# 人工智能技术
# Artificial Intelligence

## ——人工智能: 逻辑智能+计算智能+学习智能
## AI: Logical Intelligence + Computational Intelligence + Learning Intelligence

理论课：王鸿鹏、王润花、韩明静

实验课：许丽、靖智博

南开大学人工智能学院

# 第一部分 逻辑智能

# Logical Intelligence

## ——第五章：推理求解

## Chapter 5: Reasoning Technology

# 推理求解

## Reasoning Technology

**推理**——是运用知识解决问题的方法，是计算机思维能力的表现

## 推理概述

推理就是按某种策略由已知判断推出另一判断的思维过程。

推理包括两种判断：

① 已知判断，它包括已掌握的与求解问题有关的知识及关于问题的已知事实。

② 由已知判断推出的新判断，即推理的结论。

已知判断（事实）和知识是构成推理的两个基本要素。

已知判断（事实）又称为证据，用以指出推理的出发点及推理时应该使用的知识。知识是使推理得以向前推进，并逐步达到最终目标的依据。

# 推理

## 推理方法的分类

一、按推出结论的途径(Logical Basis)划分，推理可分为演绎推理、归纳推理和默认推理。

① 演绎推理(Deduction)是从全称判断推导出特称判断或单称判断的过程，即由一般性知识推出适合于某一具体情况的结论。是一种从一般到个别的推理。

② 归纳推理(Induction)是从足够多的事例中归纳出一般性结论的推理过程，是一种从个别到一般的推理。

③ 默认推理又称为缺省推理。它是在知识不完全的情况下假设某些条件已经具备所进行的推理。在人工智能的缺省推理中认为“在没有证据可以证明某事件不存在的情况下”，就认为它是存在的。

## 推理方法的分类

二、按推理时所使用知识的确定性(Certainty of Knowledge)分，推理可分为确定性推理和不确定性推理。

① 确定性推理(Certainty)是指推理时所用的知识都是精确的，推出的结论也是确定的，其真值要么为真要么为假。

② 不确定性推理(Uncertainty)是指推理时所用的知识不都是确定的，推出的结论也不完全是肯定的。其真值不限于真和假，可能还有其他值。

## 推理方法的分类

三、按推理过程中推出的结论是否单调的增加，或者说是否越来越接近最终目标(monotony of reasoning process)来划分，推理又分为单调推理和非单调推理。

① 单调推理(Monotonic)是指推理过程中随着推理的向前推进以及新知识的加入，推出结论的数目呈单调递增的趋势，并且越来越接近最终目标。如基于经典逻辑的归结推理过程就属于单调推理。

② 非单调推理(Non-monotonic)是指在推理过程中由于新知识加入，不仅没有加强已推出的结论，反而要否定它，使得推理退回到前面的某一步，重新开始。如默认推理是非单调推理。

四、按推理中是否运用与问题有关的启发式知识划分，推理可分为启发式推理和非启发式推理。

# 推理控制策略(Control Strategies in Reasoning)

推理的控制策略主要包括推理方向、搜索策略、求解策略、冲突消解策略及限制策略等。

① 推理方向(Inference Direction)：用于确定推理的驱动方式，分为：

Facts → Conclusions (Forward chain, Data-driven, 正向推理)

Facts ← Conclusions (Backward chain, Goal-driven, 逆向推理)

Facts ↔ Conclusions (Bi-directional, 混合推理)

② 搜索策略(Searching)，第三章已讲述。

③ 求解策略：指推理是求一个解，还是求所有解以及最优解等。

④ 冲突消解(Conflict Resolution)：见产生式系统部分介绍。

Sort knowledge for improving reasoning efficiency

② 限制策略：指在推理过程中对各种资源的限制条件，如对推理的深度、宽度、时间和空间等进行限制。

# 推理是如何进行的?

推理过程多种多样

① 例1：

如果今天不下雨，我就去你家

今天没有下雨

② 例2：

小王说他下午或者去图书馆或者在家休息

小王没去图书馆

③ 计算机如何选择？

# Reasoning Technology 1:

## 确定性推理

## （Certainty Reasoning Technology）

1. 自然演绎推理 (Deduction)
2. 消解原理(Resolution principle)(归结原理)
3. 规则演绎系统(Rule-based Deduction Systems)
4. 产生式系统(Production System)
5. 非单调推理(Non-monotonic Reasoning)

# 推理

## 确定性推理——自然演绎推理

（补充预备知识）：From 离散数学

① 定义1：命题：具有确定真值的陈述句。

② 定义2：谓词：句子是由主语和谓语组成的，主语一般是客体，用于刻划客体的性质或关系的即是谓词。

③ 定义3：连接词：否定（¬/~），合取（ ∧ ），析取（ ∨ ），蕴含（条件）（ →），双条件（↔）

④ 定义4：全称量词（$\forall x$），存在量词（$\exists x$）

# 推理

## 确定性推理——自然演绎推理

（补充预备知识）：From 离散数学

举例：写出谓词表达式

① 所有教练员是运动员。

② 某些运动员是大学生。

③ 某些教练是年老的，但是健壮的。

④ 不是所有运动员都是教练员。

⑤ 金教练既不年老，也不健壮。

⑥ 所有运动员都钦佩某些教练。

⑦ 有些大学生不钦佩运动员。

⑧ 极限的定义

# 自然演绎推理

自然演绎推理是从一组已知为真的事实出发，直接运用经典推理规则，推出结论的过程。

基本的推理规则有P规则、T规则、假言推理和拒取式推理等。

① P规则是指在推理的任何步骤上都可以引入前提，继续进行推理。

② T规则是指在推理时，如果前面步骤中有一个或多个公式永真蕴含S，则可以把S引入到推理过程中。

③ 全称指定规则： $(\forall x)P(x) \Rightarrow P(c)$

④ 存在推广规则： $P(c) \Longrightarrow (\exists x)P(x)$

⑤ 假言推理（见后页）。

⑥ 拒取式推理（见后页）。

# 自然演绎推理

假言推理：

一般形式： $\mathbf{P}, \mathbf{P} \rightarrow \mathbf{Q} \Rightarrow \mathbf{Q}$

上式表示：由$\mathbf{P} \rightarrow \mathbf{Q}$及$\mathbf{P}$为真，可推出$\mathbf{Q}$为真。

例如：由“如果$x$是食物，则$x$能吃”及“馒头是食物”可推出“馒头能吃”的结论。

拒取式推理：

一般形式： $\mathbf{P} \rightarrow \mathbf{Q}, \neg \mathbf{Q} \Rightarrow \neg \mathbf{P}$

上式表示：由$\mathbf{P} \rightarrow \mathbf{Q}$真及$\mathbf{Q}$为假，可推出$\mathbf{P}$为假

例如，由“如果下雨，则地上湿”及“地上不湿”，可推出“没有下雨”的结论。

# 自然演绎推理

【例题1】：设已知以下事实如下：①. 凡是容易的课程小王都喜欢；②. C班的课程都是容易的；③. ds是C班的一门课程。求证：小王喜欢ds这门课程。

证明：

(1) 定义谓词：

$\mathrm{EASY}(x)$： $x$是容易的

$\mathrm{LIKE}(x,y)$： $x$喜欢$y$

$\mathrm{C}(x)$： $x$是C班的一门课程

(2) 将上述事实及待求的问题用谓词公式表示为：

$(\forall x)(\mathrm{EASY}(x) \rightarrow \mathrm{LIKE}(Wang,x))$ //凡是容易的课程小王都喜欢

$(\forall x)(\mathrm{C}(x) \rightarrow \mathrm{EASY}(x))$ //C班的课程都是容易的

$\mathrm{C}(\mathrm{ds})$ //ds是C班的一门课程

$\mathrm{LIKE}(Wang,ds)$ //小王喜欢ds这门课程

Cont.

# 自然演绎推理

【例题1】：设已知以下事实如下：①. 凡是容易的课程小王都喜欢；②. C班的课程都是容易的；③. $ds$是C班的一门课程。求证：小王喜欢$ds$这门课程。

(3) 应用推理规则进行推理：

因为： $(\forall x)(\mathrm{C}(x) \rightarrow \mathrm{EASY}(x))$

所以： $\mathrm{C}(\mathrm{ds}) \rightarrow \mathrm{EASY}(\mathrm{ds})$

所以： $\mathrm{C}(ds), \mathrm{C}(ds) \rightarrow \mathrm{EASY}(ds) \Rightarrow \mathrm{EASY}(ds)$

因为： $(\forall x)(\mathrm{EASY}(x) \rightarrow \mathrm{LIKE}(Wang, x))$

所以： $\mathrm{EASY}(ds) \rightarrow \mathrm{LIKE}(Wang, ds)$

所以： $\mathrm{EASY}(ds), \mathrm{EASY}(\mathrm{ds}) \rightarrow \mathrm{LIKE}(Wang, \mathrm{ds}) \Rightarrow \mathrm{LIKE}(Wang, ds)$

即小王喜欢$ds$这门课程。证毕。

# 归结反演推理(消解)

归结反演推理概述：

研究用计算机实现定理证明的机械化，已是人工智能研究的一个重要领域。对于定理证明问题，如果用一阶谓词逻辑表示的话，就是要求对前提P和结论Q证明 $P \longrightarrow Q$是永真的。然而，要证明这个谓词公式的永真性，必须对所有个体域上的每一个解释进行验证，这是极其困难的。为了化简问题，和数学上常采用的方法一样，我们考虑反证法。即，我们先否定逻辑结论Q，再由否定后的逻辑结论¬Q及前提条件P出发推出矛盾，即可证明原问题。

鲁滨逊

美国数学家鲁滨逊

提出消解原理（1965年）

基本的出发点：要证明一个命题为真都可以通过证明其否命题为假来得到。

将多样的推理规则简化为一个

——消解。

# 归结反演推理(消解)

（补充预备知识）：From 离散数学

① 定义1：不含有任何连接词的谓词公式叫原子公式，简称原子，而原子或原子的否定统称文字(Literal)

- Atomic sentences and their negation
- E.g. $P$ , $\sim P$, $Q(x, y)$, $\sim R(f(x, y), z)$

② 定义2：子句(Clause)就是由一些文字组成的析取式。

- disjunction of literals
- E.g. $P(x) \vee Q(x)$ , $\sim R(x, f(y)) \vee S(x, g(x))$

③ 定义3：不包含任何文字的子句称为空子句，记为$NIL$。

- unsatisfiable

④ 定义4：由子句构成的集合称为子句集。

⑤ 定义5：若P是原子谓词公式或原子命题，则称P与¬P为互补文字。

# 归结反演推理(消解)

（补充知识）：

## 1、前束形范式

一个谓词公式，如果它的所有量词均非否定地出现在公式的最前面，且它的辖域一直延伸到公式之末，同时公式中不出现连接词→及↔，这种形式的公式称作前束形范式。

任意一个谓词公式，均和一个前束形范式等价。

# 归结反演推理(消解)

（补充知识）：

## 2、斯克林（ Skolem ）范式与斯克林标准型

斯克林范式对前束形范式进行了改进，使得首标中所出现的量词具有一定的规则，即每个存在量词均出现在全称量词之前。

从前束形范式中消去全部存在量词所得到的公式即为Skolem标准型。其一般形式为：$(\forall x1)(\forall x2)\ldots(\forall xn)M(x1,x2,\ldots,xn)$

其中，$M(x1,x2,\ldots,xn)$是一个合取范式，称为Skolem标准型的母式。

# 冲突消解策略

- 已知事实与知识的三种匹配情况

① 恰好匹配成功(一对一)

② 不能匹配成功

③ 多种匹配成功（一对多，多对一，多对多）

# 什么是消解

- 例1：
  - 小王说他下午或者去图书馆或者在家休息
  - 小王没去图书馆

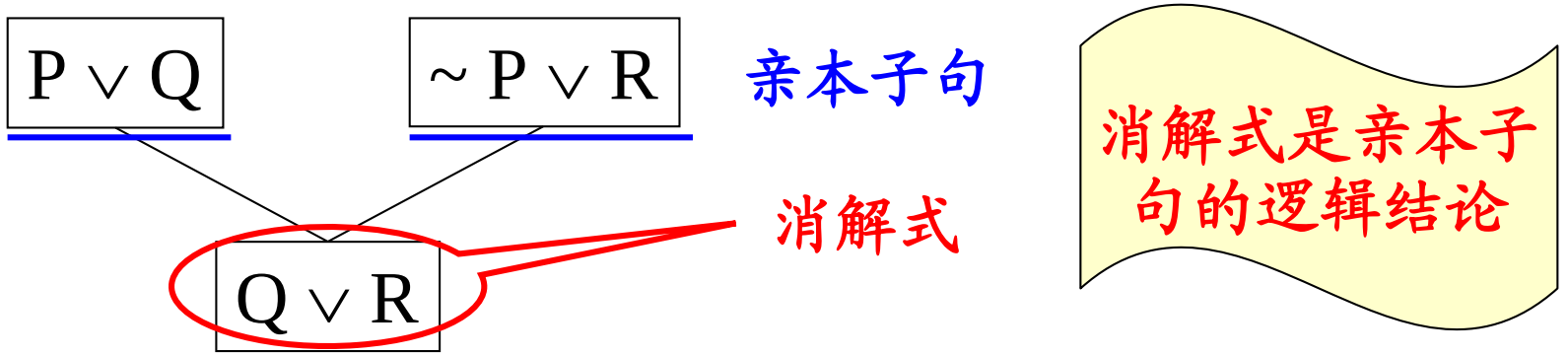
  R——小王下午去图书馆

  S——小王下午在家休息

$$\left.\begin{array}{l} R \vee S \\ \sim R \end{array}\right\} \Rightarrow S$$

- 例2：
  - 如果今天不下雨，我就去你家　　$\sim P \rightarrow Q \Leftrightarrow P \vee Q$
  - 今天没有下雨　　$\sim P$

# 什么是消解




- 消解只能在仅含否定和析取连接词的公式（子句）间进行
- 必须先把公式化成规范的形式（范式，子句集）

# 含变量的消解

- 例：苏格拉底论断

  凡人都会死. $(\forall x)(\text{Man}(x) \Rightarrow \text{Mortal}(x))$

  苏格拉底是人. $\text{Man}(\text{Socrates})$

  如何得到结论：苏格拉底会死. $\text{Mortal}(\text{Socrates})$

- 要完成消解还面临几个问题
  - “∀”和“⇒”必须去掉
    - $\text{Man}(x) \Rightarrow \text{Mortal}(x) \Leftrightarrow \sim \text{Man}(x) \vee \text{Mortal}$
    - “∀”怎么办？

    } 化为子句集

  - 如果能去掉“∀”，$\sim \text{Man}(x)$ 和 $\text{Man}(\text{Socrates})$ 也不能构成互补对，形式不一样，怎么办？

    } 置换与合一

# 子句集的求取(Conversion to Clause Form)

在谓词逻辑中，任何一个谓词公式都可以通过等价关系及推理规则化成相应的子句集。

将谓词公式G化为子句集的步骤：

（1） 消去谓词公式G中的蕴涵（→）和双条件符号（ ↔ ），以¬A∨B代替A→B，以(A∧B)∨(¬A∧¬B)替换A↔B。

（2） 减少否定符号¬的辖域，使否定符号最多只作用到一个谓词上。

（3）重新命名变元名，使所有的变元的名字均不同，并且自由变元及约束变元亦不同。

（4） 消去存在量词。这里分两种情况，一种情况是存在量词不出现在全称量词的辖域内，此时，只要用一个新的个体常量替换该存在量词约束的变元，就可以消去存在量词；另一种情况是，存在量词位于一个或多个全称量词的辖域内，这时需要用一个Skolem函数替换存在量词而将其消去。

# 子句集的求取(Conversion to Clause Form)

将谓词公式G化为子句集的步骤：（续）

（5）把全称量词全部移到公式的左边，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）公式化为Skolem标准型：任何公式都可以写成由一些谓词公式和谓词公式否定的析取的有限集组成的合取。

需要指出的是，由于在化解过程中，消去存在量词时作了一些替换，一般情况下，G的Skolem标准型与G并不等值。

（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。

（9）消去合取词，得到子句集。

# 子句集的求取(Conversion to Clause Form)

- Example

将下列谓词演算公式化为一个子句集

$$(\forall x)\{P(x)\Rightarrow\{(\forall y)[P(y)\Rightarrow P(f(x,y))]\wedge\sim(\forall y)[Q(x,y)\Rightarrow P(y)]\}\}$$

Start:

$$(1)\ (\forall x)\ \{\sim P(x)\vee\{(\forall y)\ [\sim P(y)\vee P(f(x,y))]\wedge\sim(\forall y)\ [\sim Q(x,y)\vee P(y)]\ \}\}$$

(1)Eliminate implication symbols

只应用∨和～符号，以～A∨B替换A ⇒ B。

# 子句集的求取(Conversion to Clause Form)

**(2)** $(\forall x)\{\sim P(x)\vee\{(\forall y)[\sim P(y)\vee P(f(x,y))]\wedge(\exists y)[Q(x,y)\wedge\sim P(y)]\}\}$

(2) Reduce the scopes of negation symbols by repeatedly using DeMorgan's laws 每个否定符号只用到一个谓词符号上

**(3)** $(\forall x)\{\sim P(x)\vee\{(\forall y)[\sim P(y)\vee P(f(x,y))]\wedge(\exists w)[Q(x,w)\wedge\sim P(w)]\}\}$

(3) Standardize variables: each quantifier should use a different one

对哑元（虚构变量）改名，以保证每个量词有其自己唯一的哑元。

# 子句集的求取(Conversion to Clause Form)

(4) Eliminate existenti

(4) $(\forall x)\{\sim P(x)\vee\{(\forall y)[\sim P(y)\vee P(f(x,y))]\wedge[Q(x,g(x))\wedge\sim(g(x))]\}\}$

式中，$w=g(x)$为一Skolem函数。

- 2 situations
- The "∃" is within the scope of some "∀"
  - E.g. $\forall x\ \exists y$ Height$(x, y)$
  - $y$ that "exists" might depend on $x$
  - Let this dependence be explicitly defined by some function $f(x)$, such a function is called a *Skolem function*.
  - Eliminate the "∃" and write $\forall x$ Height$(x, f(x))$
  - More e.g. $\forall x\ \forall y\ \exists z$ P$(x, y, z)$=> $\forall x\ \forall y$ P$(x, y, f(x, y))$
- The "∃" is not within the scope of a "∀"
  - E.g. $\exists x\ \forall y$ P$(x, y)$
  - Use a constant, $\exists x\ \forall y$ P$(x, y)$ becomes $\forall y$ P$(A, y)$

# 子句集的求取(Conversion to Clause Form)

(5) Convert to prenex form 化为前束形

Move all of the "∀" to the front of the *wff* and let the scope of each quantifier include the entirety of the wff

prenex form :

前束形={前缀} {母式}

全称量词串 无量词公式

(5) $(\forall x)(\forall y)\{\sim P(x)\vee\{[\sim P(y)\vee P(f(x,y))]\wedge[Q(x,g(x))\wedge\sim P(g(x))]\}\}$

# 子句集的求取(Conversion to Clause Form)

(5) $(\forall x)(\forall y)\{\sim P(x)\vee\{[\sim P(y)\vee P(f(x,y))]\wedge[Q(x,g(x))\wedge\sim P(g(x))]\}\}$

(6) $(\forall x)(\forall y)\{[\sim P(x)\vee\sim P(y)\vee P(f(x,y))]\wedge[\sim P(x)\vee Q(x,g(x))]\wedge[\sim P(x)\vee\sim(g(x))]\}$

(6) Convert the matrix into a conjunctive normal form;

把母式化为合取范式

(7) $\{[\sim P(x)\vee\sim P(y)\vee P(f(x,y))]\wedge[\sim P(x)\vee Q(x,g(x))]\wedge[\sim P(x)\vee\sim(g(x))]\}$

**(7)** Remove the prefix, eliminate universal quantifiers;消去全称量词

# 子句集的求取(Conve

(8) $\{\sim P(x)\vee\sim P(y)\vee P(f(x,y)),$
$\sim P(x)\vee Q(x,g(x)),$
$\sim P(x)\vee\sim P(g(x))\}$

(8) Eliminate conjunction symbols $\wedge$ and create a separate clause;

用{A,B}代替(A∧B)，消去符号∧。最后得到一个有限集，其中每个公式是文字的析取。

(9) Rename variables and standardize them

No variable symbol appears in more than one clause.

(9) $\{\sim P(x1)\vee\sim P(y)\vee P[f(x1,y)],$
$\sim P(x2)\vee Q[x2,g(x2)],$
$\sim P(x3)\vee\sim P[g(x3)]\}$

# Question

- Convert the following to clause form:

"Everyone who loves all animals is loved by someone."

$\forall x \{[\forall y (Animal(y) \Rightarrow Loves(x,y))] \Rightarrow \exists y\ Loves(y,x)\}$

**Answer**

1. Eliminate implications 消去蕴含符合

$\forall x\ (\sim \forall y\ (\sim Animal(y) \vee Loves(x, y)) \vee \exists y\ Loves(y, x))$

2. Move ~ inwards减小否定辖域范围

$\forall x\ (\exists y\ \sim(\sim Animal(y) \vee Loves(x, y)) \vee \exists y\ Loves(y, x))$

$\forall x\ (\exists y\ (Animal(y) \wedge \sim Loves(x,y\ )) \vee \exists y\ Loves(y, x))$

3. Standardize variables 变量标准化

$\forall x\ (\exists y\ (Animal(y) \wedge \sim Loves(x, y)) \vee \exists \mathbf{z}\ Loves(\mathbf{z}, x))$

4. Skolemize 消去存在量词

$\forall x\ ((Animal(\boldsymbol{f(x)}) \wedge \sim Loves(x, \boldsymbol{f(x)})) \vee Loves(\boldsymbol{g(x)}, x))$

5. Convert to CNF 化为合取范式

$\forall x\ (\ (Animal(f(x)) \vee Loves(g(x), x)) \wedge (\sim Loves(x, f(x)) \vee Loves(g(x), x))\ )$

6. Drop universal quantifiers 消去全称量词

$(Animal(f(x)) \vee Loves(g(x), x)) \wedge (\sim Loves(x, f(x)) \vee Loves(g(x), x))$

7. The clauses set is: 消去连词符合变子句集

$\{Animal(f(x)) \vee Loves(g(x), x),\ \sim Loves(x, f(x)) \vee Loves(g(x), x)\}$

8. Standardize variables 变量标准化

(1) $Animal(f(x1)) \vee Loves(g(x1), x1)$

(2) $\sim Loves(x2, f(x2)) \vee Loves(g(x2), x2)\}$

# 消解推理规则(Resolution Inference Rules)

- **Resolvent Definition 消解式的定义**
  - 令$L_1$, $L_2$为两任意原子公式；$L_1$和$L_2$具有相同的谓词符号，但一般具有不同的变量。
  - 已知两子句$L_1 \vee \alpha$和$\sim L_2 \vee \beta$,如果$L_1$和$L_2$具有最一般合一者$\sigma$，那么通过消解可以从这两个父辈子句推导出一个新子句$(\alpha \vee \beta)\sigma$。这个新子句叫做消解式。

# 消解规则证明(Proof of Resolution Rules)

Resolution:

$$\left.\begin{array}{l} C_1 = L \vee C_1' \\ C_2 = \neg L \vee C_2' \end{array}\right\} \rightarrow C_{12} = C_1' \vee C_2'$$

Proof:

$$\because C_1' \vee L \Leftrightarrow \sim C_1' \Rightarrow L \qquad \sim L \vee C_2' \Leftrightarrow L \Rightarrow C_2'$$

$$\therefore C_1 \wedge C_2 = (\sim C_1' \Rightarrow L) \wedge (L \Rightarrow C_2')$$

$$[(\sim C_1' \Rightarrow L) \wedge (L \Rightarrow C_2')] \Rightarrow [\sim C_1' \Rightarrow C_2']$$

$$\because \sim C_1' \Rightarrow C_2' \Leftrightarrow C_1' \vee C_2' = C_{12} \qquad \therefore C_1 \wedge C_2 \Rightarrow C_{12}$$

# 消解式举例 (Resolution Examples)

- Resolvent Examples 消解式例子

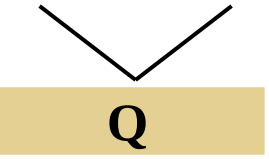
(a) NIL Clause空子句

父辈子句　　P　　~P

NIL

(b) Modus ponens 假言推理

P　　~P ∨ Q　（P ⇒ Q）



Q

(c) Combination 合并

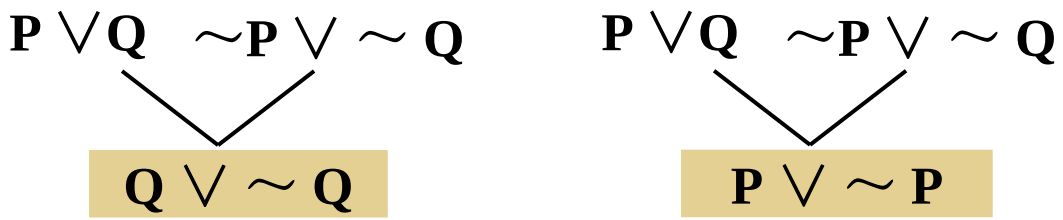
P ∨ Q　　~P ∨ Q

Q ∨ Q = Q

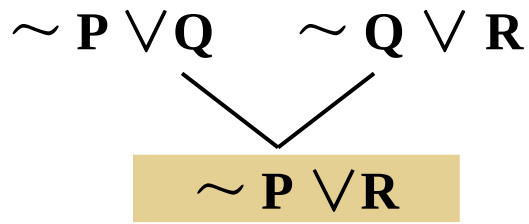
# 消解式举例 (Resolution Examples)

Note: Each time only one pair of literal can be eliminate

(d) Tautologies 重言式




(e) Chain 链式（三段论）



# 含有变量的消解式

- 含有变量的子句之消解式
  - 要把消解推理规则推广到含有变量的子句，必须找到一个作用于父辈子句的置换，使父辈子句含有互补文字。
- Example

$$P[x,f(y)] \vee Q(x) \vee R[f(a),y] \qquad \sim P[f(f(a)),z] \vee R(z,w)$$

$$\sigma = \{f(f(a))/x,\ f(y)/z\}$$

$$Q[f(f(a))] \vee R(f(a),\ y) \vee R(f(y),\ w)$$

# 消解反演求解过程(Solving Process of Resolution Refutation)

- Resolution Refutation 消解反演

Given a set of WFFs {*S*} and goal WFF *L* which will be proved by resolution refutation.

1. Convert *S* to clause form
2. Convert the negation of *L* (~ *L*) to clause form
3. Combine the clauses resulting from steps 1 and 2 into a single set *T*.
4. Iteratively apply resolution to the clauses in *T* and add the results to *T* either until there are no more resolvents that can be added or until the empty clause NIL is produced

# 消解反演求解过程(Solving Process of Resolution Refutation)

- Resolution Refutation 消解反演

  给出公式集{$S$}和目标公式$L$

  - 否定$L$，得~$L$;
  - 把~$L$添加到$S$中去;
  - 把新产生的集合 $T$ = { ~$L$，$S$} 化成子句集;
  - 应用消解原理，力图推导出一个表示矛盾的空子句

# 消解反演过程举例

【例题2】：已知：每个储蓄钱的人都获得利息。求证：如果没有利息，那么就没有人去储蓄？

# 消解反演过程举例

【例题2】：已知：每个储蓄钱的人都获得利息。求证：如果没有利息，那么就没有人去储蓄？

证明：

(1) 定义谓词：

$S(x,y)$： $x$储蓄$y$

$M(x)$： $x$是钱

$I(x)$： $x$是利息

$E(x,y)$： $x$获得$y$

(2) 将已知事实及待证明的结论用谓词公式表示为：

$$(\forall x)[(\exists y)(S(x,y)) \wedge M(y)] \Rightarrow [((\exists y)(I(y)) \wedge E(x,y)] \text{ //已知事实}$$

$$\neg\,(\exists x)I(x) \Rightarrow (\forall x)(\forall y)(M(y) \rightarrow \neg S(x,y)) \text{//待证结论}$$

Cont.

# 消解反演过程举例

【例题2】：已知：每个储蓄钱的人都获得利息。求证：如果没有利息，那么就没有人去储蓄？

证明：

(3) 把前提化为子句形：

$$\sim S(x,y) \vee \sim M(y) \vee I(f(x))$$

$$\sim S(x,y) \vee \sim M(y) \vee E(x,f(x))$$

把结论的否定化为子句形

$$\sim I(z)$$

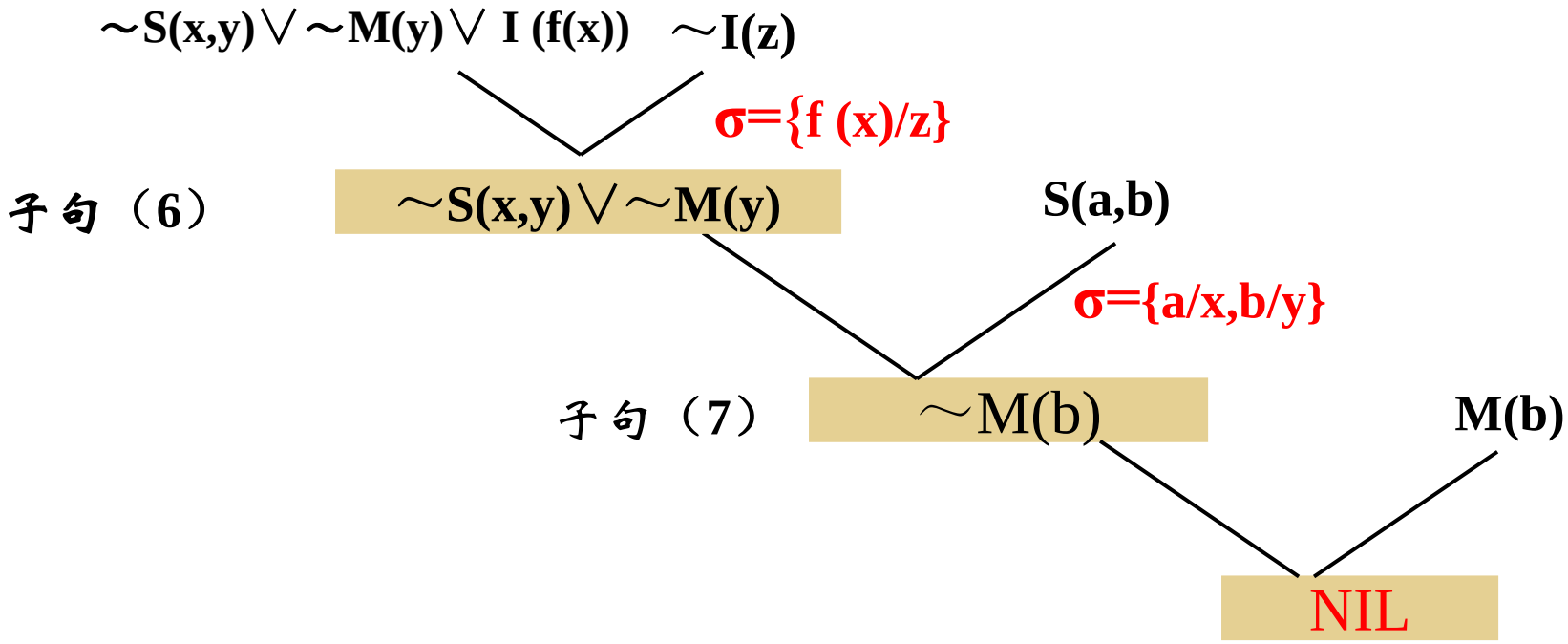
$$S(a,b)$$

$$M(b)$$

Cont.

# 消解反演过程举例

(4) 消解反演求空子句（NIL）




储蓄问题反演树

# 反演求解过程

- 反演求解Solving Process of Resolution Refutation
  - 把由目标公式的否定产生的每个子句添加到目标公式否定之否定的子句中去。Add every clause that is generated by the negation of goal clause into clause of goal negation of negation.
  - 按照反演树，执行和以前相同的消解，直至在根部得到某个子句止。Resolve these clauses together, producing new clauses that logically.
  - The substitutions used to produce the goal clause is the answer.用根部的子句作为一个回答语句
- 实质
  - 把一棵根部有NIL的反演树变换为根部带有回答语句的一棵证明树。

# 应用归结原理对问题求解步骤

应用归结原理对问题求解步骤：

① 把已知前提条件用谓词公式表示出来，并化成相应的子句集，设该子句集的名字为$S_1$。

② 把待求解的问题也用谓词公式表示出来，然后将其否定，并与一谓词ANSWER构成析取式。谓词ANSWER是一个专为求解问题而设置的谓词，其变量必须与问题公式的变量完全一致。

③ 把问题公式与谓词ANSWER构成的析取式化为子句集，并把该子句集与$S_1$合并构成子句集$S$。

④ 对子句集$S$应用谓词归结原理进行归结，在归结的过程中，通过合一置换，改变ANSWER中的变元。

⑤ 如果得到归结式ANSWER，则问题的答案即在ANSWER谓词中。

# 消解反演过程举例

【例题3】：任何兄弟都有同一个父亲，John和Peter是兄弟，且John的父亲是David，问Peter的父亲是谁？

解：

Step1：将已知条件用谓词公式表示出来，并化成子句集，先定义谓词：

(1) 定义谓词：

设：Father$(x, y)$表示$x$是$y$的父亲

Brother$(x, y)$表示$x$是$y$的兄弟

(2) 将上述事实及待求的问题用谓词公式表示为：

F1：$(\forall x)(\forall y)(\forall z)(\text{Brother}(x, y) \wedge \text{Father}(z, x) \rightarrow \text{Father}(z, y))$

//任何兄弟都有同一个父亲

F2：Brother(John, Peter) //John和Peter是兄弟

F3：Father(David, John) //John的父亲是David

Cont.

# 消解反演过程举例

【例题3】：任何兄弟都有同一个父亲，John和Peter是兄弟，且John的父亲是David，问Peter的父亲是谁？

解：

(3) 将它们化成子句集得：

$S_1 = \{\neg \text{Brother}(x, y) \vee \neg \text{Father}(z, x) \vee \text{Father}(z, y), \text{Brother}(\text{John}, \text{Peter}), \text{Father}(\text{David}, \text{John})\}$

Step2：把问题用谓词公式表示出来，并将其否定与谓词ANSWER作析取：

设Peter的父亲是$u$，则有： $\text{Father}(u, \text{Peter})$。

将其否定与ANSWER作析取，得：

G: $\neg \text{Father}(u, \text{Peter}) \vee \text{ANSWER}(u)$

Cont.

# 消解反演过程举例

【例题3】：任何兄弟都有同一个父亲，John和Peter是兄弟，且John的父亲是David，问Peter的父亲是谁？

解：

Step3：将上述公式G化为子句集$S_2$,并将$S_1$和$S_2$合并到S：

$$S_2 = \{\neg \text{Father}(u, \text{Peter}) \vee \text{ANSWER}(u)\}$$

$$S = S_1 \cup S_2$$

将$S$中各子句列出如下：

①. $\neg \text{Brother}(x, y) \vee \neg \text{Father}(z, x) \vee \text{Father}(z, y)$

②. $\text{Brother}(\text{John}, \text{Peter})$

③. $\text{Father}(\text{David}, \text{John})$

④. $\neg \text{Father}(u, \text{Peter}) \vee \text{ANSWER}(u)$

Cont.

# 消解反演过程举例

【例题3】：任何兄弟都有同一个父亲，John和Peter是兄弟，且John的父亲是David，问Peter的父亲是谁？

解：

Step4：应用归结原理进行归结：

⑤.$\neg \text{Brother}(\text{John}, y) \vee \text{Father}(\text{David}, y)$

①与③归结 $\sigma = \{\text{David}/z, \text{John}/x\}$

⑥. $\neg \text{Brother}(\text{John}, \text{Peter}) \vee \text{ANSWER}(\text{David})$

④与⑤归结 $\sigma = \{\text{David}/\text{u}, \text{Peter}/y\}$

⑦. $\text{ANSWER}(\text{David})$

②与⑥归结

Step5：得到了归结式ANSWER(David)，答案即在其中，所以 $u = \text{David}$。即Peter的父亲是David。求解完毕。

Cont.

# 消解反演过程举例

【例题4】：如果无论约翰(John)到哪里，菲多(Fido)也就去那里，那么如果约翰在学校里，菲多在那里呢？

解：

Step1：将已知条件用谓词公式表示出来，并化成子句集，先定义谓词：

(1) 定义谓词：

设：$\mathrm{AT}(x,y)$表示$x$在$y$地方

(2) 将上述事实及待求的问题用谓词公式表示为：

事实：$(\forall x)\,(\mathrm{AT}(John,x) \rightarrow \mathrm{At}(Fido,x))$

//无论约翰(John)到哪里，菲多(Fido)也就去那里

$\mathrm{AT}(John,\mathrm{School})$ //约翰(John)在学校里

目标：$(\exists x)\mathrm{At}(Fido,x)$//求：菲多在那里呢?

Cont.

# 消解反演过程举例

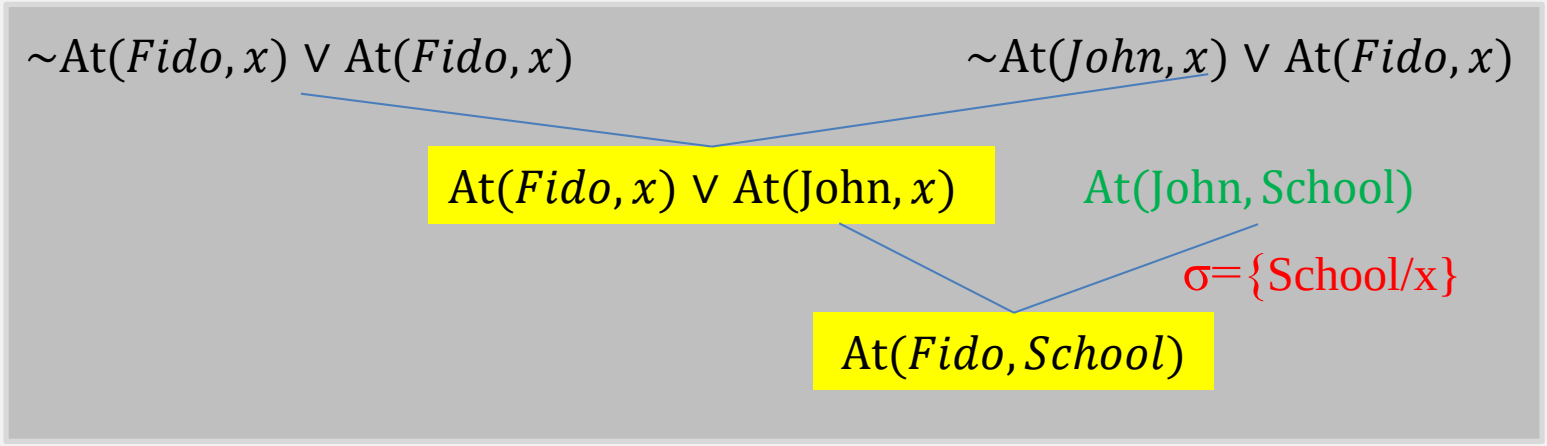
【例题4】：如果无论约翰(John)到哪里，菲多(Fido)也就去那里，那么如果约翰在学校里，菲多在那里呢？

(3) 用反演求解过程(2)：

目标否定子句：$\sim \mathrm{At}(Fido, x)$

将其添加到目标否定之否定的子句中：

$$\sim \mathrm{At}(Fido, x) \vee \mathrm{At}(Fido, x)$$




(4) 从根部求得答案：$\mathrm{At}(Fido, School)$

# 规则演绎系统

# （Rule-based Deduction Systems）

# 规则演绎系统 Rule-based Deduction Systems

- 对于许多公式来说，子句形是一种低效率的表达式，一些重要信息可能在求取子句形过程中丢失
- 例
  - 如果人们发烧并感到肚子疼那很可能是感染了(if the person has fever and feels tummy-pain then she may have an infection)
  - **$(\forall x)[ (has\_fever(x) \wedge tummy\_pain(x)) \Rightarrow has\_an\_infection(x)]$**
  - **clausal form: $\sim has\_fever(x) \vee \sim tummy\_pain(x) \vee has\_an\_infection(x)$**
  - 两者在逻辑上是等价的
  - 但从子句形中无法判断谁是因谁是果，丢失了重要的信息
- Rule
  - General form: If … then …

# 规则演绎系统 Rule-based Deduction Systems

- 基本思想
  - 充分利用蕴涵式的优势
  - 直接使用规则的形式进行推理
- 系统中包括两类信息
  - IF-THEN 规则
  - 事实(Facts)
- 根据推理方向，可分为
  - 正向推理(正向链推理,Forward chaining)
  - 逆向推理(逆向链推理,Backward chaining)

# 规则正向演绎系统 Forward Rule-based Deduction Systems

- 从if部分向then部分推理的过程
- 从事实或状况向目标或动作进行操作
- 事实驱动或数据驱动
- 事实被表示成与或形(AND/OR form)
  - 仅含有合取(与)∧和析取(或) ∨联接词
  - An expression in AND/OR form is not in clausal form

# 规则正向演绎系统 Forward Rule-based Deduction Systems

- Solving steps
  - (1) Transform fact expressions as AND/OR form

    事实表达式的与或形变换
  - (2) AND/OR tree of facts expression

    事实表达式的与或树
  - (3) F rule transform

    F规则变换
  - (4) Goal formula as termination condition

    作为终止条件的目标公式

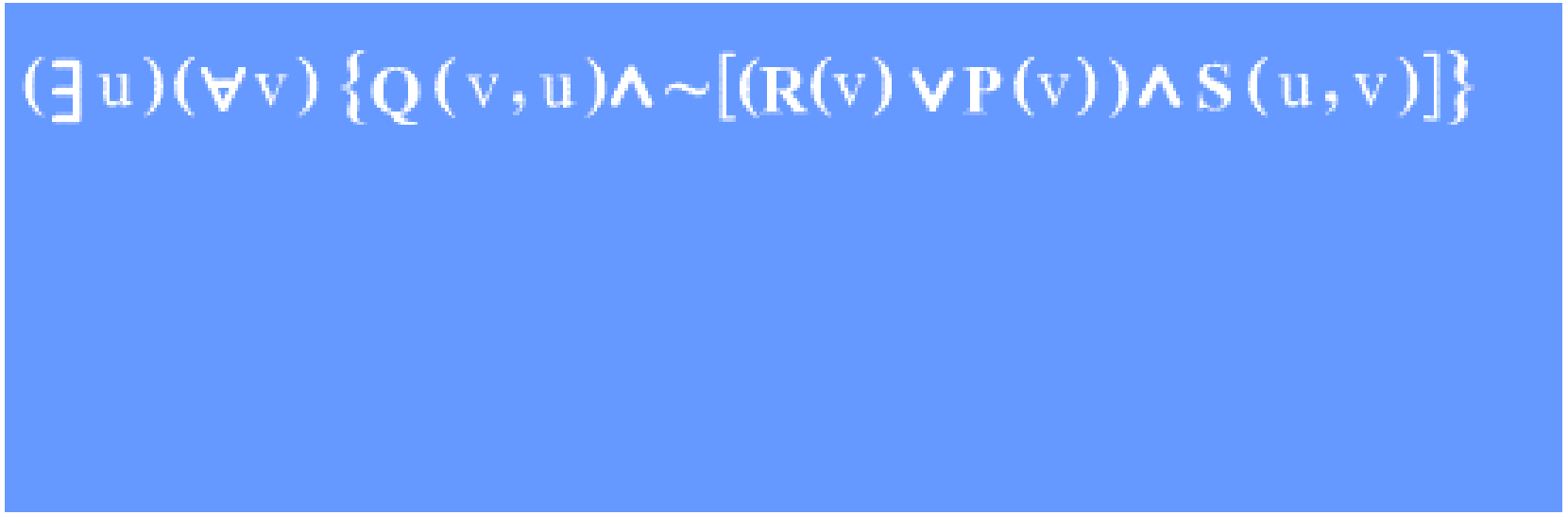
# 规则正向演绎系统 Forward Rule-based Deduction Systems

- (1) 事实表达式的与或形变换
  - 1）消除蕴含符号（暂时）
  - 2）减少否定辖域范围
  - 3） *Skolem*化
  - 4） 化前束形，并去掉全称量词
  - 5） 变量标准化

# 规则正向演绎系统 Forward Rule-based Deduction Systems

- Example:

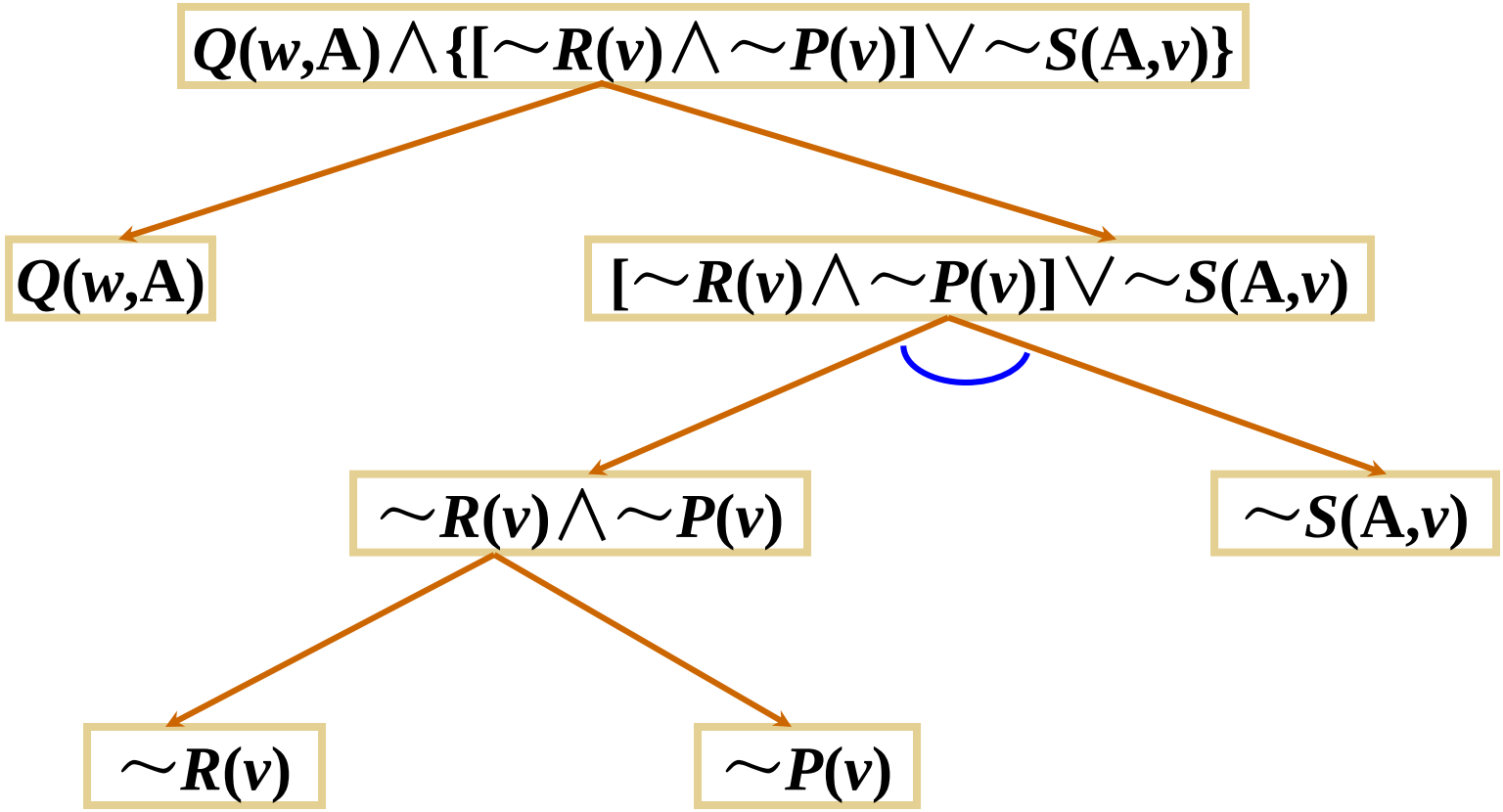
  $(\exists u)(\forall v)\{Q(v,u) \wedge [(R(v) \vee P(v)) \wedge S(u, v)]\}$




- In this systems, facts were represented by non-implicative AND/OR form, as total database of systems.

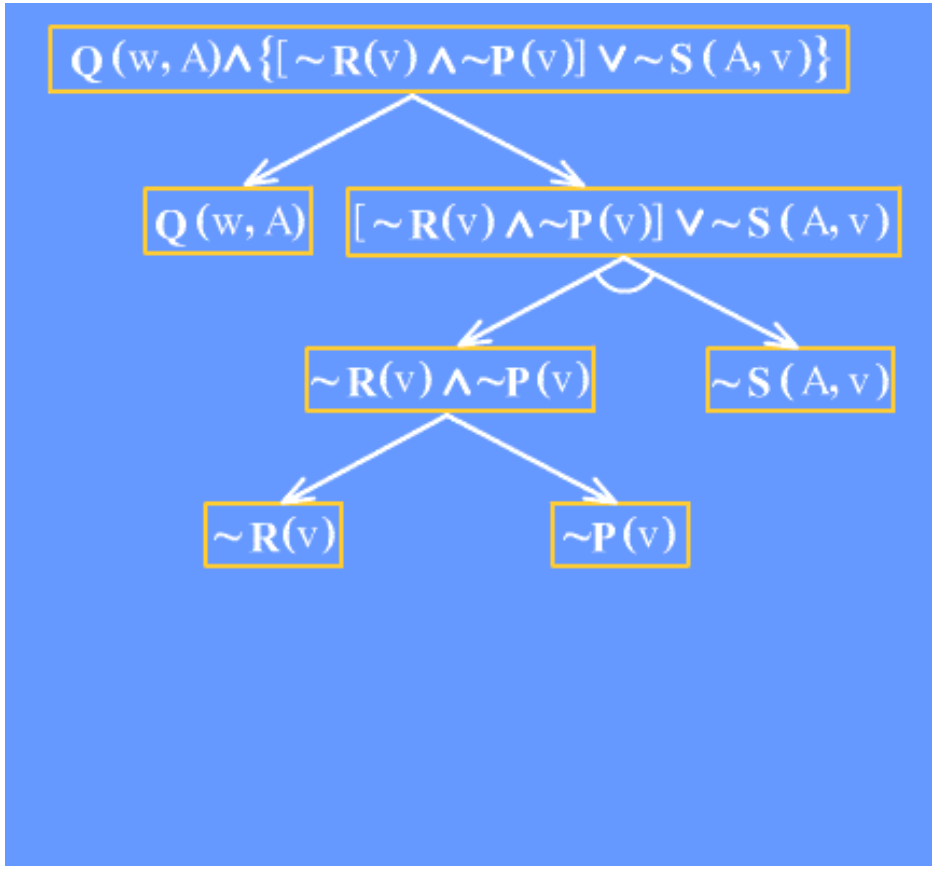
# 规则正向演绎系统 Forward Rule-based Deduction Systems

- (2) 事实表达式的与或树



# 规则正向演绎系统 Forward Rule-based Deduction Systems

- Obtained clause from AND/OR tree of a fact expression



# 规则正向演绎系统 Forward Rule-based Deduction Systems

- (3) F rule transform

  F规则

  $$L \Rightarrow W$$

  where： L was a single literal; W was a unique formula of AND/OR from.

  - Assumed
    - Any variable in implication formulae is universal quantified on whole implication.
    - Some variables of this facts and rules were separated and standardized, so as to non variables appear in more than one rule, and rule variables are different with fact variables.

# 规则正向演绎系统 Forward Rule-based Deduction Systems

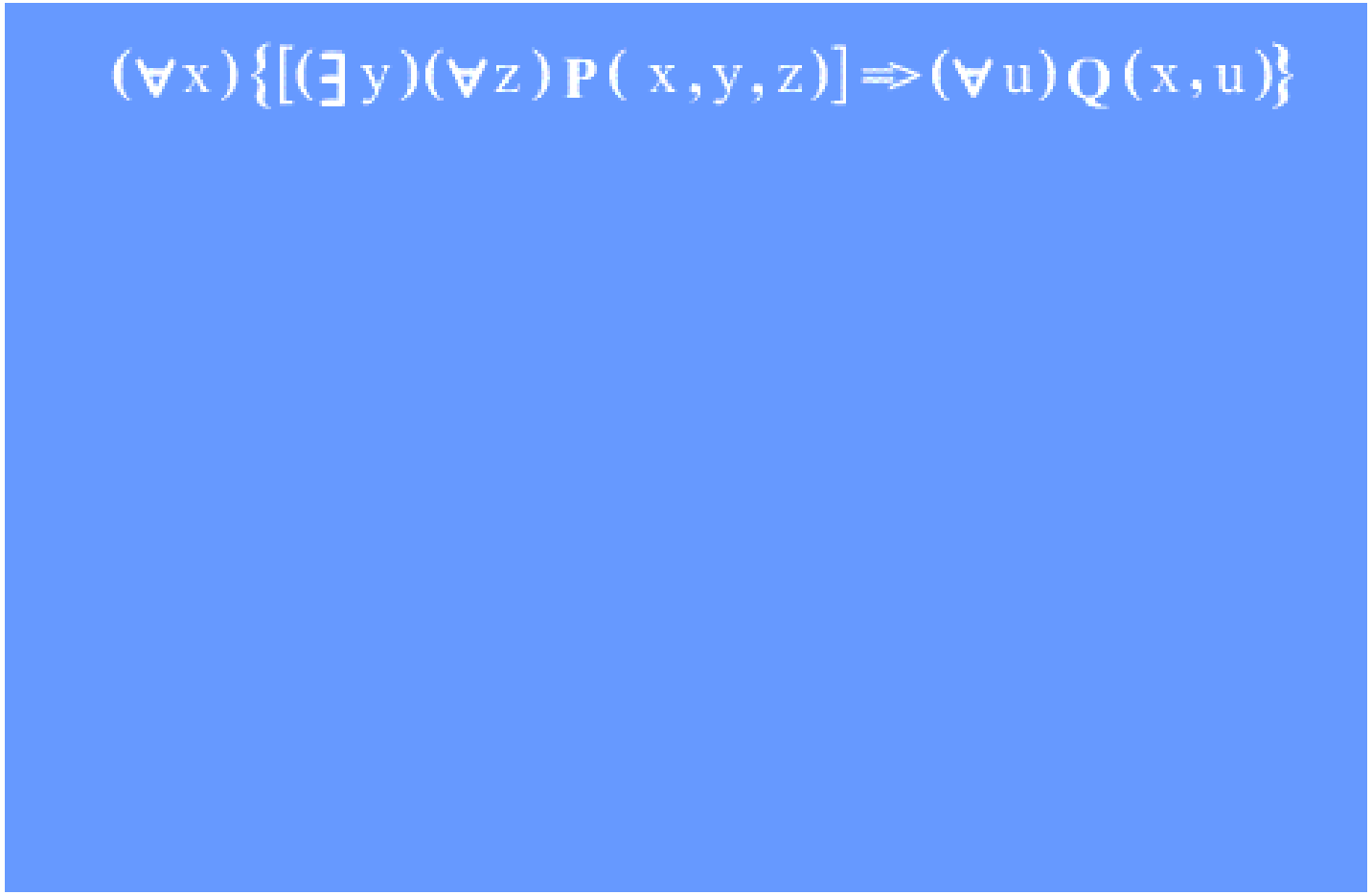
- F规则变换步骤
  - Eliminate (temporarily) implication symbols.
  - I
  - S
  - M
  - E
  - I

$$(\forall x)\{[(\exists y)(\forall z)P(x,y,z)]\Rightarrow(\forall u)Q(x,u)\}$$

# 规则正向演绎系统 Forward Rule-based Deduction Systems

- For example:

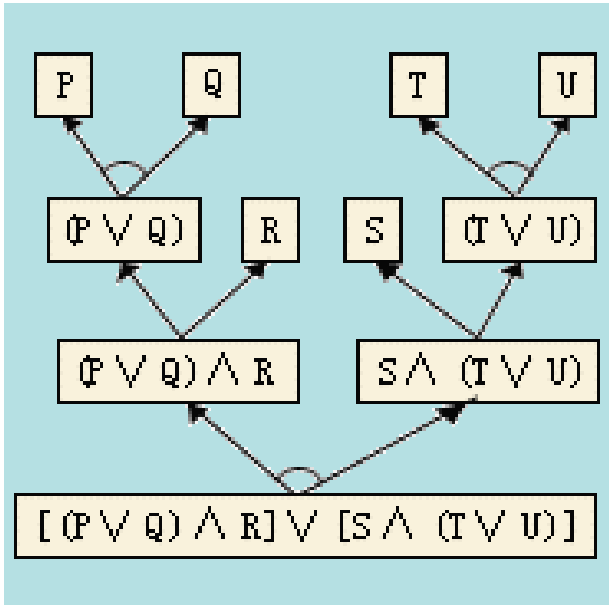
$$(\forall x)\{[(\exists y)(\forall z)P(x, y, z)] \Rightarrow (\forall u)Q(x, u)\}$$

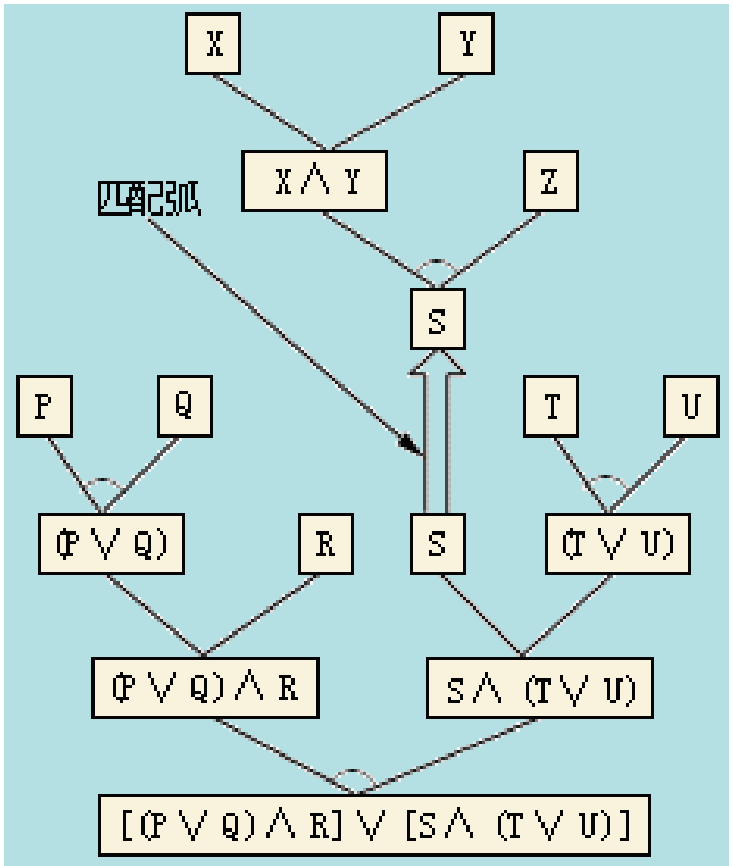


# 规则正向演绎系统 Forward Rule-based Deduction Systems

- F规则应用到与或树

Rule1: $S \Rightarrow (X \wedge Y) \vee Z$



# 规则正向演绎系统 Forward Rule-based Deduction Systems

- （4） Goal formula as termination condition

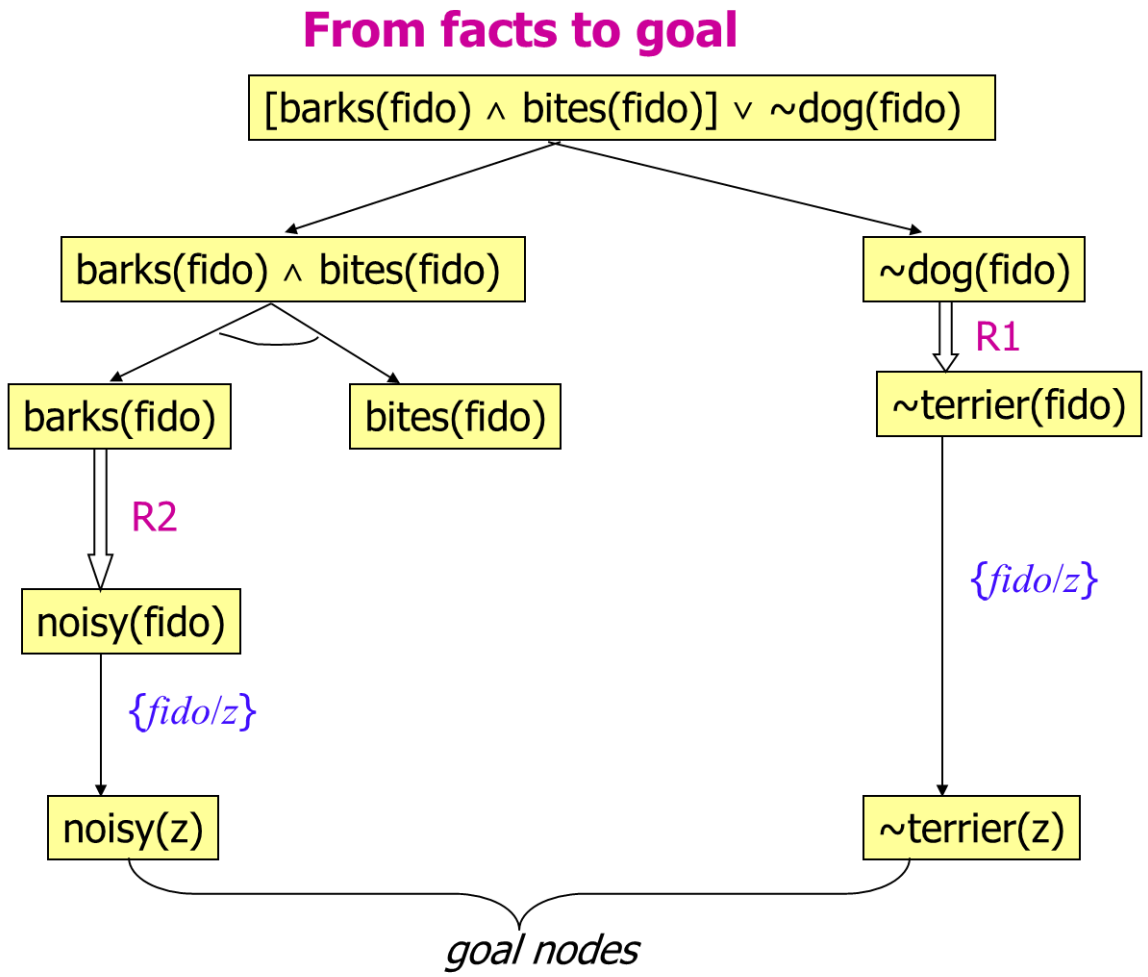
  作为终止条件的目标公式

  - The goal formula is limited in provable formula, especially those of literal disjunctive form.

# Example of forward chaining

- Fact: Fido barks and bites, or Fido is not a dog.
- F rules :
  - R1: All terriers are dogs.
  - R2: Anyone who barks is noisy.
- Goal: “there exists someone who is not a terrier or who is noisy.”
- Logic representation:
  - (barks(fido) ∧ bites(fido)) ∨ ~dog(fido)
  - R1: terrier(x) → dog(x)
  - R2: barks(y) → noisy(y)
  - goal: ∃w (~terrier (w) ∨ noisy(w))

# Example of forward chaining



# forward chaining

- F规则的目的在于从某个事实公式和某个规则集出发来证明某个目标公式
- 目标表达式只限于可证明的文字析取形的目标公式
- 目标文字和规则可用来对与或图添加后继节点
- 当一个目标文字与该图中文字节点n上的一个文字相匹配时，就对该图添加这个节点n的新后裔，并标记为匹配的目标文字，这个后裔叫做目标节点，目标节点都用匹配弧分别接到其父节点上。
- 当产生一个与或图，并包含有终止在目标节点上的一个解图时，系统成功结束。

# 规则逆向演绎系统

- Backward rule-based deduction systems are reasoning from THEN to IF ( from *goal* or *action* to *fact* or *situation condition*).
- Solving procedure
  - (1) AND/OR form of goal expressions
  - (2) B rule transform of AND/OR graph $W \Rightarrow L$
  - (3) Consistent solving graph of fact node as termination condition

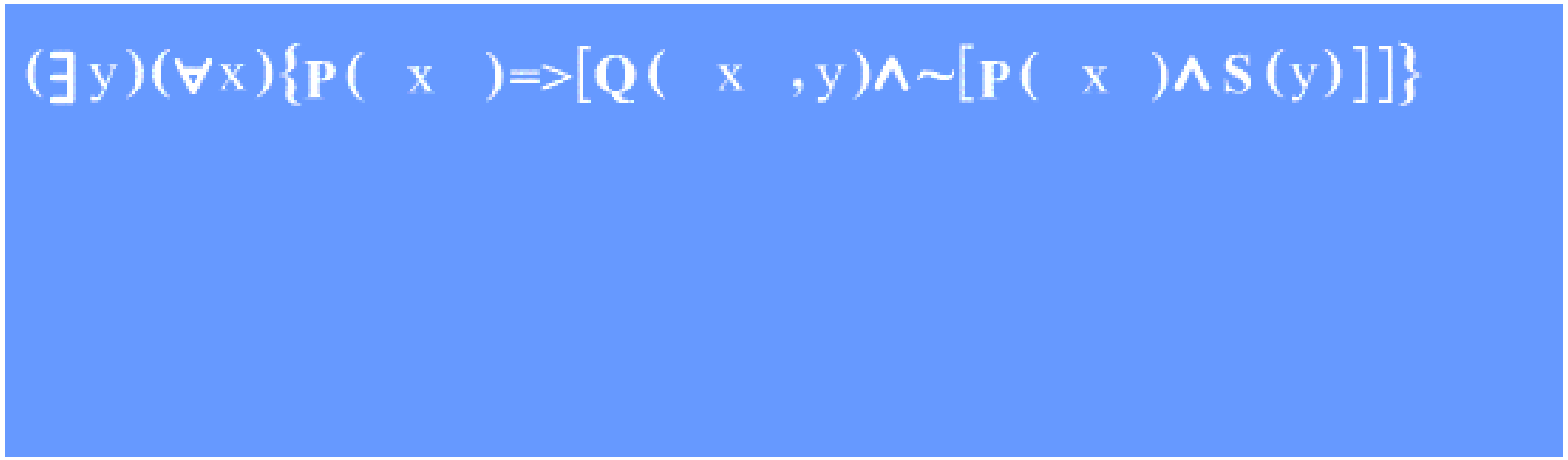
# 规则逆向演绎系统

- (1) AND/OR form of goal expressions
  - Backward deduction systems can deal with any goal expressions form.
  - Transform steps
    - Eliminated implication symbol ;
    - Moved negative symbol into bracket;
    - Skolemizated universal quantifiers and delete existential quantifiers.
    - Assumed that all variables which were remained AND/OR form has been quantified by existential quantifiers.

# 规则逆向演绎系统

- Exp: goal expression

$$(\exists y)(\forall x)\{P(x) \Rightarrow [Q(x, y) \wedge \sim[P(x) \wedge S(y)]]\}$$




Transform to AND/OR form

$$\sim P(f(y)) \vee \{Q(f(y),y) \wedge [\sim P(f(y)) \vee \sim S(y)]\}$$

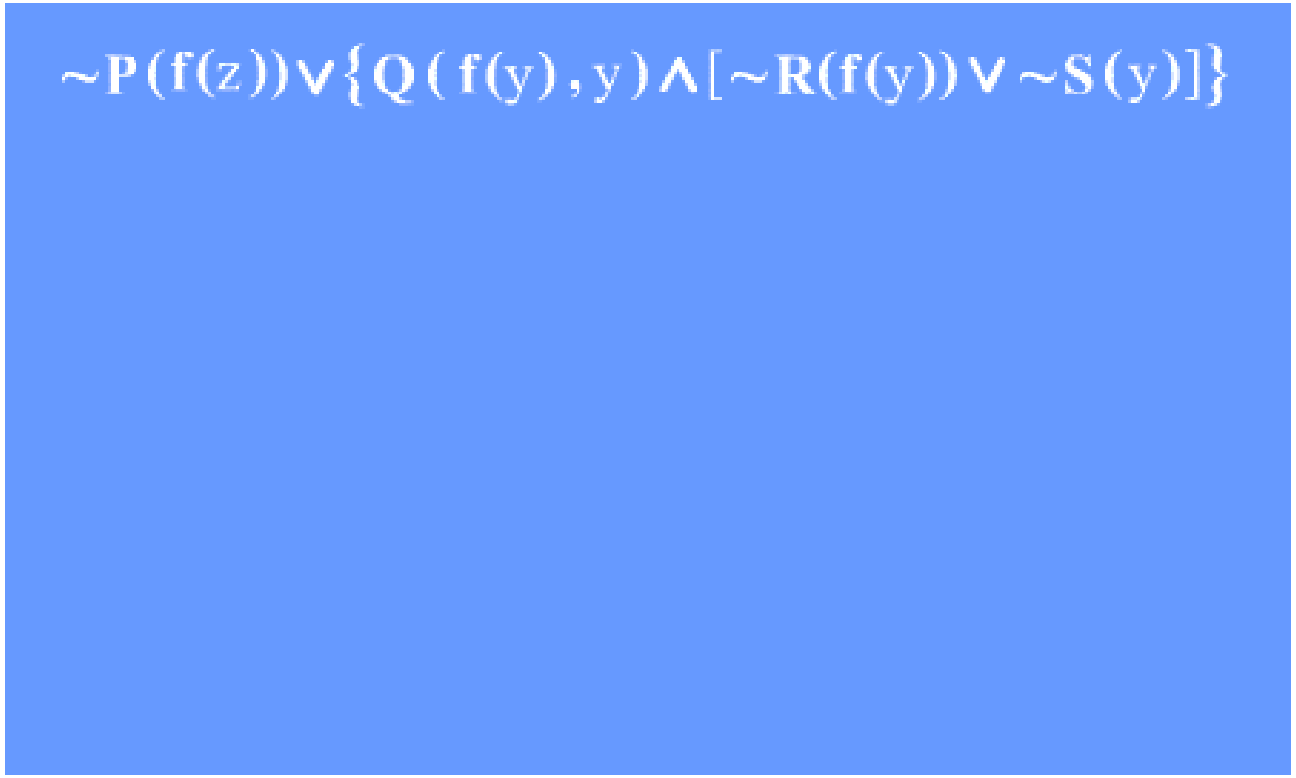
Here, f(y) is a Skolem function.

Separated and standardized variables in main conjunction form of goal:

$$\sim P(f(z)) \vee \{Q(f(y),y) \wedge [\sim P(f(y)) \vee \sim S(y)]\}$$

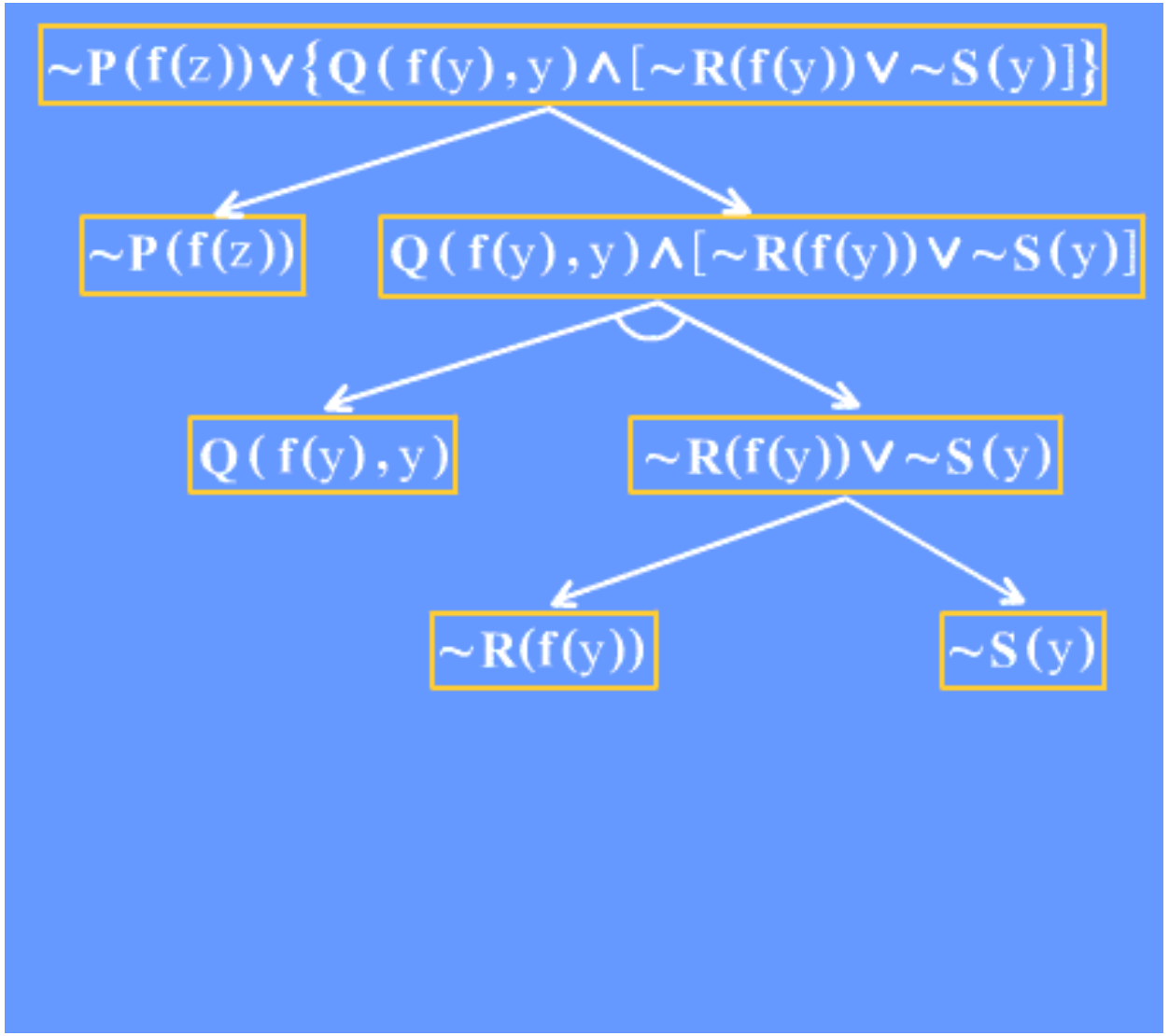
# 规则逆向演绎系统

- （2）AND/OR graph of goal expression
  - Different with AND/OR graph of facts expression, k-line connector was used to depart conjunctive sub-expression for goal expression.



# 规则逆向演绎系统

The clause set of goal expression can be read from AND/OR graph:




Goal clauses are literal conjunction, and that disjunction of the clauses form the goal formula.

# 规则逆向演绎系统

- （3）Applied B rule

  B rule is restricted as: $W \Rightarrow L$

  - W is arbitrary AND/OR form, L is literal.
  - Moreover, implication formulae liked $W \Rightarrow (L1 \wedge L2)$ can be transformed as two rules $W \Rightarrow L1$ and $W \Rightarrow L2$

- （4）Consistent solving graph of fact node as termination condition

  - Successful termination condition of backward system is that AND/OR graph includes fact nodes.

# 规则逆向演绎系统

**Fact**：

F1: DOG(FIDO)； The dog’s name is Fido

F2: ～BARKS(FIDO)； Fido does not bark

F3: WAGS TAIL(FIDO)； Fido wags its tail

F4: MEOWS(MYRTLE)； Meows is Myrtle

**Rules**：

R1: [WAGS TAIL(x1) ∧ DOG(x1)] ⇒ FRIENDLY(x1)；

The dog who wags its tail is a friendly dog.

R2: [FRIENDLY(x2) ∧ ～BARKS(x2)] ⇒ ～AFRAID(y2,x2)；

We should not be afraid the thing who is friendly and non-barking.

R3: DOG(x3) ⇒ ANIMAL(x3)； Dogs are animals

R4: CAT(x4) ⇒ ANIMAL(x4)； Cats are animals

R5: MEOWS(x5) ⇒ CAT(x5)； Meows is a cat

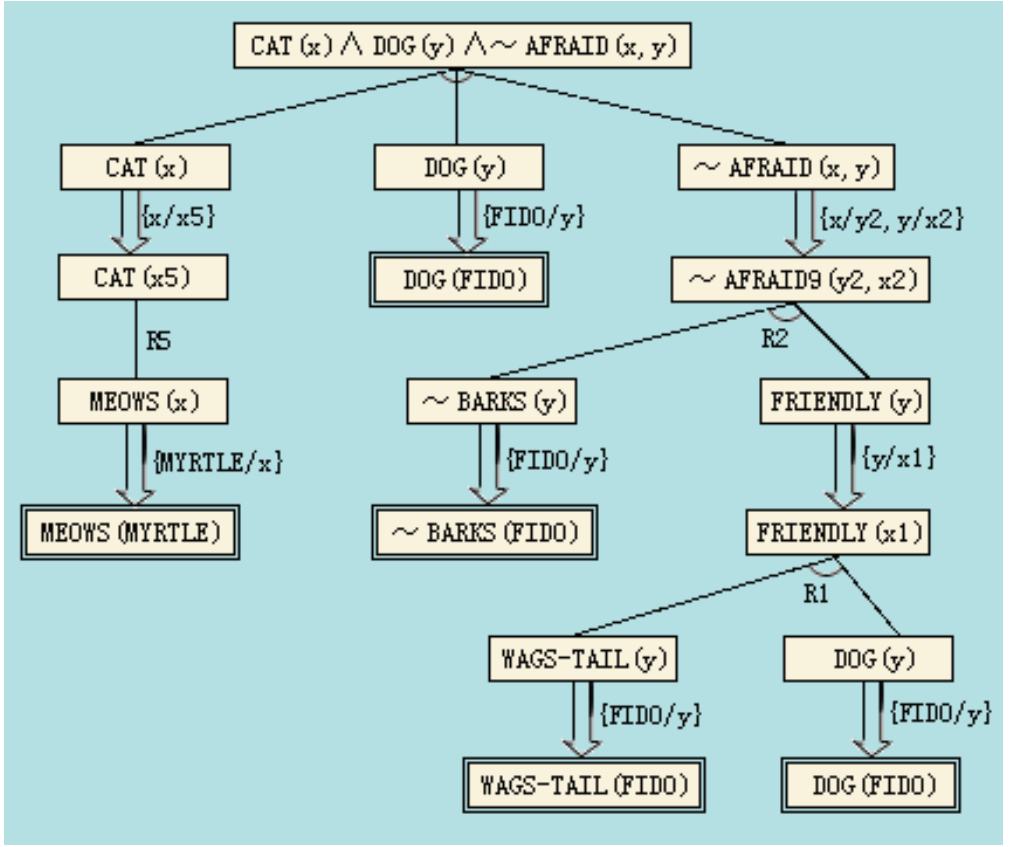
**Question**： Is there a cat and a dog, and the cat is not afraid the dog?

# 规则逆向演绎系统

用目标表达式表示此问题为：

$$(\exists x)(\exists y)[CAT(x) \wedge DOG(y) \wedge \sim AFRAID(x, y)]$$

用双线框表示事实节点，用规则编号R1、R2和R5等来标记所应用的规则。此解图中有八条匹配弧，每条匹配弧上都有一个置换。这些置换为{x/x5}，{MYRTLE/x}，{FIDO/y}，{x/y2,y/x2}，{FIDO/y} ({FIDO/y}重复使用四次)。由图可见，终止在事实节点前的置换为{MYRTLE/x}和{FIDO/y}。把它应用到目标表达式，我们就得到该问题的回答语句如下：




$$[CAT(MYRTLE) \wedge DOG(FIDO) \wedge \sim AFRAID(MYRTLE, FIDO)]$$

# 规则逆向演绎系统

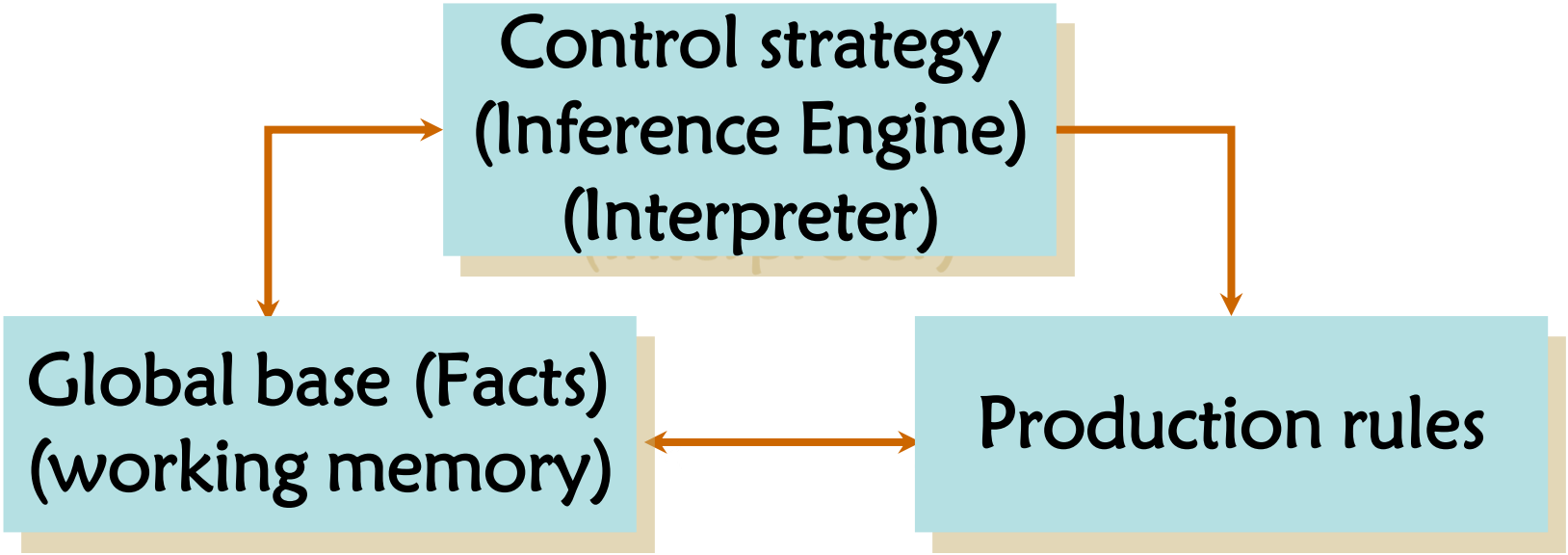
- 逆向系统中的事实表达式均限制为文字合取形
- 当一个事实文字和标在该图文字节点上的文字相匹配时，就可把相应的后裔事实节点添加到该与或图中去。这个事实节点通过标有mgu的匹配弧与匹配的子目标文字节点连接起来。同一个事实文字可以多次重复使用(每次用不同变量)，以便建立多重事实节点。
- 逆向系统成功的终止条件是与或图包含有某个终止在事实节点上的一致解图

# 产生式系统
# （Production System）

# 产生式系统(Production System)

- **Definition：**
  - Also called Rule-based System.
  - Proposed by Post in 1943. DENDRAL
  - Describes several different things that based on a same basic concept.
- **Essential：** Knowledge separated 2 parts
  - facts represented static knowledge, exp: object, event and relation between them;
  - production rules represented inference process and action.

# 产生式系统的组成


Control strategy
(Inference Engine)
(Interpreter)
Global base (Facts)
(working memory)
Production rules

# Production-rule System

- 总数据库(Global Database )
  - 又称综合数据库、上下文、黑板等
  - 存放求解过程中各种当前信息的数据，如：问题的初始状态、事实或证据、中间推理结论和结果等。
- 产生式规则（规则库Rule base）
  - 存放于求解问题相关的某个领域知识的规则集合
  - 完整性、一致性、准确性、灵活性和合理性
- 控制策略（推理机Inference Engine ）
  - 由一组程序组成，用来控制产生式系统的运行

# 控制策略的主要任务

① 按一定策略从规则库中选择与总数据库中的已知事实相匹配的规则，即把所选规则的前提与总数据库中的已知事实进行比较，若事实与所选规则前提一致，则匹配成功，该规则可被使用；否则，匹配失败，该规则不可用于当前推理。

② 当存在多条匹配成功的规则时，控制策略能够按照某种策略从中选出一条合适的规则去执行。

③ 如果要执行规则的右部不是问题的目标，且为一个或多个结论时，则把这些结论加入到总数据库中；当其为一个或多个操作时，执行这些操作。

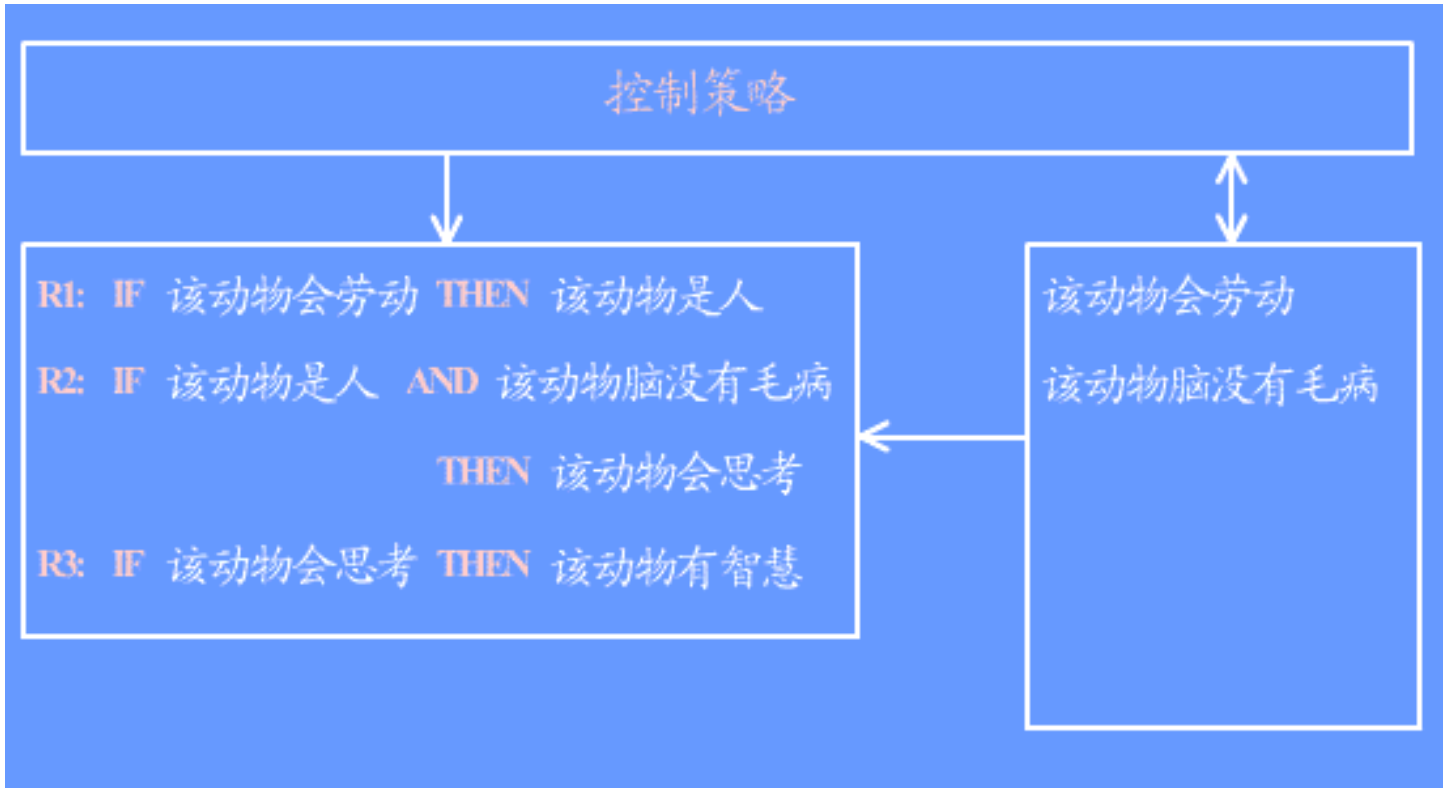
④ 如果要执行规则的右部满足问题的结束条件时，则停止推理。

⑤ 记住问题求解过程应用过的规则序列，以便求解结束时能够给出问题的解答路径。

# Production-rule System

- 选择规则到执行操作的步骤
  - 匹配
    - 把当前数据库与规则的条件部分相匹配。
  - 冲突
    - 当有一条以上规则的条件部分和当前数据库相匹配时，就需要决定首先使用哪一条规则，这称为冲突解决。
  - 操作
    - 执行规则的操作部分

# Example:


控制策略
R1: IF 该动物会劳动 THEN 该动物是人
R2: IF 该动物是人 AND 该动物脑没有毛病
THEN 该动物会思考
R3: IF 该动物会思考 THEN 该动物有智慧
该动物会劳动
该动物脑没有毛病

# 产生式系统的推理(Inference of Producton Sys.)

- Forward Inference
- Backward Inference
- Bidirectional Inference

# An example production system

- You want a program that can answer questions and make inferences about food items
- Like:
  - What is purple and perishable?
  - What is packed in small containers?
  - What is green and weighs 5 kg?

# A simple production rule system making inferences about food

**WORKING MEMORY (WM)**
Initially WM = (green, weighs-5-kg)

**RULE BASE**

R1: **IF** green **THEN** produce
R2: **IF** packed-in-small-container **THEN** delicacy
R3: **IF** [refrigerated **OR** produce ] **THEN** perishable
R4: **IF** [weighs-5-kg **AND** inexpensive **AND NOT** perishable] **THEN** staple
R5: **IF** [weighs-5-kg **AND** produce] **THEN** watermelon

**INTERPRETER**

1. Find all productions whose condition parts are true
2. Deactivate productions that would add a duplicate symbol
3. Execute the lowest numbered production (or quit)
4. Repeat until there is no rule to execute

# First cycle of execution

**WORKING MEMORY**

WM = (green, weighs-5-kg)

**CYCLE 1**

1. Productions whose condition parts are true: ***R1***
2. No production would add duplicate symbol
3. Execute ***R1***.

   This gives: WM = (***produce,*** green, weighs-5-kg)

**RULE BASE**

R1: **IF** green **THEN** produce
R2: **IF** packed-in-small-container **THEN** delicacy
R3: **IF** [refrigerated **OR** produce ] **THEN** perishable
R4: **IF** [weighs-5-kg **AND** inexpensive **AND** **NOT** perishable]
**THEN** staple
R5: **IF** [weighs-5-kg **AND** produce] **THEN** watermelon

**INTERPRETER**

1. Find all productions whose condition parts are true
2. Deactivate productions that would add a duplicate symbol
3. Execute the lowest numbered production (or quit)
4. Repeat

# Second cycle of execution

**WORKING MEMORY**

WM = (produce, green, weighs-5-kg)

**CYCLE 2**

1. Productions whose condition parts are true: ***R1, R3, R5***
2. Production R1 would add duplicate symbol, so ***deactivate R1***
3. Execute ***R3*** because it is the lowest numbered production.

   This gives: WM = (***perishable,*** produce, green, weighs-5-kg)

**RULE BASE**

R1: **IF** green **THEN** produce
R2: **IF** packed-in-small-container **THEN** delicacy
R3: **IF** [refrigerated **OR** produce ] **THEN** perishable
R4: **IF** [weighs-5-kg **AND** inexpensive **AND NOT** perishable]
**THEN** staple
R5: **IF** [weighs-5-kg **AND** produce] **THEN** watermelon

**INTERPRETER**

1. Find all productions whose condition parts are true
2. Deactivate productions that would add a duplicate symbol
3. Execute the lowest numbered production (or quit)
4. Repeat

# Third cycle of execution

**WORKING MEMORY**

WM = (perishable, produce, green, weighs-5-kg)

**CYCLE 3**

1. Productions whose condition parts are true: ***R1, R3, R5***
2. Productions ***R1, R3*** would add duplicate symbol, so deactivate them
3. Execute ***R5***.

This gives: WM = (***watermelon***, perishable, produce, green, weighs-5-kg)

**RULE BASE**

R1: **IF** green **THEN** produce
R2: **IF** packed-in-small-container **THEN** delicacy
R3: **IF** [refrigerated **OR** produce ] **THEN** perishable
R4: **IF** [weighs-5-kg **AND** inexpensive **AND NOT** perishable]
**THEN** staple
R5: **IF** [weighs-5-kg **AND** produce] **THEN** watermelon

**INTERPRETER**

1. Find all productions whose condition parts are true
2. Deactivate productions that would add a duplicate symbol
3. Execute the lowest numbered production (or quit)
4. Repeat

# Fourth cycle of execution

**WORKING MEMORY**

WM = (watermelon, perishable, produce, green, weighs-5-kg)

**CYCLE 4**

1. Productions whose condition parts are true: ***R1, R3, R5***
2. Productions ***R1, R3, R5*** would add duplicate symbol, so ***deactivate them***
3. ***Quit.***

***What are the conclusions?***

**RULE BASE**

R1: **IF** green **THEN** produce
R2: **IF** packed-in-small-container **THEN** delicacy
R3: **IF** [refrigerated **OR** produce ] **THEN** perishable
R4: **IF** [weighs-5-kg **AND** inexpensive **AND NOT** perishable]
**THEN** staple
R5: **IF** [weighs-5-kg **AND** produce] **THEN** watermelon

**INTERPRETER**

1. Find all productions whose condition parts are true
2. Deactivate productions that would add a duplicate symbol
3. Execute the lowest numbered production (or quit)
4. Repeat

Section 2:

# 不确定性推理

## 不确定性的类型

按性质划分，不确定性大致可分为随机性、模糊性、不完全性、不一致性和时变性等几种类型

① 随机性：随机性就是一个命题(亦即所表示的事件)的真实性不能完全肯定，而只能对其为真的可能性给出某种估计。例如，

如果乌云密布并且电闪雷鸣，则很可能要下暴雨。
如果头痛发烧，则大概是患了感冒。

② 模糊性：模糊性就是一个命题中所出现的某些言词，从概念上讲，无明确的内涵和外延，即是模糊不清的。例如，

小王是个高个子。
张三和李四是好朋友。
如果向左转，则身体就向左稍倾。

## 不确定性的类型

按性质划分，不确定性大致可分为随机性、模糊性、不完全性、不一致性和时变性等几种类型

③ 不完全性：不完全性就是对某事物来说，关于它的信息或知识还不全面、不完整、不充分。

④ 不一致性：不一致性就是在推理过程中发生了前后不相容的结论；或者随着时间的推移或者范围的扩大，原来一些成立的命题变得不成立、不适合了。

## 不确定性推理方法

① **模型方法**是对确定性推理框架的一种扩展。

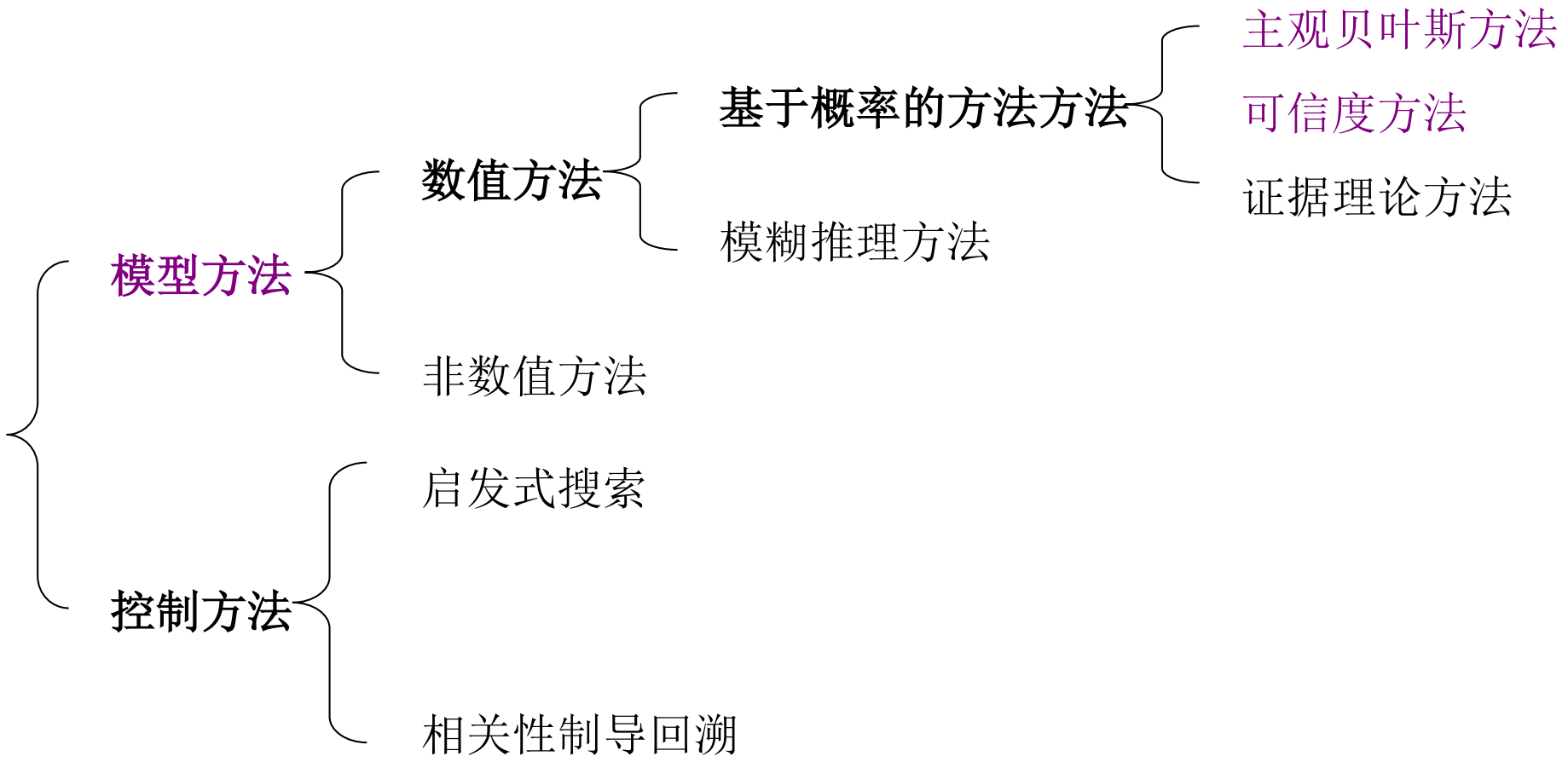
模型方法把不确定性证据和不确定性知识分别与某种度量标准对应起来，给出了更新结论不确定的算法，从而构成相应的不确定性推理的模型。

模型方法又分为数值方法和非数值方法两种，数值方法就是对不确定性进行定量表示和处理的方法。非数值方法是指除数值方法以外的其他各种处理不确定性的方法。

② **控制方法**主要在控制策略一级来处理不确定性。

控制方法通过识别领域中引起不确定性的某些特征及相应的控制策略来限制或者减少不确定性对系统产生的影响。这类方法没有处理不确定性的统一模型，其效果极大地依赖于控制策略。目前常用的控制方法有启发式搜索、相关性制导回溯和机缘控制等。

# 不确定性推理方法


模型方法
数值方法
基于概率的方法方法
主观贝叶斯方法
可信度方法
证据理论方法
模糊推理方法
非数值方法
控制方法
启发式搜索
相关性制导回溯

Section:

# 概率推理

## Probabilistic Reasoning

概率推理是一种使用较多的不确定性推理模型

## 1、概率的基本性质和计算公式

令$A$表示一个事件，则其概率记为$\mathrm{P}(A)$。

① 对于任一事件$A$，有

$$0 \le \mathrm{P}(A) \le 1$$

② 必然事件D的概率$\mathrm{P}(\mathrm{D}) = 1$，不可能事件$\Phi$的概率$\mathrm{P}(\Phi) = 0$

③ 若$A, B$是两个事件，则

$$\mathrm{P}(A \cup B) = \mathrm{P}(A) + \mathrm{P}(B) - \mathrm{P}(A \cap B)$$

④ 若事件$A_1, A_2, \cdots, A_{\mathrm{k}}$是两两互不相容（或称互斥）的事件，则有

$A_i \cap A_j = \emptyset (i \ne j)$，则

$$\mathrm{P}\left(\bigcup_{i=1}^{\mathrm{k}} A_i\right) = \mathrm{P}(A_1) + \mathrm{P}(A_2) + \cdots + \mathrm{P}(A_k)$$

若事件$A, B$互斥，则$\mathrm{P}(A \cup B) = \mathrm{P}(A) + \mathrm{P}(B)$

## 1、概率的基本性质和计算公式

⑤ 若$A,B$是两个事件，且$A \supset B$（表示事件$B$ 的发生必然导致事件$A$的发生），则

$$\mathrm{P}(A \backslash B) = \mathrm{P}(A) - \mathrm{P}(B)$$

其中事件$A \backslash B$事件$A$发生而事件$B$ 不发生。

⑥ 对任一事件$A$，有

$$\mathrm{P}(\overline{A}) = 1 - \mathrm{P}(A)$$

$\overline{A}$表示事件$A$的逆，即事件$A$和事件$\overline{A}$有且仅有一个发生

## 1、概率的基本性质和计算公式

令$A$表示一个事件，则其概率记为$\mathrm{P}(A)$。

① 条件概率与乘法公式

在事件$B$发生的条件下，事件$A$发生的概率称为事件$A$在事件$B$已发生的条件下的条件概率，记作$\mathrm{P}(A|B)$。当$\mathrm{P}(B)>0$时，规定

$$\mathrm{P}(A|B)=\frac{\mathrm{P}(A\cap B)}{\mathrm{P}(B)}$$

当$\mathrm{P}(B)=0$时，规定$\mathrm{P}(A|B)=0$，由此得出乘法公式

$$\mathrm{P}(A\cap B)=\mathrm{P}(B)\mathrm{P}(A|B)=\mathrm{P}(A)\mathrm{P}(B|A)$$

$$\mathrm{P}(A_1A_2\cdots A_\mathrm{n})=\mathrm{P}(A_1)\mathrm{P}(A_2|A_1)\mathrm{P}(A_3|A_1A_2)\cdots\mathrm{P}(A_\mathrm{n}|A_1A_2\cdots A_{n-1})$$

其中$\mathrm{P}(A_1A_2\cdots A_{n-1})>0$

## 1、概率的基本性质和计算公式

令$A$表示一个事件，则其概率记为$\mathrm{P}(A)$。

② 独立性公式

若事件$A$与$B$满足$\mathrm{P}(A|B)=\mathrm{P}(A)$，则称事件$A$关于事件$B$是独立的，独立性是相互的性质，即$A$关于$B$独立，$B$也一定关于$A$独立，或称$A$与$B$相互独立。

$$\mathrm{P}(A\cap B)=\mathrm{P}(A)\mathrm{P}(B)$$

③ 全概率公式

若事件$\mathrm{B}_1,\mathrm{B}_2,\cdots$满足$\mathrm{B}_i\cap\mathrm{B}_j=\emptyset(i\neq j)$，$\mathrm{P}\left(\bigcup_{i=1}^{\infty}\mathrm{B}_i\right)=1,\mathrm{P}(\mathrm{B}_i)>0,\mathrm{i}=1,2,\cdots$

则对于任一事件$A$，有

$$\mathrm{P}(A)=\sum_{i}^{\infty}\mathrm{P}(A|B_i)\mathrm{P}(B_i)$$

若$\mathrm{B}_i$只有$n$个，则此公式也成立，此时右端只有$n$项相加。

## 1、概率的基本性质和计算公式

令$A$表示一个事件，则其概率记为$\mathrm{P}(A)$。

④ 贝叶斯(Bayes)公式

事件$\mathrm{B}_1, \mathrm{B}_2, \cdots$满足全概率公式条件，则对于任一事件$A(\mathrm{P}(A) > 0)$，有

$$\mathrm{P}(B_i|A) = \frac{\mathrm{P}(B_i)\mathrm{P}(A|B_i)}{\sum_{i=1}^{\infty}\mathrm{P}(B_i)\mathrm{P}(A|B_i)}$$

若$\mathrm{B}_i$只有$n$个，则此公式也成立，此时右端只有$n$项相加。

## *、先验概率与后验概率

- 先验概率（prior probability）是指根据以往经验和分析得到的概率，如全概率公式中的$\mathrm{P}(B_i)$，它往往作为“由因求果”问题中的“因”出现的概率。
- 后验概率是指在得到“结果”的信息后重新修正的概率，如贝叶斯公式中的$\mathrm{P}(B_i|A)$，是“执果寻因”问题中的"果"。
- 先验概率与后验概率有不可分割的联系，后验概率的计算要以先验概率为基础。

## 2、概率推理方法

有如下产生式规则：　　$IF\ E\ THEN\ H$，　则证据（或前提条件）$E$不确定性的概率为$\mathrm{P}(E)$。概率方法不精确推理的目的就是求出在证据$E$下结论$H$发生的概率$\mathrm{P}(H|E)$

(1:1) 贝叶斯方法的原始条件：已知前提$E$的概率$\mathrm{P}(E)$和$H$的先验概率$\mathrm{P}(H)$，并已知$H$成立时$E$出现的条件概率$\mathrm{P}(E|H)$。

$$\mathrm{P}(H|E)=\frac{\mathrm{P}(H)\mathrm{P}(E|H)}{\mathrm{P}(E)}$$

(1:n) 若一个证据$E$ 支持多个假设$H_1,H_2,\cdots H_n$，即$IF\ E\ THEN\ H_i$则可得如下贝叶斯公式

$$\mathrm{P}(H_i|E)=\frac{\mathrm{P}(H_i)\mathrm{P}(E|H_i)}{\sum_{j=1}^{n}\mathrm{P}(H_j)\mathrm{P}(E|H_j)},i=1,2,\cdots n$$

## 2、概率推理方法

(m:n) 若有多个证据$E_1, E_2, \cdots, E_m$和多个结论$H_1, H_2, \cdots, H_n$，并且每个证据都以一定程度支持结论，则

$$\mathrm{P}(H_i|E_1E_2E_m) = \frac{\mathrm{P}(E_1|H_i)\mathrm{P}(E_2|H_i)\cdots\mathrm{P}(E_m|H_i)\mathrm{P}(H_i)}{\sum_{j=1}^{n}\mathrm{P}(E_1|H_j)\mathrm{P}(E_2|H_j)\cdots\mathrm{P}(E_m|H_j)\mathrm{P}(H_j)}$$

这时，只要已知$H_i$的先验概率$\mathrm{P}(H_i)$及$H_i$成立时证据$E_1, E_2, \cdots, E_m$出现的条件概率$\mathrm{P}(E_1|H_i)$，$\mathrm{P}(E_2|H_i)$，…，$\mathrm{P}(E_m|H_i)$，就可以利用上述公式计算出在$E_1, E_2, \cdots, E_m$出现的情况下的$H_i$条件概率$\mathrm{P}(H_i|E_1E_2E_m)$。

## 例1、

设$H_1, H_2, H_3$为3个结论，$E$是支持这些结论的证据，且已知：

$$P(H_1) = 0.3, \quad P(H_2) = 0.4, \quad P(H_3) = 0.5$$

$$P(E|H_1) = 0.5, \ P(E|H_2) = 0.3, \ P(E|H_3) = 0.4$$

求：$P(H_1|E)$，$P(H_2|E)$及$P(H_3|E)$的值。

**解**：根据(1:n) 贝叶斯公式可得：

$$\mathrm{P}(H_1|E) = \frac{\mathrm{P}(H_1)\mathrm{P}(E|H_1)}{\sum_{j=1}^{3}\mathrm{P}(H_j)\mathrm{P}(E|H_j)} = \frac{0.3\times0.5}{0.3\times0.5+0.4\times0.3+0.5\times0.4}=0.32$$

$$\mathrm{P}(H_2|E)=0.26$$

$$\mathrm{P}(H_3|E)=0.43$$

计算结果表明，由于证据$E$的出现，$H_1$成立的可能性略有增加，而$H_2, H_3$成立的可能性却有不同程度的下降。

## 例2、

已知：$P(H_1)=0.4,\quad P(H_2)=0.3,\quad P(H_3)=0.3$

$P(E_1|H_1)=0.5,\quad P(E_1|H_2)=0.6,\quad P(E_1|H_3)=0.3$

$P(E_2|H_1)=0.7,\quad P(E_2|H_2)=0.9,\quad P(E_2|H_3)=0.1$

求：$P(H_1|E_1E_2)$，$P(H_2|E_1E_2)$及$P(H_3|E_1E_2)$的值。

**解**：根据(m:n) 贝叶斯公式可得：

$$\mathrm{P}(H_1|E_1E_2)=\frac{\mathrm{P}(E_1|H_1)\mathrm{P}(E_2|H_1)\mathrm{P}(H_1)}{\sum_{j=1}^{3}\mathrm{P}(E_1|H_j)\mathrm{P}(E_2|H_j)\mathrm{P}(H_j)}=0.45$$

$$\mathrm{P}(H_2|E_1E_2)=0.52$$

$$\mathrm{P}(H_3|E_1E_2)=0.03$$

计算结果表明，由于证据$E_1E_2$的出现，$H_1$和$H_2$成立的可能性略有增加，而$H_3$成立的可能性却有不同程度的下降。

## 3、主观贝叶斯方法

问题提出：直接用贝叶斯公式求取$P(H_i|E)$，需要已知$\mathrm{P}(H_i)$及$\mathrm{P}(E|H_i)$。对于实际应用，这时难以做到的。Duda和Hart等人基于贝叶斯公式，于1976年提出主观贝叶斯方法，建立了不精确推理模型，并成功应用。

**3.1 知识不确定性表示**：用产生式规则：　　$IF\ E\ THEN\ (LS, LN)\ H$　表示知识

式中$(LS, LN)$表示知识的静态强度，$LS$称为“充分性因子”，$LN$称为“必要性因子”，分别衡量$E$对$H$的支持程度和$\sim E$对$H$的支持程度。$LS$和$LN$的取值范围为$[0, +\infty)$，其具体数值由领域专家决定。

$$LS = \frac{\mathrm{P}(E|H)}{\mathrm{P}(E|\sim H)} \qquad LN = \frac{\mathrm{P}(\sim E|H)}{\mathrm{P}(\sim E|\sim H)} = \frac{1 - \mathrm{P}(E|H)}{1 - \mathrm{P}(E|\sim H)}$$

主观贝叶斯方法的不精确推理过程就是根据前提$E$的概率$\mathrm{P}(E)$，利用规则的$LS$和$LN$，把结论$H$的先验概率$\mathrm{P}(H)$更新为后验概率$\mathrm{P}(H|E)$的过程。

## 3.1 知识不确定性表示

定义概率函数：

$$O(X)=\frac{P(X)}{1-P(X)}=\frac{P(X)}{P(\sim X)}$$

概率函数可以把取值在$[0,1]$的$P(X)$放大为取值在$[0,+\infty)$的$O(X)$

$$P(X)=\frac{O(X)}{1+O(X)}$$

主观贝叶斯公式：

$$O(H|E)=LS\cdot O(H);\qquad O(H|\sim E)=LN\cdot O(H)$$

当$E$为真时，可利用$LS$将$H$的先验几率$O(H)$更新为其后验几率$O(H|E)$；当$E$为假时，可利用$LN$将$H$的先验几率$O(H)$更新为其后验几率$O(H|\sim E)$。

## 3.2 证据不确定性的表示

引入可信度的概念，让用户在−5~5之间的11个数中根据实际情况选一个数作为初始证据的可信度C(*E*|*S*)，表示对所提供证据可以相信的程度，其中*E* 为证据， S为观察：

$$P(E|S)=\begin{cases}\dfrac{C(E|S)+P(E)\times(5-C(E|S))}{5} & ，若0\le \mathrm{C}(E|S)\le 5\\ \dfrac{P(E))\times(C(E|S)+5)}{5} & ，若-5\le \mathrm{C}(E|S)\le 0\end{cases}$$

$$C(E|S)=\begin{cases}5\times\dfrac{P(E|S)-P(E)}{1-P(E)} & ，若P(E)\le P(E|S)\le 1\\ 5\times\dfrac{P(E|S)-P(E)}{P(E)} & ，若0\le P(E|S)\le P(E)\end{cases}$$

## 3.2 证据不确定性的表示

当证据不确定时，计算H的后验概率$P(H|S)$

EH公式

$$P(H|S)=\begin{cases} P(H|\sim E)+\dfrac{P(H)-P(H|\sim E)}{P(E)}\times P(E|S) & ，若0\le P(E|S)<P(E) \\ P(H)+\dfrac{P(H|E)-P(H)}{1-P(E)}\times[P(E|S)-\mathrm{P}(E)] & ，若P(E)\le P(E|S)\le 1 \end{cases}$$

CP公式

$$P(H|S)=\begin{cases} P(H|\sim E)+[P(H)-P(H|\sim E)]\times[\dfrac{1}{5}C(E|S)+1] & ，若C(E|S)\le 0 \\ P(H)+[P(H|E)-P(H)]\times\dfrac{1}{5}C(E|S) & ，若C(E|S)>0 \end{cases}$$

## 3、主观贝叶斯方法——公式总结

当知识不确定时：

① 概率函数：给出$P(X)$和$O(X)$的关系

$$O(X)=\frac{P(X)}{1-P(X)} \qquad P(X)=\frac{O(X)}{1+O(X)}$$

② 主观贝叶斯公式：给出先验$O(H)$和后验$O(H|E)/O(H|\sim E)$的关系

$$O(H|E)=LS\cdot O(H); \qquad O(H|\sim E)=LN\cdot O(H)$$

当证据不确定时：

③ 可信度$\mathrm{C}(E|S)$与概率$\mathrm{P}(E|S)$的关系

④ EH公式：给出$\mathrm{P}(H|S)$和$\mathrm{P}(E|S)$的关系

⑤ CP公式：给出$\mathrm{P}(H|S)$和$\mathrm{C}(E|S)$的关系

## 3.3 主观贝叶斯方法的推理过程

当采用初始证据进行推理时，通过提问用户得到C$(E|S)$，通过$CP$公式就可求出P$(H|S)$。

当采用推理过程中得到的中间结论作为证据进行推理时，通过EH公式可求得P$(H|S)$。

如果有$n$条知识都支持同一结论$\boldsymbol{H}$，而且每条知识的前提条件分别是$n$个相互独立的证据$E_1, E_2, \cdots, E_n$，而这些论据又分别与观察$S_1, S_2, \cdots, S_n$相对应，这时，首先对每条知识分别求出$H$的后验几率O$(H|S_i)$，然后按下述公式求出所有观察下$H$的后验几率：

$$O(H|S_1, S_2, \cdots, S_n) = \frac{O(H|S_1)}{O(H)} \times \frac{O(H|S_2)}{O(H)} \times \cdots \times \frac{O(H|S_n)}{O(H)} \times O(H)$$

## 例3、

已知下列规则：

$R_1$：IF $E_1$ THEN $(2, 0.000001)$ $H_1$

$R_2$：IF $E_2$ THEN $(100, 0.000001)$ $H_1$

$R_3$：IF $H_1$ THEN $(65, 0.01)$ $H_2$

$R_4$：IF $E_3$ THEN $(300, 0.0001)$ $H_2$

先验概率 $O(H_1) = 0.1, O(H_2) = 0.01$

通过用户得到：

$$C(E_1|S_1) = 3, \qquad C(E_2|S_2) = 1, \qquad C(E_3|S_3) = -2$$

求：$O(H_2|S_1S_2S_3)$

# 3.4 主观贝叶斯方法的优缺点比较

优点：

- 主观贝叶斯方法的计算公式大多是在概率论的基础上推导出来的，具有比较坚实的理论基础；
- 规则的$LS$和$LN$是由领域专家根据实践经验给出的，避免了大量的数据统计工作。$LS$和$LN$比较全面地反映了证据与结论之间的因果关系，符合现实世界中某些领域的实际情况，使推出的结论具有比较准确的确定性。
- 主观贝叶斯方法不仅给出了在证据确定情况下由$H$的先验概率更新为后验概率的方法，而且还给出了在证据不确定情况下更新先验概率为后验概率的方法。由其推理过程还可以看出，它确定实现了不确定性的逐级传递。因此，可以说主观贝叶斯方法是一种比较实用而又灵活的不确定性推理方法，已成功的应用在专家系统里。

缺点：

- 该方法要求领域专家在给出规则的同时，给出$H$的先验概率$P(H)$，这是比较困难的；
- 贝叶斯定理中关于事件间独立性的要求使主观贝叶斯方法的应用受到一定的限制。

## 4、可信度方法

可信度方法是Shortlifffe等人在确定性理论基础上结合概率论等理论提出的一种不精确推理模型，它对许多实际应用都是一个合理而有效地推理模式，在专家系统等领域获得较广泛的应用。

4.1 知识不确定性表示：用产生式规则： $IF\ E\ THEN\ H\ (\mathrm{CF}\ (H,E))$

式中$\mathrm{CF}\ (H,E)$表示规则的可信度，称为可信度因子或规则强度。$\mathrm{CF}\ (H,E)$的作用域为$[-1,1]$，$\mathrm{CF}\ (H,E)>0$则表示该证据增加了结论为真的程度，且$\mathrm{CF}\ (H,E)$的值越大，结论$H$越真。若$\mathrm{CF}\ (H,E)=1$，则表示该证据使结论为真。反之，若$\mathrm{CF}\ (H,E)<0$，则表示该证据增加了结论为假的程度，且$\mathrm{CF}\ (H,E)$的值越小，结论$H$越假。$\mathrm{CF}\ (H,E)=-1$表示该证据使结论为假。$\mathrm{CF}\ (H,E)=0$表示证据$E$和结论$H$没有关系

## 4.1 知识不确定性表示

定义1：

$$CF(H,E) = MB(H,E) - MD(H,E)$$

式中$MB$(measure belief)为信任增长度，表示因证据$E$的出现而增加对假设$H$为真的信任增加程度，即当$MB(H,E) > 0$时，有$P(H|E) > P(H)$。$MD$)measure disbelief)为不信任增长度，表示因证据$E$的出现而增加对假设$H$为假的信任增加程度，即当$MD(H,E) > 0$时，有$P(H|E) < P(H)$。

定义2：

$$\mathrm{M}B(H,E) = \begin{cases} 1, & \text{若} P(H) = 1 \\ \dfrac{\max\{P(H|E), P(H)\} - P(H)}{1 - P(H)}, & \text{else} \end{cases}$$

定义3：

$$\mathrm{M}D(H,E) = \begin{cases} 1, & \text{若} P(H) = 0 \\ \dfrac{\min\{P(H|E), P(H)\} - P(H)}{-P(H)}, & \text{else} \end{cases}$$

## 4.2 证据不确定性表示

在可信度方法中，证据$E$的不确定性用证据的可信度$\mathrm{CF}(E)$表示，初始证据的可信度由用户在系统运行时提供，中间结果的可信度由不精确推理算法求得。

证据$E$的可信度$\mathrm{CF}(E)$的取值范围与$\mathrm{CF}(H,E)$相同，即$[-1,1]$。当证据以某种程度为真时，$\mathrm{CF}(E)>0$；当证据肯定为真时，$\mathrm{CF}(E)=1$；当证据以某种程度为假时，$\mathrm{CF}(E)<0$；当证据肯定为假时，$\mathrm{CF}(E)=-1$；当证据一无所知时，$\mathrm{CF}(E)=0$。

## 4.3 可信度方法的推理算法

### a) 组合证据的不确定性算法

① 合取证据

当组合证据为多个单一证据的合取时： $\mathrm{E} = E_1\ AND\ E_2\ AND \cdots AND\ E_n$

若已知$\mathrm{CF}\ (E_1), \mathrm{CF}\ (E_2) \cdots \mathrm{CF}\ (E_n)$，则有

$$\mathrm{CF}\ (E) = \min\{\mathrm{CF}\ (E_1), \mathrm{CF}\ (E_2) \cdots \mathrm{CF}\ (E_n)\}$$

即对于多个证据合取的可信度，取其可信度最小的那个证据的CF值作为组合证据的可信度。

② 析取证据

当组合证据为多个单一证据的析取时： $\mathrm{E} = E_1\ OR\ E_2\ OR \cdots OR\ E_n$

若已知$\mathrm{CF}\ (E_1), \mathrm{CF}\ (E_2) \cdots \mathrm{CF}\ (E_n)$，则有

$$\mathrm{CF}\ (E) = \max\{\mathrm{CF}\ (E_1), \mathrm{CF}\ (E_2) \cdots \mathrm{CF}\ (E_n)\}$$

即对于多个证据析取的可信度，取其可信度最大的那个证据的CF值作为组合证据的可信度。

## 4.3 可信度方法的推理算法

### b) 不确定性的传递算法

不确定性的传递算法就是根据证据和规则的可信度求其结论的可信度。若已知规则为：IF $E$ $THEN$ $H$ $(CF(H,E))$

且证据$E$的可信度为$\mathrm{CF}(E)$，则结论$H$ 的可信度$\mathrm{CF}(H)$ 为：

$$\mathrm{CF}(H) = \mathrm{CF}(H,E)\max\{0, \mathrm{CF}(E)\}$$

当$\mathrm{CF}(E) > 0$，即证据以某种程度为真时，则$\mathrm{CF}(H) = \mathrm{CF}(H,E)\ \mathrm{CF}(E)$。若$\mathrm{CF}(E) = 1$，则证据为真，则$\mathrm{CF}(H) = \mathrm{CF}(H,E)$。这说明，当证据$E$为真时，结论$H$ 的可信度为规则的可信度。当$\mathrm{CF}(E) < 0$，即证据以某种程度为假时，规则不能使用时，则$\mathrm{CF}(H) = 0$。可见，在可信度方法的不精确推理中，并没有考虑证据为假对结论$H$ 产生的影响。

# 4.3 可信度方法的推理算法

## c) 多个独立证据推出同一假设的合成算法

如果两条不同规则推出同一结论，但可信度各不相同，则可用合成算法计算综合可信度。已知如下两条规则：

$$\text{IF} \quad E_1 \quad THEN \quad H \quad (CF(H,E_1))$$

$$\text{IF} \quad E_2 \quad THEN \quad H \quad (CF(H,E_2))$$

其结论$H$得综合可信度可按如下步骤求得：

① 根据不确定性的传递算法，可求出：

$$\text{CF}(H_1) = \text{CF}(H,E_1)\max\{0, \text{CF}(E_1)\}$$
$$\text{CF}(H_1) = \text{CF}(H,E_2)\max\{0, \text{CF}(E_2)\}$$

# 4.3 可信度方法的推理算法

## c) 多个独立证据推出同一假设的合成算法

② 求出$E_1$和$E_2$对$H$ 的综合影响所形成的可信度$\mathrm{CF}_{1,2}(H)$：

$$\mathrm{CF}_{1,2}(H)=\begin{cases}\mathrm{CF}_1(H)+\mathrm{CF}_2(H)-\mathrm{CF}_1(H)\cdot\mathrm{CF}_2(H) & ，若\mathrm{CF}_1(H)\geq 0，\ \mathrm{CF}_2(H)\geq 0\\ \mathrm{CF}_1(H)+\mathrm{CF}_2(H)+\mathrm{CF}_1(H)\cdot\mathrm{CF}_2(H) & ，若\mathrm{CF}_1(H)<0，\ \mathrm{CF}_2(H)<0\\ \mathrm{CF}_1(H)+\mathrm{CF}_2(H) & ，若\mathrm{CF}_1(H)\cdot\mathrm{CF}_2(H)<0\end{cases}$$

在MYCIN系统的基础上形成的专家系统工具EMYCIN中，对上式做了如下修改：

$$\mathrm{CF}_{1,2}(H)=\begin{cases}\mathrm{CF}_1(H)+\mathrm{CF}_2(H)-\mathrm{CF}_1(H)\cdot\mathrm{CF}_2(H) & ，若\mathrm{CF}_1(H)\geq 0，\ \mathrm{CF}_2(H)\geq 0\\ \mathrm{CF}_1(H)+\mathrm{CF}_2(H)+\mathrm{CF}_1(H)\cdot\mathrm{CF}_2(H) & ，若\mathrm{CF}_1(H)<0，\ \mathrm{CF}_2(H)<0\\ \dfrac{\mathrm{CF}_1(H)+\mathrm{CF}_2(H)}{1-\min\{|\mathrm{CF}_1(H)|,|\mathrm{CF}_2(H)|\}} & ，若\mathrm{CF}_1(H)\cdot\mathrm{CF}_2(H)<0\end{cases}$$

当组合两个以上的独立证据时，可首先组合其中的两个，再将其组合结果与第三个证据进行组合，如此继续进行组合，直至组合完成为止。

## 例4、

已知下列规则：

$R_1$：IF $E_1$ THEN $H$ (0.8)

$R_2$：IF $E_2$ THEN $H$ (0.6)

$R_3$：IF $E_3$ THEN $H$ (-0.5)

$R_4$：IF $E_4$ AND ( $E_5$ $OR$ $E_6$) THEN $E_1$(0.7)

$R_5$：IF $E_7$ AND $E_8$ THEN $E_3$(0.9)

通过用户得到：

$\mathrm{CF}(E_2) = 0.8$ $\mathrm{CF}(E_4) = 0.5$ $\mathrm{CF}(E_5) = 0.6$

$\mathrm{CF}(E_6) = 0.7$ $\mathrm{CF}(E_7) = 0.6$ $\mathrm{CF}(E_8) = 0.9$

求：H的综合可信度CF(H)

# Q&A

# THANKS!