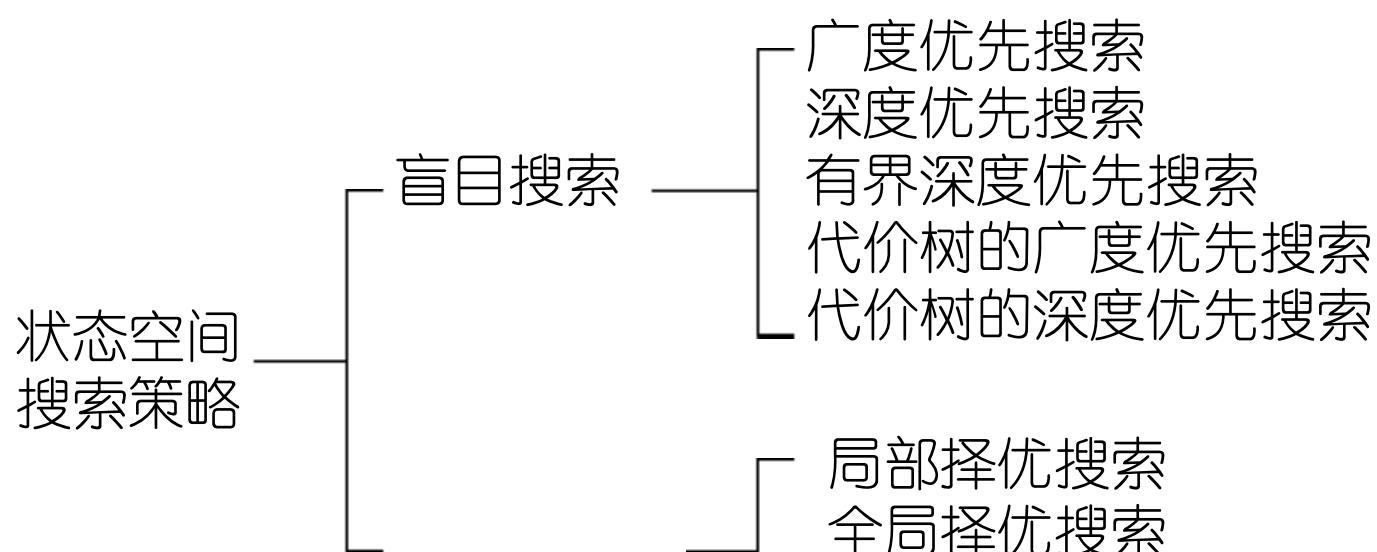
$$A \cup \vee 0.6) \mu_1 \vee 0.4) \mu_2 \vee 0.7) \mu_3$$

$$u_1 u_2 u_3$$

$$A = (1 - 0.3) \mu_1 + (1 - 0.8) \mu_2 + (1 - 0.6) \mu_3$$

$$u_1 u_2 u_3$$

3. 3 搜索—广度优先、深度优先、全局择优三选一



状态空间的一些根本概念

- 1) 很多问题的求解过程都可以看作是一个搜索过程。问题及其求解过程可以用状态空间表示法来表示。
- 2) 状态空间用“状态”和“算符”来表示问题。

状态

状态用以描述问题在求解过程中不同时刻的状态，一般用一个向量表示：

$$SK = (Sk_0, Sk_1, \dots)$$

算符

使问题从一个状态转变为另一个状态的操作称为算符。在产生式系统中，一条产生式规则就是一个算符。

状态空间

由所有可能出现的状态及一切可用算符所构成的集合称为问题的状态空间。

- 3) 采用状态空间求解问题，可以用下面的一个三元组表示：

$$(S, F, G)$$

其中 S 是问题初始状态的集合； F 是算符的集合； G 是目标状态的集合。

采用状态空间表示方法，首先要把问题的一切状态都表示出来，其次要定义一组算符。

问题的求解过程是一个不断把算符作用于状态的过程。如果在使用某个算符后得到的新状态是目标状态，就得到了问题的一个解。这个解

就是从初始状态到目标状态所采用算符的序列。使用算符最少的解称为最优解。

对任何一个状态，可使用的算符可能不止一个。这样由一个状态所生成的后继状态就可能有多个。此时首先对哪一个状态进行操作，就取决于搜索策略。

OPEN 表和 CLOSE 表

OPEN 表用于存放刚生成的节点。对于不同的搜索策略，节点在 OPEN 表中的排列顺序是不同的。

CLOSE 表用于存放将要扩展的节点。对一个节点的扩展是指：用所有可适用的算符对该节点进行操作，生成一组子节点

OPEN表		CLOSE表		
状态节点	父节点	编号	状态节点	父节点

搜索的一般过程

1. 把初始节点 S_0 放入 OPEN 表，并建立目前只包含 S_0 的图，记为 G ；
2. 检查 OPEN 表是否为空，假设为空那么问题无解，退出；
3. 把 OPEN 表的第一个节点取出放入 CLOSE 表，并计该节点为 n ；
4. 考察节点 n 是否为目标节点。假设是，那么求得了问题的解，退出；
5. 扩展节点 n ，生成一组子节点。把其中不是节点 n 先辈的那些子节点记做集合 M ，并把这些子节点作为节点 n 的子节点参加 G 中；
6. 针对 M 中子节点的不同情况，分别进行如下处理：
 - 1) 对于那些未曾在 G 中出现过的 M 成员设置一个指向父节点（即节点 n ）的指针，并把它们放入 OPEN 表；（不在 OPEN 表）
 - 2) 对于那些先前已经在 G 中出现过的 M 成员，确定是否需要修改它指向父节点的指针；（在 OPEN 表中）
 - 3) 对于那些先前已在 G 中出现并且已经扩展了的 M 成员，确定是否需要修改其后继节点指向父节点的指针；（在 CLOSE 表中）

7. 按某种搜索策略对 OPEN 表中的节点进行排序；
8. 转第 2 步。

一些说明

一个节点经一个算符操作后一般只生成一个子节点。但适用于一个节点的算符可能有多个，此时就会生成一组子节点。这些子节点中可能有些是当前扩展节点的父节点、祖父节点等，此时不能把这些先辈节点作为当前扩展节点的子节点。

一个新生成的节点，它可能是第一次被生成的节点，也可能是先前已作为其它节点的子节点被生成过，当前又作为另一个节点的子节点被再次生成。此时，它究竟应选择哪个节点作为父节点？一般由原始节点到该节点的代价来决定，处于代价小的路途上的那个节点就作为该节点的父节点。

在搜索过程中，一旦某个被考察的节点是目标节点就得到了一个解。该解是由从初始节点到该目标节点路径上的算符构成。

如果在搜索中一直找不到目标节点，而且 OPEN 表中不再有可供扩展的节点，那么搜索失败。

通过搜索得到的图称为搜索图，搜索图是状态空间图的一个子集。

由搜索图中的所有节点及反向指针所构成的集合是一棵树，称为搜索树。根据搜索树可给出问题的解。

盲目搜索

广度优先搜索

广度优先搜索按照“先扩展出的节点先被考察”的原则进行搜索；

根本思想

从初始节点 S_0 开始，逐层地对节点进行扩展并考察它是否为目标节点。在第 n 层的节点没有全部扩展并考察之前，不对第 $n + 1$ 层的节点进行扩展。

OPEN 表中节点总是按进入的先后顺序排列，先进入的节点排在前面，后进入的排在后面。

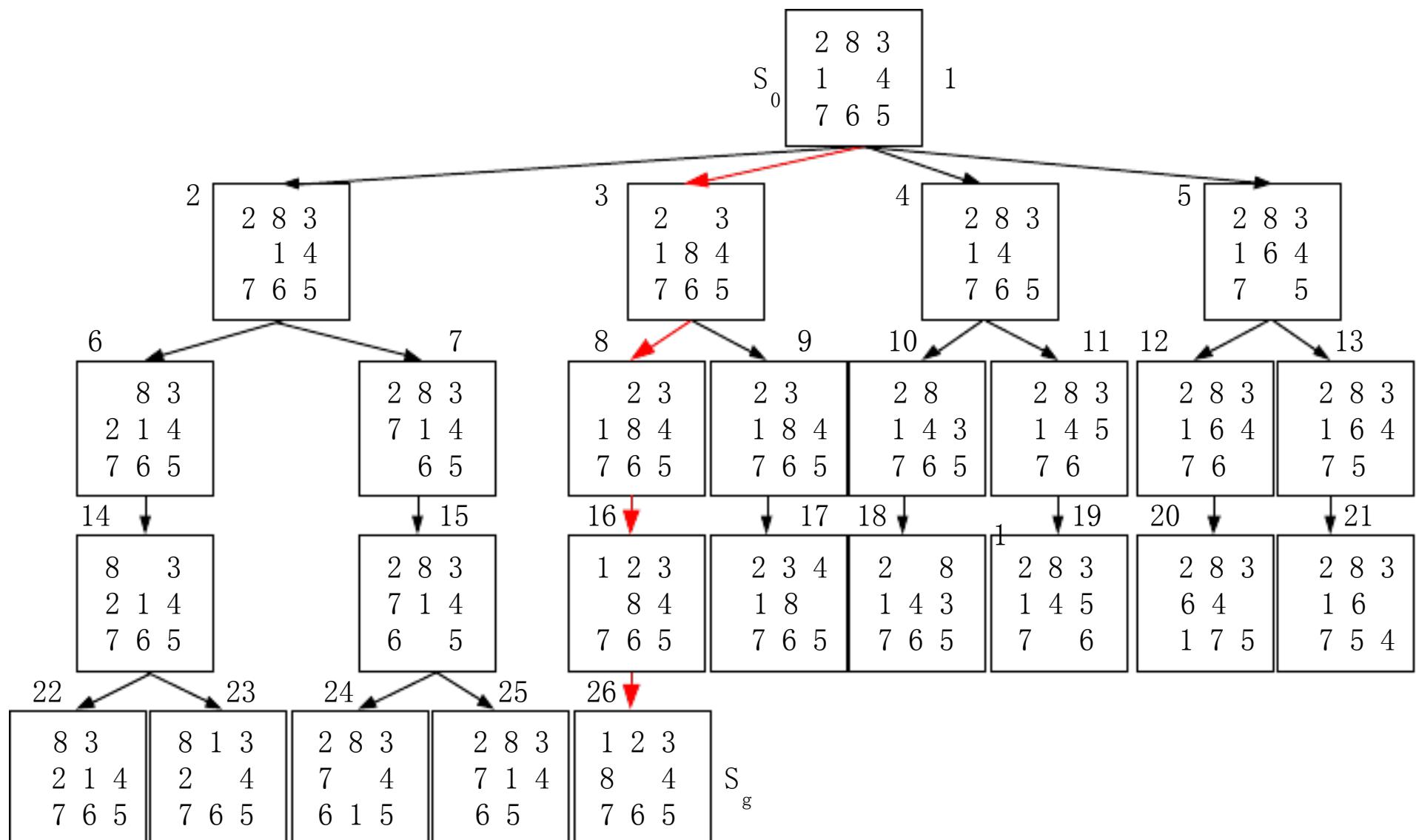
广度优先搜索过程

- 1) 把初始节点 S_0 放入 OPEN 表。
- 2) 如果 OPEN 表为空，那么问题无解，退出。
- 3) 把 OPEN 表的第一个节点（记为节点 n）取出放入 CLOSE 表。
- 4) 考察节点 n 是否为目标节点。假设是，那么求得了问题的解，退出。
- 5) 假设节点 n 不可扩展，那么转第 2 步。
- 6) 扩展节点 n，将其子节点放入 OPEN 表的尾部，并为每一个子节点都配置指向父节点的指针，然后转第 2 步。

广度优先搜索的本质是，以初始节点为根节点，在状态空间图中按照广度优先的原则，生成一棵搜索树

优点：总可以得到解，且是路径最短的解。缺点：盲目、效率低。

重排九宫的广度优先搜索



深度优先搜索——考有界深度优先搜索的可能性很大

深度优先搜索按照“后扩展出的节点先被考察”的原则进行搜索；

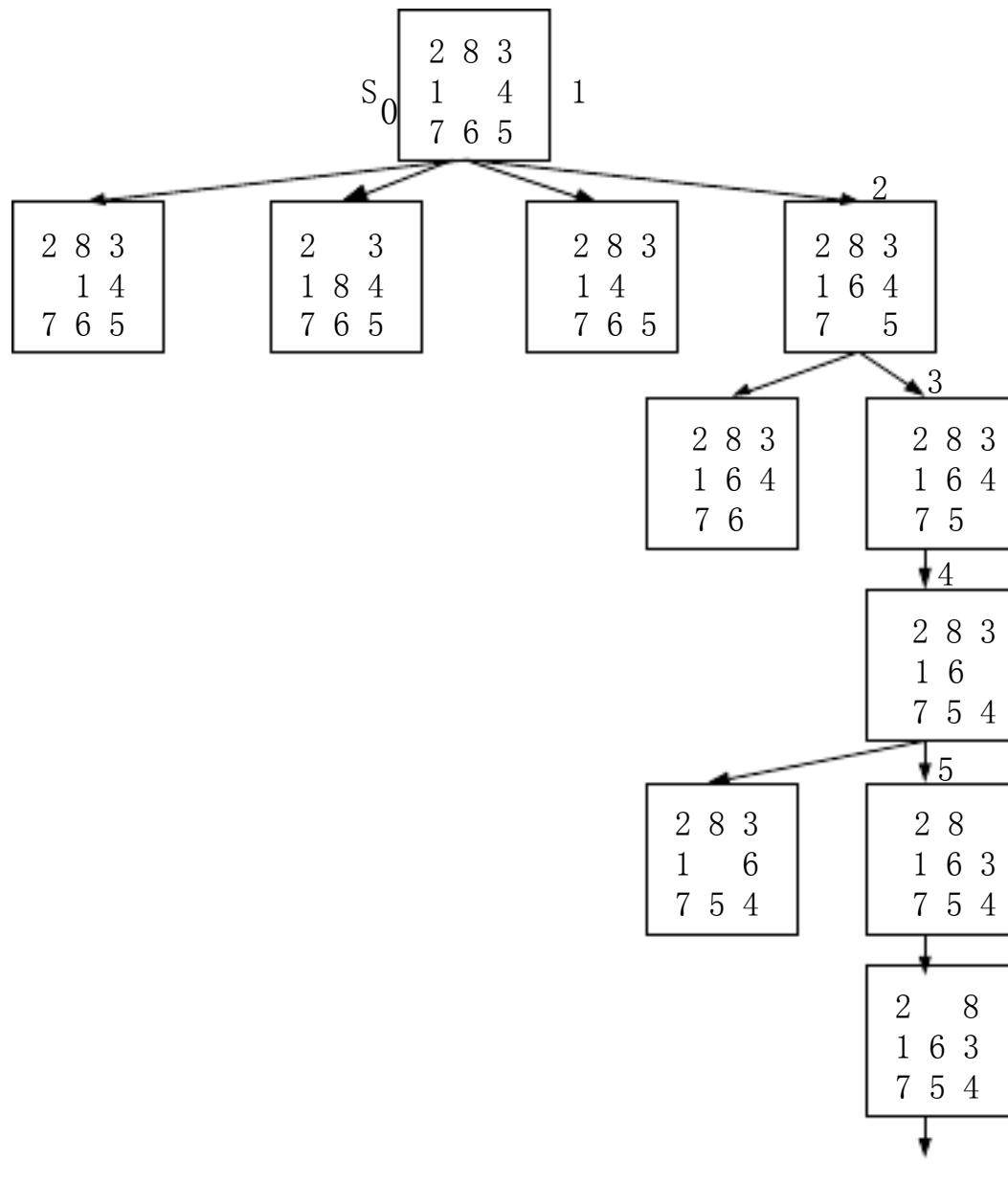
深度优先搜索与广度优先搜索的区别是：广度优先搜索是将节点 n 的子节点放入到 OPEN 表的尾部，而深度优先搜索是把节点 n 的子节点放入到 OPEN 表的首部。

深度优先搜索过程

- 1) 把初始节点 S_0 放入 OPEN 表。
- 2) 如果 OPEN 表为空，那么问题无解，退出。
- 3) 把 OPEN 表的第一个节点（记为节点 n）取出放入 CLOSE 表。
- 4) 考察节点 n 是否为目标节点。假设是，那么求得了问题的解，退出。
- 5) 假设节点 n 不可扩展，那么转第 2 步。
- 6) 扩展节点 n，将其子节点放入 OPEN 表的首部，并为每一个子节点都配置指向父节点的指针，然后转第 2 步。

深度优先搜索的本质：以初始节点为根节点，在状态空间图中按照深度优先的原则，生成一棵搜索树。

重排九宫的深度优先搜索



有界深度优先搜索：

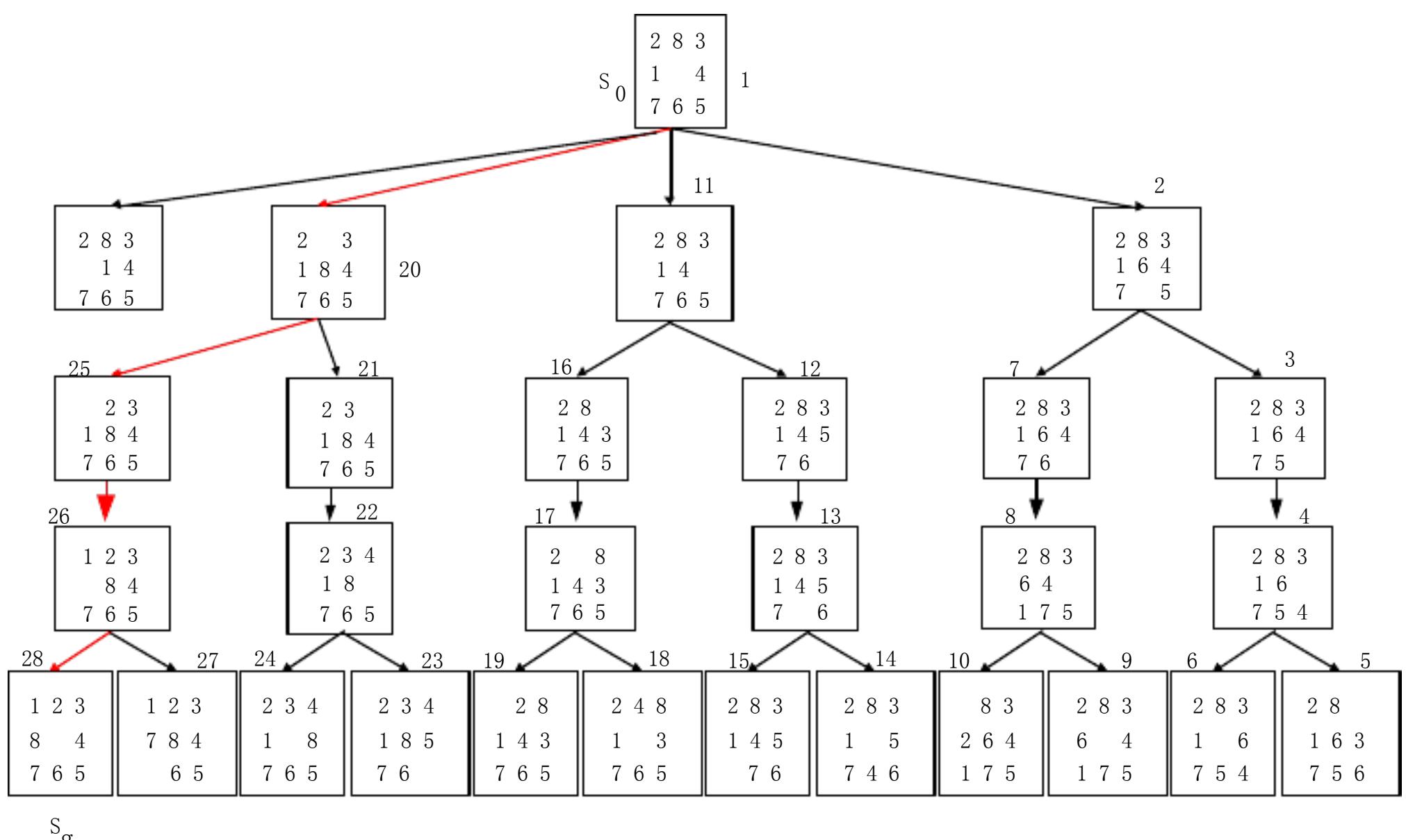
根本思想

对深度优先搜索引入搜索深度的界限 [设为 d_m]，当搜索深度到达了深度界限，而仍未出现目标节点时，就换一个分支进行搜索。

搜索过程

- 1) 把初始节点 S_0 放入 OPEN 表中，置 S_0 的深度 $d(S_0)=0$
- 2) 如果 OPEN 表为空，那么问题无解，退出。
- 3) 把 OPEN 表的第一个节点 [记为节点 n] 取出放入 CLOSE 表。
- 4) 考察节点 n 是否为目标节点。假设是，那么求得了问题的解，退出。
- 5) 假设节点 n 的深度 $d(n)=d_m$ ，那么转第 2 步 [此时节点 n 位于 CLOSE 表，但并未进行扩展]。
- 6) 假设节点 n 不可扩展，那么转第 2 步。
- 7) 扩展节点 n，将其子节点放入 OPEN 表的首部，为每一个子节点都配置指向父节点的指针，将每一个子节点的深度设置为 $d(n)+1$ ，然后转第 2 步。

重排九宫的有界深度优先搜索 [设深度界限 =]



启发式搜索

启发式搜索采用问题自身的特性信息，以指导搜索朝着最有希望的方向前进。这种搜索针对性较强，因而效率较高。

启发性信息与估价函数：

可用于指导搜索过程，且与具体问题有关的信息称为启发性信息。

用于评估节点重要性的函数称为估价函数。其一般形式为：

$$f(x) = g(x) + h(x)$$

其中 $g(x)$ 表示从初始节点 S_0 到节点 x 的代价； $h(x)$ 是从节点 x 到目标节点 S_g 的最优路径的代价的估计，它表达了问题的启发性信息。 $h(x)$ 称为启发函数。

$g(x)$ 有利于搜索的完备性，但影响搜索的效率。 $h(x)$ 有利于提高搜索的效率，但影响搜索的完备性。

全局择优搜索

全局择优搜索按照“哪个节点到目标节点的估计代价小就先考察哪个节点”的原则进行搜索；

广度优先搜索、代价树的广度优先搜索是全局择优搜索的特例。

根本思想

每当要选择下一个节点进行考察时，全局择优搜索每次总是从 OPEN 表的全体节点中选择一个估价值最小的节点。

搜索过程

- 1) 把初始节点 S_0 放入 OPEN 表，计算 $f(S_0)$ 。
- 2) 如果 OPEN 表为空，那么问题无解，退出。
- 3) 把 OPEN 表的第一个节点（记为节点 n ）取出放入 CLOSE 表。
- 4) 考察节点 n 是否为目标节点。假设是，那么求得了问题的解，退出。
- 5) 假设节点 n 不可扩展，那么转第 2 步。
- 6) 扩展节点 n ，用估价函数 $f(x)$ 计算每个子节点的估价值，并为每一个子节点都配置指向父节点的指针。把这些子

节点都送入 OPEN 表中，然后对 OPEN 表中的全部节点按估价值从小至大的顺序进行排序，然后转第 2 步。

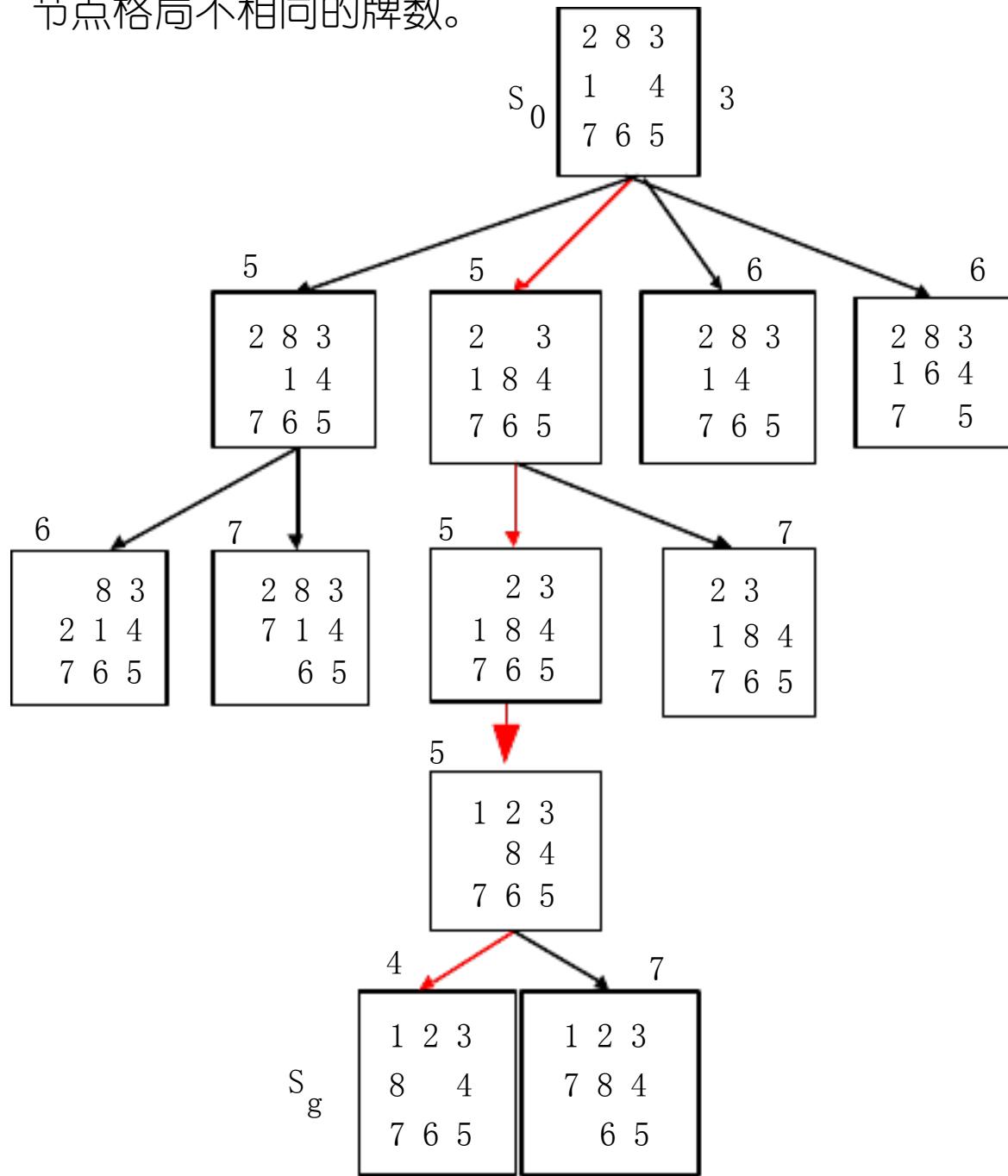
重排九宫问题的全局择优搜索树

设估价函数为：

$$f(x) = d(x) + h(x)$$

其中， $d(x)$ 表示节点 x 的深度， $h(x)$ 表示节点 x 的格局与目标

节点格局不相同的牌数。



第四章 神经网络与遗传计算

1. 神经网络 [只考概念]

*此处老师重点为：神经元的工作特性。但无奈两个课本+ppt 均找不到原话。

所以按下边的内容自己总结一下吧。

1) 生物神经元的根本工作机制

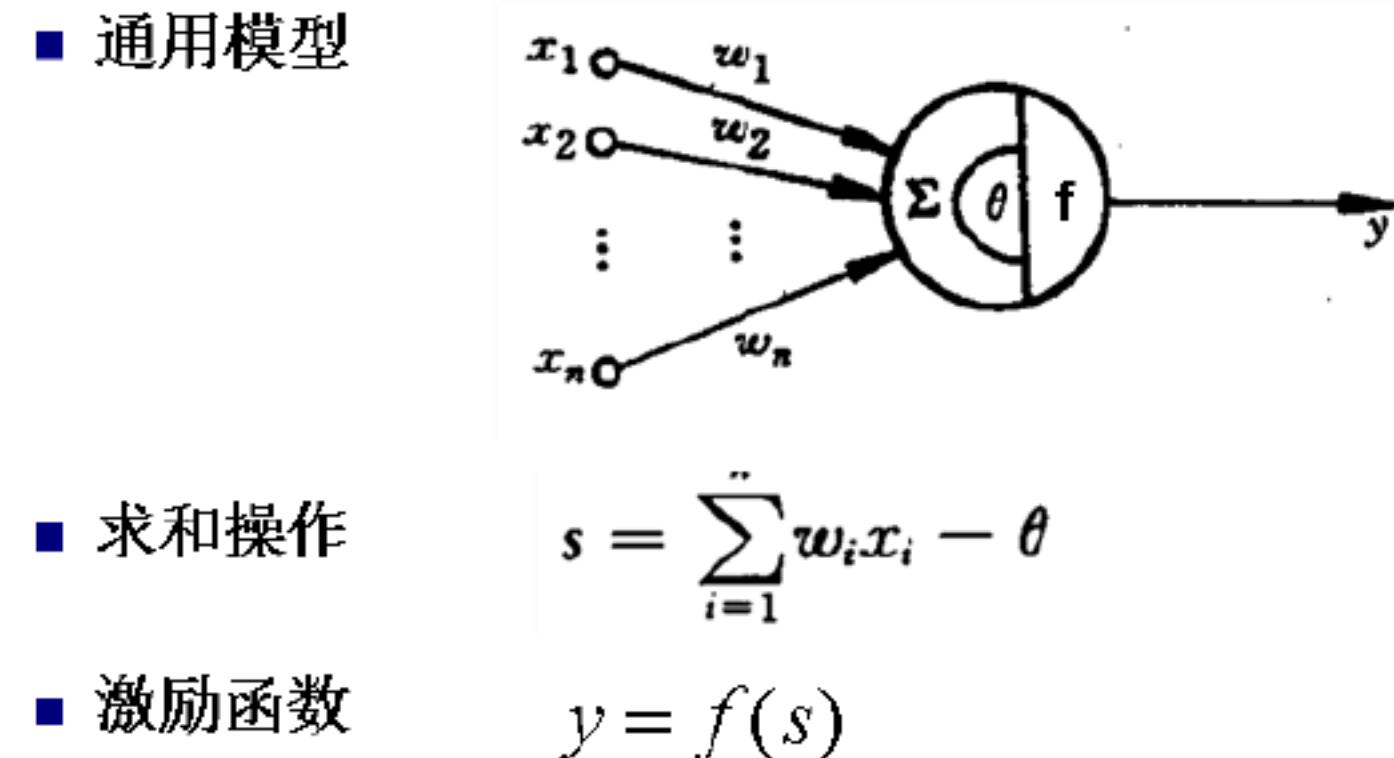
一个神经元有两种状态—兴奋和抑制。平时处于抑制状态的神经元，其树突和胞体接受其它神经元经由突触传来的兴奋电位，多个输入在神经元中以

代数和的方式叠加；如输入兴奋总量超过阈值，神经元被激发进入兴奋状态，发出输出脉冲，由轴突的突触传递给其它神经元。

2) 生物神经特性

- (1) 并行分布处理的工作模式
- (2) 神经系统的可塑性和自组织性。
- (3) 信息处理与信息存贮合二为一。
- (4) 信息处理的系统性
- (5) 能接受和处理模糊的、模拟的、随机的信息。
- (6) 求满意解而不是精确解。人类处理日常行为时，往往都不是一定要按最优或最精确的方式去求解，而是以能解决问题为原那么，即求得满意解就行了。
- (7) 系统具有鲁棒性和容错性

3) 人工神经元模型



2. 遗传计算 [此处为简答题，但老师没有给简答题的范围，所以看着重要的背]

1) 进化计算包括：

遗传算法 (genetic algorithm GA)

进化策略 (evolution strategies)

进化编程 (evolutionary programming)

遗传编程 (genetic programming)

2) 遗传操作

简单遗传算法的遗传操作主要有三种：选择(selection)、交叉(crossover)、变异(mutation)。改良的遗传算法大量扩充了遗传操作，以到达更高的效率。

选择操作也叫复制操作，根据个体的适应度函数值所度量的优、劣程度决定它在下一代是被淘汰还是被遗传。

交叉操作的简单方式是将选择出的两个个体 P1 和 P2 作为父母个体，将两者的局部码值进行交换。

变异操作的简单方式是改变数码串的某个位置上的数码。

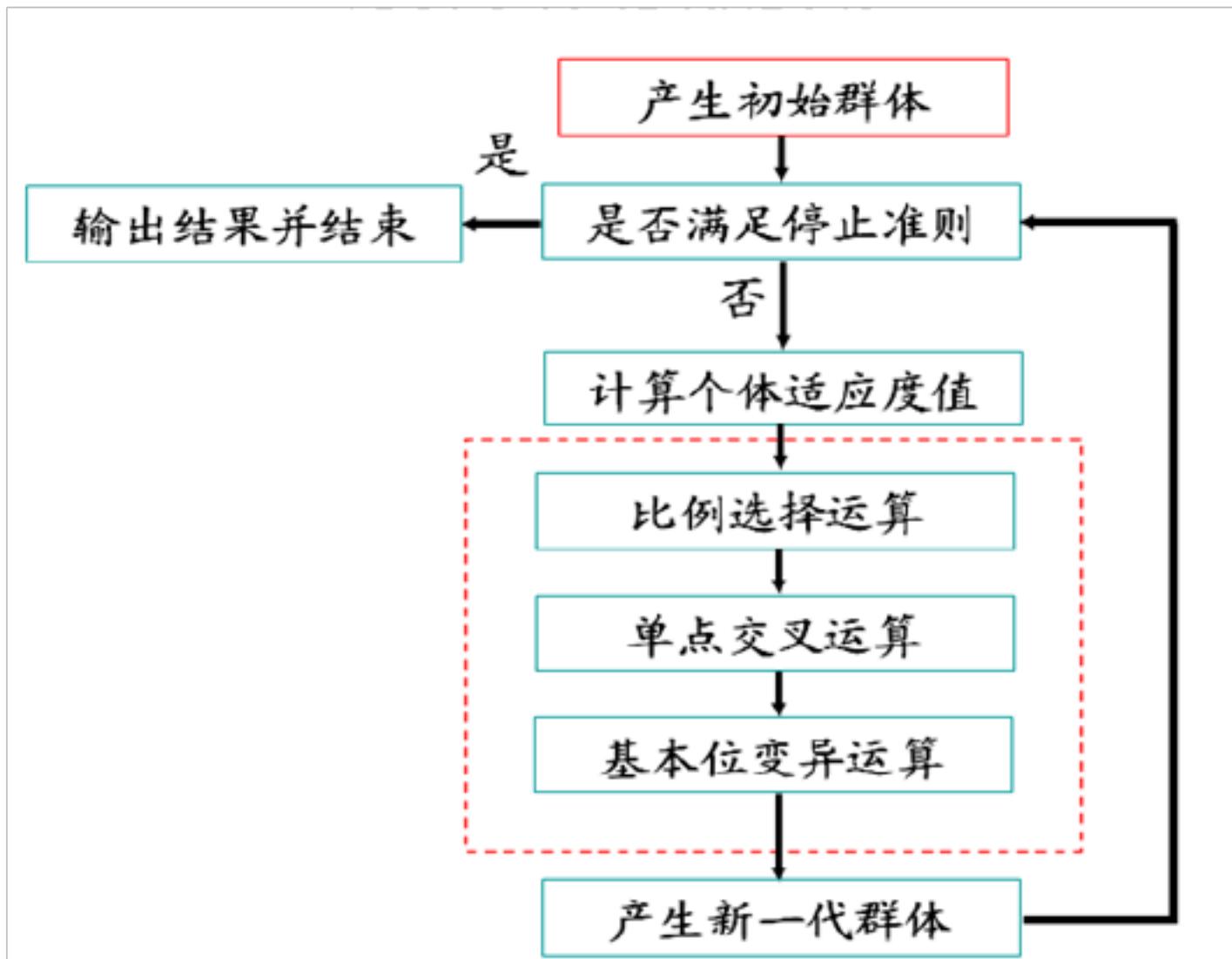
3) 遗传算法的特点

6. 遗传算法是对参数集合的编码而非针对参数本身进行变化；
7. 遗传算法是从问题解的编码组开始而非从单个解开始搜索；
8. 遗传算法利用目标函数的适应度这一信息而非利用导数或其他辅助信息来指导搜索；
9. 遗传算法利用选择、交叉、变异等算子而不是利用确定性规则那么进行随机操作。

4) 算法停止条件：

- D) 完成了预先给定的进化代数那么停止；
- E) 种群中的最优个体在连续假设干代没有改良或平均适应度在连续假设干代根本没有改良时停止。

5) 遗传算法流程



6) 遗传算法的根本机理

一般的遗传算法由四个局部组成：

编码机制、控制参数、适应度函数、遗传算子

第五章 机器学习

此处只考概念，假设 11 班学了这一章，就考；假设 11 班没学，那么不考。

C) 类比学习

类比学习是利用二个不同领域〔源域、目标域〕中的知识相似性，可以通过类比，从源域的知识〔包括相似的特征和其它性质〕推导出目标域的相应知识，从而实现学习。

所以，类比学习系统可以使一个已有的计算机应用系统转变为适应于新的领域，来完成原先没有设计的相类似的功能。

D) 解释学习

解释学习兴起于 20 世纪 80 年代中期，根据任务所在领域知识和正在学习的概念知识，对当前实例进行分析和求解，得出一个表征求解过程的因果解释树，以获取新的知识。

E) 机械学习

又称为记忆学习或死记硬背式的学习。这种学习方法直接记忆或存储环

境提供的新知识，并在以后通过对知识库的检索来直接使用这些知识，而不再需要进行任何的计算和推导。

感谢 lee 年复一年月复一月日复一日的拷 ppt，感谢 One Piece 牺牲自习室看妹子的时机回来整理重点，感谢 Armo 对重点中最为复杂的一章的总结。

1. : lee 第二章: One Piece

第三章: Armo 第四、五章: 大雪无痕

我想说的是，重点不是万能的，概念题简答题肯定有意想不到的，抛开课本背重点是犯傻的，两手都要抓两手都要硬才是无敌的。

来自#10 I-226