人工智能的不同研究流派：符号主义/逻辑主义学派--符号智能；连接主义--计算智能；行为主义-低级智能。

人工智能的主要**研究领域**

（一）自动推理（二）专家系统（三）机器学习（四）自然语言理解（五）机器人学和智能控制（六）模式识别（七）基于模型的诊断

产生式系统是人工智能系统中常用的一种程序结构，是一种知识表示系统。

三部分组成：综合数据库:**存放问题的状态描述的数据结构，动态变化的**。产生式规则集、控制系统。

/ 产生式规则集/ 控制系统

产生式规则形式： IF 前提条件 THEN 操作

八数码难题的产生式系统表示

综合数据库：以状态为节点的有向图。

状态描述： 3×3 矩阵

产生式规则：

IF<空格不在最左边 >Then<左移空格 >；依次

控制系统：

选择规则:按左、上、右、下的顺序移动空格。

终止条件：匹配成功。

产生式系统的基本过程：

Procedure PROCUCTION

1. DATA←初始状态描述
2. until DATA 满足终止条件，do:
3. begin
4. 在规则集合中，选出一条可用于 DATA 的规则 R（**步骤 4 是不确定的，只要求选出一条可用的规则 R，至于这条规则如何选取，却没有具体说明。**）
5. DATA←把 R 应用于 DATA 所得的结果
6. End

产生式系统的特点：1.模块性强，2.产生式规则相互独立，3.规则的形式与逻辑推理相近，易懂。

产生式系统的控制策略：1.不可撤回的控制策略：优点是空间复杂度小、速度快；缺点是多数情况找不到解 2.试探性控制策略：回溯方式：占用空间小，多数情况下能找到解；缺点是如果深度限制太低就找不到解；和图搜索方式：优点总能找到解，缺点时间空间复杂度高。

产生式系统工作方式：正向、反向和双向产生式系统

可交换产生式系统：1.可应用性，每一条对 D 可应用的规则，对于对 D 应用一条可应用的规则后，所产生的状态描述仍是可应用的。2.可满足性，如果 D 满足目标条件，则对 D 应用任何一条可应用的规则所产生的状态描述也满足目标条件。3.无次序性，对 D 应用一个由可应用于 D 的规则所构成的规则序列所产生的状态描述不因序列的次序不同而改变。

可分解的产生式系统：能够把产生式系统综合数据库的状态描述分解为若干组成部分，产生式规则可以分别用在各组成部分上，并且整个系统的终止条件可以用在各组成部分的终止条件表示出来的产生式系统，称为可分解的产生式系统。**基本过程**:

Procedure SPLIT

1.DATA ← 初始状态描述
2.{Di} ← DATA 的分解结果；每个 Di 看成是独立的状态描述
3.until 对所有的 Di {Di}， Di 都满足终止条件，do:
4.begin
5. 在{Di}中选择一个不满足终止条件的 D*
6. 从{Di}中删除 D*
7.从规则集合中选出一个可应用于 D*的规则 R
8.D ← 把 R 应用于 D*的结果
9.{di} ← D 的分解结果
10.把{di}加入{Di}中
11.end

回溯算法 BACKTRACK 过程：Recursive

Procedure BACKTRACK(DATA)

1.if TERM(DATA),return NIL;
2.if DEADEND(DATA),return FAIL;
3.RULES←APPRULES(DATA);
4.LOOP:if NULL(RULES),return FAIL;
5.R←FIRST(RULES);
6.RULES←TAIL(RULES);
7.RDATA←R (DATA) ;
8.PATH←BACKTRACK(RDATA);
9. if PATH=FAIL,go PATH;
10.return CONS(R,PATH).

Procedure GRAPHSEARCH

1. G←{s}， OPEN ← (s) .
2. CLOSED ←NIL.
3. LOOP: IF OPEN=NIL,THEN FAIL.
4. n ← FIRST(OPEN)，OPEN ← TAIL(OPEN),CONS(n, CLOSED) .
5. IF TERM(n)，THEN 成功结束

(解路径可通过追溯 G 中从 n 到 s 的指针获得）。

6. 扩展节点 n，

令 M={m | m 是 n 的子节点，且 m 不是 n 的祖先} ， G ←G ⋃M

7. (设置指针，调整指针）对于 m M，

(1)若 m CLOSED，m OPEN，建立 m 到 n 的指针，并 CONS(m，OPEN).

(2)(a)m OPEN，考虑是否修改 m 的指针.

(b)m CLOSED,考虑是否修改 m 及在 G 中后裔的指针。

8. 重排 OPEN 表中的节点（按某一任意确定的方式或者根据探索信息）。

9. GO LOOP

无信息的图搜索过程：深度优先搜索：排列 OPEN 表中的节点时按它们在搜索树中的深度递减排序 。深度最大的节点放在表的前面，

深度相等的节点以任意方式排序。**宽度优先搜索**：在排列 OPEN 表中节点时按它们在搜索图中的深度递增顺序，深度最小的节点放在表的前面。

A 算法：使用估价函数 f(n)=g(n)+h(n)排列 OPEN 表中节点顺序的 GRAPHSEARCH 算法。

其中，g(n)：对 g*(n)的一个估计 是当前的搜索图 G 中 s 到 n 的最优路径费用 g(n)⩾g*(n)

h(n)：对 h*(n)的估计，称为启发函数。(Note:若 h(n)=0，g(n)=d，则 f(n)=d，为宽度优先)。

A*算法:对任何节点 n 都有 h(n)⩽h*(n)的 A 算法。

定义: **如果一个搜索算法对于任何具有解路径的图都能找到一条最佳路径，则称此算法为可采纳的。**

可以证明：A*算法是可采纳的（如果解路径存在，A*一定由于找到最佳解路径而结束）

A*算法的可采纳性：定理 1 GRAPHSEARCH 对有限图必然终止。定理 2 若存在 s 到目标的路，则算法 A*终止前的任何时刻，OPEN 表中总存在一个节点 n’，n’在从 s 到目标的最佳路径上，且满足 f(n’)⩽f*(s)

定理 3 若存在从 s 到目标的解路，则算法 A*必终止。

定理 4 算法 A*是可采纳的（即如果解路径存在，A*一定找到最佳解路径而终止）.

定理 5 算法 A*选择的任意扩展点都有 f(n)⩽f*(s)

可采纳的条件：1.与或图有解图，2.对图中所有节点 n 有 h(n)⩽h*(n)，3.启发函数满足单调性。则 AO*必然终止并找出最佳解路径。

影响算法 A 启发能力的三个重要因素：

(1) 算法 A 所找到的解路径的费用。

(2) 算法 A 在寻找这条解路径的过程中所需要 扩展的节点数。

(3) 计算启发函数所需要的计算量。

启发能力的度量：渗透度 P = L / T 其中，L 是算法发现的解路径的长度，

T 是算法在寻找这条解路径期间所产生的节点数（不包括初始节点，包括目标节点）

有效分枝系数是 B，则有 $B+B_2+\cdots+B_L=T$ 或 $B(B_L-1)/(B-1)=T$

8 数码启发函数 h(n)=P(n)+3S(n),p(n)是每个硬纸片离开目标位置的和，S(n)是如果一个硬纸片后面的纸片不是它的目标后继则记 2，否则记 0，如果中心有硬纸片记 1，否则记 0，然后求和。

渗透度，搜索算法的性能的度量：P = L / T，L 是算法发现的解路径的长度，T 是算法在寻找这条解路径期间所产生的节点数（不包括初始节点，包括目标节点）。

有效分枝数 B，反映目标搜索的集中程度：设搜索树的深度是 L，算法所产生的总节点数为 T，则 $B+B2+\cdots+BL=T$ 或 $B(BL-1)/(B-1)=T$

与/或图是一种超图．在超图中父亲节点和一组后继节点用超弧连接． 超弧又叫 k-连接符．

k-连接符：一个父节点指向一组 k 个有与关系的后继节点，这样一组弧线称为一个 k-连接符．

极小极大原则：MAX 节点在其 MIN 子节点的倒推值中选 max；MIN 节点在其 MAX 子节点的倒推值中选 min

剪枝规则：（1）α 剪枝：如果一个 MIN 节点的 β 值小于或等于它的某一个 MAX 祖先节点的 α 值，则剪枝发生在该 MIN 节点之下：中止这个 MIN 节点以下的搜索过程。这个 MIN 节点最终的倒推值就确定为这个 β 值。

（2）β 剪枝：如果一个 MAX 节点的 α 值大于或者等于它的某一个 MIN 祖先节点的 β 值，则剪枝发生在该 MAX 节点之下．中止这个 MAX 节点以下的搜索过程。该 MAX 节点的最终返回值可以置成它的 α 值.

ND=2(B^D/2)-1 (D 为偶数)

ND=B^(D+1)/2+B^(D-1)/2-1 (D 为奇数)

D 为深度，B 为平均后继。

定理 1 任意公式 G 都等价于一个前束范式.

证明 通过如下四个步骤即可将公式 G 化为前束范式.

步骤 1：使用基本等价式

F H=(F→H) ∧(H→F) F→H=¬F∨H

可将公式 G 中的 和→删去。

步骤 2：使用¬(¬F)=F 和 De. Morgan 律及引理 1，

可将公式中所有否定号¬放在原子之前。

步骤 3：如果必要的话，则将约束变量改名.

步骤 4：使用引理 1 和引理 2 又将所有量词都提到公式的最左边。

G= x y z u v wP(x,y,z,u,v,w) 则用 a 代替 x，用 f(y，z)代替 u，用 g(y，z，v)代替 w，得公式 G 的 Skolem 范式：

y z vP(a,y,z,f(y,z),v,g(y,z,v))

定理 2 设 S 是公式 G 的子句集．于是，G 是不可满足的，当且仅当 S 是不可满足的.

医生骗子问题：S＝ {P(a), ¬D(y) L(a,y), P(x) ¬Q(y) ¬L(x,y),D(b),Q(b)}

引理 1 设 G 是仅含有自由变量 x 的公式，记以 G（x），H 是不含变量 x 的公式，于是有

(1) x(G(x)∨H)= xG(x)∨H

(1)’ x(G(x)∨H)= xG(x)∨H

(2) x(G(x)∧H)= xG(x) ∧H

(2)’ x(G(x) ∧H)= xG(x) ∧H

(3) ¬(xG(x))= x (¬G(x))

(4) ¬(xG(x))= x (¬G(x))

引理 2 设 H, G 是两个仅含有自由变量 x 的公式, 分别记以 H(x), G(x), 于是有:

(1) $\forall x G(x) \wedge \forall x H(x) = \forall x (G(x) \wedge H(x))$

(2) $\forall x G(x) \vee \forall x H(x) = \forall x (G(x) \vee H(x))$

(3) $\forall x G(x) \vee \forall x H(x) = \forall x \forall y (G(x) \vee H(y))$

(4) $\forall x G(x) \wedge \forall x H(x) = \forall x \forall y (G(x) \wedge H(y))$

基本等价式

1) $(G \to H) = (\neg G \vee H) \wedge (H \to G)$;

2) $(G \leftrightarrow H) = (\neg G \vee H)$;

3) $G \vee G = G$, $G \wedge G = G$; (等幂律)

4) $G \vee H = H \vee G$, $G \wedge H = H \wedge G$; (交换律)

5) $G \vee (H \vee S) = (G \vee H) \vee S$,
$G \wedge (H \wedge S) = (G \wedge H) \wedge S$; (结合律)

6) $G \vee (G \wedge H) = G$, $G \wedge (G \vee H) = G$; (吸收律)

7) $G \vee (H \wedge S) = (G \vee H) \wedge (G \vee S)$,
$G \wedge (H \vee S) = (G \wedge H) \vee (G \wedge S)$; (分配律)

8) $G \vee F = G$, $G \wedge T = G$; (同一律)

9) $G \vee F = F$, $G \wedge T = T$; (零一律)

10) $\neg(G \vee H) = \neg G \wedge \neg H$,
$\neg(G \wedge H) = \neg G \vee \neg H$。 (De Morgan 律)

11) $G \vee \neg G = T$; $G \wedge \neg G = F$ (互补律)

12) $\neg\neg G = G$ (双重否定律)

将公式 $\forall x \forall y (A(x) \to B(x,y)) \to (\exists y C(y) \to \exists z D(z))$ 化为前束范式

解: (1) 消去 联结词。

$\forall x \forall y (A(x) \to B(x,y)) \to (\exists y C(y) \to \exists z D(z))$

$= \neg\forall x \forall y (\neg A(x) \vee B(x,y)) \vee (\neg \exists y C(y) \vee \exists z D(z))$

(2) 将公式中所有否定号 ¬ 放在原子之前。

$\forall x \forall y (\neg A(x) \vee B(x,y)) \vee (\neg \exists y C(y) \vee \exists z D(z))$

$= \forall x \forall y(A(x) \wedge \neg B(x,y)) \vee (\forall y \neg C(y) \vee \exists z D(z))$

(3) 将约束变量改名.

$\forall x \forall y(A(x) \wedge \neg B(x,y)) \vee (\forall y \neg C(y) \vee \exists z D(z))$

$= \forall x \forall y(A(x) \wedge \neg B(x,y)) \vee (\forall t \neg C(t) \vee \exists z D(z))$

(4) 将量词提到整个公式前。

$\forall x \forall y(A(x) \wedge \neg B(x,y)) \vee (\forall t \neg C(t) \vee \exists z D(z))$

$= \forall x \forall y \forall t \exists z ((A(x) \wedge \neg B(x,y)) \vee \neg C(t) \vee D(z))$

$= \forall x \forall y \forall t \exists z((A(x) \wedge \neg C(t) \vee D(z)) \vee (\neg B(x,y) \vee \neg C(t) \vee D(z)))$

用 a 代替 x, 用 b 代替 y, 用 f(t)代替 z, 得公式的 Skolem 范式:

$\forall t((A(a) \wedge \neg C(t) \vee D(f(t))) \vee (\neg B(a,b) \vee \neg C(t) \vee D(f(t))))$

例: 1) $G = \forall x(P(f(x)) \vee Q(x, f(a)))$

2) $H = \forall x(P(x) \wedge Q(x, a))$

设解释 I: D={2, 3},

a	f(2)	f(3)
2	3	2

P(2)	P(3)	Q(2, 2)	Q(2, 3)	Q(3, 2)	Q(3, 3)
F	T	T	T	F	T

$T_I(G) = T_I((P(f(2)) \vee Q(2, f(2))) \wedge (P(f(3)) \vee Q(3, f(2))))$

$= T_I((P(3) \vee Q(2, 3)) \wedge (P(2) \vee Q(3, 3)))$

$=(T \vee T) \wedge (F \vee T)$

$=T$

$T_I(H) = T_I(P(2) \wedge Q(2, 2) \wedge P(3) \vee Q(3, 2))$

$=F \wedge T \vee T \wedge F$

$=F$

(Herbrand 域) 设 S 为子句集, 令 H0 是出现于子句集 S 的常量符号集。如果 S 中无常量符号出现, 则 H0 由一个常量符号 a 组成。 对于 i＝1, 2, …, 令 $H_i = H_{i-1}$ {所有形如 f(t1, …, tn)的项} 其中 f(t1, …, tn)是出现在 S 中的所有 n 元函数符号, $t_j \in H_{i-1}$, j＝1, …, n. H_i 为 S 的 i 级常量集, H 称为 S 的 Herbrand 域, 简称 S 的 H 域。

{P(x), Q (f(y)∨R(y)) }, 于是 S 的 H 域 {a,f(a),f(f(a))…}原子集

{P(a),Q(a),R(a),P(f(a))..} 子句的一个基例

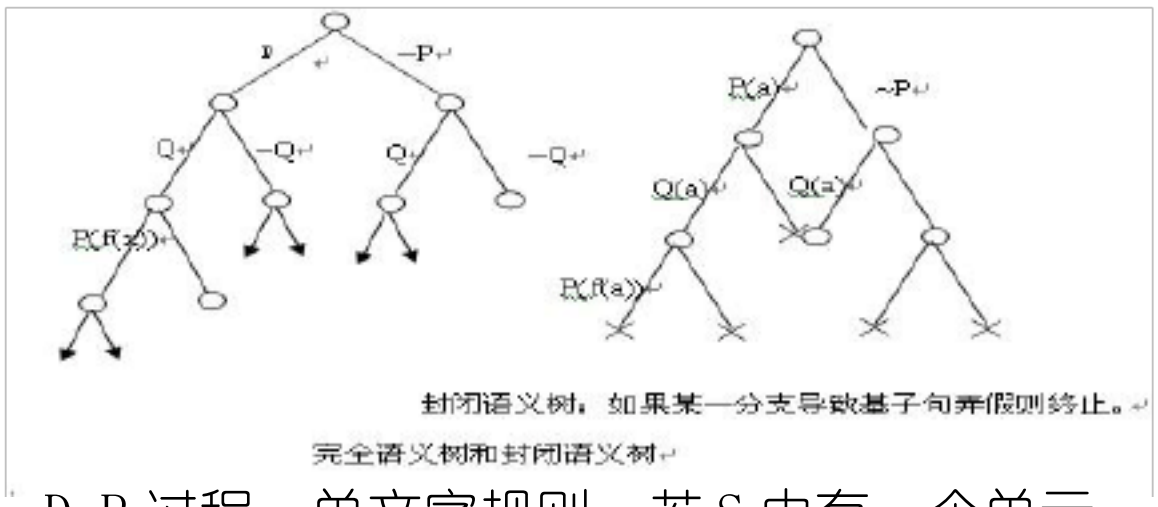
={Q(f(a))∨R(a),Q(f(f(a)))∨R(F(a))..};

该 S 的 H 解释与原子集相同。

H 解释与普通解释的关系: 1、子句集 S 的 H 解释是 S 的普通解释。2、S 的普通解释不一定是 S 的 H 解释: 普通解释不是必须定义在 H 域上, 即使定义在 H 域上, 也不一定是一个 H 解释。3、任取普通解释 I, 依照 I, 可以按如下方法构造 S 的一个 H 解释 I*, 使得若 S 在 I 下为真则 S 在 I*下也为真。

定理: 如果某区域 D 上的解释 I 满足子句集 S, 则对应于 I 的任意一个 H 解释 I*也满足 S。

语义树: S={¬P(x)∨Q(x),P(f(x)), ¬Q(f(x))}分别画出 S 的完全语义树与 封闭语义树。



封闭语义树：如果某一分支导致基子句弄假则终止。
完全语义树和封闭语义树

D-P 过程：单文字规则：若 S 中有一个单元基子句 L，令 S’为删除 S 中包含 L 的所有基子句所剩子句集，则：1）若 S’为空集，则 S 可满足。(2) 否则，令 S’’为删除 S’中所有文字¬L 所得子句集（若 S’中有单元基子句¬L，则删文字¬L 得空子句），于是，S 恒假 iff S’’恒假。定义（纯文字）：称 S 的基子句中文字 L 是纯的，如果¬L 不出现在 S 中。纯文字规则设 L 是 S 中纯文字，且 S’为删除 S 中所有包含 L 的基子句所剩子句集，则(1)若 S’为空集，则 S 可满足。(2) 否则，S 恒假 iff S’恒假。

分裂规则若 S=(A1 L) … (Am L) (B1 ¬L) … (Bn ¬L) R 其中 A i , Bi ,R 都不含 L 或¬L，令 S1 =A1 … Am R，S2= B1 … Bn R 则 S 恒假 iff S1 , S2 同时恒假。

归结式 对任意两个基子句 C1 和 C2。如果 C1 中存在文字 L1，C2 中存在文字 L2，且 L1 = ¬L2，则从 C1 和 C2 中分别删除 L1 和 L2，将 C1 和 C2 的剩余部分析取起来构成的子句，称为 C1 和 C2 的归结式，记为 R(C1, C2)。

（归结演绎） 设 S 是子句集。从 S 推出子句 C 的一个归结演绎是如下一个有限子句序列：

C1，C2，…，Ck 其中 Ci 或者是 S 中子句，或者是 Cj 和 Cr 的归结式（j i，r i）；并且 Ck＝C。定理：如果基子句集 S 是不可满足的，则存在从 S 推出空子句的归结演绎。

归结式简化:若 S={P∨C1’，…，P∨Ci’，¬P∨Ci+1’，…，¬P∨Cj’，Cj+1，…，Cn}则

S’={Ci+1’，…，Cj’，Cj+1，…，Cn}

S’’={C1’，…，Ci’，Cj+1，…，Cn}

合一算法：定义(替换)一个替换是形如{t1/v1，…，tn/vn }的一个有限集合，其中 vi 是变量符号，ti 是不同于 vi 的项。

合一算法步骤：W={Q(f(a)，g(x))，Q(y，y)}，求 W 的 mgu。

步骤 1： k=0，W0=W， 0= 。

步骤 2： D0 ={f(a)，y}。

步骤 3：有 v0= y D0，v0 不出现在 t0＝f(a)中。

步骤 4：令 1= 0 {t0/v0}={f(a)/y}，W1={Q(f(a)，g(x))，Q(f(a)，f(a))}

步骤 2： D1 ={g(x)，f(a) }。

步骤 3：D1 中无变量符号，算法停止，W 不可合一。

归结定理的几种改进：支架集归结：子句集 S 的子集 T 称为 S 的支架集，如果（S-T）是可满足的。一个支架集归结是一个不同时属于（S-T）的两个子句的归结。

语义归结：(1) ¬P∨¬Q∨R(2)P∨R(3)Q∨R(4) ¬R 令 I={¬P，¬Q，¬R}，P Q R。于是在 I 下为假的子句只有两个：Si ={P∨R，Q∨R}我们可得如下

PI-演绎：(5)R 由(2)、(3)、(1)

(6) 由(5)、(4)

（线性归结）设 S 是一个子句集，C0 是 S 中的一个子句。以 C0 为顶子句，从 S 到 Cn 的线性归结演绎是如下一个演绎：对于 i=0，1，…，n-1，Ci+1 是 Ci 和 Bi 的归结式。每个 Bi 或者属于 S，或者是一个 Cj，其中 j i。

输入归结：边子句都属于 S 的线性归结演绎称为输入归结演绎.

（锁归结式） 将子句中的每个文字配上一个整数。最小锁原则（归结最小锁的子句），合并规则（保留最小锁原则）

基于规则的产生式系统：1、如果子表达式 E1,…,Ek 的父亲节点是（E1∨…∨Ek），则用 k-连接符（带弧）把这些子节点与父节点连接起来，否则用 1-连接符（不带弧）。

（替换的相容性集合、合一复合替换）设有替换集合{σ1 ，σ2 ，……，σn}，每一σi 具有如下形式：σi ={ti1/vi1 ，……，tim(i)/vim(i)}其中 ti1，……，tim(i) 是项，vi1，……，vim(i)是变量；我们用这些替换构造两个表达式 U1 和 U2 如下：

U1 ={v11，……，v1m(1)，……，vn1，……，vnm(n) }

U2 ={t11，……，t1m(1)，……，tn1，……，tnm(n)) }

如果 U1 和 U2 是可合一的，则替换集合{σ1 ，σ2 ，……，σn}称作是相容的，否则称作是不相容的，U1 和 U2 的最一般合一替换也叫做{σ1 ，σ2 ，……，σn}的合一复合替换。

事实:P(x)∨Q(x)

规则:P(A)→R(A) Q(B)→R(B)

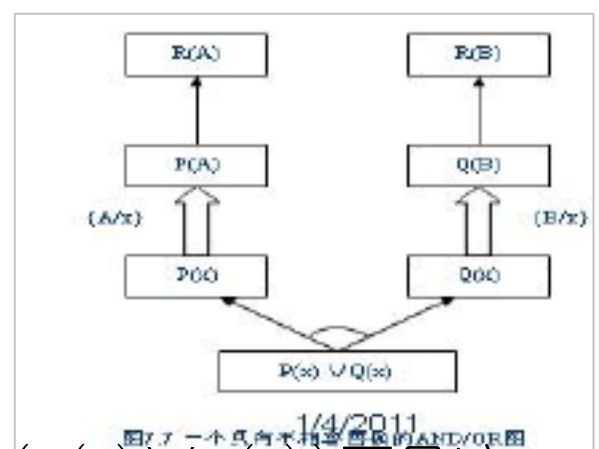


图 7 一个负向不相容 AND/OR 图

(R(A)∨R(B))不是该 AND/OR 图的一个子句表示。