3.6 Rule-based Deduction Systems

规则演绎系统

- 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 …

3.6 规则演绎系统

- 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)

3.6.1 规则正向演绎系统

- 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

求解步骤

- (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

(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

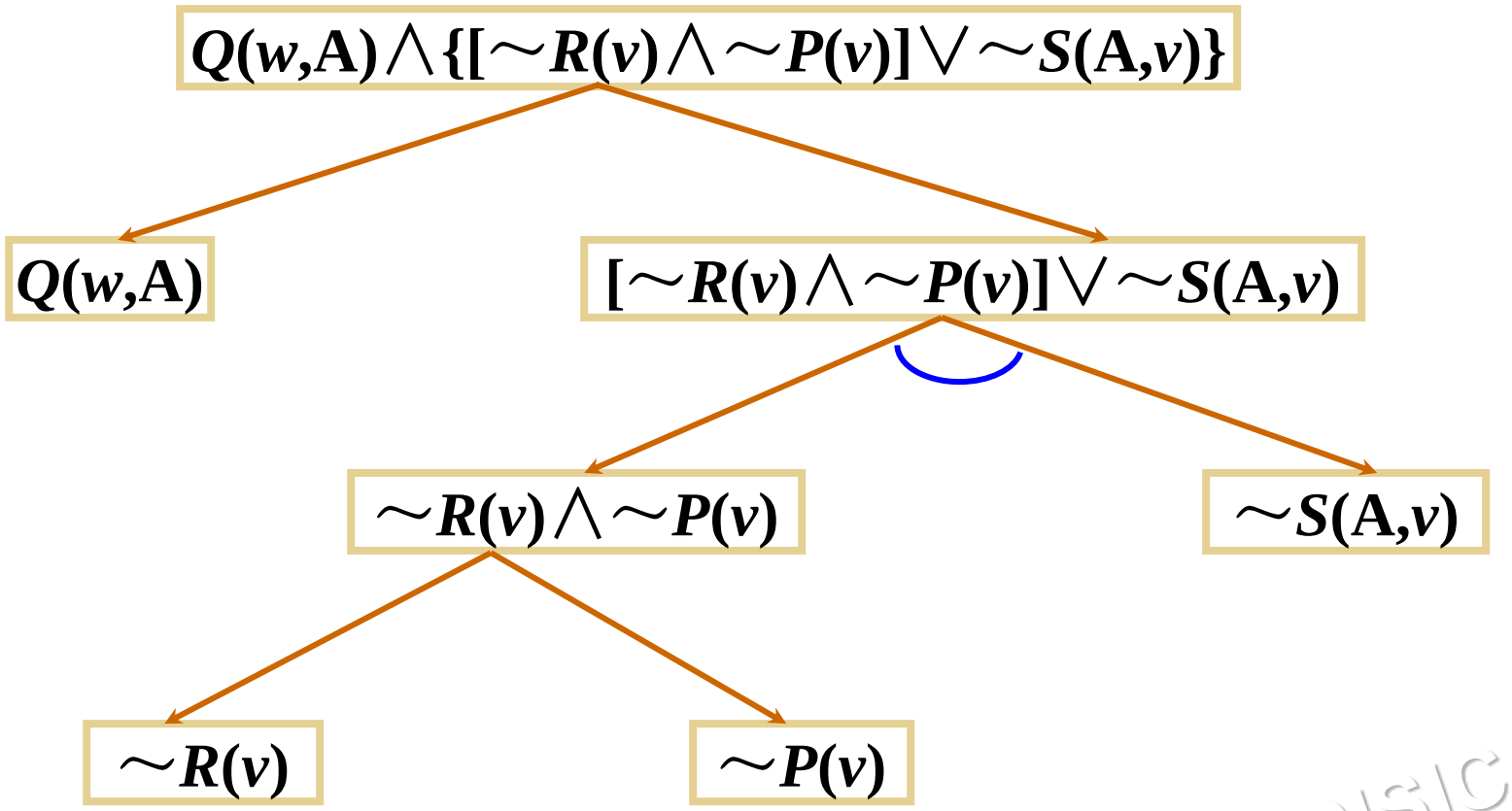
- Example:

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

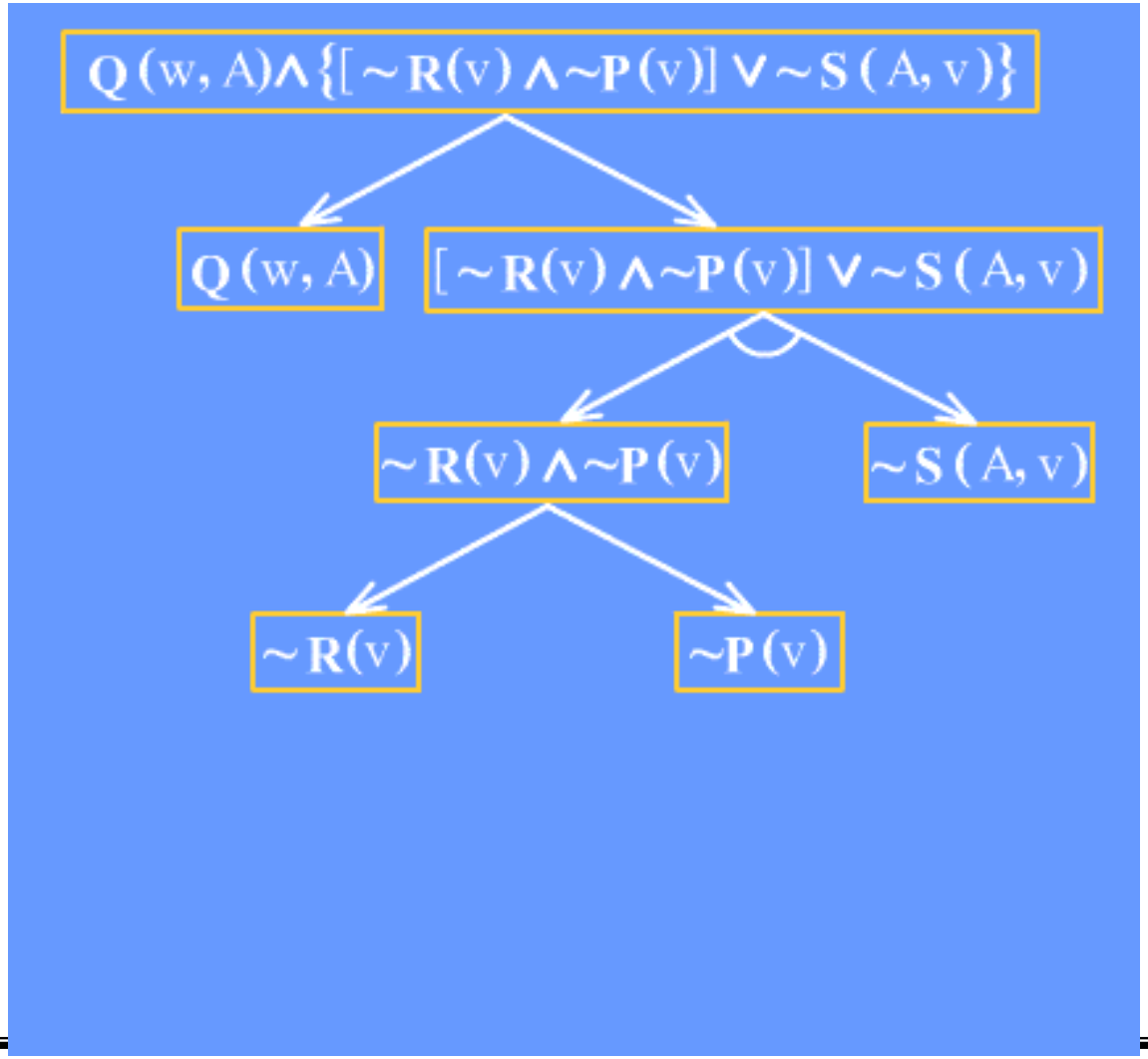
$$(\exists u)(\forall v)\{Q(v,u)\wedge\sim[(R(v)\vee P(v))\wedge S(u,v)]\}$$

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

(2) 事实表达式的与或树



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



(3) 与或图的F规则变换

$$L \Rightarrow W$$

这里L为单文字，W为与或形的唯一公式.

- 假设
  - 出现在蕴涵式中的任意变量都有**全称量化**作用于整个蕴涵式.
  - 这些事实和规则中的一些变量被分离标准化，使得没有一个变量出现在一个以上的规则中，而且使规则变量不同于事实变量.

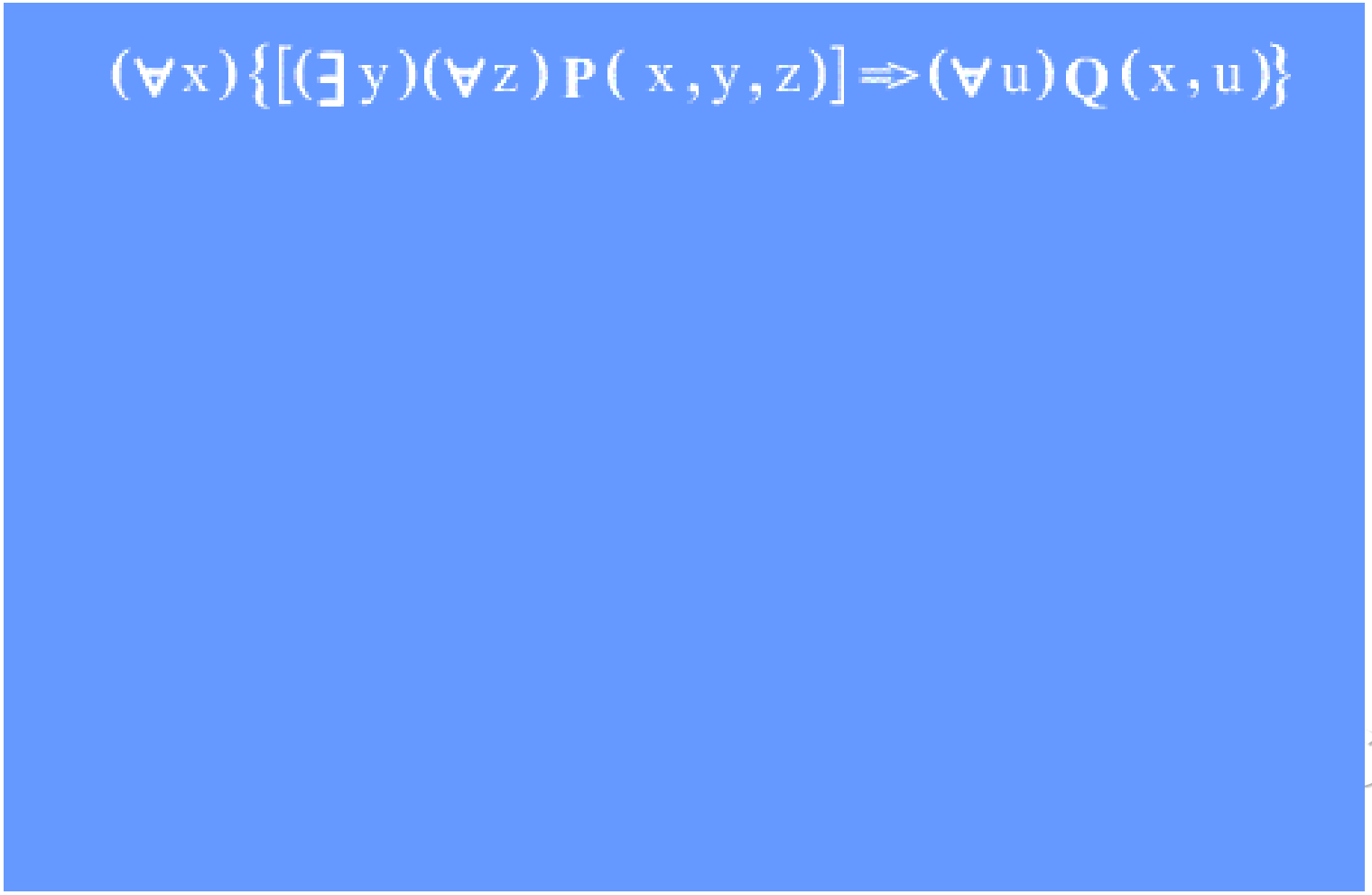
- F规则变换步骤
  - 暂时消去蕴涵符号.
  - 把否定符号移进第一个析取式内，调换变量的量词.
  - 进行Skolem化.
  - 把所有全称量词移至前面，然后消去.
  - 恢复蕴涵式.

  - For example:

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

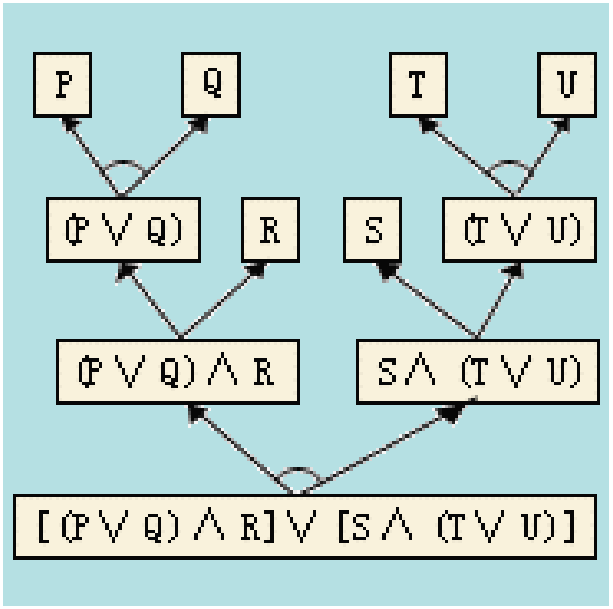
- For example:

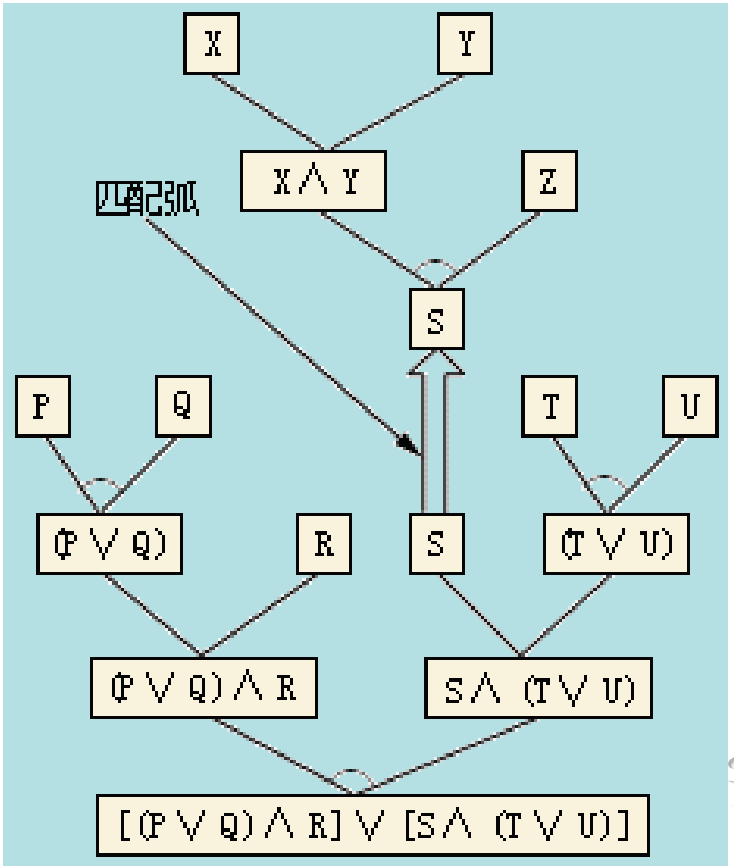
$(\forall x)\{[(\exists y)(\forall z)P(x, y, z)] \Rightarrow (\forall u)Q(x, u)\}$



F规则应用到与或树

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





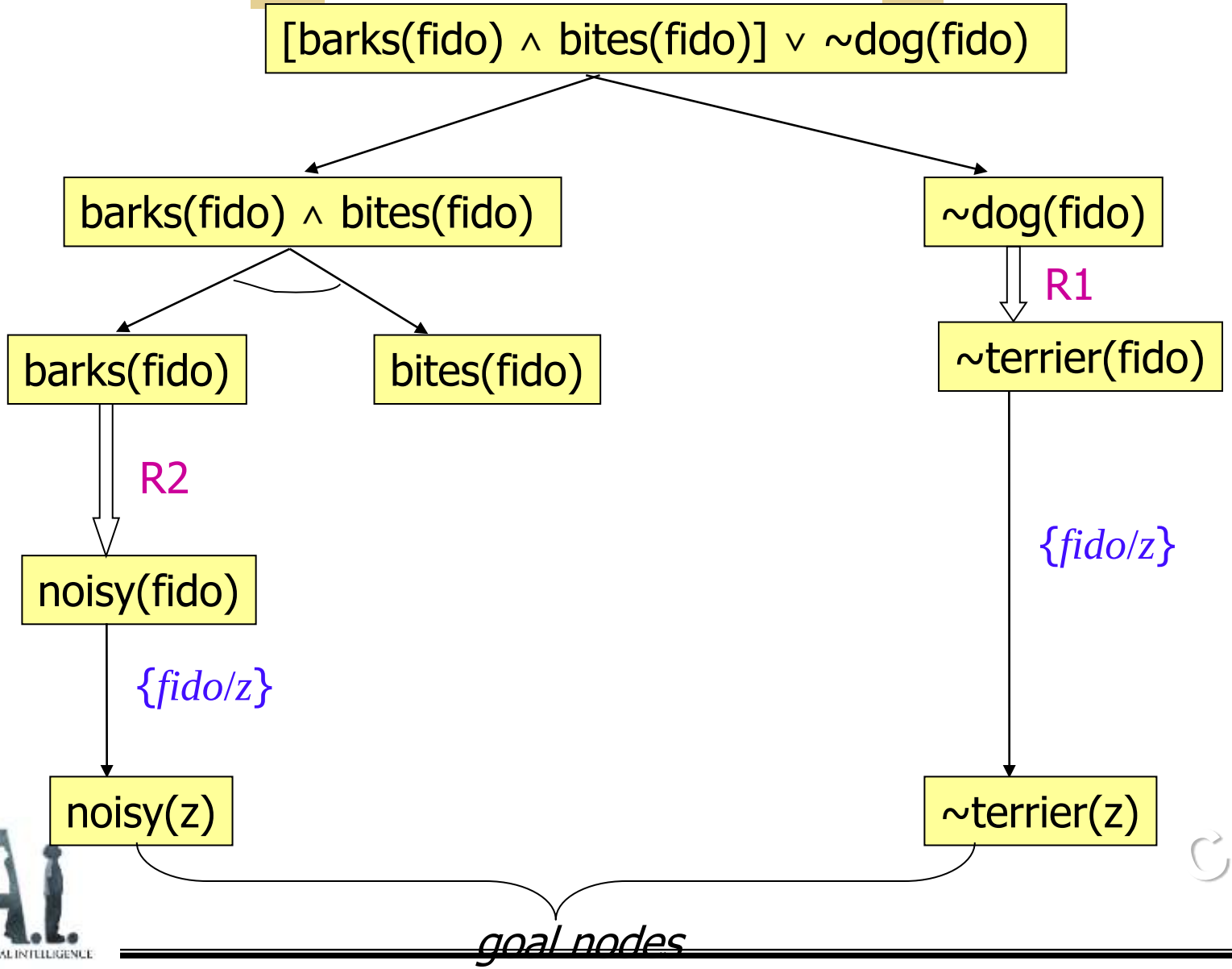
（4） 作为终止条件的目标公式

- 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) $\wedge$ bites(fido)) $\vee$ ~dog(fido)
  - R1: terrier(x) → dog(x)
  - R2: barks(y) → noisy(y)
  - goal: $\exists$w (~terrier (w) $\vee$ noisy(w))

From facts to goal



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。

3.6.2规则逆向演绎系统（Backward Rule-based Deduction Systems）

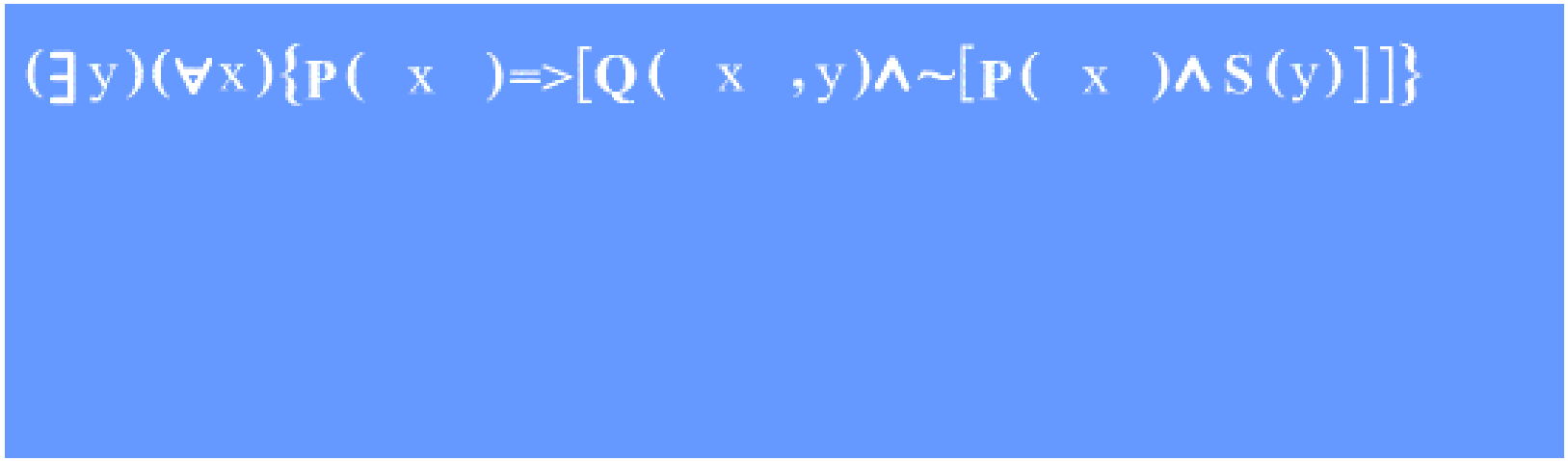
- Backward rule-based deduction systems are reasoning from THEN to IF (from *goal* or *action* to *fact* or *situation condition*).
- 步骤
  - (1)目标表达式的与或形式
  - (2)与或图的B规则变换 $W \Rightarrow L$
  - (3)作为终止条件的事实节点的一致解图

(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.

- 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:

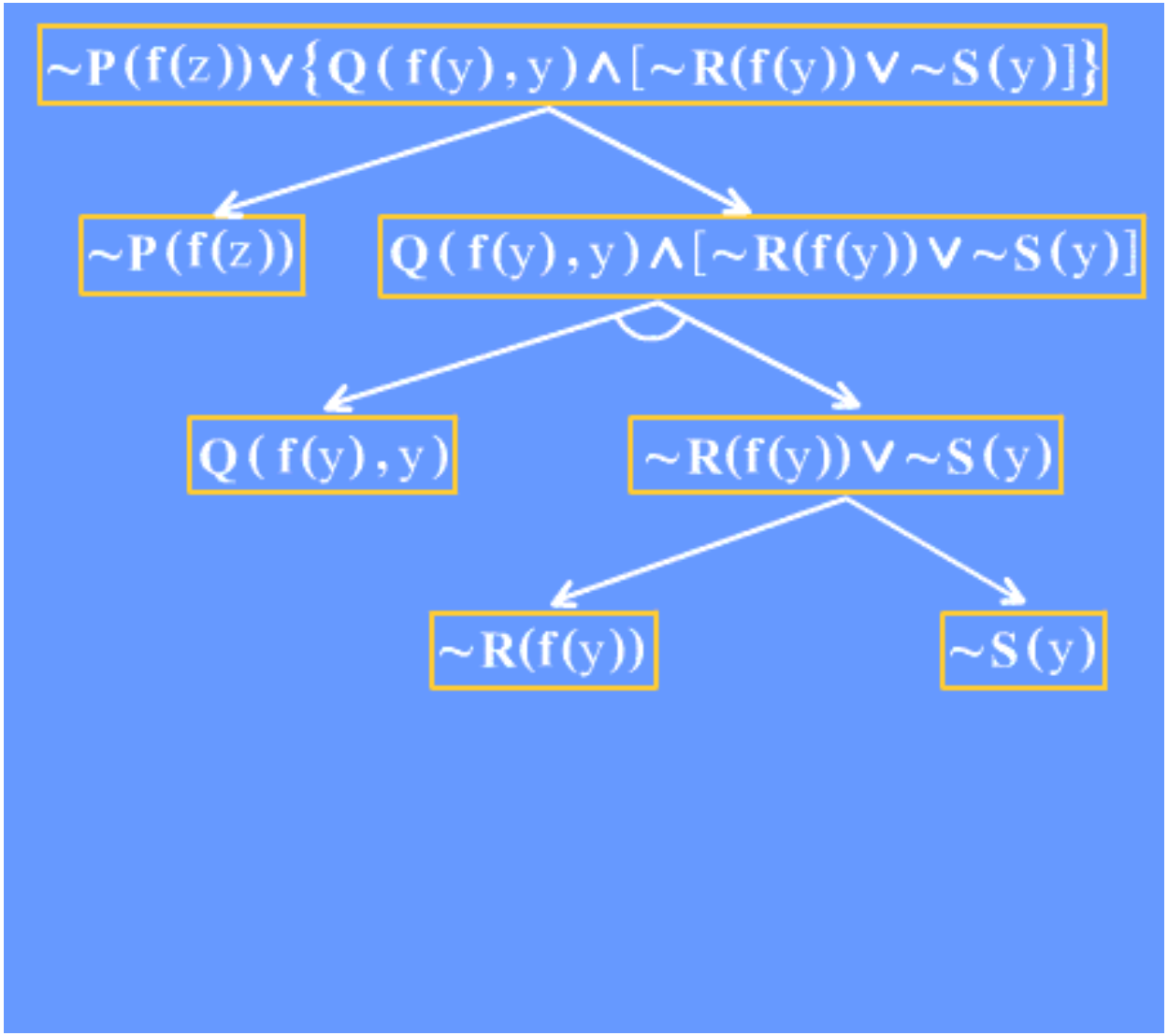
$$\sim P(f(z)) \vee \{Q(f(y),y) \wedge [\sim P(f(y)) \vee \sim S(y)]\}$$

(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.

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

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.

- (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.

Example of backward chaining

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

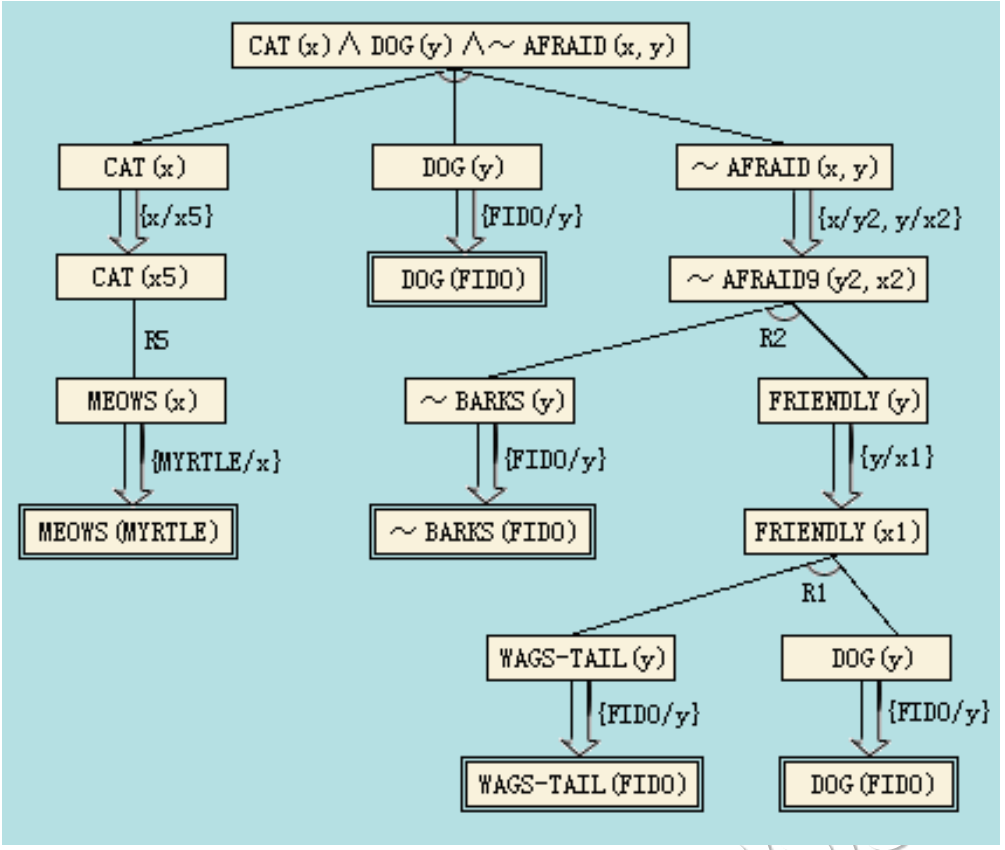
R5: MEOWS(x5) ⇒ CAT(x5); Meows is a cat

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

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：



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

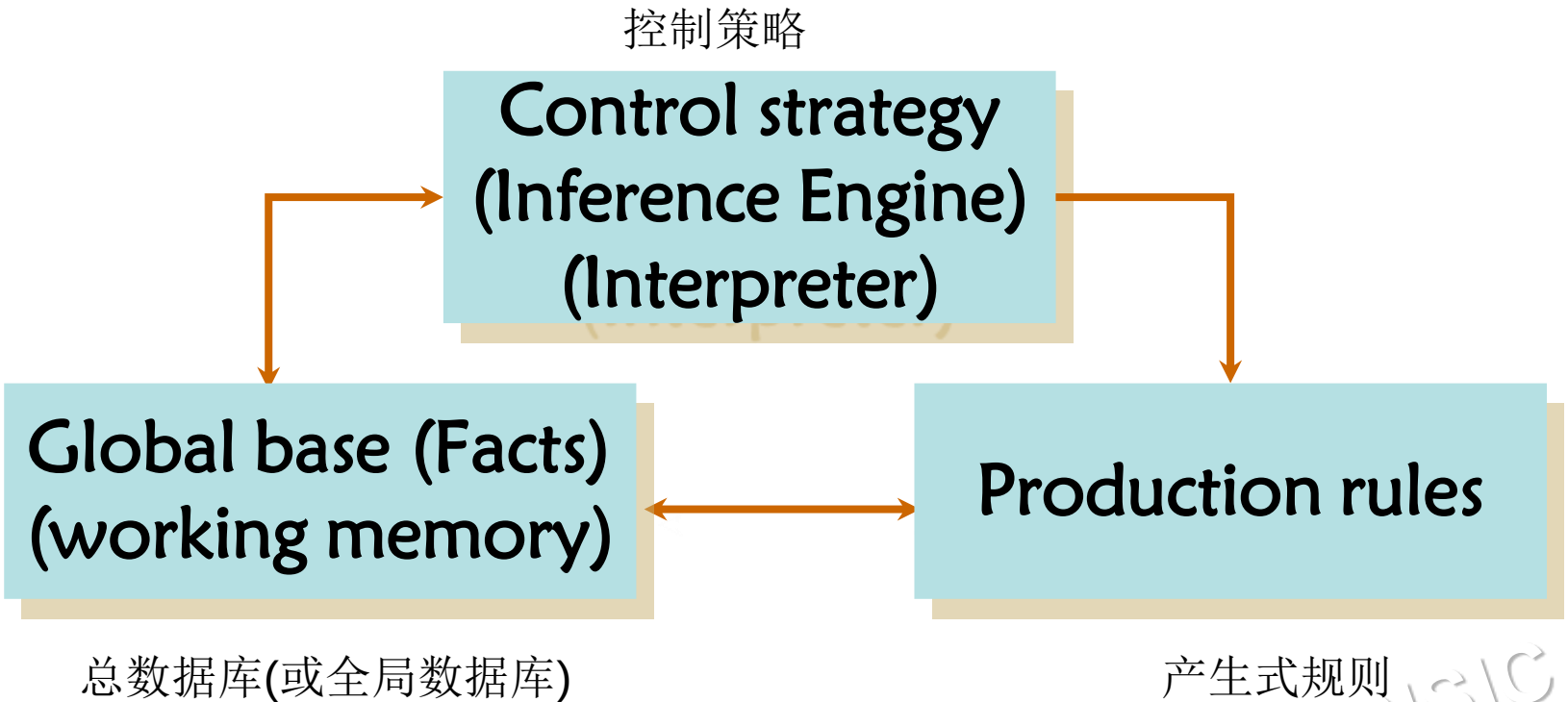
Backward chaining

- 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

3.7 Production System
产生式系统

- 定义：
  - Also called Rule-based System.
  - 由波斯特 于1943年提出的产生式规则而得名. DENDRAL
  - 产生式系统用来描述若干个不同的以一个基本概念为基础的系统.
- 在产生式系统中，论域的知识分为两部分：
  - 用事实表示静态知识，如事物、事件和它们之间的关系；
  - 用产生式规则表示推理过程和行为.

3.7.1 Architecture of Production System (产生式系统的组成)



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

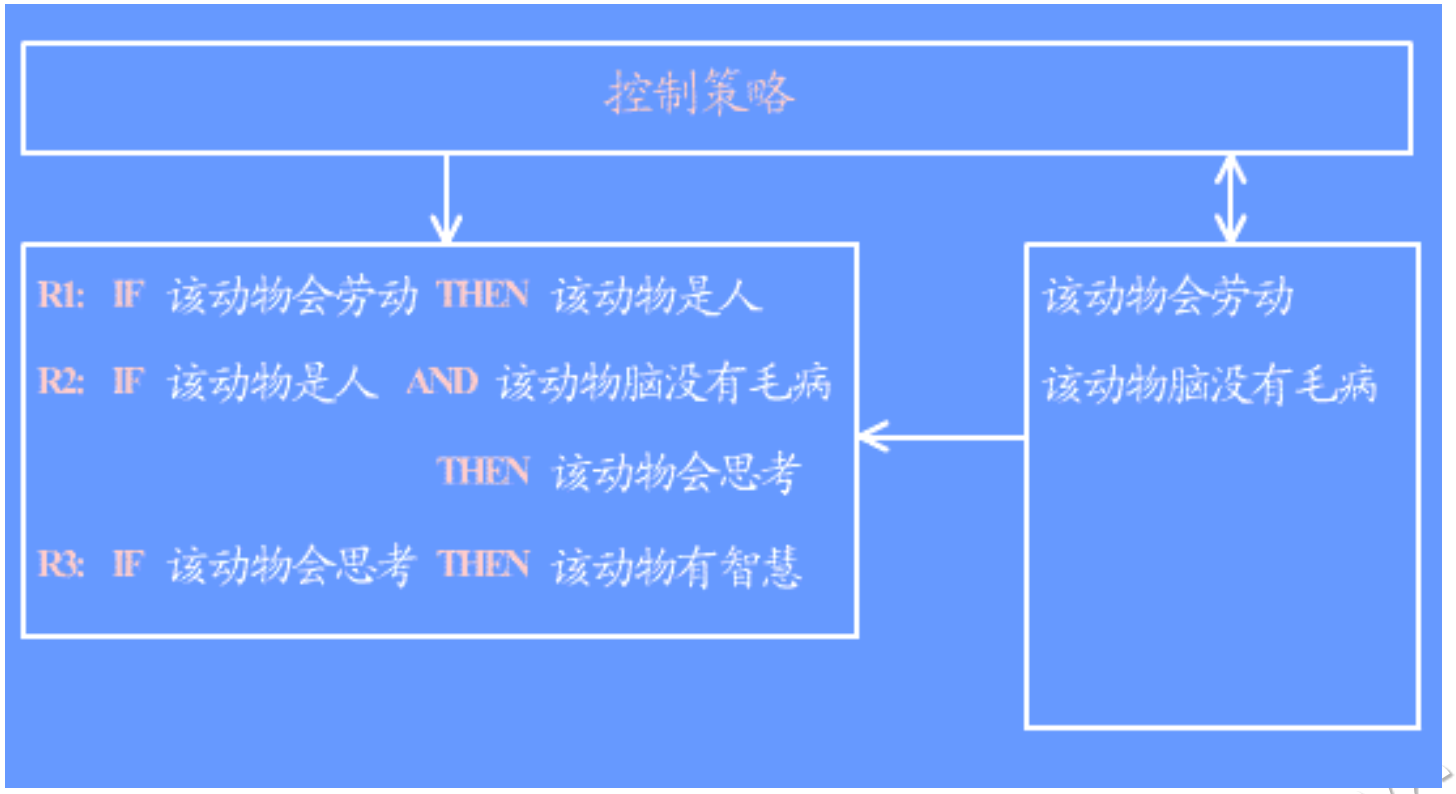
Production-rule System

- 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

Rule Matching → Conflict Resolution → Rule Execution

Stop Conditions

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

3.7.2 Inference of Production System
产生式系统的推理

- 正向推理
  - 从一组表示事实的谓词或命题出发，使用一组产生式规则，用以证明该谓词公式或命题是否成立。
- 逆向推理
  - 从表示目标的谓词或命题出发，使用一组产生式规则证明事实谓词。
- 双向推理（正反向混合推理）
  - 同时从目标向事实推理和从事实向目标推理，并在推理过程中的某个步骤，实现事实与目标的匹配。

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

3.8 Reasoning with Uncertainty

不确定性推理

- Reasoning with uncertainty is a powerful tool to research incompleteness and uncertainty of complex system
- Two kind uncertainties
  - Proof
  - Conclusion, or Rule
- Generally, the uncertainties are represented by a coefficient between 0 to 1.
  - 1 -- complete certainty
  - 0 -- complete uncertainty.
  - This coefficient is called reliability.

- Commonly Methods
  - Probabilistic and Subjective Bayes Method

    概率推理和主观贝叶斯推理
  - C-F (Certain Factor) Method

    可信度方法
  - Evidence (D-S, Dempster- Shafer) Theory

    证据理论
  - Fuzzy Method

    模糊推理

- Subjective Bayes Method
  - The application of Bayes theory demands of collecting amount of event samples to statistics the probability (certain degree) of the events.
  - In many situations, the frequency that same events appeared (or collected) is not high which lead that cann't make probability statistics.
  - But, according to observed data, expert can make some subjective judgments by experience, called subjective probability.

Subjective Bayes Method

- Bayes Theory
  - Conditional Probability 条件概率
    - Assumed the rule P ⇒ Q is uncertainty，and the uncertainty can be described as Conditional Probability(条件概率) p(Q/P)

$$p(Q/P) = \frac{p(P/Q) \cdot p(Q)}{p(P)}$$

    - p(P) and p(Q) are Prior Probability(先验概率) which represents the probability of condition and conclusion respectively（independent to the rule P ⇒ Q）
    - p(P/Q）is Posterior Probability (后验概率)，and denotes the probability of P on the condition of Q effective.

Subjective Bayes Method

- Example：令P为汽车轮子发出的刺耳噪声，Q为汽车刹车失调。P可视为征兆，Q则指示引起P的原因。设想根据经验，刹车调整不好会引起刺耳噪声，并估计 p(P／Q) = 0.7，若同时又获得先验概率p(P) = 0.04，p(Q) = 0.05

$$p(Q/P) = \frac{0.7 \times 0.05}{0.04} = 0.88$$

即每当我们发现车轮的刺耳噪声时，可以推测有0.88的可能性是刹车失调。

Subjective Bayes Method

- Probability of rule strength 规则强度的似然性
  - Likelihood Ratio 似然比O（几率函数）
    - 表示x的出现概率与不出现概率之比，显然随P(x)的加大O(x)也加大

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

    - 当P(x)=0时，有O(x) = 0
    - 当P(x)=1时，有 O(x) = ∞
      - 于是，取值于[0,1]的P(x)被放大为取值于[0, ∞]的O (x)。

Subjective Bayes Method

- Sufficiency Level of the rule

充分性度量LS

- 对于规则P ⇒ Q，引入Q的先验似然比O(Q)和条件似然比 O(Q／P)：

$$O(Q) = p(Q) / p(\neg Q) = \frac{p(Q)}{1 - p(Q)}$$

$$O(Q / P) = \frac{p(Q / P)}{p(\neg Q / P)} = \frac{p(P / Q)}{p(P / \neg Q))} \cdot O(Q)$$

- 令 $LS = \frac{p(P / Q)}{p(P / \neg Q)}$ ，则有

$$O(Q / P) = LS \cdot O(Q)$$

LS称为规则P ⇒ Q成立的充分性度量，用以指示规则强度的似然率，即P成立对Q成立的影响力

Subjective Bayes Method

结论Q的条件似然比可以由其先验似然比和规则的充分性度量LS来计算，并进一步计算出p(Q／P)。

$$O(Q/P) = LS \cdot O(Q) \Rightarrow O(Q/P) = \frac{p(Q/P)}{1-p(Q/P)} \Rightarrow p(Q/P) = \frac{O(Q/P)}{1+O(Q/P)}$$

- Necessity Level of the rule 必要性度量LN

令 $LN = \dfrac{p(\neg P/Q)}{p(\neg P/\neg Q)}$，则有

$$O(Q/\neg P) = LN \cdot O(Q)$$

LN称为规则P ⇒ Q成立的**必要性度量**，用以指示P不成立时对Q成立的影响力

Subjective Bayes Method

- LS和LN的不同取值范围对规则强度的影响

$$LS=\begin{cases}1, & \text{则}O(Q/P)=O(Q),\text{意指}P\text{对}Q\text{无影响}\\ >1, & \text{则}O(Q/P)>O(Q),\text{意指}P\text{在一定程度上支持}Q\\ <1, & \text{则}O(Q/P)<O(Q),\text{意指}P\text{在一定程度上不支持}Q\end{cases}$$

$$LN=\begin{cases}1, & \text{则}O(Q/\neg P)=O(Q),\text{意指}\neg P\text{对}Q\text{无影响}\\ >1, & \text{则}O(Q/\neg P)>O(Q),\text{意指}\neg P\text{在一定程度上支持}Q\\ <1, & \text{则}O(Q/\neg P)<O(Q),\text{意指}\neg P\text{在一定程度上不支持}Q\end{cases}$$

在缺乏大量统计数据的情况下，有经验的专家仍然能作出近似的估计。

把基于专家主观估计的LS（和LN）值而演算出来的条件概率p(Q／P)（经由O(Q／P)）称为主观概率，而这种推算主观概率的方法称为主观Bayes。

Subjective Bayes Method

- 例：P为汽车轮子发出的刺耳噪声，Q为汽车刹车失调。

  若p(Q) = 0.05，则可以推算出O(Q) = 0.053

  专家又估计LS = 120，LN=0.3，则

  $$O(Q/P) = 6.4,\ O(Q/\neg P) = 0.016;$$

  进一步可求出p(Q／P) = 0.87，p(Q／¬P) = 0.016。

Subjective Bayes Method

- Transfer of uncertainty 不确定性的传递
  - 若有规则可传递有P' ⇒ P ⇒ Q，则p(Q／P') 为：
    - p(Q／P')=p(Q／P)·p(P／P') + p(Q／¬P)·p(¬P／P')
    - 当p(P／P') = p(P)时，p(Q／P) = p(Q) 。

例： P为汽车轮子发出的刺耳噪声，Q为汽车刹车失调。设p(P/P') = 0.8。

由于已算出p(Q／P) = 0.87，p(Q／¬P) = 0.016；

又 p(¬P／P') = 0.2，

则：

$$p(Q／P') = 0.87 \times 0.8 + 0.016 \times 0.2 = 0.70$$

3.9 Nonmonotonic Reasoning
非单调推理

- 定义

非单调推理用来处理那些不适合用谓词逻辑表示的知识。

它能够较好地处理不完全信息、不断变化的情况以及求解复杂问题过程中生成的假设，具有较为有效的求解效率。

- 缺省推理 (Default Reasoning)
  - 定义1：如果X不知道，那么得结论Y。
  - 定义2：如果X不能被证明，那么得结论Y。
  - 定义3：如果X不能在某个给定的时间内被证明，那么得结论Y。
- 非单调推理系统(Nonmonotonic Reasoning System)
  - 正确性维持系统用以保持其它程序所产生的命题之间的相容性。一旦发现某个不相容，它就调出自己的推理机制，面向从属关系的回溯，并通过修改最小的信念集来消除不相容。