谓词演算：用于人工智能的一种表示语言

优点：明确定义的形式化语义以及可靠和完备的推理规则

局限性：无法描述客观事物及逻辑特征，难以表述共同的特点

合取：conjunction，析取：disjunction，蕴涵：implication，

等价：equivalence，合法的语句：合式公式（Well-formed formula）

常用公式：$P \vee Q \equiv (\neg P \rightarrow Q)$，$P \rightarrow Q \equiv \neg P \vee Q$，$\{\wedge, \vee\}$均可交换，可结合，可分配。

$P \leftrightarrow Q \equiv (\neg P \vee Q) \wedge (\neg Q \vee P)$

元数(arity)：函数的参数个数

friend(friend(X), X) 也是原子语句

plus(two, three) 是函数，而不是原子语句（无法判定真假）

哑元(dummy)：全称量词，可以被替换为任何东西的变量。

常用公式：

$\neg \exists X p(X) \equiv \forall X \neg p(X)$，$\neg \forall X p(X) \equiv \exists X \neg p(X)$

$\exists X p(X) \equiv \exists Y p(Y)$，$\forall X q(X) \equiv \forall Y q(Y)$ $\Rightarrow$ 哑元

$\forall X (p(X) \wedge q(X)) \equiv \forall X p(X) \wedge \forall Y p(Y)$

$\exists X (p(X) \vee q(X)) \equiv \exists X p(X) \vee \exists Y q(Y)$，

若~~一条语句~~一种解释使一条语句为真，那么就说它满足这条语句

若一个解释满足表达式集S中所有成员~~语句~~，~~解释~~，那么它满足这个集合

如果满足谓词表达式集合S的$\boxed{\text{所有解释}}$，也满足表达式X，那么X逻辑跟随S

若从表达式集S推理产生的表达式也跟随自S，说明这个推理是可靠的。

若一个推理规则可以推出S的所有表达式，这个规则是完备的。

合一：unify，是判断什么样的替换可以使产生的两个谓词表达式匹配的算法

要求所有变量是全称量化的，用常量/函数替换存在量化的变量（斯柯伦化）

（ground instance）

（前者/后者）常量不可被替代，两个不同的常量也不能被一个变量替代

变量不能与含有这个变量的项合一

复合：θδ入是先将入的替换对θ应用，再删去原有的项。

{ 符合结合率
不符合交换率

合一过程：(E1,E2)

1.若两个元素是常量且相同，则不替换，若不同则失败

2.若两个元素有一个是变量，将变量换成常量，但两个元素不能相互包含，包含的话就失败。

3.若E1,E2均为列表，先对表头合一，将得到的替换式应用于表尾，再对表尾进行合一。

搜索策略：

{ 数据驱动：用已有的条件生成更多条件．{ 问题初始时给出所有或大部分数据 / 且目标数量过多 / 难以组成目标或假设

目标驱动：用目标找需要的事实（DFS）{ 易于形式化，可利用目标推理来确认及排除 / 有很多底层的结论与事实相符（易剪枝）/ 数据未给定，要在搜索时求解．

（DFS）

对图搜索：CS：当前节点，SL：从根结点到CS的路径（简单路径）（栈）（State list）

NSL：待访问结点，将与CS同一层的结点加入，~~访问过后删去~~，从前面

DE：不会再访问的结点（从NSL中删除的）

（BFS）= open：表的第一个元素是当前结点，结束后将儿子加入到表末（队列）

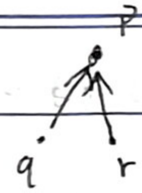
closed：结束的元素。

（DFS）= open：表的表头是当前结点，结束后将儿子加入到表头（stack）

closed：结束的元素，

迭代加深的DFS：先对空间进行深度为1的DFS，找不到就再进行深度为2的DFS…．

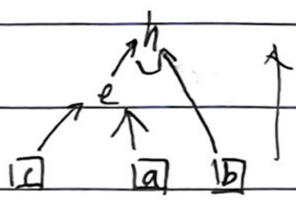
与或图:



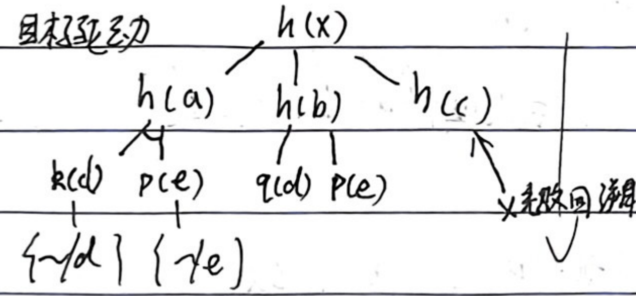
$(q \vee r \rightarrow p)$

$(q \wedge r \rightarrow p)$.

数据驱动:



目标驱动



采用启发的情况:

(1) 在问题现有数据和问题陈述存在模糊性，所以问题可能没有精确解.

(2) 可能有精确解，但代价太大.

启发式搜索 = 启发度量 + 进行空间搜索的算法.

爬山法: 扩展当前状态后，选一个最优的儿子扩展 (greddy)，易陷入局部最优

在AI中，最小 edit distance 里中替换的代价是2，不是1。

Best-First-Search: 将 queue 替换为 priority queue，将最佳的状态放在前面.

评估函数 $f(n) = g(n) + h(n)$

从任意状态n到起始状态的实际路径长度，状态n到目标距离的启发性估计.

专家系统的搜索可以对每条规则设置一个置信度.

若一个算法满足 $h(n) \le h^*(n)$，那么将它称为 A^* 算法.

若无启发 硬搜 $h(n)=0$，

是从n到目标的最短实际代价

n_j是n_i的后继.

启发函数 h单调的条件是，$h(n_i)-h(n_j)\le cost(n_i,n_j)$ ← 实际花费

若单调，第一次发现的路径一定最短。 P103

在算法是A*的前提下，若有 $h_1(n)\le h_2(n)$，则 h_2 比 h_1 有更高的信息度(发现最短路的能力)

博弈：在可穷举图上的极大极小过程，MAX层(式)与MIN层(对手)(与)交替。给MAX层的父状态赋与儿子的最大值，反之~

可用□表示最大点，○表示最小点.

井字棋：tic-tac-toe

α-β剪枝，α为最大值临时值(不减小)，β反之(不增大)，当β<α就可剪掉.

产生式系统：

组成：①产生式规则集合：条件-动作对.

② 工作内存：对当前状态的描述（已经得到的所有条件）

③ 识别-动作循环：将得到的模式与产生式的规则相比较，条件与工作内存中的匹配则是冲突集合，从冲突集合中选择一条激发。

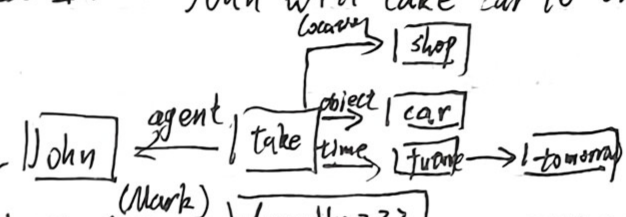
规则形式：$\forall X,Y\,[path(X,Y)\leftarrow \exists Z[move(X,Z)\wedge path(Z,Y)]]$

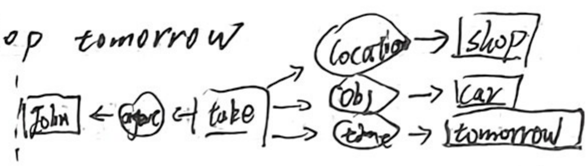
$p\wedge q\to goal$，从goal出发是目标驱动

$start\to v\wedge r\wedge q$ 从start出发是数据驱动.

概念图：John will take car to shop tomorrow

网络：





概念图：John 可以写为 person:john 表达类型.

限定(restrict) 唯一标识(Mark)：dog:#2333，不指定个体：dog:*

Agent
Object
Location
time
recipient

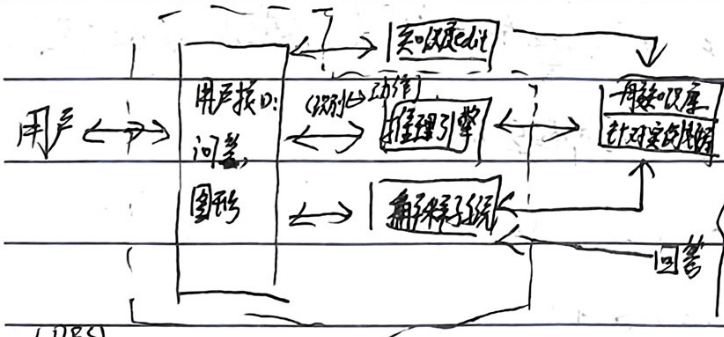
类型层次：T：通用类型，所有类型的超类.(universal type)

其中 $c\le a$，c是a的子类型.

⊥：荒谬类型，所有类型的子类(absurd type)

专家系统：快速原型化，检验和修改，不断发现不足，主要的瓶颈在知识获取。

知识库与控制系统是分离的。



how：~~推导~~如何得到这个结果
why：为什么需要这个信息 ⇒ {已确认：~；若~，则~}

{若~&~&~；则~&~&~}

(DFS)
目标驱动推理：先将结果加入工作内存，将规则匹配的放入工作内存，直到工作内存所有子目标均为真。

(一般BFS)
数据驱动推理：按规则库中的顺序将内存中的内容进行匹配，将激发的规则加入工作内存。

规则推理的优点：① 直接从人类专家获取经验知识 ② 可将规则映射为状态空间，解释功能支持调试

③ 知识与控制分离

缺点：① 难以处理意外值 ② 在知识边缘快速退化 ③ 无法进行理论解释

规划：STRIPS. P：前提集合，满足才能操作

A：增加列表，当前操作符导致的状态描述增加 ← 共同描述状态变动

D：删除列表，完成操作符后，从状态描述中删除的项目。

Standford 确信度代数：{MD(H|E) —— 不可信度量；MB(H|E) 为给定证据E时H的可信度量。} 在(0,1)内。

$CF(H|E) = MB(H|E) - MD(H|E)$ 当 {CF→1 时倾向于支持假设，CF→0 ~ 只有极少证据支持或不支持，或两边证据相当；CF→−1 —— 不支持假设}

→ 反映了专家对这条规则可信程度的信息

计算：
$CF(P1\ and\ P2) = MIN(CF(P1), CF(P2))$
$CF(P1\ or\ P2) = MAX(CF(P1), CF(P2))$

例：$(P1^{0.6}\ and\ P2^{0.4})\ or\ P3^{0.2} \rightarrow R1(0.7)\ and\ R2(0.3)$

CF(条件) = MAX(MIN(0.6,0.4),0.2) = 0.4

R1的确信度 = CF(cond) × CF(R1) = 0.28，R2确~ = 0.3×0.4 = 0.12

多~~条件~~规则支持R时的合并：

同号：均为正：$CF(R1)+CF(R2)-CF(R1)\times CF(R2)$，均为负 $CF(R1)+CF(R2)+CF(R1)\times CF(R2)$

异号：$\dfrac{CF(R1)+CF(R2)}{1-MIN(|CF(R1)|,|CF(R2)|)}$

信任度与不信任度：$MB(h|e)=\dfrac{\max(P(h|e),P(h))-P(h)}{1-P(h)}$

$MD(h|e)=\dfrac{\min(P(h|e),P(h))-P(h)}{-P(h)}$

$$\Rightarrow CF(h|e)=\begin{cases}1 & P(h)=1\\ \dfrac{P(h|e)-P(h)}{1-P(h)} & 若\ P(h|e)>P(h)\\ 0 & P(h|e)=P(h)\ 无关\\ \dfrac{P(h|e)-P(h)}{P(h)} & 若\ P(h|e)<P(h)\\ -1 & P(h)=0\end{cases}$$

性质：1、若 $MB(h|e)>0$，则 $MD(h|e)=0$，证据e不能同时支持和不支持某结论h

$MD(h|e)>0$，则 $MB(h|e)=0$

2、若 $P(h|e)>P(h)$ 表明e出现增加了h成立的信任程度，不成立的信任程度反之

3、若 $P(h|e)=P(h)$ 表明证据e与结论h之间相互独立

马尔可夫模型：$P(S_t)=P(S_t|S_{t-1},S_{t-2},S_{t-3}\cdots)$

一阶马尔可夫链：~~P(St|St~~ $P(S_t)=P(S_t|S_{t-1})$

状态转移矩阵：$a_{ij}=P(S_t=s_j|S_{t-1}=s_i)$ 从行数转移到列数

$a_{ij}\geq 0$，$\sum a_{ij}=1$

计算时先列出观察列表 $O=(S_a,S_b\sim S_{\dots})$，列出 $P(O|M)=P(S_a,S_b\sim S\sim|M)=\overset{第一天}{P(S_a)}P(S_b|S_a)P(S_c|S_b)\cdots$

若O中有t+1个S_i，随后变为其它状态，$P(O|M)=1\times \overset{自转移}{a_{ii}^{t}}\times(1-a_{ii})$

可用于计算某种状态的期望天数：$d_i=\sum d\times(a_{ii})^{d-1}\times(1-a_{ii})$，$\because d\to\infty$ $\therefore$ $\boxed{d_i=\dfrac{1}{1-a_{ii}}}$ 其期望天数

候选排除算法：

从特殊到一般：初始化S为第一个正例
N为目前的所有反例集合

对所有正例p：若S中的项不匹配p，用匹配p的最小泛化替换之，删去S中较其它假设更一般的假设，删去S中存在N内有反例的假设

对each负类n：删去S中与N匹配的项，将n加入集合N。

从一般到特殊：初始化G为最泛化概念，P中有所有正例

对each反例n：将G中与n匹配的项用不匹配的最一般特化替代，删去G中较其它假设更特殊的项，删去G中无法与P完全匹配的项。

对each正例p：删去G中不能匹配p的所有假设，将p加入P。

双向搜索：初始化G为空间中最一般概念，初始化S为第一个正例

• 对每个新正例p：删去G中不能与p匹配的所有成员，将S中不匹配p的项目用最特殊的泛化替换S，删去S中更一般的假设，删去S中比G中更一般的假设。

• 对每个负例n：排除S中比G中某些假设更一般的假设。

对每个新负例n：删去匹配n的S成员，对每一个匹配n的G中的项用最一般的特化替换之。排除G中比S中更特殊的所有假设

• 最后：G=S并两个集合只有一个概念，则找到与所有数据一致的概念。

标签的数量

在AI中为I[D]

ID3决策树：信息熵：$entropy(D) = -\sum_{j=1}^{|C|} Pr(c_j)\log_2(Pr(c_j))$ 当Pr(正)=0.5，Pr(负)=0.5时熵最大。熵越小，数据越纯。

在AI中为E[P]

用A_i划分，$entropy_{A_i}(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times entropy(D_j)$ ← 用A_i划分后子集的熵。

$gain(P) = I[C] - E[P]$，选择最大的信息增益。

感知机学习：仅有一层的神经网络，无法分类线性不可分模型。 感知机输出 $\begin{cases} \sum w_i x_i \geq t \Rightarrow 1 \\ \sum w_i x_i < t \Rightarrow -1 \end{cases}$

阈值

权值更新，$\Delta w_i = c(d - sign(\sum x_i w_i)) \cdot x_i$

lr expection

因为W有多个，所以写成向量形式 $\vec{W_t} = \vec{W}_{t-1} + c(d^{t-1} - sign(\vec{W}_{t-1} \cdot \vec{X}_{t-1})) \cdot \vec{X}_{t-1}$，使用矩阵运算。

加入logistic函数(delta rule) $f(net) = \frac{1}{1+e^{-\lambda \cdot net}}$ (sigmoid) ⇒ 连续且可微

平方网络误差：$Error = \frac{1}{2}\sum(d_i - O_i)^2$ d为期望，O为实际，i为结点编号。

$\frac{\partial Error}{\partial O_i} = \frac{\partial \frac{1}{2}\sum(d_i - O_i)^2}{\partial O_i} = -(d_i - O_i)$，$\frac{\partial Error}{\partial w_k} = \frac{\partial Error}{\partial O_i} \cdot \frac{\partial O_i}{\partial w_k} = -(d_i - O_i) \times \frac{\partial O_i}{\partial w_k}$ ∵ $O_i = f(W X_i)$ ∴ $\frac{\partial O_i}{\partial w_k} = f'(net_i) x_k$

$\therefore \Delta w_k = -c\frac{\partial Error}{\partial w_k} = c(d_i - O_i) f'(net_i) \times x_k$

微分后的Input

输出 $O_1 \sim O_j$ ~
↕ w_{ij}
隐层 $O_1 \sim O_i$ ~
↕ w_{ki}
输入 $O_1 \sim O_k$ ~

~~$-delta_i = \frac{\partial Error}{\partial net_i} = \frac{\partial Error}{\partial net_i}$~~ , $\frac{\partial net}{\partial O_i} = w_{ij}$

反向传播：从输出向输入传播。

① 输出→隐层：$net_j = \sum_j w_{ij} O_i$, $\frac{\partial net_i}{\partial O_i} = w_{ij}$, $\Delta W_{ij} = -c\frac{\partial Error}{\partial w_{ij}} = -c\frac{\partial Error}{\partial O_i}\cdot\frac{\partial O_i}{\partial net_i}\cdot\frac{\partial net_i}{\partial w_{ij}}$

$= c(d_i - O_i)f'(net_i) \times x_k$.

当前层输入

遗传：适应度函数 $f(x_i^t)$，返回t时刻候选个体的适应度。

新个体由双亲的分量组成新个体

交叉：分解每个个体，交换分量（例 取A的前半与B的后半，取B的前半和A的后半）

变异：选一位取反，如 (0⇔1)

有序交叉：P1(192|4657|83)，P2(459|1876|23) ⇒ C1(???|4657|??)，C2(???|1876|??)

将P2与C1配对，删去C1中存在的部分，有 C1(239|4657|18)

不变

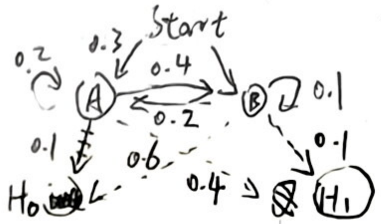
既将P2去掉后有 P2'(23459|18|76)

葛莱码：首位相同，每两位异或后加到首位之后。

(不能被直接观测到)

隐马尔可夫模型：分层，自回归

St为马尔可夫，当且仅当 $P(S_{t+1}|S_t) = P(S_{t+1}|S_1 \sim S_t)$，知道当前状态可抛弃历史状态。



可直接观测

无法直接观测

$(\forall x)A(x) \to B(y) \Leftrightarrow (\exists x)(A(x) \to B(y))$

$(\exists x)A(x) \to B(y) \Leftrightarrow (\forall x)(A(x) \to B(y))$

归结：归结反驳，将要证明的命题取反后加入已知的集合。

1.转化为子句形式 2.将要证明的命题取反，转为子句

3.归结子句，导出新子句 4.用生成空子句得出矛盾 5.用将推出空子句的代换应用于命题

(1) $a \to b \Leftrightarrow \neg a \vee b$ 消去"→"。
(2) 减少"¬"的作用域
(3) 将同名变量改成不同名字
(4) 将量词移到最左边
(5) 存在量词斯柯伦化：看在几个变元的作用范围中，将其换为函数（从∃x定义到第一次使用）
(6) 去掉全称量化
(7) 转换为析取子句的合取形式：(~)∧(~)
(8) 变量名标准化：(Xa) (~)
(Xb) (~) 子句间不能有重名变量。

在归结的时候选择尽量能消掉的句子。

归结时提取相反的变元

$L_1 = \sim$, $L_2 = \sim$

且 $L_1 = \sim L_2$

归结C1和C2得C12为 $(\{L, \sim\} - \{L_1\}) \cup (\{\sim, \sim\} - \{L_2\})$

既将相反的元素去掉取并集，可能存在多组 L_1, L_2。

注：C1与C2不能有同名变元。