

# 第3章 确定性推理

- 智能系统的推理过程实际上就是一种思维过程，即运用知识推理和求解问题。
  - 按照推理过程所用知识的确定性，推理可分为确定性推理和不确定性推理。
  - 3.1 推理的基本概念
  - 3.2 推理的逻辑基础
  - 3.3 自然演绎推理
  - 3.4 归结演绎推理
- ~~3.5 基于规则的演绎推理（不做要求）~~

# 3.1 推理的基本概念

- 3.1.1 什么是推理
- 3.1.2 推理方法及其分类
- 3.1.3 推理的控制策略及其分类
- 3.1.4 正向推理
- 3.1.5 逆向推理
- 3.1.6 混合推理

### 3.1.1 什么是推理

- 推理的概念
- 是指按照某种策略从已知事实出发去推出结论的过程。
- 推理所用的事实：
  - 初始证据：推理前用户提供的
  - 中间结论：推理过程中所得到的
- 推理过程：由推理机来完成，所谓推理机就是智能系统中用来实现推理的那些程序。
  - 例如，医疗专家系统，专家知识保存在知识库中。推理开始时，先把病人的症状和检查结果放到综合数据库中，然后再从综合数据库的初始证据出发，按照某种策略在知识库中寻找，并使用知识，直到推出最终结论为止。
- 推理的两个基本问题
  - 推理的方法：解决前提和结论的逻辑关系，不确定性传递
  - 推理的控制策略：解决推理方向，冲突消解策略

## 3.1.2 推理方法及其分类

### 1. 按推理的逻辑基础分类(1/4)

- 可分为演绎推理、归纳推理等
- 演绎推理
  - 演绎推理是从已知的一般性知识出发，去推出蕴含在这些已知知识中的适合于某种个别情况的结论。是一种由一般到个别的推理方法，其核心是三段论，如假言推理、拒取式和假言三段论。
  - 例：假言三段论  $A \rightarrow B, B \rightarrow C \Rightarrow A \rightarrow C$
  - 常用的三段论是由一个大前提、一个小前提和一个结论这三部分组成的。其中，大前提是已知的一般性知识或推理过程得到的判断；小前提是关于某种具体情况或某个具体实例的判断；结论是由大前提推出的，并且适合于小前提的判断。
    - 例：
      - ① 计算机系的学生都会编程；（一般性知识）
      - ② 程强是计算机系的一位学生；（具体情况）
      - ③ 程强会编程。（结论）
    - ①是大前提，②是小前提；③是经演绎推出来的结论。
    - 结论是蕴含在大前提中的。

## 3.1.2 推理方法及其分类

### 1. 按推理的逻辑基础分类(2/4)

- 归纳推理：是一种由个别到一般的推理方法。
- 归纳推理的类型
  - 按照所选事例的广泛性可分为完全归纳推理和不完全归纳推理
  - 按照推理所使用的方法可分为枚举、类比、统计和差异归纳推理等
- 完全归纳推理
  - 是指在进行归纳时需要考察相应事物的全部对象，并根据这些对象是否都具有某种属性，推出该类事物是否具有此属性。如，计算机质量检验。
- 不完全归纳推理
  - 是指在进行归纳时只考察了相应事物的部分对象，就得出了关于该事物的结论。例如，计算机，随机抽查。
- 枚举归纳推理
  - 是指在进行归纳时，如果已知某类事物的有限可数个具体事物都具有某种属性，则可推出该类事物都具有此种属性。
- 例：
  - 王强是计算机系学生，他会编程序；
  - 高华是计算机系学生，她会编程序；
  - .....        .....
  - 当这些具体事例足够多时，就可归纳出一个一般性的知识：
  - 凡是计算机系的学生，就一定会编程序。

## 3.1.2 推理方法及其分类

### 1. 按推理的逻辑基础分类(3/4)

- 类比归纳推理

- 是指在两个或两类事物有许多属性都相同或相似的基础上，推出它们在其他属性上也相同或相似的一种归纳推理。

- 设A、B分别是两类事物的集合：

- $A=\{a_1, a_2, \dots\}$        $B=\{b_1, b_2, \dots\}$

并设 $a_i$ 与 $b_i$ 总是成对出现，且当 $a_i$ 有属性P时， $b_i$ 就有属性Q与此对应，即

$$P(a_i) \rightarrow Q(b_i) \quad i=1, 2, \dots$$

则当A与B中有一新的元素对出现时，若已知 $a'$ 有属性P， $b'$ 有属性Q，即

$$P(a') \rightarrow Q(b')$$

- 类比归纳推理的基础是相似原理，其可靠程度取决于两个或两类事物的相似程度以及这两个或两类事物的相同属性与推出的那个属性之间的相关程度。

## 3.1.2 推理方法及其分类

### 1. 按推理的逻辑基础分类(4/4)

- 演绎推理与归纳推理的区别
- 演绎推理是在已知领域内的一般性知识的前提下，通过演绎求解一个具体问题或者证明一个结论的正确性。它所得出的结论实际上早已蕴含在一般性知识的前提中，演绎推理只不过是将已有事实揭露出来，因此它不能增殖新知识。
- 归纳推理所推出的结论是没有包含在前提内容中的。这种由个别事物或现象推出一般性知识的过程，是增殖新知识的过程。
- 例如，一位计算机维修员，从书本知识，到通过大量实例积累经验，是一种归纳推理方式。运用这些一般性知识去维修计算机的过程则是演绎推理。

### 3.1.2 推理方法及其分类

- 2. 按所用知识的确定性分类
    - 确定性推理 vs. 不确定性推理
  - 3. 按推理过程的单调性分类
    - 单调推理 vs. 非单调推理
    - 单调：不会由于新知识的加入否定前面的结论
      - 不能取消原有的结论 (理想化)
      - 知道的条件越多,能推出的结论也就越多
    - 非单调：知识不完全情况下，某些新知识的加入会否定前面的结论 为使推理继续，需先进行一些假设，以此为基础进行推理；当引入某些新知识后，发现该假设不正确时，需撤销该假设及由该假设推出的一切结论

# 非单调推理 (Minsky, 1975)

- 鸟类辨识例子
  - 如果给出一个~~是鸟的~~动物，并且不知道其他事情，就假定它会飞。如果后来知道这个动物其实是企鹅/鸵鸟，这个事实（该动物会飞）无论如何都必须被撤销。
  - “水里游的” & “鱼”：鲸鱼？
- 看到~~潮湿的草地~~的最可能的解释是~~下雨了~~；但是在知道了草地潮湿的真正原因是~~浇水了~~的时候，这个解释应当被撤销。因为获得了增加的知识(~~洒水车经过了~~)，旧的解释(~~下雨了~~)被撤销了，这种逻辑就是非单调的。

# 非单调推理

- 非常适合刻画常识推理的最一般性特征
  - 知识不完全（一般情况下，鸟是会飞的；但企鹅和鸵鸟是例外）
  - 对不知道的知识做猜测，并不断修正自己的结论
- 缺省/默认推理
  - 事实可以是已知的，只是因为缺乏反面的证据，如：鸟会飞，有罪判定
  - 当且仅当没有事实证明 $S$ 不成立时， $S$ 总是成立的。
- 演绎推理
- 信念修正
  - 新知识可以和旧信仰矛盾，如：科学假说

# 非单调推理

- 最具有代表性的理论
  - McDermott的非单调逻辑
    - 加入“unless”、“is consistent with”意义的逻辑扩展
  - J. McCarthy的界限理论
  - Reiter的缺省理论
    - 加入“unless abnormal”的规则
  - Doyle的正确性维持系统
- 代表性的非单调逻辑系统
  - McDermott和Doyle的非单调模态逻辑系统
  - Moore的自认知逻辑系统
  - Gabby的基于直觉逻辑的模态非单调逻辑系统
  - Turner的基于三值逻辑的模态非单调逻辑系统
  - 基于网络和框架的非单调系统

# 非单调推理

- 实际应用

- 在专家系统中的应用

- 专家系统所给出的知识常常是不完备的。
    - 在专家系统运行中，新推出的事实在和原有的假设矛盾，就必须消除矛盾，方法是撤销原来的假设及由其推导出的事实
    - 非单调推理在土木工程专家系统，法律专家系统，心理专家系统，职业测评专家系统等等都已经有了广泛的应用。

- 对其它理论的影响

- 语义网中的继承推理的形式化研究使用非单调逻辑来描述
    - 非单调推理与数值优化理论的结合及应用

### 3.1.3 推理的控制策略及其分类

- 推理的控制策略
- 推理过程不仅依赖于所用的推理方法，同时也依赖于推理的控制策略。
  - 推理的控制策略是指如何使用领域知识使推理过程尽快达到目标的策略。
- 控制策略的分类
  - 由于智能系统的推理过程一般表现为一种搜索过程，因此，推理的控制策略又可分为推理策略和搜索策略。

### 3.1.3 推理的控制策略及其分类

- 推理策略

主要解决推理方向、冲突消解等问题，如推理方向控制策略、求解策略、限制策略、冲突消解策略等

推理方向控制策略用于确定推理的控制方向，可分为正向推理、逆向推理、混合推理及双向推理。

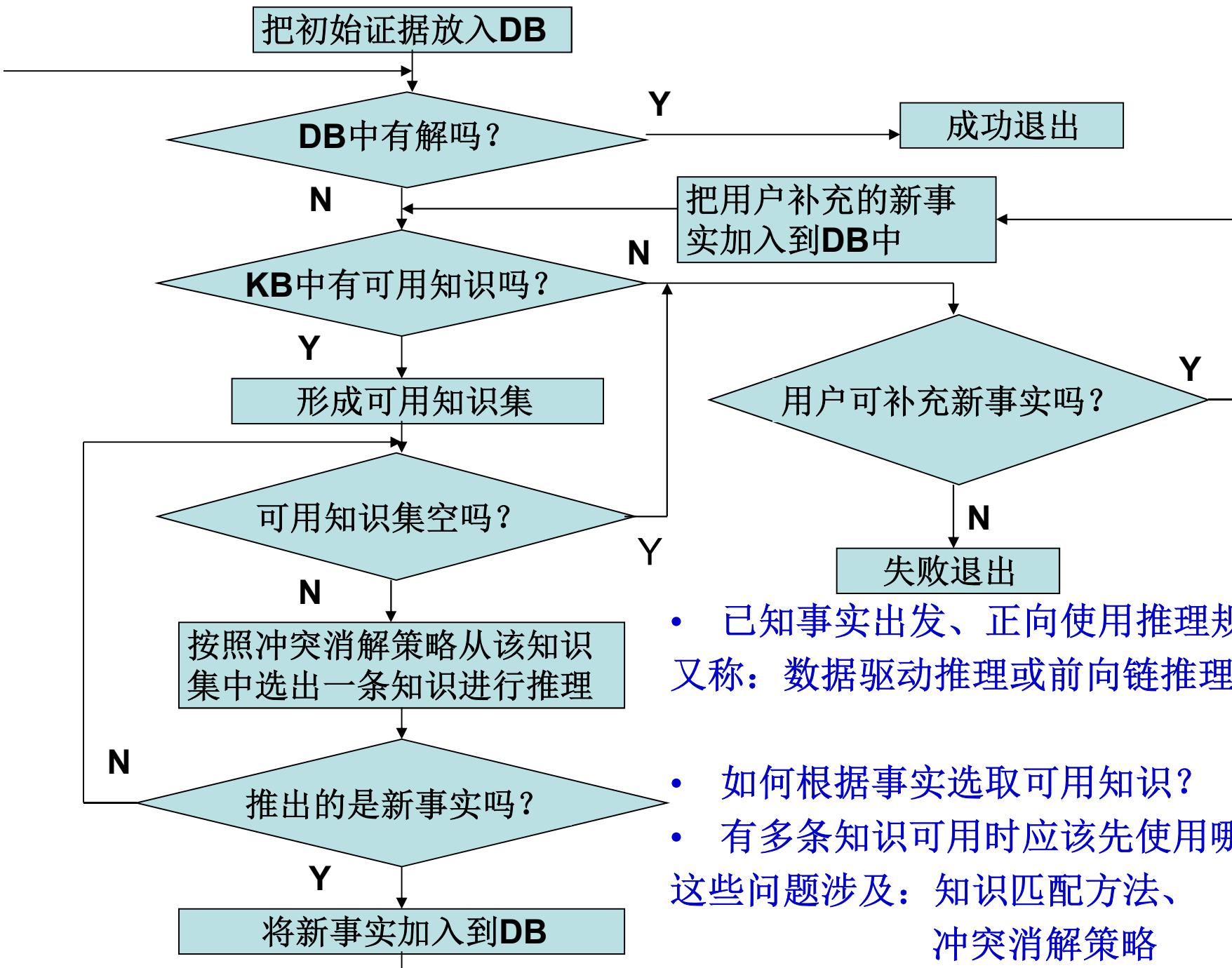
求解策略是指仅求一个解，还是求所有解或最优解等。

限制策略是指对推理的深度、宽度、时间、空间等进行的限制。

冲突消解策略是指当推理过程有多条知识可用时，如何从这多条可用知识中选出一条最佳知识用于推理的策略。

- 搜索策略

主要解决推理线路、推理效果、推理效率等问题。本章主要讨论推理策略，至于搜索策略将放到第4章单独讨论。



- 已知事实出发、正向使用推理规则  
又称：数据驱动推理或前向链推理
  - 如何根据事实选取可用知识？
  - 有多条知识可用时应该先使用哪一条？
- 这些问题涉及：知识匹配方法、  
冲突消解策略

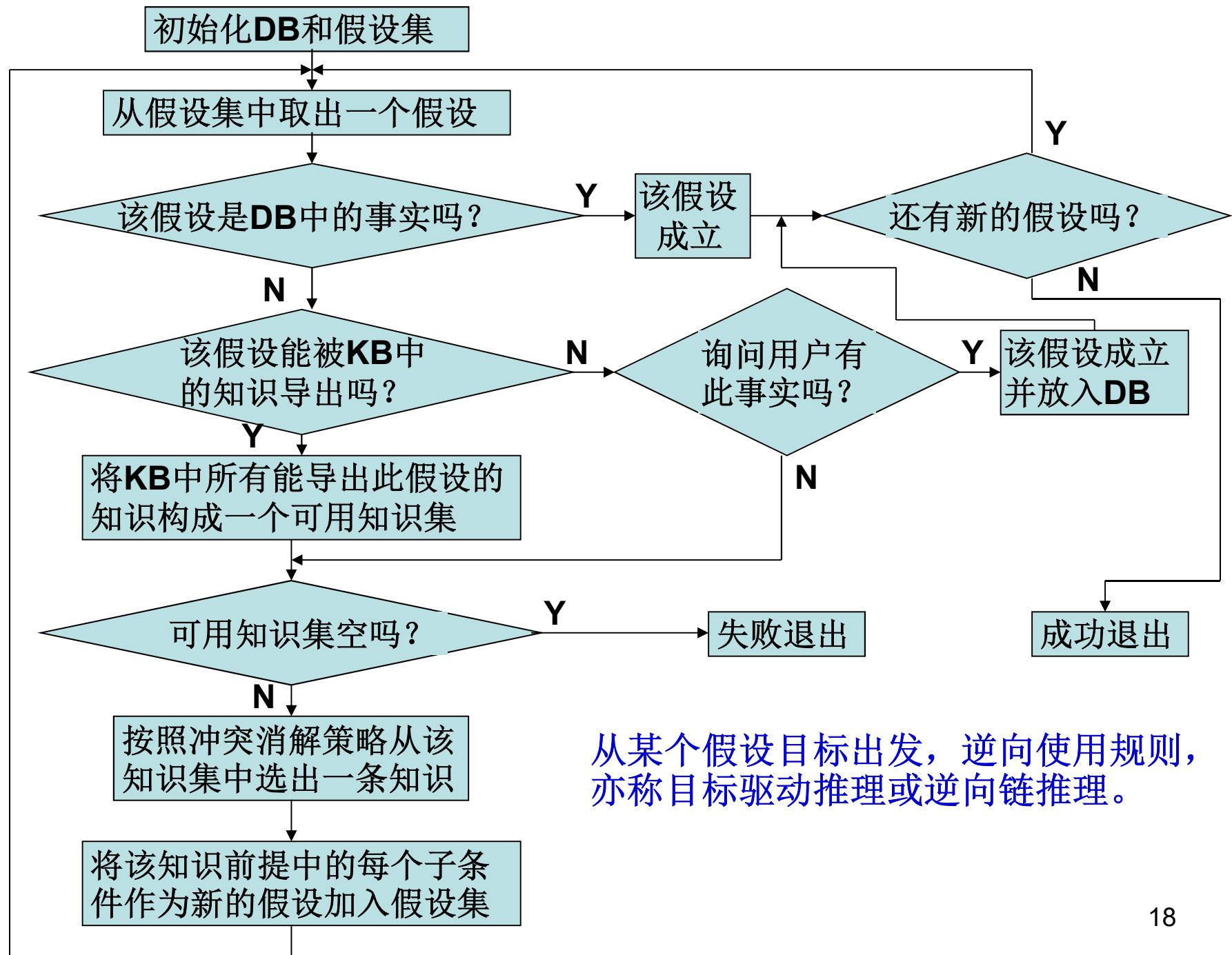
### 3.1.4 正向推理

推理例子

- 例3.1请用正向推理完成以下问题的求解
- 假设知识库中包含有以下2条规则：
  - $r_1: \text{IF } B \text{ THEN } C$
  - $r_2: \text{IF } A \text{ THEN } B$
- 已知初始证据A，求证目标C。
- 解：（略）

### 3.1.4 正向推理 优缺点

- 正向推理的主要优点
- 比较直观，允许用户主动提供有用的事实信息，适合于诊断、设计、预测、监控等领域的问题求解。
- 正向推理的主要缺点
- 推理无明确目标，求解问题是可能会执行许多与解无关的操作，导致推理效率较低。

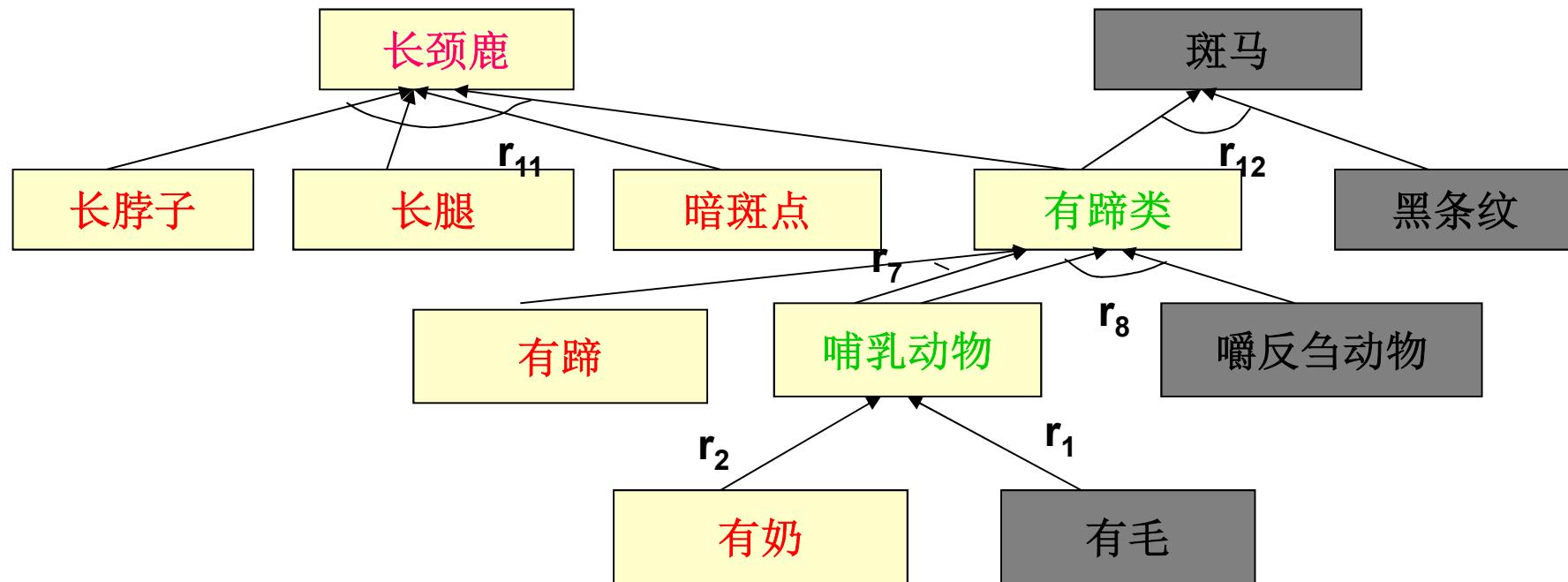


### 3.1.5 逆向推理 优缺点

- 逆向推理的主要优点
  - 不必寻找和使用那些与假设目标无关的信息和知识
  - 推理过程的目标明确
  - 也有利于向用户提供解释，在诊断性专家系统中较为有效。
- 逆向推理的主要缺点
  - 当用户对解的情况认识不清时，由系统自主选择假设目标的盲目性比较大，若选择不好，可能需要多次提出假设，会影响系统效率。

# 例子（动物识别专家系统）

- 初始综合数据库包含的事实有：
- 动物有暗斑点，有长脖子，有长腿，有奶，有蹄
- 目标：该动物为何种动物？



## 3.1.6 混合推理

- 混合推理的概念
- 把正向推理论和逆向推理论结合起来所进行的推理论称为混合推理论。是一种解决较复杂问题的方法。
- 混合推理论的方法
  - 1. 先正向后逆向
    - 这种方法先进行正向推理论，从已知事实出发推出部分结果，然后再用逆向推理论对这些结果进行证实或提高它们的可信度。
  - 2. 先逆向后正向
    - 这种方法先进行逆向推理论，从假设目标出发推出一些中间假设，然后再用正向推理论对这些中间假设进行证实。
  - 3. 双向混合
    - 是指正向推理论和逆向推理论同时进行，使推理论过程在中间的某一步结合起来。

# 第3章 确定性推理

- 3.1 推理的基本概念
- 3.2 推理的逻辑基础
- 3.3 自然演绎推理
- 3.4 归结演绎推理
- 3.5 基于规则的演绎推理

## 3.2 推理的逻辑基础

- 3.2.1 谓词公式的解释
- 3.2.2 谓词公式的永真性和可满足性
- 3.2.3 谓词公式的等价性和永真蕴含性
- 3.2.4 谓词公式的范式
- 3.2.5 置换与合一

## 3.2.1 谓词公式的解释 概念

- 命题逻辑 vs. 谓词逻辑
  - 命题演算中，命题符号P, Q代表一定的命题
  - P: 星期四下雨了 Q: 张三是李四的老师
  - 局限性：看不出关系，表达不出共同特征
  - 而谓词可表示日期与天气的关系： Weather(Tuesday, Rain)
  - 允许包含变元 Weather (X, Rain)
- 谓词公式的解释
- 在谓词逻辑中，谓词公式的一个解释就是对该谓词公式中各个变元的一次真值指派。
- 有了谓词公式的解释，就可据此求出该谓词公式的真值。
- 谓词公式的解释
- 由于谓词公式中可能包含由个体常量、变元或函数，因此，必须先考虑这些个体常量和函数在个体域上的取值，然后才能根据它们的具体取值为谓词分别指派真值。

## 3.2.1 谓词公式的解释

### 概念

- 定义3.1 设D是谓词公式P的非空个体域，若对P中的个体常量、函数和谓词按如下规定赋值：
  - (1) 为每个个体常量指派D中的一个元素；
  - (2) 对每一个变元，指派D的一个非空集合，这是该变元的变域(个体域)；
  - (3) 为每个n元函数指派一个从 $D^n$ 到D的一个映射，其中
$$D^n = \{(x_1, x_2, \dots, x_n) \mid x_1, x_2, \dots, x_n \in D\}$$
  - (4) 为每个n元谓词指派一个从 $D^n$ 到{F, T}的映射。
  - 则称这些指派为P在D上的一个解释I。

# 谓词演算表达式的真值

- 设有表达式**E**和在非空论域**D**上对**E**的一个解释**I**, **E**的真值按以下规律决定:
  - 1) 一个常元的值是根据**I**指派给它的**D**的一个元素。
  - 2) 一个变元的值是根据**I**指派给它的**D**的一个元素集合。
  - 3) 一个函数的值是根据由**I**指派给它的参数值计算得到的**D**的元素。
  - 4) 真值符号**true**的值是**T**, **false**的值是**F**。
  - 5) 原子命题的值或者为**T**, 或者为**F**, 取决于解释**I**。
  - 6) 如果一个命题的值为**F**, 则其否定式为**T**, 否则为**F**。
  - 7~10) 如果...合取, 析取, 蕴含, 等价... (真值表)
  - 11) 如果对于在解释**I**下的**X**的每一个指派, **S**的值为**T**, 则 $\forall_x S$ 为**T**, 否则为**F**。
  - 12) 如果在解释**I**下存在**X**的一个指派使得**S**的值为**T**, 则 $\exists_x S$ 为**T**, 否则为**F**。

## 3.2.1 谓词公式的解释

### 例子一(1/2)

- 例3.2 设个体域 $D=\{1, 2\}$ , 求公式 $A=(\forall x)(\exists y)P(x, y)$ 在 $D$ 上的解释, 并指出在每一种解释下公式 $A$ 的真值。
- 解: 由于公式 $A$ 中没有包含个体常量和函数, 故可直接为谓词指派真值, 设有

$P(1,1)$	$P(1,2)$	$P(2,1)$	$P(2,2)$
T	F	T	F

- 这就是公式 $A$  在 $D$  上的一个解释。从这个解释可以看出:
- 当 $x=1$ 、 $y=1$ 时, 有 $P(x,y)$ 的真值为T;
- 当 $x=2$ 、 $y=1$ 时, 有 $P(x,y)$ 的真值为T;
- 即对 $x$ 在 $D$ 上的任意取值, 都存在 $y=1$ 使 $P(x,y)$ 的真值为T。因此, 在此解释下公式 $A$ 的真值为T。

## 3.2.1 谓词公式的解释

### 例子一(2/2)

- 说明：一个谓词公式在其个体域上的解释不是唯一的。例如，对公式A，若给出另一组真值指派

$P(1,1)$	$P(1,2)$	$P(2,1)$	$P(2,2)$
T	T	F	F

- 这也是公式A在D上的一个解释。从这个解释可以看出：
- 当x=1、y=1时，有 $P(x,y)$ 的真值为T；
- 当x=2、y=1时，有 $P(x,y)$ 的真值为F；
- 即对x在D上的任意取值，不存在一个y使得 $P(x,y)$ 的真值为T。因此，在此解释下公式A的真值为F。
- 实际上，A在D上共有16种解释，这里就不一一列举。

## 3.2.1 谓词公式的解释

### 例子二

- 例3.3 设个体域 $D=\{1, 2\}$ , 求公式 $B=(\forall x)P(f(x), a)$ 在 $D$ 上的解释, 并指出在该解释下公式 $B$ 的真值。
- 解: 设对个体常量 $a$ 和函数 $f(x)$ 的值指派为:

a	f(1)	F(2)
1	1	2

- 对谓词的真值指派为:

P(1,1)	P(1,2)	P(2,1)	P(2,2)
T	X	T	X

- 由于已指派 $a=1$ , 因此 $P(1,2)$ 和 $P(2,2)$ 不可能出现, 故没给它们指派真值。
- 上述指派是公式 $B$ 在 $D$ 上的一个解释。在此解释下有
  - 当 $x=1$ 时,  $a=1$ 使 $P(1,1)=T$
  - 当 $x=2$ 时,  $a=1$ 使 $P(2,1)=T$
- 即对 $x$ 在 $D$ 上的任意取值, 都存在 $a=1$ 使 $P(f(x), a)$ 的真值为 $T$ 。因此, 在此解释下公式 $B$ 的真值为 $T$ 。
- 由上面的例子可以看出, 谓词公式的真值都是针对某一个解释而言的, 它可能在某一个解释下真值为 $T$ , 而在另一个解释下为 $F$ 。29

### 3.2.2 谓词公式的永真性和可满足性

- 为了以后推理的需要，下面先定义：
- 定义3.2 如果谓词公式P对非空个体域D上的任一解释都取得真值T，则称P在D上是永真的；如果P在任何非空个体域上均是永真的，则称P永真。
- 可见，要判定一谓词公式为永真，必须对每个非空个体域上的每个解释逐一进行判断。当解释的个数为有限时，尽管工作量大，公式的永真性毕竟还可以判定，但当解释个数为无限时，其永真性就很难判定了。

### 3.2.2 谓词公式的永真性和可满足性

- 定义3.3 对于谓词公式P，如果至少存在D上的一个解释，使公式P在此解释下的真值为T，则称公式P在D上是可满足的。
- 谓词公式的可满足性也称为相容性。
- 定义3.4 如果谓词公式P对非空个体域D上的任一解释都取真值F，则称P在D上是永假的；如果P在任何非空个体域上均是永假的，则称P永假。
  - 谓词公式的永假性又称不可满足性或不相容。
  - 后面我们要给出的归结推理，就是采用一种逻辑上的反证法，将永真性转换为不可满足性的证明。

### 3.2.3 谓词公式的等价性和永真蕴含性

#### 1. 等价式(1/2)

- 谓词公式的等价性和永真蕴含性可分别用相应的等价式和永真蕴含式来表示，这些等价式和永真蕴含式都是演绎推理的主要依据，因此也称它们为推理规则。
- 谓词公式的等价式可定义如下：
- 定义3.5 设P与Q是D上的两个谓词公式，若对D上的任意解释，P与Q都有相同的真值，则称P与Q在D上是等价的。如果D是任意非空个体域，则称P与Q是等价的，记作 $P \Leftrightarrow Q$ 。

### 3.2.3 谓词公式的等价性和永真蕴含性

#### 1. 常用等价式(2/2)

- (1) 双重否定律  $\neg\neg P \Leftrightarrow P$
- (2) 交换律  $(P \vee Q) \Leftrightarrow (Q \vee P), (P \wedge Q) \Leftrightarrow (Q \wedge P)$
- (3) 结合律  $(P \vee Q) \vee R \Leftrightarrow P \vee (Q \vee R)$   
 $(P \wedge Q) \wedge R \Leftrightarrow P \wedge (Q \wedge R)$
- (4) 分配律  $P \vee (Q \wedge R) \Leftrightarrow (P \vee Q) \wedge (P \vee R)$   
 $P \wedge (Q \vee R) \Leftrightarrow (P \wedge Q) \vee (P \wedge R)$
- (5) 德摩根(否定)律  $\neg(P \vee Q) \Leftrightarrow \neg P \wedge \neg Q$   
 $\neg(P \wedge Q) \Leftrightarrow \neg P \vee \neg Q$  置换律:  
 $(P \rightarrow Q) \Leftrightarrow (\neg Q \rightarrow \neg P)$
- (6) 吸收律  $P \vee (P \wedge Q) \Leftrightarrow P$   
 $P \wedge (P \vee Q) \Leftrightarrow P$
- (7) 补余律  $P \vee \neg P \Leftrightarrow T, P \wedge \neg P \Leftrightarrow F$
- (8) 连词化归律  $P \rightarrow Q \Leftrightarrow \neg P \vee Q$   
 $P \leftrightarrow Q \Leftrightarrow (P \rightarrow Q) \wedge (Q \rightarrow P)$   
 $P \leftrightarrow Q \Leftrightarrow (P \wedge Q) \vee (\neg Q \wedge \neg P)$
- (9) 量词转换律  $\neg(\exists x)P \Leftrightarrow (\forall x)(\neg P)$   
 $\neg(\forall x)P \Leftrightarrow (\exists x)(\neg P)$
- (10) 量词分配律  $(\forall x)(P \wedge Q) \Leftrightarrow (\forall x)P \wedge (\forall x)Q$   
 $(\exists x)(P \vee Q) \Leftrightarrow (\exists x)P \vee (\exists x)Q$

### 3.2.3 谓词公式的等价性和永真蕴含性

#### 2. 永真蕴含式

- 定义3.6 对谓词公式P和Q，如果 $P \rightarrow Q$ 永真，则称P 永真蕴含Q，且称Q为P的逻辑结论，P为Q的前提，记作 $P \Rightarrow Q$ 。
- 常用的永真蕴含式如下：
  - (1) 化简式  $P \wedge Q \Rightarrow P, P \wedge Q \Rightarrow Q$
  - (2) 附加式  $P \Rightarrow P \vee Q, Q \Rightarrow P \vee Q$
  - (3) 析取三段论  $\neg P, P \vee Q \Rightarrow Q$
  - (4) 假言推理  $P, P \rightarrow Q \Rightarrow Q$
  - (5) 拒取式  $\neg Q, P \rightarrow Q \Rightarrow \neg P$
  - (6) 假言三段论  $P \rightarrow Q, Q \rightarrow R \Rightarrow P \rightarrow R$
  - (7) 二难推理  $P \vee Q, P \rightarrow R, Q \rightarrow R \Rightarrow R$
  - (8) 全称固化  $(\forall x)P(x) \Rightarrow P(y)$   
其中，y是个体域中的任一个体，依此可消去谓词公式中的全称量词。
  - (9) 存在固化  $(\exists x)P(x) \Rightarrow P(y)$   
其中，y是个体域中某一个可以使 $P(y)$ 为真的个体，依此可消去谓词公式中的存在量词。
- 以上等价式和永真蕴含式也称 推理规则

### 3.2.4 谓词公式的范式

- 范式是谓词公式的标准形式。在谓词逻辑中，范式分为两种：
- 前束范式
- 定义3.7 设F为一谓词公式，如果其中的**所有量词均非否定地**出现在公式的最前面，且它们的辖域为整个公式，则称F为前束范式。一般形式：
$$(Q_1x_1) \dots (Q_nx_n)M(x_1, x_2, \dots, x_n)$$
- 其中， $Q_i(i=1, 2, \dots, n)$ 为前缀，它是一个由全称量词或存在量词组成的量词串； $M(x_1, x_2, \dots, x_n)$ 为母式，它是一个不含任何量词的谓词公式。
- 例如， $(\forall x)(\forall y)(\exists z)(P(x) \wedge Q(y, z) \vee R(x, z))$ 是前束范式。
- 任一谓词公式均可化为与其对应的前束范式，其化简方法将在后面子句集的化简中讨论。
- Skolem范式
- 定义3.8 如果前束范式中所有的存在量词都在全称量词之前，则称这种形式的谓词公式为Skolem范式。
- 例如， $(\exists x)(\exists z)(\forall y)(P(x) \vee Q(y, z) \wedge R(x, z))$ 是Skolem范式。
- 任一谓词公式均可化为与其对应的Skolem范式，其化简方法也将在后面子句集的化简中讨论。

## 3.2.5 置换与合一

### 概念

- 在不同谓词公式中，往往会出现谓词名相同但其个体不同的情况，此时推理过程是不能直接进行匹配的，需要先进行置换。
- 例如，可根据全称固化推理论和假言推理由谓词公式 $W_1(A)$  和  $(\forall x)(W_1(x) \rightarrow W_2(x))$
- 推出 $W_2(A)$ 。
- 要使用假言推理，首先需要找到项A对变元x的置换，使 $W_1(A)$ 与 $W_1(x)$ 一致。
- 这种寻找项对变元的置换，使谓词一致的过程叫做合一的过程。
- 下面讨论置换与合一的有关概念与方法。

## 3.2.5 置换与合一

### 1. 置换(1/2)

- 置换可简单的理解为是在一个谓词公式中用置换项去替换变量。
- 定义3.9 置换是形如
- $\{t_1/x_1, t_2/x_2, \dots, t_n/x_n\}$
- 的有限集合。其中， $t_1, t_2, \dots, t_n$ 是项； $x_1, x_2, \dots, x_n$ 是互不相同的变元； $t_i/x_i$ 表示用 $t_i$ 替换 $x_i$ 。并且要求 $t_i$ 与 $x_i$ 不能相同， $x_i$ 不能循环地出现在另一个 $t_i$ 中。
- 例如， $\{a/x, c/y, f(b)/z\}$ 是一个置换。
- 但 $\{g(y)/x, f(x)/y\}$ 不是一个置换。原因是它在x与y之间出现了循环置换现象。即当用 $g(y)$ 置换x,用 $f(g(y))$ 置换y时，既没有消去x，也没有消去y。
- 若改为 $\{g(a)/x, f(x)/y\}$ 即可，原因是用 $g(a)$ 置换x，用 $f(g(a))$ 置换y，则消去了x和y。
- 通常，置换是用希腊字母 $\theta$ 、 $\sigma$ 、 $\alpha$ 、 $\lambda$ 等来表示的。
- 定义3.10 设 $\theta = \{t_1/x_1, t_2/x_2, \dots, t_n/x_n\}$ 是一个置换，F是一个谓词公式，把公式F中出现的所有 $x_i$ 换成 $t_i$ ( $i=1, 2, \dots, n$ )，得到一个新的公式G，称G为F在置换 $\theta$ 下的例示，记作 $G=F\theta$ 。
- 一个谓词公式的任何例示都是该公式的逻辑结论。

## 3.2.5 置换与合一

### 2. 置换(2/2)

- 定义3.11 设
- $\theta = \{t_1/x_1, t_2/x_2, \dots, t_n/x_n\}$
- $\lambda = \{u_1/y_1, u_2/y_2, \dots, u_m/y_m\}$
- 是两个置换。则  $\theta$  与  $\lambda$  的合成也是一个置换，记作  $\theta \circ \lambda$ 。它是从集合
  - $\{t_1\lambda/x_1, t_2\lambda/x_2, \dots, t_n\lambda/x_n, u_1/y_1, u_2/y_2, \dots, u_m/y_m\}$中删去以下两种元素
  - ① 当  $t_i\lambda = x_i$  时，删去  $t_i\lambda/x_i$  ( $i=1, 2, \dots, n$ );
  - ② 当  $y_j \in \{x_1, x_2, \dots, x_n\}$  时，删去  $u_j/y_j$  ( $j=1, 2, \dots, m$ )最后剩下的元素所构成的集合。
- 例3.4 设  $\theta = \{f(y)/x, z/y\}$ ,  $\lambda = \{a/x, b/y, y/z\}$ , 求  $\theta$  与  $\lambda$  的合成。
- 解: 先求出集合
  - $\{f(b)/x, (y/z)/y, a/x, b/y, y/z\} = \{f(b)/x, y/y, a/x, b/y, y/z\}$其中,  $f(b)/x$  中的  $f(b)$  是置换  $\lambda$  作用于  $f(y)$  的结果;  $y/y$  中的  $y$  是置换  $\lambda$  作用于  $z$  的结果。在该集合中,  $y/y$  满足定义中的条件①, 需要删除;  $a/x$  和  $b/y$  满足定义中的条件②, 也需要删除。最后得
  - $\theta \circ \lambda = \{f(b)/x, y/z\}$

## 3.2.5 置换与合一

### 2. 合一

- 合一可理解为是寻找项对变量的置换，使两个谓词公式一致。可定义为：
- 定义3.12 设有公式集 $F=\{F_1, F_2, \dots, F_n\}$ ，若存在一个置换 $\theta$ ，可使 $F_1\theta=F_2\theta=\dots=F_n\theta$ ，
- 则称 $\theta$ 是 $F$ 的一个合一。称 $F_1, F_2, \dots, F_n$ 是可合一的。
- 例如，设有公式集 $F=\{P(x,y,f(y)), P(a,g(x),z)\}$ ，则 $\lambda=\{a/x, g(a)/y, f(g(a))/z\}$ 是它的一个合一。
- 一般来说，一个公式集的合一不是唯一的。
- 定义3.13 设 $\sigma$ 是公式集 $F$ 的一个合一，如果对 $F$ 的任一个合一 $\theta$ 都存在一个置换 $\lambda$ ，使得 $\theta=\sigma^{\circ} \lambda$ ，则称 $\sigma$ 是一个最一般合一。
- 一个公式集的最一般合一是唯一的。
- 对如何求取最一般合一的问题，不再讨论。

# 第3章 确定性推理

- 3.1 推理的基本概念
- 3.2 推理的逻辑基础
- 3.3 自然演绎推理
- 3.4 归结演绎推理
- 3.5 基于规则的演绎推理

### 3.3 自然演绎推理

- 自然演绎推理
- 从一组已知为真的事实出发，直接运用经典逻辑中的推理规则推出结论的过程称为自然演绎推理。
- 自然演绎推理最基本的推理规则是三段论推理，它包括：
  - 假言推理  $P, P \rightarrow Q \Rightarrow Q$
  - 拒取式  $\neg Q, P \rightarrow Q \Rightarrow \neg P$
  - 假言三段论  $P \rightarrow Q, Q \rightarrow R \Rightarrow P \rightarrow R$
- 自然演绎推理的例子
- 例3.5 设已知如下事实： $A, B, A \rightarrow C, B \wedge C \rightarrow D, D \rightarrow Q$
- 求证： $Q$ 为真。
- 证明：因为  $A, A \rightarrow C \Rightarrow C$  假言推理
- $B, C \Rightarrow B \wedge C$  引入合取词
- $B \wedge C, B \wedge C \rightarrow D \Rightarrow D$  假言推理
- $D, D \rightarrow Q \Rightarrow Q$  假言推理
- 因此， $Q$ 为真

### 3.3 自然演绎推理

- 例3.6 设已知如下事实:
  - (1) 只要是需要编程序的课，王程都喜欢。
  - (2) 所有的程序设计语言课都是需要编程序的课。
  - (3) C是一门程序设计语言课。
- 求证：王程喜欢C这门课。
- 证明：首先定义谓词
  - $\text{Prog}(x)$   $x$ 是需要编程序的课。
  - $\text{Like}(x, y)$   $x$ 喜欢 $y$ 。
  - $\text{Lang}(x)$   $x$ 是一门程序设计语言课
- 把已知事实及待求解问题用谓词公式表示如下:
  - $\text{Prog}(x) \rightarrow \text{Like}(\text{Wang}, x)$
  - $(\forall x)(\text{Lang}(x) \rightarrow \text{Prog}(x))$
  - $\text{Lang}(C)$
- 应用推理规则进行推理:
  - $\text{Lang}(y) \rightarrow \text{Prog}(y)$  全称固化
  - $\text{Lang}(C), \text{Lang}(y) \rightarrow \text{Prog}(y) \Rightarrow \text{Prog}(C)$  假言推理  $\{\text{C}/y\}$
  - $\text{Prog}(C), \text{Prog}(x) \rightarrow \text{Like}(\text{Wang}, x) \Rightarrow \text{Like}(\text{Wang}, C)$  假言推理  $\{\text{C}/x\}$
- 因此，王程喜欢C这门课。

# 应用:一个基于逻辑的金融投资 辅助决策程序

- 依据以下原则为投资者推荐投资方案:
  - (1) 投资者如果没有足够的存款, 则不管其收入多少都应把钱存入银行。
  - (2) 拥有足够的存款和收入的人应该考虑股票投资。
  - (3) 拥有较少收入和足够存款的人可能想把他们多余的钱分别用于存款和买股票。
- 存款和收入是否足够取决于投资者需要赡养的人口数目, 规则是:
  - (a) 对每个要赡养的人, 银行里必须至少有5000美元;
  - (b) 一个足够的收入必须是一个稳定的收入, 每年提供15000美元加上每个被赡养人需要的4000美元。

# 应用:一个基于逻辑的金融投资辅助决策程序

- 一位有三个人需赡养,有**\$22000**存款,每年有**\$25000**的稳定收入的投资者的情况,可产生一个由下列命题组成的逻辑系统:
  - 1.**savings (inadequate)** => **investment (savings)**.
  - 2.**savings (adequate)  $\wedge$  income (adequate )**  
=> **investment (stocks)**.
  - 3. **Savings (adequate)  $\wedge$  income (inadequate)**  
=> **investment (combination)**.
  - 4.  **$\forall X \text{ amountsaved}(X) \wedge \exists Y (\text{dependents}(Y) \wedge \text{greater}(X, \text{minsavings}(Y)))$**   
=> **savings (adequate)**.
  - 5.  **$\forall X \text{ amountsaved}(X) \wedge \exists Y (\text{dependents }(Y) \wedge \neg \text{greater}(X, \text{minsavings}(Y)))$**   
=> **savings (inadequate)**.

## 应用:一个基于逻辑的金融投资辅助决策程序

- 6.  $\forall X \text{ earnings}(X, \text{steady}) \wedge \exists Y (\text{dependents}(Y) \wedge \text{greater}(X, \text{minincome}(Y))) \Rightarrow \text{income}(\text{adequate}).$
- 7.  $\forall X \text{ earnings}(X, \text{steady}) \wedge \exists Y (\text{dependents}(Y) \wedge \neg \text{greater}(X, \text{minincome}(Y))) \Rightarrow \text{income}(\text{inadequate}).$
- 8.  $\forall X \text{ earnings}(X, \text{unsteady}) \Rightarrow \text{income}(\text{inadequate}).$
- 9. amountsaved(22000).
- 10. earnings(25000, steady).
- 11. dependents(3).

其中:  $\text{minsavings}(X) = 5000 * X;$

$\text{minincome}(X) = 15000 + (4000 * X)$

## 应用:一个基于逻辑的金融投资辅助决策程序

- 第一步把第10、11与第7的前提合一, 得:
- **12. Income (inadequate) .**
- 第二步把第9、11与第4的前提合一, 得:
- **13. savings (adequate)**

将第12、13条命题检验第3条命题得到其前提为真。  
于是运用假言推理后得到结论:  
**investment (combination),**  
这就是给投资者的建议。

(应该把他们多余的钱分别用于存款和买股票，在增加存款做保险的同时试图通过做股票以增加收入。 )

### 3.3 自然演绎推理

- 优点：定理证明过程自然，易于理解，并且有丰富的推理规则可用。
- 缺点：是容易产生知识爆炸，推理过程中得到的中间结论一般按指数规律递增，对于复杂问题的推理不利，甚至难以实现。

# 第3章 确定性推理

- 3.1 推理的基本概念
- 3.2 推理的逻辑基础
- 3.3 自然演绎推理
- 3.4 归结演绎推理
- 3.5 基于规则的演绎推理

## 3.4 归结演绎推理

- 归结演绎推理是一种基于鲁宾逊（Robinson）归结原理的机器推理技术。鲁宾逊归结原理亦称为消解原理，是鲁宾逊于1965年在海伯伦（Herbrand）理论的基础上提出的一种基于逻辑的“反证法”。
- 在人工智能中，几乎所有的问题都可以转化为一个定理证明问题。定理证明的实质，就是要对前提P和结论Q，证明 $P \rightarrow Q$ 永真。
- 由3.2节可知，要证明 $P \rightarrow Q$ 永真，就是要证明 $P \rightarrow Q$ 在任何一个非空的个体域上都是永真的。这将是非常困难的，甚至是不可实现的。
- 为此，人们进行了大量的探索，后来发现可以采用反证法的思想，把关于永真性的证明转化为关于不可满足性的证明。
- 即要证明 $P \rightarrow Q$ 永真，只要能够证明 $P \wedge \neg Q$ 是不可满足的就可以了（原因是 $\neg(P \rightarrow Q) \Leftrightarrow \neg(\neg P \vee Q) \Leftrightarrow P \wedge \neg Q$ ）。
- 这方面最有成效的工作就是鲁宾逊归结原理。它使定理证明的机械化成为现实。

## 3.4 归结演绎推理

- 3.4.1 子句集及其化简
- 3.4.2 鲁滨逊归结原理
- 3.4.3 归结反演推理的归结策略
- 3.4.4 用归结反演求取问题的答案

# 3.4.1 子句集及其化简

## 1. 子句和子句集

- 鲁滨逊归结原理是在子句集的基础上讨论问题的。因此，讨论归结演绎推理之前，需要先讨论子句集的有关概念。
- 子句和子句集
- 定义3.14 原子谓词公式及其否定统称为文字。
- 例如， $P(x)$ 、 $Q(x)$ 、 $\neg P(x)$ 、 $\neg Q(x)$ 等都是文字。
- 定义3.15 任何文字的析取式称为子句。
- 例如， $P(x) \vee Q(x)$ ， $P(x, f(x)) \vee Q(x, g(x))$ 都是子句。
- 定义3.16 不含任何文字的子句称为空子句。
- 由于空子句不含有任何文字，也就不能被任何解释所满足，因此空子句是永假的，不可满足的。
- 空子句一般被记为 $\square$ 或NIL。
- 定义3.17 由子句或空子句所构成的集合称为子句集。

## 3.4.1 子句集及其化简

### 2. 子句集的化简 (1/6)

- 在谓词逻辑中，任何一个谓词公式都可以通过应用等价关系及推理规则化成相应的子句集。其化简步骤如下：
  - (1) 消去连接词 “ $\rightarrow$ ” 和 “ $\leftrightarrow$ ”
  - 反复使用如下等价公式：
    - $P \rightarrow Q \Leftrightarrow \neg P \vee Q$
    - $P \leftrightarrow Q \Leftrightarrow (P \wedge Q) \vee (\neg P \wedge \neg Q)$
  - 即可消去谓词公式中的连接词 “ $\rightarrow$ ” 和 “ $\leftrightarrow$ ”。
  - 例如公式
    - $(\forall x)((\forall y)P(x,y) \rightarrow \neg(\forall y)(Q(x,y) \rightarrow R(x,y)))$
  - 经等价变化后为
    - $(\forall x)(\neg(\forall y)P(x,y) \vee \neg(\forall y)(\neg Q(x,y) \vee R(x,y)))$

## 3.4.1 子句集及其化简

### 2. 子句集的化简 (2/6)

- (2) 减少否定符号的辖域
- 反复使用双重否定律
  - $\neg(\neg P) \Leftrightarrow P$
- 摩根定律
  - $\neg(P \wedge Q) \Leftrightarrow \neg P \vee \neg Q$
  - $\neg(P \vee Q) \Leftrightarrow \neg P \wedge \neg Q$
- 量词转换律
  - $\neg(\forall x)P(x) \Leftrightarrow (\exists x) \neg P(x)$
  - $\neg(\exists x)P(x) \Leftrightarrow (\forall x) \neg P(x)$
- 将每个否定符号 “ $\neg$ ” 移到仅靠谓词的位置，使得每个否定符号最多只作用于一个谓词上。
- 例如，上式经等价变换后为
  - $(\forall x)((\exists y) \neg P(x, y) \vee (\exists y)(Q(x, y) \wedge \neg R(x, y)))$

## 3.4.1 子句集及其化简

### 2. 子句集的化简 (3/6)

- (3) 对变元标准化
  - 在一个量词的辖域内，把谓词公式中受该量词约束的变元全部用另外一个没有出现过的任意变元代替，使不同量词约束的变元有不同的名字。
  - 例如，上式经变换后为

$$(\forall x)((\exists y)\neg P(x, y) \vee (\exists z)(Q(x, z) \wedge \neg R(x, z)))$$

- (4) 化为前束范式
  - 化为前束范式的方法：把所有量词都移到公式的左边，并且在移动时不能改变其相对顺序。由于第(3)步已对变元进行了标准化，每个量词都有自己的变元，这就消除了任何由变元引起冲突的可能，因此这种移动是可行的。
  - 例如，上式化为前束范式后为

$$(\forall x)(\exists y)(\exists z)(\neg P(x, y) \vee (Q(x, z) \wedge \neg R(x, z)))$$

## 3.4.1 子句集及其化简

### 2. 子句集的化简 (4/6)

- (5) 消去存在量词
- 消去存在量词时，需要区分以下两种情况：
- 若存在量词不出现在全称量词的辖域内（即它的左边没有全称量词），只要用一个新的个体常量替换受该存在量词约束的变元，就可消去该存在量词。
- 若存在量词位于一个或多个全称量词的辖域内，例如 $(\forall x_1) \dots (\forall x_n) (\exists y) P(x_1, x_2, \dots, x_n, y)$
- 则需要用**Skolem**函数 $f(x_1, x_2, \dots, x_n)$ 替换受该存在量词约束的变元y，然后再消去该存在量词。
- 例如，上步所得公式中存在量词 $(\exists y)$ 和 $(\exists z)$ 都位于 $(\forall x)$ 的辖域内，因此都需要用**Skolem**函数来替换。设替换y和z的**Skolem**函数分别是 $f(x)$ 和 $g(x)$ ，则替换后的式子为 $(\forall x)(\neg P(x, f(x)) \vee (Q(x, g(x)) \wedge \neg R(x, g(x))))$

## 3.4.1 子句集及其化简

### 2. 子句集的化简 (5/6)

- (6) 化为Skolem标准形
- Skolem标准形的一般形式为 $(\forall x_1)\dots(\forall x_n) M(x_1, x_2, \dots, x_n)$
- 其中， $M(x_1, x_2, \dots, x_n)$ 是Skolem标准形的母式，它由子句的合取所构成。
- 把谓词公式化为Skolem标准形需要使用以下等价关系
- $P \vee (Q \wedge R) \Leftrightarrow (P \vee Q) \wedge (P \vee R)$
- 例如，前面的公式化为Skolem标准形后为 $(\forall x)((\neg P(x, f(x)) \vee Q(x, g(x))) \wedge (\neg P(x, f(x)) \vee \neg R(x, g(x))))$
- (7) 消去全称量词
- 由于母式中的全部变元均受全称量词的约束，并且全称量词的次序已无关紧要，因此可以省掉全称量词。但剩下的母式，仍假设其变元是被全称量词量化的。
- 例如，上式消去全称量词后为 $(\neg P(x, f(x)) \vee Q(x, g(x))) \wedge (\neg P(x, f(x)) \vee \neg R(x, g(x)))$

# 3.4.1 子句集及其化简

## 2. 子句集的化简 (6/6)

- **(8) 消去合取词**

在母式中消去所有合取词，把母式用子句集的形式表示出来。其中，子句集中的每一个元素都是一个子句。

- 例如，上式的子句集中包含以下两个子句

$$\neg P(x, f(x)) \vee Q(x, g(x))$$

$$\neg P(x, f(x)) \vee \neg R(x, g(x))$$

- **(9) 更换变量名称**

对子句集中的某些变量重新命名，使任意两个子句中不出现相同的变量名。由于每一个子句都对应着母式中的一个合取元，并且所有变元都是由全称量词量化的，因此任意两个不同子句的变量之间实际上不存在任何关系。这样，更换变量名是不会影响公式的真值的。

- 例如，对前面的公式，可把第二个子句集中的变元名x更换为y，得到如下子句集

$$\neg P(x, f(x)) \vee Q(x, g(x))$$

$$\neg P(y, f(y)) \vee \neg R(y, g(y))$$

## 练习：子句集化简

设： $X, Y, Z, W$  表示变元；  
 $l, m, n$  表示常元；  
 $a, b, c, d, e$  表示谓词名。

对下式进行子句集化简：

$$(\forall X) ([a(X) \wedge b(X)] \rightarrow [c(X, l) \wedge (\exists Y) ((\exists Z) [c(Y, Z)] \rightarrow d(X, Y))]) \vee (\forall X) (e(X))$$

$$\begin{aligned} & (\forall X) \\ & ( [a(X) \wedge b(X)] \rightarrow \\ & [c(X, l) \wedge (\exists Y) ((\exists Z) [c(Y, Z)] \rightarrow d(X, Y))] ) \\ & \vee \\ & (\forall X) (e(X)) \end{aligned}$$

# 练习：子句集化简

$$1. (\forall X) ([a(X) \wedge b(X)] \rightarrow [c(X, I) \wedge (\exists Y) ((\exists Z) [c(Y, Z) \rightarrow d(X, Y)])] ) \vee (\forall X) (e(X))$$

$$2. (\forall X) (\sim [a(X) \wedge b(X)] \vee [c(X, I) \wedge (\exists Y) ((\sim \exists Z) [c(Y, Z)] \vee d(X, Y))] ) \vee (\forall X) (e(X))$$

$$3. (\forall X) ([\sim a(X) \vee \sim b(X)] \vee [c(X, I) \wedge (\exists Y) ((\forall Z) [\sim c(Y, Z)] \vee d(X, Y))] ) \vee (\forall X) (e(X))$$

$$4. (\forall X) ([\sim a(X) \vee \sim b(X)] \vee [c(X, I) \wedge (\exists Y) ((\forall Z) [\sim c(Y, Z)] \vee d(X, Y))] ) \vee (\forall W) (e(W))$$


---

所有量词移到最左边而不改变其次序

$$5. (\forall X) (\exists Y) (\forall Z) (\forall W) ([\sim a(X) \vee \sim b(X)] \vee [c(X, I) \wedge [\sim c(Y, Z)] \vee d(X, Y)]) \vee e(W))$$

前束范式

斯柯伦标准化去掉所有的存在量词

# 练习：子句集化简

$$5. (\forall X)(\exists Y)(\forall Z)(\forall W) ([\sim a(X) \vee \sim b(X)] \vee [c(X, I) \wedge [\sim c(Y, Z) \vee d(X, Y)]])) \vee e(W)$$

斯柯伦标准化后

$$6. (\forall X)(\forall Z)(\forall W) ([\sim a(X) \vee \sim b(X)] \vee [c(X, I) \wedge [\sim c(f(X), Z) \vee d(X, f(X))]]) \vee e(W)$$

去掉全称量词

$$7. [\sim a(X) \vee \sim b(X)] \vee [c(X, I) \wedge (\sim c(f(X), Z) \vee d(X, f(X)))] \vee e(W)$$

转换成析取式的合取

$$8. [\sim a(X) \vee \sim b(X) \vee c(X, I) \vee e(W)] \wedge [\sim a(X) \vee \sim b(X) \vee \sim c(f(X), Z) \vee d(X, f(X)) \vee e(W)]$$

每个合取式为一个分离的子句

$$9a: \sim a(X) \vee \sim b(X) \vee c(X, I) \vee e(W)$$

$$9b: \sim a(X) \vee \sim b(X) \vee \sim c(f(X), Z) \vee d(X, f(X)) \vee e(W)$$



$$10a: \sim a(X) \vee \sim b(X) \vee c(X, I) \vee e(W)$$

$$10b: \sim a(U) \vee \sim b(U) \vee \sim c(f(U), Z) \vee d(U, f(U)) \vee e(V)$$

# 3.4.1 子句集及其化简

## 3. 子句集的应用 (1/4)

- 在上述化简过程中，由于在消去存在量词时所用的Skolem函数可以不同，因此化简后的标准子句集是不唯一的。
- 这样，当原谓词公式为非永假时，它与其标准子句集并不等价。但当原谓词公式为永假（或不可满足）时，其标准子句集则一定是永假的，即Skolem化并不影响原谓词公式的永假性。
- 这个结论很重要，是归结原理的主要依据，可用定理的形式来描述。
- 定理3.1 设有谓词公式F，其标准子句集为S，则F为不可满足的充要条件是S为不可满足的。
- 为证明此定理，先作如下说明：
- 为讨论问题方便，设给定的谓词公式F已为前束形 $(Q_1x_1)\dots(Q_rx_r)\dots(Q_nx_n)M(x_1, x_2, \dots, x_n)$
- 其中， $M(x_1, x_2, \dots, x_n)$ 已化为合取范式。
- 由于将F化为这种前束形是一种很容易实现的等价运算，因此这种假设是可以的。

# 3.4.1 子句集及其化简

## 3. 子句集的应用 (2/4)

- 又设( $Q_r x_r$ )是第一个出现的存在量词( $\exists x_r$ ), 即F为  
 $F = (\forall x_1) \dots (\forall x_{r-1}) (\exists x_r) (Q_{r+1} x_{r+1}) \dots (Q_n x_n) M(x_1, \dots, x_{r-1}, x_r, x_{r+1}, \dots, x_n)$
- 为把F化为Skolem形, 需要先消去这个( $\exists x_r$ ), 并引入Skolem函数, 得到  
 $F_1 = (\forall x_1) \dots (\forall x_{r-1}) (Q_{r+1} x_{r+1}) \dots (Q_n x_n) M(x_1, \dots, x_{r-1}, f(x_1, \dots, x_{r-1}), x_{r+1}, \dots, x_n)$
- 若能证明  
 $F \text{不可满足} \Leftrightarrow F_1 \text{不可满足}$
- 则同理可证  
 $F_1 \text{不可满足} \Leftrightarrow F_2 \text{不可满足}$
- 重复这一过程, 直到证明了  
 $F_{m-1} \text{不可满足} \Leftrightarrow F_m \text{不可满足}$
- 为止。
- 此时,  $F_m$ 已为F的Skolem标准形。而S只不过是 $F_m$ 的一种集合表示形式。  
因此有
- $F_m \text{不可满足} \Leftrightarrow S \text{不可满足}$
- 下面开始用反证法证明  
 $F \text{不可满足} \Leftrightarrow F_1 \text{不可满足}$

## 3.4.1 子句集及其化简

### 3. 子句集的应用 (3/4)

- 先证明  $\Rightarrow$
- 已知  $F$  不可满足，假设  $F_1$  是可满足的，则存在一个解释  $I$ ，使  $F_1$  在解释  $I$  下为真。即对任意  $x_1, \dots, x_{r-1}$  在  $I$  的设定下有
$$(Q_{r+1}x_{r+1}) \dots (Q_n x_n) M(x_1, \dots, x_{r-1}, f(x_1, \dots, x_{r-1}), x_{r+1} \dots, x_n)$$
- 为真。亦即对任意的  $x_1, \dots, x_{r-1}$  都有一个  $f(x_1, \dots, x_{r-1})$  使
$$(Q_{r+1}x_{r+1}) \dots (Q_n x_n) M(x_1, \dots, x_{r-1}, f(x_1, \dots, x_{r-1}), x_{r+1} \dots, x_n)$$
- 为真。即在  $I$  下有
$$(\forall x_1) \dots (\forall x_{r-1}) (\exists x_r) (Q_{r+1}x_{r+1}) \dots (Q_n x_n) M(x_1, \dots, x_{r-1}, x_r, x_{r+1} \dots, x_n)$$
- 为真。即  $F$  在  $I$  下为真。
- 但这与前提  $F$  是不可满足的相矛盾，即假设  $F_1$  为可满足是错误的。从而可以得出“若  $F$  不可满足，则必有  $F_1$  不可满足”。

## 3.4.1 子句集及其化简

### 3. 子句集的应用 (4/4)

- 再证明  $\Leftarrow$
- 已知  $F_1$  不可满足，假设  $F$  是可满足的。于是便有某个解释  $I$  使  $F$  在  $I$  下为真。即对任意的  $x_1, \dots, x_{r-1}$  在  $I$  的设定下都可找到一个  $x_r$  使
$$(Q_{r+1}x_{r+1}) \dots (Q_n x_n) M(x_1, \dots, x_{r-1}, x_r, x_{r+1}, \dots, x_n)$$
- 为真。若扩充  $I$ ，使它包含一个函数  $f(x_1, \dots, x_{r-1})$ ，且有
$$x_r = f(x_1, \dots, x_{r-1})$$
- 这样，就可以把所有的  $f(x_1, \dots, x_{r-1})$  映射到  $x_r$ ，从而得到一个新的解释  $I'$ ，并且在此解释下对任意的  $x_1, \dots, x_{r-1}$  都有
$$(Q_{r+1}x_{r+1}) \dots (Q_n x_n) M(x_1, \dots, x_{r-1}, f(x_1, \dots, x_{r-1}), x_{r+1}, \dots, x_n)$$
- 为真。即在  $I'$  下有
$$(\forall x_1) \dots (\forall x_{r-1}) (Q_{r+1}x_{r+1}) \dots (Q_n x_n) M(x_1, \dots, x_{r-1}, f(x_1, \dots, x_{r-1}), x_{r+1}, \dots, x_n)$$
- 为真。它说明  $F_1$  在解释  $I'$  下为真。但这与前提  $F_1$  是不可满足的相矛盾，即假设  $F$  为可满足是错误的。从而可以得出“若  $F_1$  不可满足，则必有  $F$  不可满足”
- 于是，定理得证。
- 由此定理可知，要证明一个谓词公式是不可满足的，只要证明其相应的标准子句集是不可满足的就可以了。至于如何证明一个子句集的不可满足性，由下面的海伯伦理论和鲁宾逊归结原理来解决。

## 3.4.2 鲁滨逊归结原理

### 1. 基本思想

- 注意以下两个关键
- 第一，子句集中的子句之间是合取关系。因此，子句集中只要有一个子句为不可满足，则整个子句集就是不可满足的；
- 第二，空子句是不可满足的。因此，一个子句集中如果包含有空子句，则此子句集就一定是不可满足的。
- 鲁滨逊归结原理基本思想
- 首先把欲证明问题的结论否定，并加入子句集，得到一个扩充的子句集 $S'$ 。然后设法检验子句集 $S'$ 是否含有空子句，若含有空子句，则表明 $S'$ 是不可满足的；若不含有空子句，则继续使用归结法，在子句集中选择合适的子句进行归结，直至导出空子句或不能继续归结为止。
- 鲁滨逊归结原理包括
- 命题逻辑归结原理
- 谓词逻辑归结原理

## 3.4.2 鲁滨逊归结原理

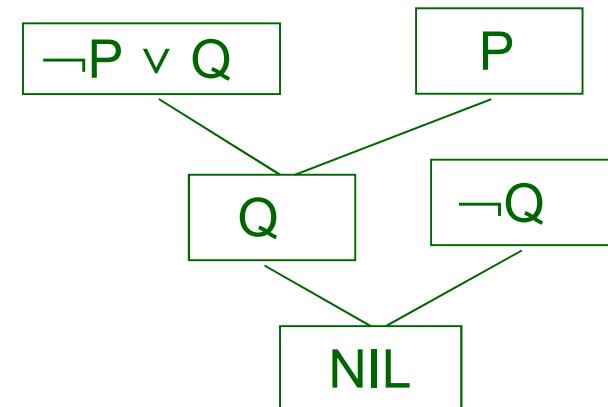
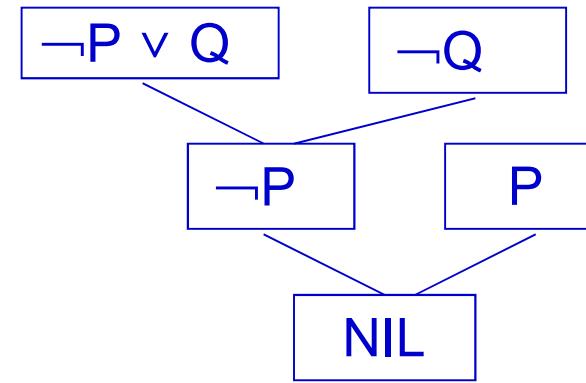
### 2. 命题逻辑的归结 (1/8)

- 归结推理的核心是求两个子句的归结式
- (1) 归结式的定义及性质
- 定义3.18 若P是原子谓词公式，则称P与 $\neg P$ 为互补文字。
- 定义3.19 设 $C_1$ 和 $C_2$ 是子句集中的任意两个子句，如果 $C_1$ 中的文字 $L_1$ 与 $C_2$ 中的文字 $L_2$ 互补，那么可从 $C_1$ 和 $C_2$ 中分别消去 $L_1$ 和 $L_2$ ，并将 $C_1$ 和 $C_2$ 中余下的部分按析取关系构成一个新的子句 $C_{12}$ ，则称这一过程为归结，称 $C_{12}$ 为 $C_1$ 和 $C_2$ 的归结式，称 $C_1$ 和 $C_2$ 为 $C_{12}$ 的亲本子句。
- 例3.7 设 $C_1=P \vee Q \vee R$ ,  $C_2=\neg P \vee S$ , 求 $C_1$ 和 $C_2$ 的归结式 $C_{12}$ 。  
解：这里 $L_1=P$ ,  $L_2=\neg P$ , 通过归结可以得到  
$$C_{12} = Q \vee R \vee S$$
- 例3.8 设 $C_1=\neg Q$ ,  $C_2=Q$ , 求 $C_1$ 和 $C_2$ 的归结式 $C_{12}$ 。  
解：这里 $L_1=\neg Q$ ,  $L_2=Q$ , 通过归结可以得到  
$$C_{12} = \text{NIL}$$

## 3.4.2 鲁滨逊归结原理

### 2. 命题逻辑的归结 (2/8)

- 例3.9 设设 $C_1 = \neg P \vee Q$ ,  $C_2 = \neg Q$ ,  $C_3 = P$ , 求 $C_1$ 、 $C_2$ 、 $C_3$ 的归结式 $C_{123}$ 。
- 解：若先对 $C_1$ 、 $C_2$ 归结，可得到
- $C_{12} = \neg P$
- 然后再对 $C_{12}$ 和 $C_3$ 归结，得到
- $C_{123} = \text{NIL}$
- 如果改变归结顺序，同样可以得到相同的结果，即其归结过程是不唯一的。
- 其归结归结过程可用右图来表示，该树称为归结树。



## 3.4.2 鲁滨逊归结原理

### 2. 命题逻辑的归结 (3/8)

- 定理3.2 归结式 $C_{12}$ 是其亲本子句 $C_1$ 和 $C_2$ 的逻辑结论。
- 证明：（按定义）设 $C_1=L \vee C_1'$ ,  $C_2=\neg L \vee C_2'$ 关于解释I为真，则只需证明 $C_{12}=C_1' \vee C_2'$ 关于解释I也为真。
  - 对于解释I,  $L$ 和 $\neg L$ 中必有一个为假。
  - 若 $L$ 为假，则必有 $C_1'$ 为真，不然就会使 $C_1$ 为假，这将与前提假设 $C_1$ 为真矛盾，因此只能有 $C_1'$ 为真。
  - 同理，若 $\neg L$ 为假，则必有 $C_2'$ 为真。
  - 因此，必有 $C_{12}=C_1' \vee C_2'$ 关于解释I也为真。即 $C_{12}$ 是 $C_1$ 和 $C_2$ 的逻辑结论。

## 3.4.2 鲁滨逊归结原理

### 2. 命题逻辑的归结 (4/8)

- 上述定理是归结原理中的一个重要定理，由它可得到以下两个推论：
- 推论1：设 $C_1$ 和 $C_2$ 是子句集 $S$ 中的两个子句， $C_{12}$ 是 $C_1$ 和 $C_2$ 的归结式，若用 $C_{12}$ 代替 $C_1$ 和 $C_2$ 后得到新的子句集 $S_1$ ，则由 $S_1$ 的不可满足性可以推出原子句集 $S$ 的不可满足性。即：  
 $S_1$ 的不可满足性  $\Rightarrow S$ 的不可满足性
- 证明：设 $S=\{C_1, C_2, C_3, \dots, C_n\}$ ， $C_{12}$ 是 $C_1$ 和 $C_2$ 的归结式，则用 $C_{12}$ 代替 $C_1$ 和 $C_2$ 后可得到一个新的子句集  
 $S_1=\{C_{12}, C_3, \dots, C_n\}$
- 设 $S_1$ 是不可满足的，则对不满足 $S_1$ 的任一解释 $I$ ，都可能有以下两种情况：
  - ① 解释 $I$ 使 $C_{12}$ 为真，则 $C_3, \dots, C_n$ 中必有一个为假，即 $S$ 是不可满足的。
  - ② 解释 $I$ 使 $C_{12}$ 为假，即 $\neg C_{12}$ 为真，根据定理3.2有 $\neg(C_1 \wedge C_2)$ 永真，即 $\neg C_1 \vee \neg C_2$ 永真，它说明解释 $I$ 使 $C_1$ 为假，或 $C_2$ 为假。即 $S$ 也是不可满足的。
- 因此可以得出 置换律：  
 $(P \rightarrow Q) \Leftrightarrow (\neg Q \rightarrow \neg P)$
- $S_1$ 的不可满足性  $\Rightarrow S$ 的不可满足性

## 3.4.2 鲁滨逊归结原理

### 2. 命题逻辑的归结 (5/8)

- **推论2：**设 $C_1$ 和 $C_2$ 是子句集 $S$ 中的两个子句， $C_{12}$ 是 $C_1$ 和 $C_2$ 的归结式，若把 $C_{12}$ 加入 $S$ 中得到新的子句集 $S_2$ ，则 $S$ 与 $S_2$ 的不可满足性是等价的。即：  
 $S_2$ 的不可满足性 $\Leftrightarrow S$ 的不可满足性
- 先证明  $\Leftarrow$
- 设 $S=\{C_1, C_2, C_3, \dots, C_n\}$ 是不可满足的，则 $C_1, C_2, C_3, \dots, C_n$ 中必有一子句为假，因而 $S_2=\{C_{12}, C_1, C_2, C_3, \dots, C_n\}$ 必为不可满足。
- 再证明  $\Rightarrow$
- 设 $S_2$ 是不可满足的，则对不满足 $S_2$ 的任一解释 $I$ ，都可能有以下两种情况：
- ① 解释 $I$ 使 $C_{12}$ 为真，则 $C_1, C_2, C_3, \dots, C_n$ 中必有一个为假，即 $S$ 是不可满足的。
- ② 解释 $I$ 使 $C_{12}$ 为假，即 $\neg C_{12}$ 为真，根据定理3.2有  $\neg(C_1 \wedge C_2)$  永真，即  $\neg C_1 \vee \neg C_2$  永真，它说明解释 $I$ 使 $C_1$ 为假，或 $C_2$ 为假。即 $S$ 也是不可满足的。
- 由此可见 $S$ 与 $S_2$ 的不可满足性是等价的。即  
 $S$ 的不可满足性 $\Leftrightarrow S_2$ 的不可满足性

## 3.4.2 鲁滨逊归结原理

### 2. 命题逻辑的归结 (6/8)

- 上述两个推论说明，为证明子句集 $S$ 的不可满足性，只要对其中可进行归结得子句进行归结，并把归结式加入到子句集 $S$ 中，或者用归结式代替他的亲本子句，然后对新的子句集证明其不可满足性就可以了。
- 如果经归结能得到空子句，根据空子句的不可满足性，即可得到原子句集 $S$ 是不可满足的结论。
- 在命题逻辑中，对不可满足的子句集 $S$ ，其归结原理是完备的。
  - 这种不可满足性可用如下定理描述：
  - 定理3.3** 子句集 $S$ 是不可满足的，当且仅当存在一个从 $S$ 到空子句的归结过程。

# 证明

- 空子句即为“假” ( $F$ )，推出空子句即为推出 $F$
- 但归结原理是推理规则，由正确的推理形式推出了 $F$ ，则说明前提非真，即归结出空子句的两个亲本子句中至少有一个为假
- 而这两个亲本子句如果都是原子句集中的子句，考虑到子句集中各子句间为合取关系，一个子句为假即可说明原子句集是不可满足的。
- 如果这两个亲本子句不是或不全是原子句集中的子句，而是某次归结出的结果，不妨同理向前追溯，同样可以推出原子句中至少有一个子句为假
- 从而说明子句集是不可满足的。

## 3.4.2 鲁滨逊归结原理

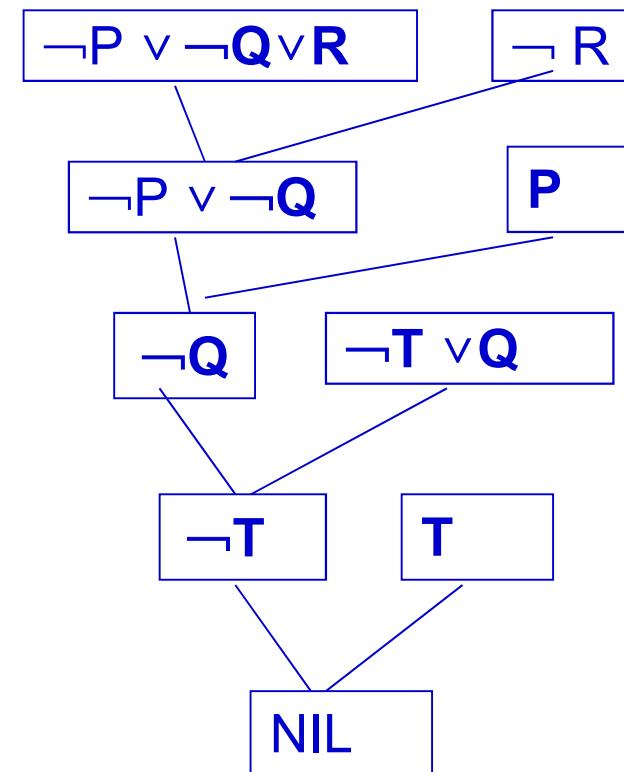
### 2. 命题逻辑的归结 (7/8)

- (2) 命题逻辑的归结反演
- 归结原理：假设F为已知前提，G为欲证明的结论，归结原理把证明G为F的逻辑结论转化为证明 $F \wedge \neg G$ 为不可满足。
- 再根据定理3.1，在不可满足的意义上，公式集 $F \wedge \neg G$ 与其子句集是等价的，即把公式集上的不可满足转化为子句集上的不可满足。
- 应用归结原理证明定理的过程称为归结反演。  
在命题逻辑中，已知F，证明G为真的归结反演过程如下：
  - ① 否定目标公式G，得 $\neg G$ ；
  - ② 把 $\neg G$ 并入到公式集F中，得到 $\{F, \neg G\}$ ；
  - ③ 把 $\{F, \neg G\}$ 化为子句集S。
  - ④ 应用归结原理对子句集S中的子句进行归结，并把每次得到的归结式并入S中。如此反复进行，若出现空子句，则停止归结，此时就证明了G为真。

## 3.4.2 鲁滨逊归结原理

### 2. 命题逻辑的归结 (8/8)

- 例3.10 设已知的公式集为 $\{P, (P \wedge Q) \rightarrow R, (S \vee T) \rightarrow Q, T\}$ , 求证结论R。  
解: 假设结论R为假, 将 $\neg R$ 加入公式集, 并化为子句集
- $S=\{P, \neg P \vee \neg Q \vee R, \neg S \vee Q, \neg T \vee Q, T, \neg R\}$
- 其归结过程如右图的归结演绎树所示。该树根为空子句。
- 其含义为: 先假设子句集S中的所有子句均为真, 即原公式集为真,  $\neg R$ 也为真; 然后利用归结原理, 对子句集进行归结, 并把所得的归结式并入子句集中; 重复这一过程, 最后归结出了空子句。
- 根据归结原理的完备性, 可知子句集S是不可满足的, 即开始时假设 $\neg R$ 为真是错误的, 这就证明了R为真。



## 3.4.2 鲁滨逊归结原理

### 3. 谓词逻辑的归结 (1/16)

- 在谓词逻辑中，由于子句集中的谓词一般都含有变元，因此不能象命题逻辑那样直接消去互补文字。而需要先用一个最一般合一对变元进行代换，然后才能进行归结。
- 谓词逻辑的归结原理
- 谓词逻辑中的归结式可用如下定义来描述：
- 定义3.20 设 $C_1$ 和 $C_2$ 是两个没有公共变元的子句， $L_1$ 和 $L_2$ 分别是 $C_1$ 和 $C_2$ 中的文字。如果 $L_1$ 和 $L_2$ 存在最一般合一 $\sigma$ ，则称
  - $C_{12} = (\{C_1\sigma\} - \{L_1\sigma\}) \cup (\{C_2\sigma\} - \{L_2\sigma\})$
  - 为 $C_1$ 和 $C_2$ 的二元归结式，而 $L_1$ 和 $L_2$ 为归结式上的文字。

## 3.4.2 鲁滨逊归结原理

### 3. 谓词逻辑的归结 (2/16)

- 例3.11 设  $C_1 = P(a) \vee R(x)$ ,  $C_2 = \neg P(y) \vee Q(b)$ , 求  $C_{12}$
- 解: 取  $L_1 = P(a)$ ,  $L_2 = \neg P(y)$ , 则  $L_1$  和  $L_2$  的最一般合一是  $\sigma = \{a/y\}$ 。根据定义3.20, 可得
  - $C_{12} = (\{C_1\sigma\} - \{L_1\sigma\}) \cup (\{C_2\sigma\} - \{L_2\sigma\})$
  - $= (\{P(a), R(x)\} - \{P(a)\}) \cup (\{\neg P(a), Q(b)\} - \{\neg P(a)\})$
  - $= (\{R(x)\}) \cup (\{Q(b)\}) = \{R(x), Q(b)\}$
  - $= R(x) \vee Q(b)$
- 例3.12 设  $C_1 = P(x) \vee Q(a)$ ,  $C_2 = \neg P(b) \vee R(x)$ , 求  $C_{12}$
- 解: 由于  $C_1$  和  $C_2$  有相同的变元  $x$ , 不符合定义3.20的要求。为了进行归结, 需要修改  $C_2$  中变元的名字, 令  $C_2 = \neg P(b) \vee R(y)$ 。此时  $L_1 = P(x)$ ,  $L_2 = \neg P(b)$ ,  $L_1$  和  $L_2$  的最一般合一是  $\sigma = \{b/x\}$ 。则有
  - $C_{12} = (\{C_1\sigma\} - \{L_1\sigma\}) \cup (\{C_2\sigma\} - \{L_2\sigma\})$
  - $= (\{P(b), Q(a)\} - \{P(b)\}) \cup (\{\neg P(b), R(y)\} - \{\neg P(b)\})$
  - $= (\{Q(a)\}) \cup (\{R(y)\}) = \{Q(a), R(y)\}$
  - $= Q(a) \vee R(y)$

## 3.4.2 鲁滨逊归结原理

### 3. 谓词逻辑的归结 (3/16)

- 对以上讨论做以下两点说明：
- (1) 这里之所以使用集合符号和集合的运算，目的是为了说明问题的方便。
- 即先将子句 $C_i\sigma$ 和 $L_i\sigma$ 写成集合的形式，在集合表示下做减法和并集运算，然后再写成子句集的形式。
- (2) 定义中还要求 $C_1$ 和 $C_2$ 无公共变元，这也是合理的。  
例如 $C_1=P(x)$ ,  $C_2=\neg P(f(x))$ , 而 $S=\{C_1, C_2\}$ 是不可满足的。但由于 $C_1$ 和 $C_2$ 的变元相同，就无法合一了。没有归结式，就不能用归结法证明 $S$ 的不可满足性，这就限制了归结法的使用范围。
- 如果对 $C_1$ 或 $C_2$ 的变元进行换名，便可通过合一，对 $C_1$ 和 $C_2$ 进行归结。如上例，若先对 $C_2$ 进行换名，即 $C_2=\neg P(f(y))$ ，则可对 $C_1$ 和 $C_2$ 进行归结，得到一个空子句，从而证明了 $S$ 是不可满足的。
- 事实上，在由公式集化为子句集的过程中，其最后一步就是做换名处理。<sup>77</sup>因此，定义中假设 $C_1$ 和 $C_2$ 没有相同变元是可以的。

## 3.4.2 鲁滨逊归结原理

### 3. 谓词逻辑的归结 (4/16)

- 例3.13 设  $C_1 = P(x) \vee \neg Q(b)$ ,  $C_2 = \neg P(a) \vee Q(y) \vee R(z)$
- 解：对  $C_1$  和  $C_2$  通过最一般合一 ( $\sigma = \{a/x, b/y\}$ ) 的作用，可以得到两个互补对。
- 注意：求归结式不能同时消去两个互补对，这样的结果不是二元归结式。如在  $\sigma = \{a/x, b/y\}$  下，若同时消去两个互补对，所得的  $R(z)$  不是  $C_1$  和  $C_2$  的二元归结式。
- 例3.14 设  $C_1 = P(x) \vee P(f(a)) \vee Q(x)$ ,  $C_2 = \neg P(y) \vee R(b)$ , 求  $C_{12}$
- 解：对参加归结的某个子句，若其内部有可合一的文字，则在进行归结之前应先对这些文字进行合一。本例的  $C_1$  中有可合一的文字  $P(x)$  与  $P(f(a))$ ，若用它们的最一般合一  $\sigma = \{f(a)/x\}$  进行代换，可得到
- $C_1\sigma = P(f(a)) \vee Q(f(a))$
- 此时可对  $C_1\sigma$  与  $C_2$  进行归结。选  $L_1 = P(f(a))$ ,  $L_2 = \neg P(y)$ ,  $L_1$  和  $L_2$  的最一般合一是  $\sigma = \{f(a)/y\}$ ，则可得到  $C_1$  和  $C_2$  的二元归结式为
- $C_{12} = R(b) \vee Q(f(a))$
- 我们把  $C_1\sigma$  称为  $C_1$  的因子。一般来说，若子句  $C$  中有两个或两个以上的文字具有最一般合一  $\sigma$ ，则称  $C\sigma$  为子句  $C$  的因子。如果  $C\sigma$  是一个单文字，则称它为  $C$  的单元因子。
- 应用因子概念，可对谓词逻辑中的归结原理给出如下定义：

## 3.4.2 鲁滨逊归结原理

### 3. 谓词逻辑的归结 (5/16)

- 定义3.21若 $C_1$ 和 $C_2$ 是无公共变元的子句，则
  - ①  $C_1$ 和 $C_2$ 的二元归结式；
  - ②  $C_1$ 和 $C_2$ 的因子 $C_2\sigma_2$ 的二元归结式；
  - ③  $C_1$ 的因子 $C_1\sigma_1$ 和 $C_2$ 的二元归结式；
  - ④  $C_1$ 的因子 $C_1\sigma_1$ 和 $C_2$ 的因子 $C_2\sigma$ 的二元归结式。
- 这四种二元归结式都是子句 $C_1$ 和 $C_2$ 的二元归结式，记为 $C_{12}$ 。
- 例3.15 设 $C_1=P(y) \vee P(f(x)) \vee Q(g(x))$ ,  $C_2=\neg P(f(g(a))) \vee Q(b)$ , 求 $C_{12}$ 。  
解：对 $C_1$ ，取最一般合一 $\sigma=\{f(x)/y\}$ ，得 $C_1$ 的因子  
 $C_1\sigma=P(f(x)) \vee Q(g(x))$
- 对 $C_1$ 的因子和 $C_2$ 归结 ( $\sigma=\{g(a)/x\}$ )，可得到 $C_1$ 和 $C_2$ 的二元归结式  
 $C_{12}=Q(g(g(a))) \vee Q(b)$
- 说明：
  - 对谓词逻辑，定理3.2仍然适用，即归结式 $C_{12}$ 是其亲本子句 $C_1$ 和 $C_2$ 的逻辑结论。用归结式取代它在子句集 $S$ 中的亲本子句，所得到的子句集仍然保持着原子句集 $S$ 的不可满足性。
  - 此外，对谓词逻辑定理3.3也仍然适用，即从不可满足的意义上说，一阶谓词逻辑的归结原理也是完备的

## 3.4.2 鲁滨逊归结原理

### 3. 谓词逻辑的归结 (6/16)

- 谓词逻辑的归结反演
- 谓词逻辑的归结反演过程与命题逻辑的归结反演过程相比，其步骤基本相同，但每步的处理对象不同。例如，在步骤(3)化简子句集时，谓词逻辑需要把由谓词构成的公式集化为子句集；在步骤(4)按归结原理进行归结时，谓词逻辑的归结原理需要考虑两个亲本子句的最一般合一。
- 例3.16 已知
  - $F: (\forall x)((\exists y)(A(x, y) \wedge B(y)) \rightarrow (\exists y)(C(y) \wedge D(x, y)))$
  - $G: \neg(\exists x)C(x) \rightarrow (\forall x)(\forall y)(A(x, y) \rightarrow \neg B(y))$
- 求证G是F的逻辑结论。
- 证明：先把G否定，并放入F中，得到的 $\{F, \neg G\}$ 为
  - $\{(\forall x)((\exists y)(A(x, y) \wedge B(y)) \rightarrow (\exists y)(C(y) \wedge D(x, y))),$
  - $\neg(\neg(\exists x)C(x) \rightarrow (\forall x)(\forall y)(A(x, y) \rightarrow \neg B(y)))\}$

$$F: (\forall x)(\exists y)(A(x, y) \wedge B(y)) \rightarrow (\exists y)(C(y) \wedge D(x, y))$$

$$G: \neg(\exists x)C(x) \rightarrow (\forall x)(\forall y)(A(x, y) \rightarrow \neg B(y))$$

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

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

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

前束范例

$$\frac{\forall x \forall y \exists u (\neg A(x, y) \vee \neg B(y) \vee C(u))}{\neg \exists u (\neg A(x, y) \vee \neg B(y) \vee D(x, u))} \quad u = f(x)$$

消去存在量词

$$\textcircled{1} \quad \neg A(x, y) \vee \neg B(y) \vee C(f(x))$$

$$\textcircled{2} \quad \neg A(x, y) \underset{u}{\vee} \neg B(y) \underset{v}{\vee} D(x, f(x)) \quad \rightarrow \text{更换变元名}$$

$$F: (\forall x)((\exists y)(A(x, y) \wedge B(y)) \rightarrow (\exists y)(C(y) \wedge D(x, y)))$$

$$G: \neg(\exists x)C(x) \rightarrow (\forall x)(\forall y)(A(x, y) \rightarrow \neg B(y))$$

$$\neg G: \neg(\neg(\exists x)C(x) \rightarrow (\forall x)(\forall y)(A(x, y) \rightarrow \neg B(y)))$$

$$G_I: \neg \exists x C(x) \vee \forall x \forall y (\neg A(x, y) \vee \neg B(y))$$

$$\neg G: \neg \exists x C(x) \wedge \neg \forall x \forall y (\neg A(x, y) \vee \neg B(y))$$

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

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

消存在量词，用常量代替

$$\forall x \neg C(x) \wedge A(m, n) \wedge B(n)$$

③  $\neg C(\underline{x})$  和 ①, ③ 中的  $x$  要区分  $\rightarrow \neg C(z)$

④  $A(m, n)$

⑤  $B(n)$

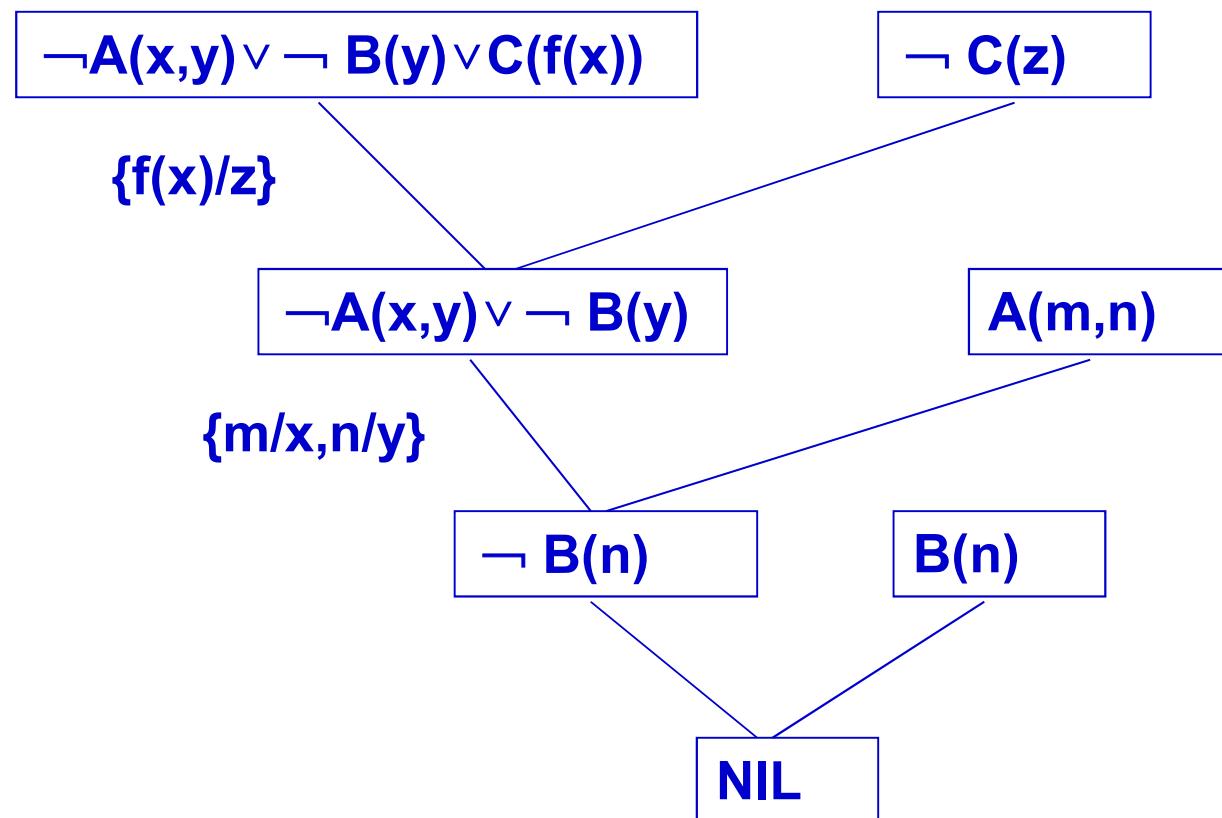
## 3.4.2 鲁滨逊归结原理

### 3. 谓词逻辑的归结 (7/16)

- 再把 $\{F, \neg G\}$ 化成子句集，得到
  - (1)  $\neg A(x,y) \vee \neg B(y) \vee C(f(x))$
  - (2)  $\neg A(u,v) \vee \neg B(v) \vee D(u,f(u))$
  - (3)  $\neg C(z)$
  - (4)  $A(m,n)$
  - (5)  $B(n)$
- 其中，(1)、(2)是由 $F$ 化出的两个子句，(3)、(4)、(5)是由 $\neg G$ 化出的3个子句。
- 最后应用谓词逻辑的归结原理对上述子句集进行归结，其过程为
  - (6)  $\neg A(x,y) \vee \neg B(y)$  由(1)和(3)归结，取 $\sigma = \{f(x)/z\}$
  - (7)  $\neg B(n)$  由(4)和(6)归结，取 $\sigma = \{m/x, n/y\}$
  - (8) **NIL** 由(5)和(7)归结
- 因此， $G$ 是 $F$ 的逻辑结论。
- 上述归结过程可用如下归结树来表示

## 3.4.2 鲁滨逊归结原理

### 3. 谓词逻辑的归结 (8/16)



## 3.4.2 鲁滨逊归结原理

### 3. 谓词逻辑的归结 (9/16)

- 为了进一步加深对谓词逻辑归结的理解，下面再给出一个经典的归结问题
- 例3.17 “快乐学生” 问题
- 假设：任何通过计算机考试并获奖的人都是快乐的，任何肯学习或幸运的人都可以通过所有考试，张不肯学习但他是幸运的，任何幸运的人都能获奖。
- 求证：张是快乐的。
- 解：先定义谓词：
  - $\text{Pass}(x, y)$   $x$ 可以通过 $y$ 考试
  - $\text{Win}(x, \text{prize})$   $x$ 能获得奖励
  - $\text{Study}(x)$   $x$ 肯学习
  - $\text{Happy}(x)$   $x$ 是快乐的
  - $\text{Lucky}(x)$   $x$ 是幸运的

## 3.4.2 鲁滨逊归结原理

### 3. 谓词逻辑的归结 (10/16)

- 再将问题用谓词表示如下：
- “任何通过计算机考试并奖的人都是快乐的”
- $(\forall x)(\text{Pass}(x, \text{computer}) \wedge \text{Win}(x, \text{prize}) \rightarrow \text{Happy}(x))$
- “任何肯学习或幸运的人都可以通过所有考试”
- $(\forall x) (\forall y) (\text{Study}(x) \vee \text{Lucky}(x) \rightarrow \text{Pass}(x, y))$
- “张不肯学习但他是幸运的”
- $\neg \text{Study}(\text{zhang}) \wedge \text{Lucky}(\text{zhang})$
- “任何幸运的人都能获奖”
- $(\forall x) (\text{Lucky}(x) \rightarrow \text{Win}(x, \text{prize}))$
- 结论“张是快乐的”的否定
- $\neg \text{Happy}(\text{zhang})$

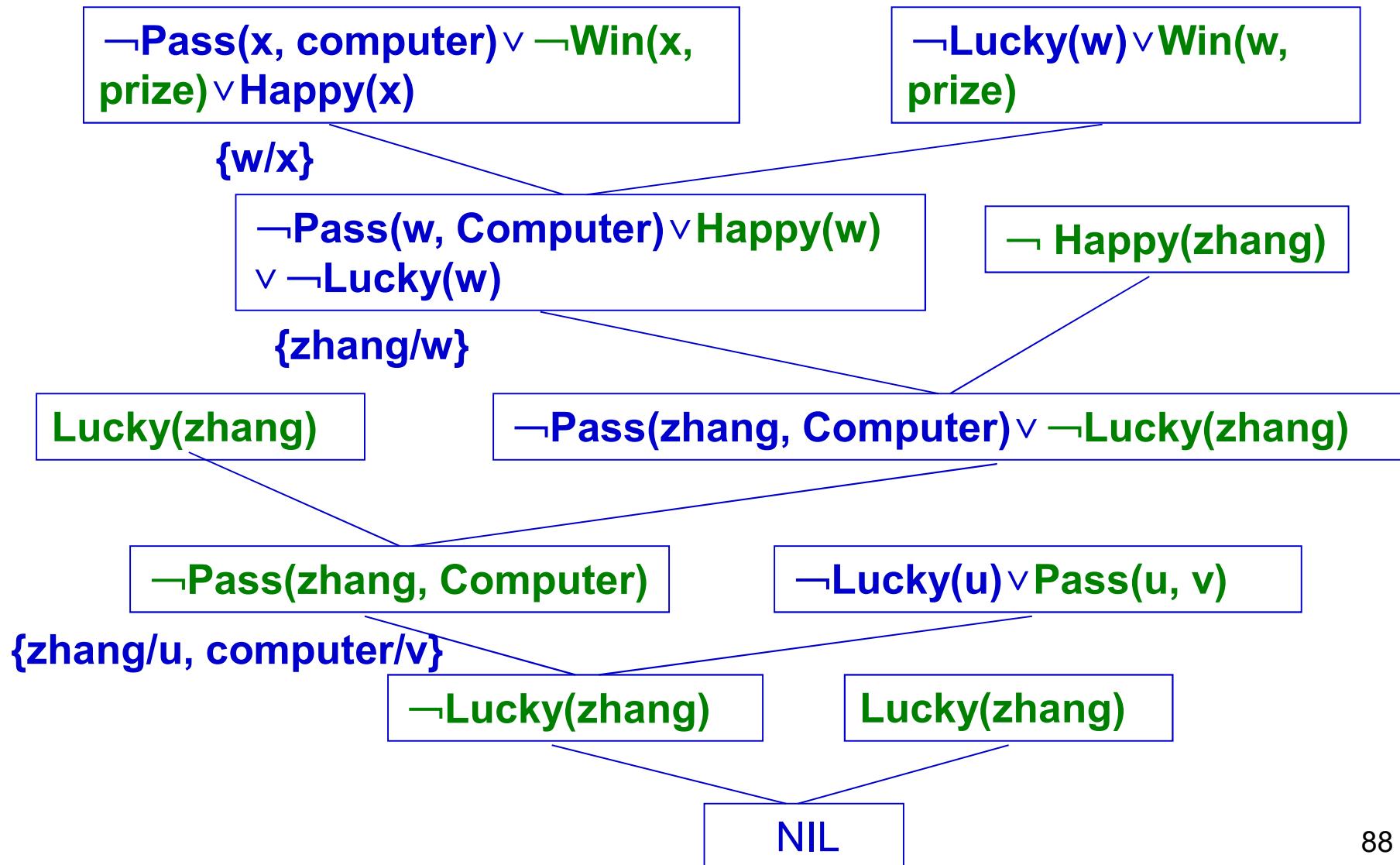
## 3.4.2 鲁滨逊归结原理

### 3. 谓词逻辑的归结 (11/16)

- 将上述谓词公式转化为子句集如下：
- (1)  $\neg \text{Pass}(x, \text{computer}) \vee \neg \text{Win}(x, \text{prize}) \vee \text{Happy}(x)$
- (2)  $\neg \text{Study}(y) \vee \text{Pass}(y, z)$
- (3)  $\neg \text{Lucky}(u) \vee \text{Pass}(u, v)$
- (4)  $\neg \text{Study}(\text{zhang})$
- (5)  $\text{Lucky}(\text{zhang})$
- (6)  $\neg \text{Lucky}(w) \vee \text{Win}(w, \text{prize})$
- (7)  $\neg \text{Happy}(\text{zhang})$  (结论的否定)

## 3.4.2 鲁滨逊归结原理

### 3. 谓词逻辑的归结 (12/16)



## 3.4.2 鲁滨逊归结原理

### 3. 谓词逻辑的归结 (13/16)

- 为了进一步加深对谓词逻辑归结的理解，下面再给出一个经典的归结问题
- 例3.18 “激动人心的生活” 问题
- 假设：所有不贫穷并且聪明的人都是快乐的，那些看书的人是聪明的。李明能看书且不贫穷，快乐的人过着激动人心的生活。
- 求证：李明过着激动人心的生活。
- 解：先定义谓词：
  - Poor(x)** x是贫穷的
  - Smart(x)** x是聪明的
  - Happy(x)** x是快乐的
  - Read(x)** x能看书
  - Exciting(x)** x过着激动人心的生活

## 3.4.2 鲁滨逊归结原理

### 3. 谓词逻辑的归结 (14/16)

- 再将问题用谓词表示如下：
- “所有不贫穷并且聪明的人都是快乐的”
- $(\forall x)((\neg \text{Poor}(x) \wedge \text{Smart}(x)) \rightarrow \text{Happy}(x))$
- “那些看书的人是聪明的”
- $(\forall y) (\text{Read}(y) \rightarrow \text{Smart}(y))$
- “李明能看书且不贫穷”
- $\text{Read}(\text{Liming}) \wedge \neg \text{Poor}(\text{Liming})$
- “快乐的人过着激动人心的生活”
- $(\forall z) (\text{Happy}(z) \rightarrow \text{Exciting}(z))$
- 目标“李明过着激动人心的生活”的否定
- $\neg \text{Exciting}(\text{Liming})$

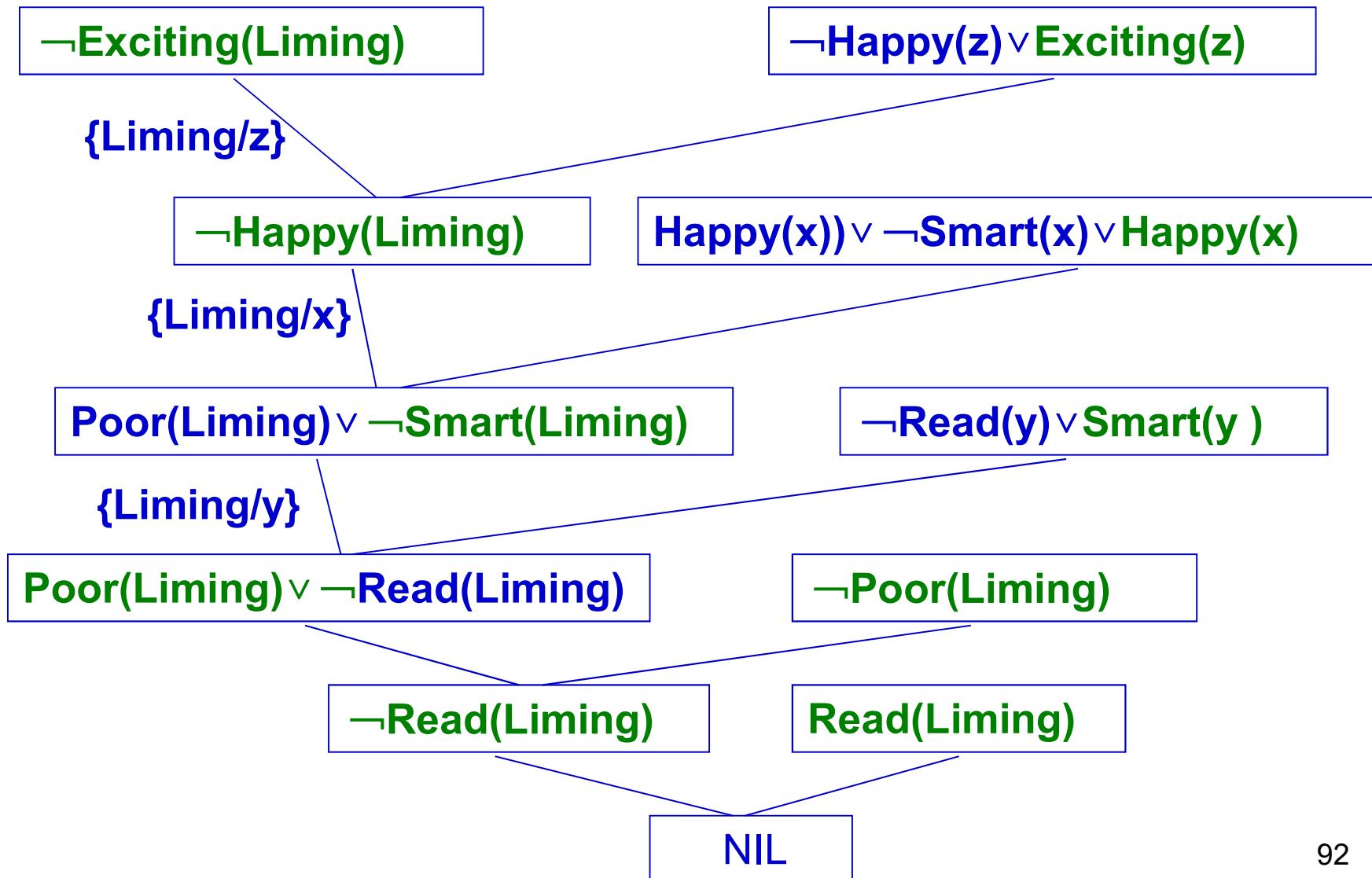
## 3.4.2 鲁滨逊归结原理

### 3. 谓词逻辑的归结 (15/16)

- 将上述谓词公式转化为子句集如下：
- (1) Poor(x)  $\vee \neg$  Smart(x)  $\vee$  Happy(x)
- (2)  $\neg$  Read(y)  $\vee$  Smart(y)
- (3) Read(Liming)
- (4)  $\neg$  Poor(Liming)
- (5)  $\neg$  Happy(z)  $\vee$  Exciting(z)
- (6)  $\neg$  Exciting(Liming) (结论的否定)

## 3.4.2 鲁滨逊归结原理

### 3. 谓词逻辑的归结 (16/16)



- 为什么要变量换名？

(1) 若  $x$  是  $y$  的父亲，则  $x$  不是女人：

$$\text{father}(x, y) \rightarrow \neg\text{woman}(x)$$

(2) 若  $x$  是  $y$  的母亲，则  $x$  是女人：

$$\text{mother}(x, y) \rightarrow \text{woman}(x) \quad \longrightarrow \quad \neg\text{mother}(w, z) \vee \text{woman}(w)$$

(3) *Chris* 是 *Mary* 的母亲：

$$\text{mother}(\text{Chris}, \text{Mary})$$

(4) *Chris* 是 *Bill* 的父亲。

$$\text{father}(\text{Chris}, \text{Bill})$$

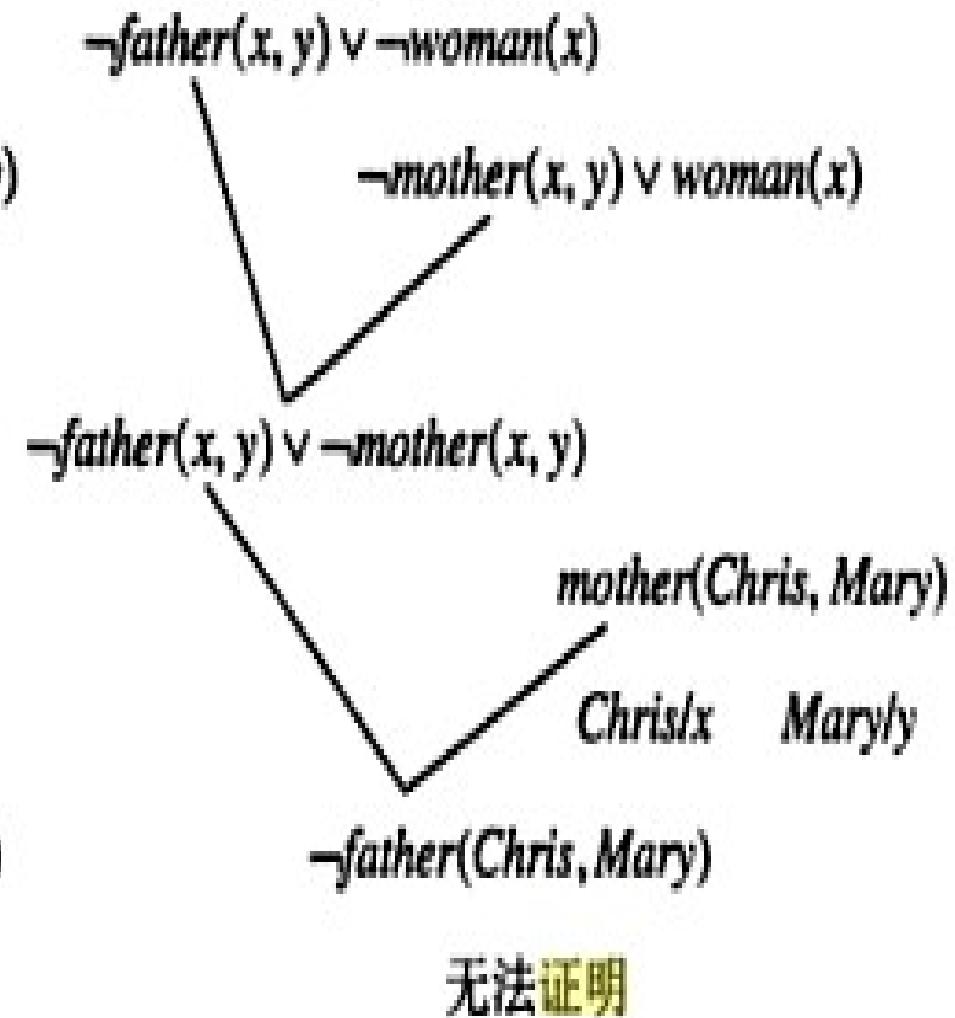
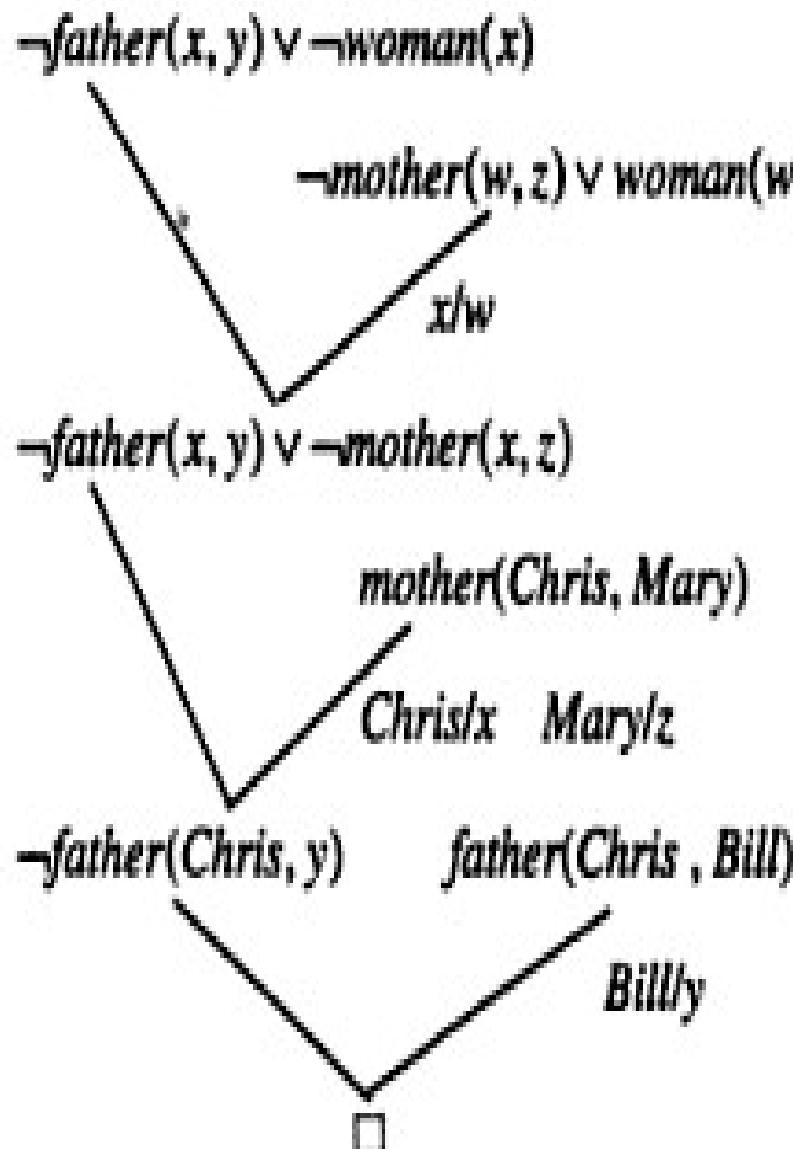
求证这些断言包含矛盾。将上述公式化成子句集为：

$$(1) \neg\text{father}(x, y) \vee \neg\text{woman}(x)$$

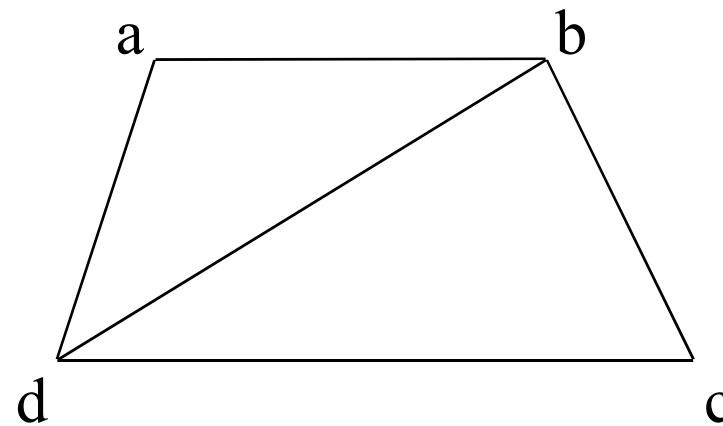
$$(2) \neg\text{mother}(x, y) \vee \text{woman}(x)$$

$$(3) \text{mother}(\text{Chris}, \text{Mary})$$

$$(4) \text{father}(\text{Chris}, \text{Bill})$$



例：证明“由梯形的对角线形成的内错角是相等的”。



首先定义谓词，并描述该问题所包含的知识。

**T(x,y,u,v)**: 表示“xyuv是左上顶点为x,右上顶点为y,右下顶点为u, 左下顶点为v的梯形”；

**P(x,y,u,v)**: 表示“线段xy平行于线段uv”；

**E(x,y,z,u,v,w)**: 表示“角xyz等于角uvw”。

于是,由几何知识,有下述公理: (由梯形定义)

$$A_1 (\forall x)(\forall y)(\forall u)(\forall v)[T(x, y, u, v) \rightarrow P(x, y, u, v)]$$

$$A_2 (\forall x)(\forall y)(\forall u)(\forall v)[P(x, y, u, v) \rightarrow E(x, y, v, u, v, y)]$$

$$A_3 T(a, b, c, d) \quad \text{(由平行线性质)}$$

由上述公理应该能够断定  $E(a, b, d, c, d, b)$

为真,即  $A_1 \wedge A_2 \wedge A_3 \rightarrow E(a, b, d, c, d, b)$

为有效的公式。

根据归结反演过程,否定该

结论并且证明  $A_1 \wedge A_2 \wedge A_3 \wedge \neg E(a, b, d, c, d, b)$

是不可满足的。

化成下述子句集：

$$S = \{\sim T(x, y, u, v) \vee P(x, y, u, v), \\ \sim P(x, y, u, v) \vee E(x, y, v, u, v, y), \\ T(a, b, c, d), \sim E(a, b, d, c, d, b)\}$$

现在用归结证明S是不可满足的：

- (1)  $\sim T(x, y, u, v) \vee P(x, y, u, v)$
- (2)  $\sim P(x, y, u, v) \vee E(x, y, v, u, v, y)$
- (3)  $T(a, b, c, d)$
- (4)  $\sim E(a, b, d, c, d, b)$
- (5)  $\sim P(a, b, c, d)$  (2) 和 (4) 的归结式
- (6)  $\sim T(a, b, c, d)$  (5) 和 (1) 的归结式
- (7)  $\square$  (3) 和 (6) 的归结式

最后一个子句是由S导出的空子句，可以断定S是不可满足的。

例：“有些患者喜欢任一医生。没有任一患者喜欢任一庸医。所以没有庸医的医生”。

定义谓词为：

$P(x)$ : “ $x$ 是患者 patient”;  $D(x)$ : “ $x$ 是医生 doctor”;

$Q(x)$ : “ $x$ 是庸医 quack”,  $L(x,y)$ : “ $x$ 喜欢 like  $y$ ”。

则前提和结论可以符号化如下：

$$A_1 : (\exists x)(P(x) \wedge (\forall y)(D(y) \rightarrow L(x, y)))$$

$$A_2 : (\forall x)(P(x) \rightarrow (\forall y)(Q(y) \rightarrow \sim L(x, y)))$$

$$G : (\forall x)(D(x) \rightarrow \sim Q(x))$$

目的是证明  $G$  是  $A_1$  和  $A_2$  的逻辑结论，即证明  
是不可满足的。首先求出子句集：

$$A_1 \wedge A_2 \wedge \sim G$$

$$\begin{aligned}
A_1 : & (\exists x)(P(x) \wedge (\forall y)(D(y) \rightarrow L(x, y))) \\
\Rightarrow & (\exists x)(\forall y)(P(x) \wedge (\sim D(y) \vee L(x, y))) \\
\Rightarrow & (\forall y)(P(a) \wedge (\sim D(y) \vee L(a, y)))
\end{aligned}$$

$$\begin{aligned}
A_2 : & (\forall x)(P(x) \rightarrow (\forall y)(Q(y) \rightarrow \sim L(x, y))) \\
\Rightarrow & (\forall x)(\sim P(x) \vee (\forall y)(\sim Q(y) \vee \sim L(x, y))) \\
\Rightarrow & (\forall x)(\forall y)(\sim P(x) \vee (\sim Q(y) \vee \sim L(x, y)))
\end{aligned}$$

$$\begin{aligned}
\sim G : & \sim (\forall x)(D(x) \rightarrow \sim Q(x)) \\
\Rightarrow & (\exists x)(D(x) \wedge Q(x)) \\
\Rightarrow & (D(b) \wedge Q(b))
\end{aligned}$$

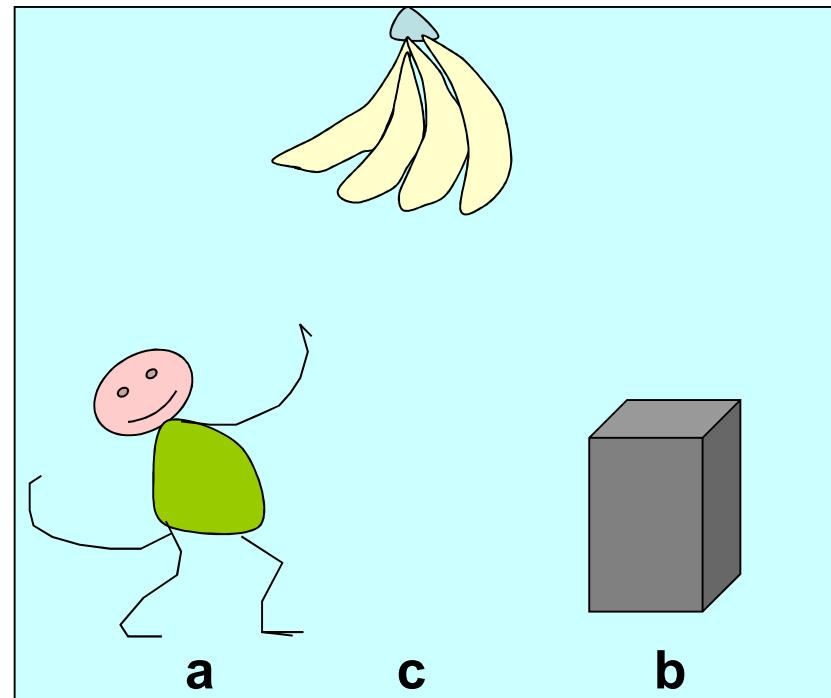
子句集合S为：

$$S = \{P(a), \sim D(y) \vee L(a, y), \sim P(x) \vee \sim Q(y) \vee \sim L(x, y), D(b), Q(b)\}$$

归结证明S是不可满足的：

- |  |  |        |
|--|--|--------|
| (1) $P(a)$                                       |  |        |
| (2) $\sim D(y) \vee L(a, y)$                     |  |        |
| (3) $\sim P(x) \vee \sim Q(y) \vee \sim L(x, y)$ | $\left. \begin{array}{c} \\ \\ \end{array} \right\} S$ |        |
| (4) $D(b)$                                       |  |        |
| (5) $Q(b)$                                       |  |        |
| (6) $L(a, b)$                                    | (2)  | (4) 归结 |
| (7) $\sim Q(y) \vee \sim L(a, y)$                | (1)  | (3) 归结 |
| (8) $\sim L(a, b)$                               | (5)  | (7) 归结 |
| (9) $\square$                                    | (6)  | (8) 归结 |

# 例：猴子摘香蕉问题



# 问题的表示

已知：

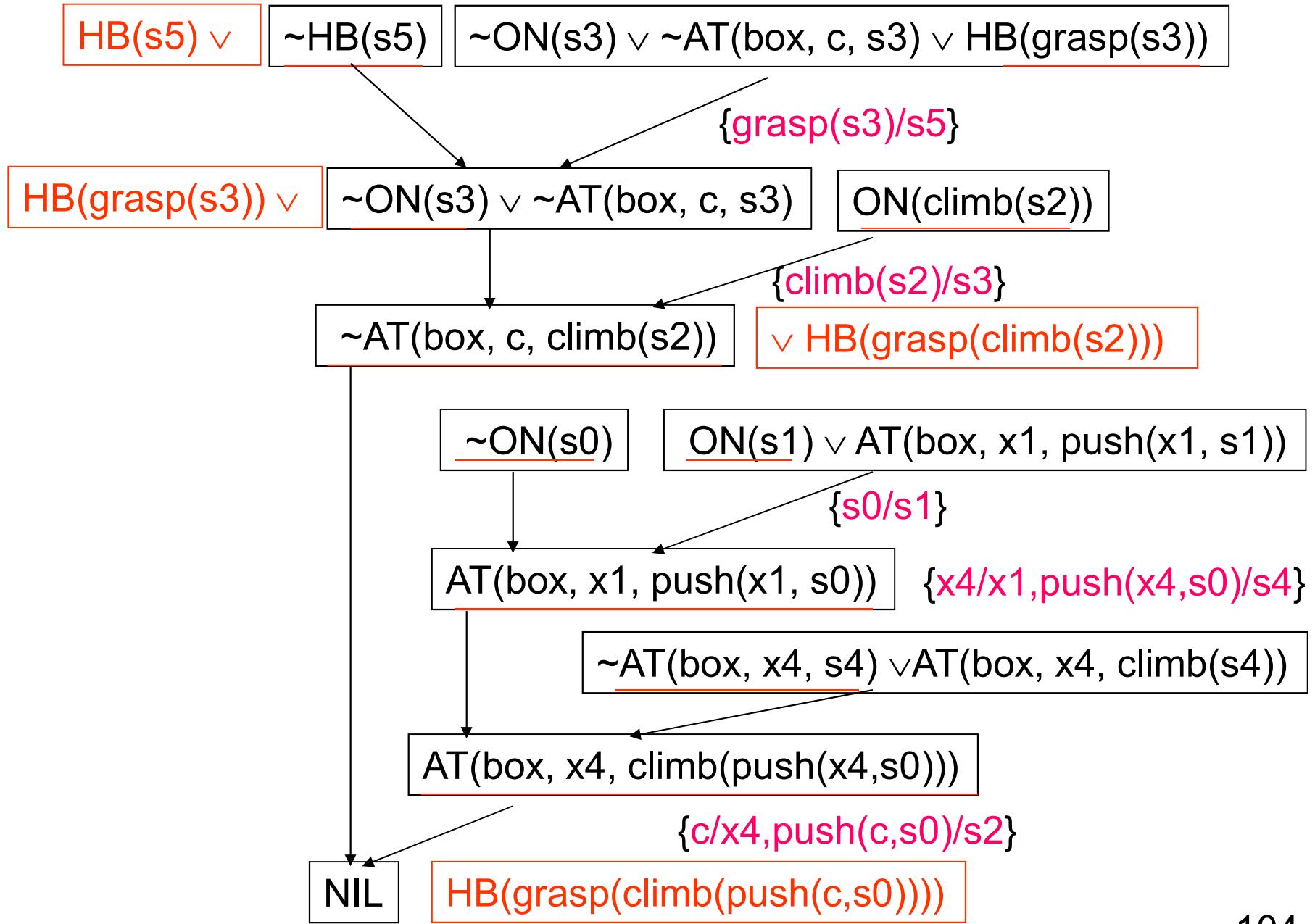
1.  $\sim \text{ON}(s_0)$  (猴子不在箱子上)
2.  $(\forall x)(\forall s)(\sim \text{ON}(s) \rightarrow \text{AT}(\text{box}, x, \text{push}(x, s)))$   
(若猴子不在箱子上，则它可以走到箱子x处，将箱子推到s处。)
3.  $(\forall s)(\text{ON}(\text{climb}(s)))$  (猴子在s处爬到箱子上)
4.  $(\forall s)((\text{ON}(s) \wedge \text{AT}(\text{box}, c, s)) \rightarrow \text{HB}(\text{grasp}(s)))$   
(若猴子在s处箱子上且箱子在c处，则猴子可以拿到香蕉)
5.  $(\forall x)(\forall s)(\text{AT}(\text{box}, x, s) \rightarrow \text{AT}(\text{box}, x, \text{climb}(s)))$   
(若猴子走到箱子x处，则它可以爬上箱子)

求解：  $(\exists s)\text{HB}(s)$

(在什么情况下猴子可以拿到香蕉？)

# 问题的子句集

- 1,  $\neg \text{ON}(s0)$
- 2,  $\text{ON}(s1) \vee \text{AT}(\text{box}, x1, \text{push}(x1, s1))$
- 3,  $\text{ON}(\text{climb}(s2))$
- 4,  $\neg \text{ON}(s3) \vee \neg \text{AT}(\text{box}, c, s3) \vee \text{HB}(\text{grasp}(s3))$
- 5,  $\neg \text{AT}(\text{box}, x4, s4) \vee \text{AT}(\text{box}, x4, \text{climb}(s4))$
- 6,  $\neg \text{HB}(s5)$



# 用消解法求更为复杂问题例子

- 例1：某记者到一孤岛采访，遇到了一个难题，即岛上许多人说假话，因而难以保证新闻报道的准确性，不过有一点他是清楚的，这个岛上的三个人，从来不说假话。一次记者遇到了孤岛上的三个人中的每一个都提了一个同样的问题，即“谁是说谎者？”结果A答“B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”，  
试问记者如何从回答中理出头绪。

# 用消解法求更为复杂问题例子

- 练习：用消解法求解上述问题
- 提示：
- 以**A**, **B**, **C**三个命题来表示**A**, **B**, **C**三个是老实人（不说谎）
- 如果**A**说真话，则**B**和**C**一定说谎，因此有：  
$$\mathbf{A} \rightarrow \neg \mathbf{B} \wedge \neg \mathbf{C}$$
- 如果**A**说假话，则**B**和**C**中至少有一人说真话，因此有：  
$$\neg \mathbf{A} \rightarrow \mathbf{B} \vee \mathbf{C}$$

# 用消解法求更为复杂问题例子

A答 “B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”

以A, B, C三个命题来表示A, B, C三个是老实人（不说谎）

►如果**A**说真话，则**B**和**C**一定说谎，因此有： $A \rightarrow \sim B \wedge \sim C$

如果A说假话，则B和C中至少有一人说真话，因此有： $\sim A \rightarrow B \vee C$

以同样的推理方式可得到：

►如果**B**说真话，

$$B \rightarrow \sim A \wedge \sim C$$

如果**B**说假话

$$\sim B \rightarrow A \vee C$$

►如果**C**说真话，

$$C \rightarrow \sim A \vee \sim B$$

如果**C**说假话

$$\sim C \rightarrow A \wedge B$$

# 用消解法求更为复杂问题例子

对以上蕴含式加以推理,并化成子句形式,可得:

$$\sim A \vee \sim B \quad (1)$$

$$\sim A \vee \sim C \quad (2)$$

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

$$\sim B \vee \sim C \quad (4)$$

$$\sim A \vee \sim B \vee \sim C \quad (5)$$

$$A \vee C \quad (6)$$

$$B \vee C \quad (7)$$

# 用消解法求更为复杂问题例子

- (1) 和 (7) 消解，得： $\sim A \vee C$  (8)
- (2) 和 (8) 消解，得： $\sim A$  (9)
- (6) 和 (9) 消解，得： $C$  (10)
- (4) 和 (10) 消解，得： $\sim B$  (11)

说明？谁是说谎者？

A和B都是说谎者，而C是老实人

注意在消解过程中，子句 (3) 和 (5) 没有用到这是自然的，实际上，(3) 是 (6) 或 (7) 的推论，(5) 是 (1)，(2) 或 (4) 的推论，这种冗余信息的存在降低了消解法的效率。因为在机械地执行消解过程时，各种可能性都是要试验的。

# 用消解法求更为复杂问题例子

►例2：四对夫妇中，王结婚时，周送了礼；周和钱是同一排球队的队员；李的爱人是陈的爱人的表哥；陈夫妇与邻居吵架，徐、周、吴的爱人都去助战；李、徐、周结婚前住一间宿舍，试用消解法求王、周、钱、陈、李、徐、吴、孙八人谁是男？谁是女？谁和谁是夫妇？

# 用消解法求更为复杂问题例子

- (1) 女 (李)
- (2) 女 (李) → 女 (徐)  
即: **NOT** 女 (李) ∨ 女 (徐)
- (3) 女 (李) → 女 (周)  
即: **NOT** 女 (李) ∨ 女 (周)
- (4) 女 (周) → 女 (钱)  
即: **NOT** 女 (周) ∨ 女 (钱)
- (5) 女 (李) ∧ 女 (徐) ∧ 女 (周) ∧ 女 (钱)  
→ 男 (王) ∧ 男 (陈) ∧ 男 (吴) ∧ 男 (孙)

►则可解得男的是：  
**王、陈、吴、孙**

►则可解得女的是：  
**李、徐、周、钱**

例2：四对夫妇中，**王结婚时，周送了礼**；周和钱是同一排球队的队员；**李的爱人是陈的爱人的表哥**；陈夫妇与邻居吵架，**徐、周、吴的爱人都去助战**；李、徐、周结婚前住一间宿舍，试用消解法求王、周、钱、陈、李、徐、吴、孙八人谁是男？谁是女？谁和谁是夫妇？

( 6) **~couple** (王, 周)

( 7) **~couple** (陈, 李)

( 8) **~ couple** (陈, 徐)

( 9) **~ couple** (陈, 周)

(10) **~ couple** (吴, 周)

(11) **~ couple** (吴, 徐)

(12) **couple** (X, Y) → 男 (X) ∧ 女(Y)

(13) **couple** (陈, 李) ∨ **couple** (陈, 徐) ∨ **couple** (陈, 周) ∨ **couple** (陈, 钱)

►男：王、陈、吴、孙

►女：李、徐、周、钱

(14) **couple** (陈, 钱) (13)(7)(8)(9)消解;  
(15) **couple**(吴,周) ∨ **couple**(吴,徐) ∨ **couple**(吴,李)  
(16) **couple** (吴,李) (15) (10) (11) 进行消解;  
(17) **couple** (王, 周) ∨ **couple** (王, 徐)  
(18) **couple** (王, 徐) (17) 与 (6) 消解  
(19) **couple** (孙, 周)

### 3.4.3 归结演绎推理的归结策略

- 归结演绎推理实际上就是从子句集中不断寻找可进行归结的子句对，并通过对这些子句对的归结，最终得出一个空子句的过程。
- 由于事先并不知道哪些子句对可进行归结，更不知道通过对哪些子句对的归结能尽快得到空子句，因此就需要对子句集中的所有子句逐对进行比较，直到得出空子句为止。这种盲目的全面进行归结的方法，不仅会产生许多无用的归结式，更严重的是会产生组合爆炸问题。
  - 组合爆炸的典型例子：**TSP**问题
    - 10城市的**TSP**问题有大约180,000个可能解
    - 20城市的**TSP**问题有大约 $60,000,000,000,000,000$ 个可能解。
    - 50城市的**TSP**问题有大约 $10^{62}$ 个可能解
- 需要：有效的归结策略

### 3.4.3 归结演绎推理的归结策略

- 目前，常用的归结策略可分为两大类，一类是删除策略，另一类是限制策略。
- 删除策略是通过删除某些无用的子句来缩小归结范围
- 限制策略是对参加归结的子句进行某些限制，来减少归结的盲目性，以尽快得到空子句。

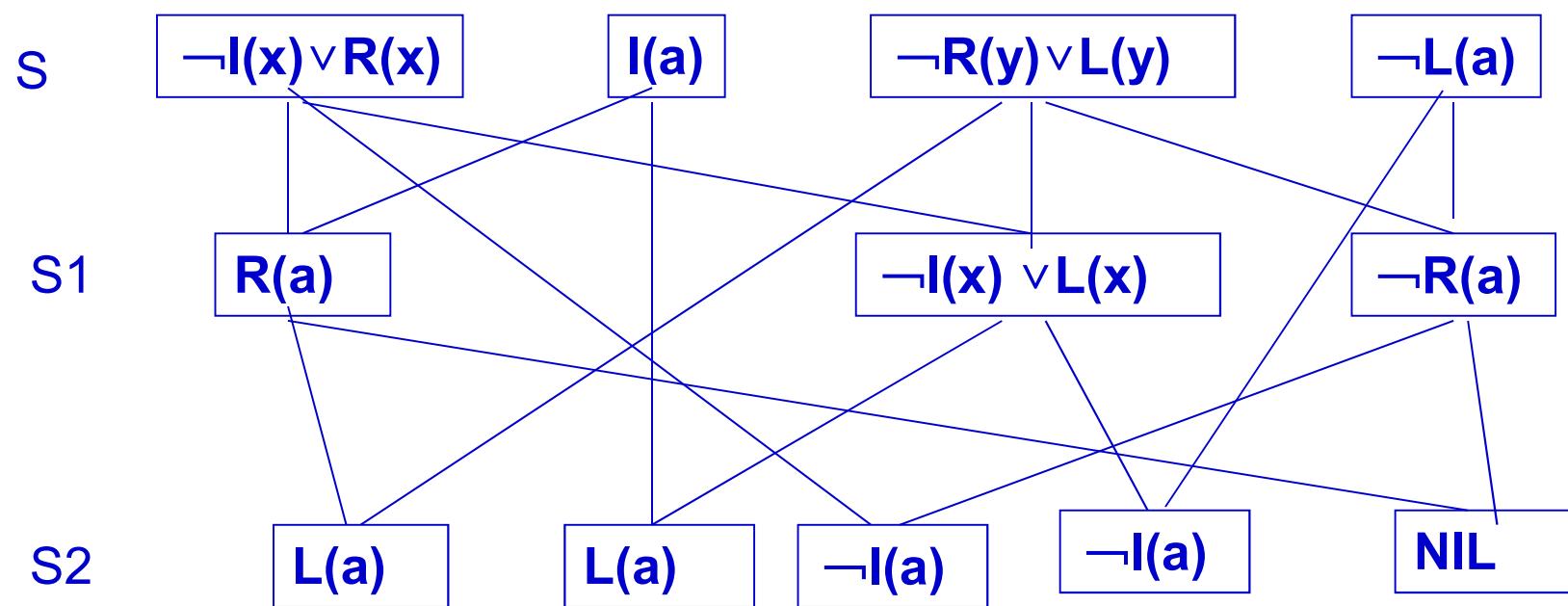
### 3.4.3 归结演绎推理的归结策略

#### 1. 广度优先策略 (1/3)

- 广度优先是一种穷尽子句比较的复杂搜索方法。设初始子句集为  $S_0$ , 广度优先策略的归结过程可描述如下:
  - (1) 从  $S_0$  出发, 对  $S_0$  中的全部子句作所有可能的归结, 得到第一层归结式, 把这些归结式的集合记为  $S_1$ ;
  - (2) 用  $S_0$  中的子句与  $S_1$  中的子句进行所有可能的归结, 得到第二层归结式, 把这些归结式的集合记为  $S_2$ ;
  - (3) 用  $S_0$  和  $S_1$  中的子句与  $S_2$  中的子句进行所有可能的归结, 得到第三层归结式, 把这些归结式的集合记为  $S_3$ ;
  - 如此继续, 直到得出空子句或不能再继续归结为止。
  - 例3.19 设有如下子句集:
    - $S = \{\neg I(x) \vee R(x), I(a), \neg R(y) \vee L(y), \neg L(a)\}$
    - 用广度优先策略证明  $S$  为不可满足。
    - 广度优先策略的归结树如下:

### 3.4.3 归结演绎推理的归结策略

#### 1. 广度优先策略 (2/3)



### 3.4.3 归结演绎推理的归结策略

#### 1. 广度优先策略 (3/3)

- 从这个例子可以看出，宽度优先策略归结出了许多无用的子句，既浪费时间，又浪费空间。但是，这种策略由一个有趣的特性，就是当问题有解时保证能找到最短归结路径。
- 因此，它是一种完备的归结策略。宽度优先对大问题的归结容易产生组合爆炸，但对小问题却仍是一种比较好的归结策略。

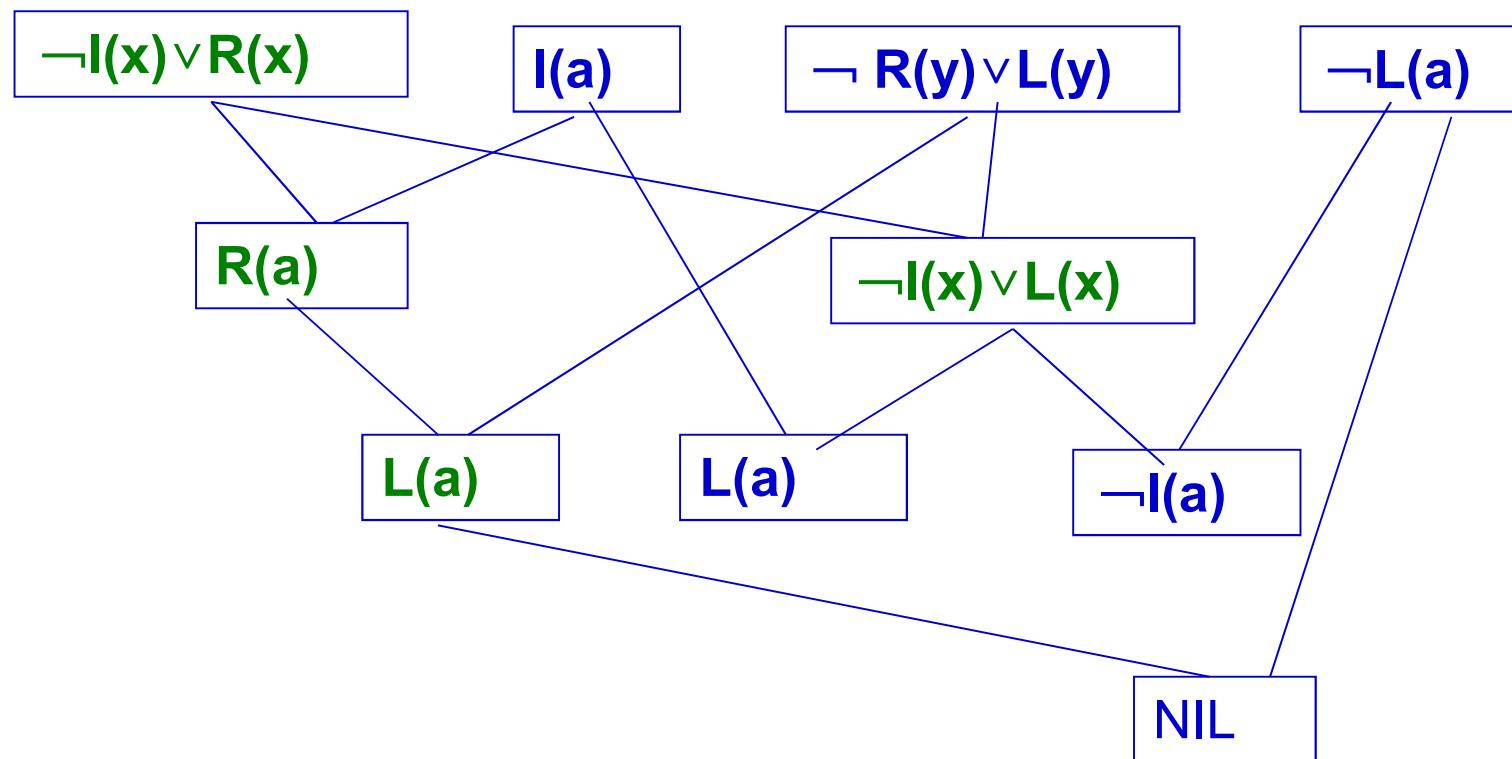
### 3.4.3 归结演绎推理的归结策略

#### 2. 支持集策略 (1/3)

- 支持集策略是沃斯(Wos)等人在1965年提出的一种归结策略。它要求每一次参加归结的两个亲本子句中，至少应该有一个是由目标公式的否定所得到的子句或它们的后裔。可以证明支持集策略是完备的，即当子句集为不可满足时，则由支持集策略一定能够归结出一个空子句。也可以把支持集策略看成是在宽度优先策略中引入了某种限制条件，这种限制条件代表一种启发信息，因而有较高的效率
- 例3.20 设有如下子句集：
- $S=\{\neg I(x) \vee R(x), I(a), \neg R(y) \vee L(y), \neg L(a)\}$
- 其中， $\neg I(x) \vee R(x)$ 为目标公式的否定。用支持集策略证明S为不可满足。

### 3.4.3 归结演绎推理的归结策略

#### 2. 支持集策略 (2/3)



### 3.4.3 归结演绎推理的归结策略

#### 2. 支持集策略 (3/3)

从上述归结过程可以看出，各级归结式数目要比宽度优先策略生成的少，但在第二级还没有空子句。就是说这种策略限制了子句集元素的剧增，但会增加空子句所在的深度。此外，支持集策略具有逆向推理的含义，由于进行归结的亲本子句中至少有一个与目标子句有关，因此推理过程可以看作是沿目标、子目标的方向前进的。

## 3.4.5 归结演绎推理的归结策略

### 3. 删除策略 (1/3)

- 主要想法是：如果在归结时能把子句集中无用的子句删除掉，这就会缩小搜索范围，减少比较次数，从而提高归结效率。
- 常用的删除方法有以下几种（纯文字、重言式、包孕）
- 纯文字删除法
- 如果某文字L在子句集中不存在可与其互补的文字 $\neg L$ ，则称该文字为纯文字。
- 在归结过程中，纯文字不可能被消除，用包含纯文字的子句进行归结也不可能得到空子句
- 对子句集而言，删除包含纯文字的子句，是不影响其不可满足性的。例如，对子句集
- $S=\{P \vee Q \vee R, \neg Q \vee R, Q, \neg R\}$
- 其中P是纯文字，因此可以将子句 $P \vee Q \vee R$ 从子句集S中删除。

### 3.4.3 归结演绎推理的归结策略

#### 3. 删除策略 (2/3)

- 重言式删除法
- 如果一个子句中包含有互补的文字对，则称该子句为重言式。例如
  - $P(x) \vee \neg P(x)$ ,  $P(x) \vee Q(x) \vee \neg P(x)$
  - 都是重言式，不管  $P(x)$  的真值为真还是为假， $P(x) \vee \neg P(x)$  和  $P(x) \vee Q(x) \vee \neg P(x)$  都均为真。
  - 重言式是真值为真的子句。对一个子句集来说，不管是增加还是删除一个真值为真的子句，都不会影响该子句集的不可满足性。因此，可从子句集中删去重言式。

### 3.4.3 归结演绎推理的归结策略

#### 3. 删除策略 (3/3)

- 包孕删除法
- 设有子句 $C_1$ 和 $C_2$ ，如果存在一个置换 $\sigma$ ，使得 $C_1\sigma \subseteq C_2$ ，则称 $C_1$ 包孕于 $C_2$ 。例如
  - $P(x)$  包孕于  $P(y) \vee Q(z)$   $\sigma=\{x/y\}$
  - $P(x)$  包孕于  $P(a)$   $\sigma=\{a/x\}$
  - $P(x)$  包孕于  $P(a) \vee Q(z)$   $\sigma=\{a/x\}$
  - $P(x) \vee Q(a)$  包孕于  $P(f(a)) \vee Q(a) \vee R(y)$   $\sigma=\{f(a)/x\}$
  - $P(x) \vee Q(y)$  包孕于  $P(a) \vee Q(u) \vee R(w)$   $\sigma=\{a/x, u/y\}$
- 对于子句集来说，把其中包孕的子句删去后，不会影响该子句集的不可满足性。因此，可以从子句集中删除包孕的子句。

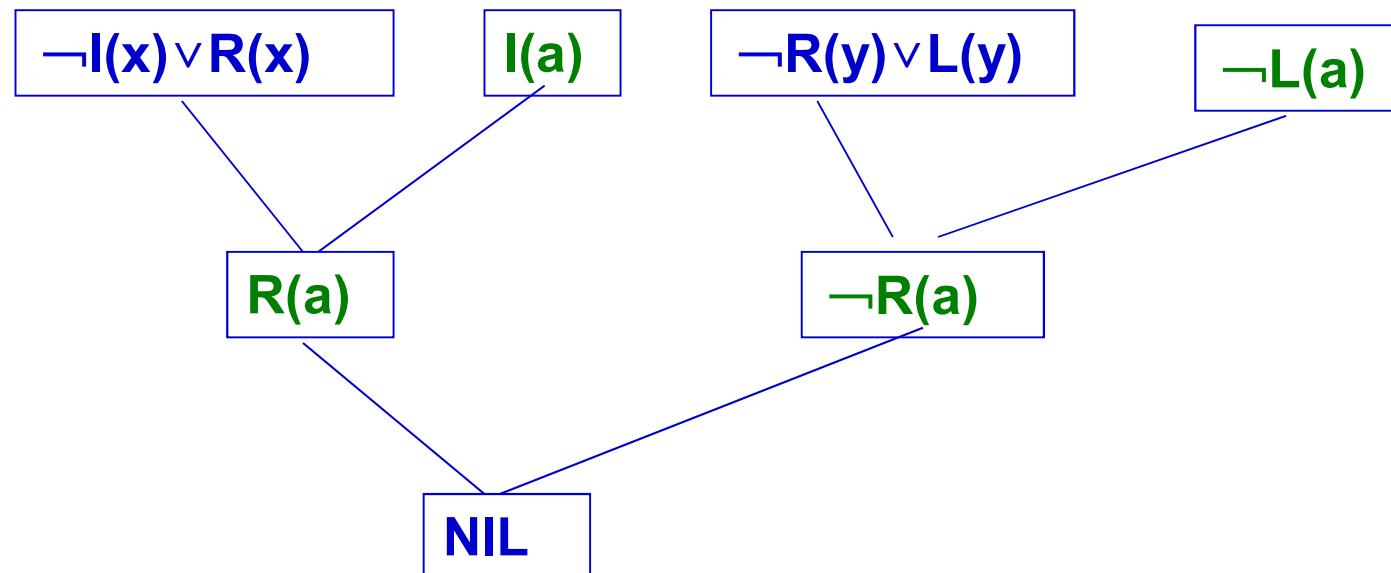
### 3.4.3 归结演绎推理的归结策略

#### 4. 单文字子句策略 (1/2)

- 如果一个子句只包含一个文字，则称此子句为单文字子句。单文字子句策略是对支持集策略的进一步改进，它要求每次参加归结的两个亲本子句中至少有一个子句是单文字子句。
- 例3.21 设有如下子句集：
- $S = \{\neg I(x) \vee R(x), I(a), \neg R(y) \vee L(y), \neg L(a)\}$
- 用单文字子句策略证明S为不可满足。
- 采用单文字子句策略，归结式包含的文字数将少于其亲本子句中的文字数，这将有利于向空子句的方向发展，因此会有较高的归结效率。
- 是不完备的，即当子句集为不可满足时，用这种策略不一定能归结出空子句。

### 3.4.3 归结演绎推理的归结策略

#### 4. 单文字子句策略 (2/2)



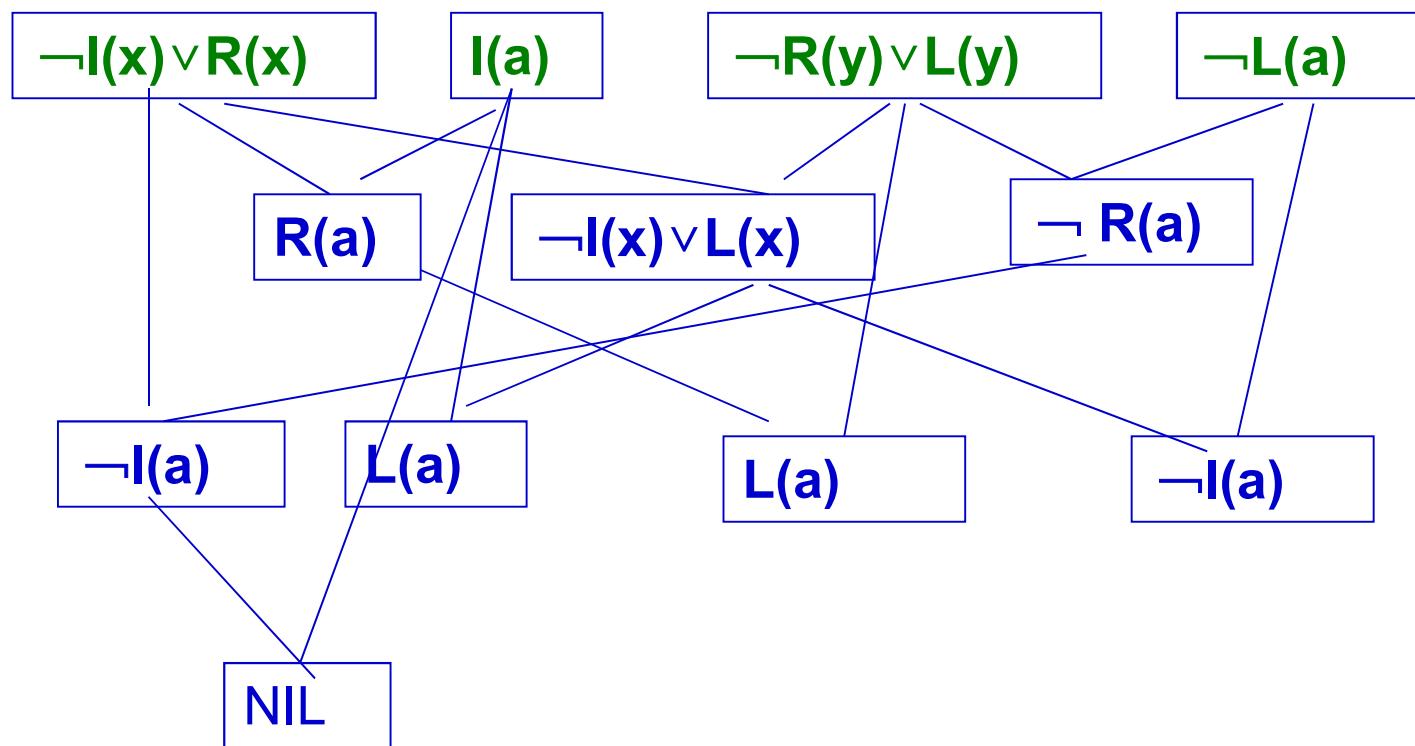
### 3.4.3 归结演绎推理的归结策略

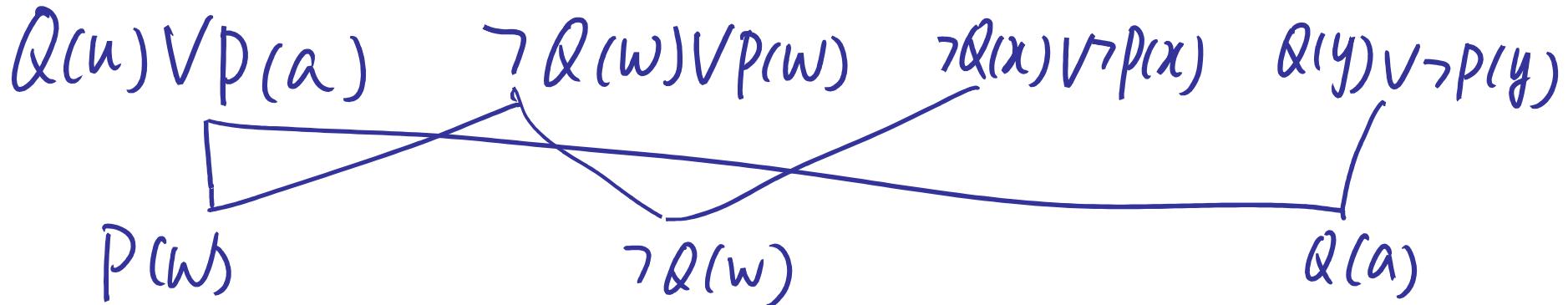
#### 5. 线性输入策略 (1/2)

- 这种策略要求每次参加归结的两个亲本子句中，至少应该有一个是初始子句集中的子句。所谓初始子句集是指开始归结时所使用的子句集。
- 例3.22 设有如下子句集：
- $S = \{\neg I(x) \vee R(x), I(a), \neg R(y) \vee L(y), \neg L(a)\}$
- 用线性输入策略证明S为不可满足。
- 线性输入策略可限制生成归结式的数目，具有简单和高效的优点。但是，这种策略也是一种不完备的策略。

### 3.4.3 归结演绎推理的归结策略

#### 5. 线性输入策略 (2/2)





例如，子句集

$S = \{Q(u) \vee P(a), \neg Q(w) \vee P(w), \neg Q(x) \vee \neg P(x), Q(y) \vee \neg P(y)\}$

从  $S$  出发很容易找到一棵归结反演树，但却不存在线性输入策略的归结反演树。

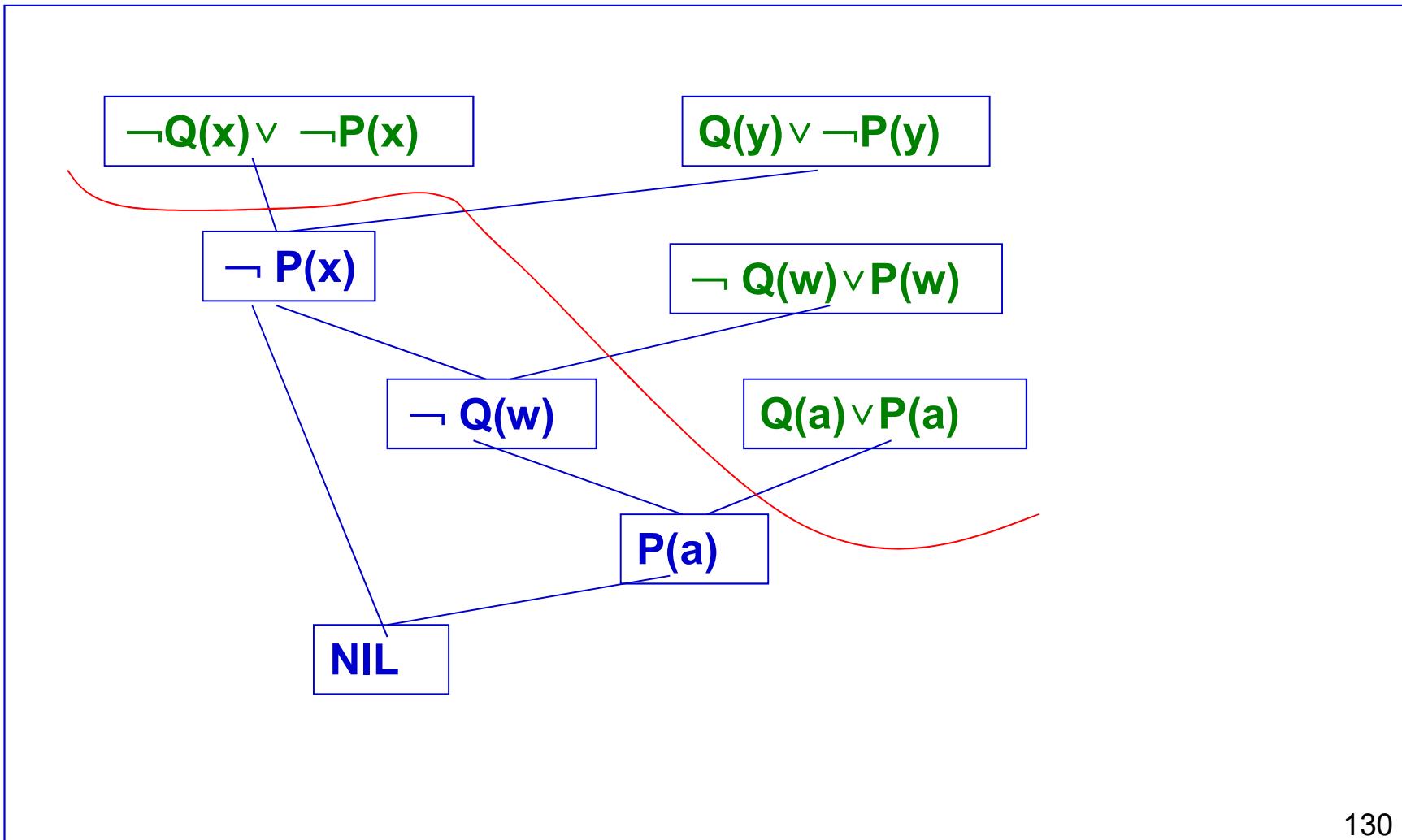
### 3.4.3 归结演绎推理的归结策略

#### 6. 祖先过滤策略 (1/2)

- 这种策略与线性输入策略有点相似，但是，放宽了对子句的限制。每次参加归结的两个亲本子句，只要满足以下两个条件中的任意一个就可进行归结：
  - (1) 两个亲本子句中至少有一个是初始子句集中的子句。
  - (2) 如果两个亲本子句都不是初始子句集中的子句，则一个子句应该是另一个子句的先辈子句。所谓一个子句(例如 $C_1$ )是另一个子句(例如 $C_2$ )的先辈子句是指 $C_2$ 是由 $C_1$ 与别的子句归结后得到的归结式。
- 例3.23 设有如下子句集：
- $S = \{\neg Q(x) \vee \neg P(x), Q(y) \vee \neg P(y), \neg Q(w) \vee P(w), Q(a) \vee P(a)\}$
- 用祖先过滤策略证明 $S$ 为不可满足
- 可以证明祖先过滤策略也是完备的。

### 3.4.3 归结演绎推理的归结策略

#### 6. 祖先过滤策略 (2/2)



### 3.4.4 用归结反演求取问题的答案

- 使目标为真的变量由每个归结步骤中所产生的变量置换给出，因此，保留各归结步骤中的变量置换信息可以给出正确答案
- 例“兴奋生活”问题
- 所有勤奋且聪明的人都是愉快的。读书人都是聪明的。李明读书且勤奋。勤奋的人过着兴奋的生活。
- 可以找出过着兴奋生活的人吗？

解：先定义谓词：

Diligent(x)	x是勤奋的
Smart(x)	x是聪明的
Happy(x)	x是愉快的
Read(x)	x能看书
Exciting(x)	x过着兴奋生活

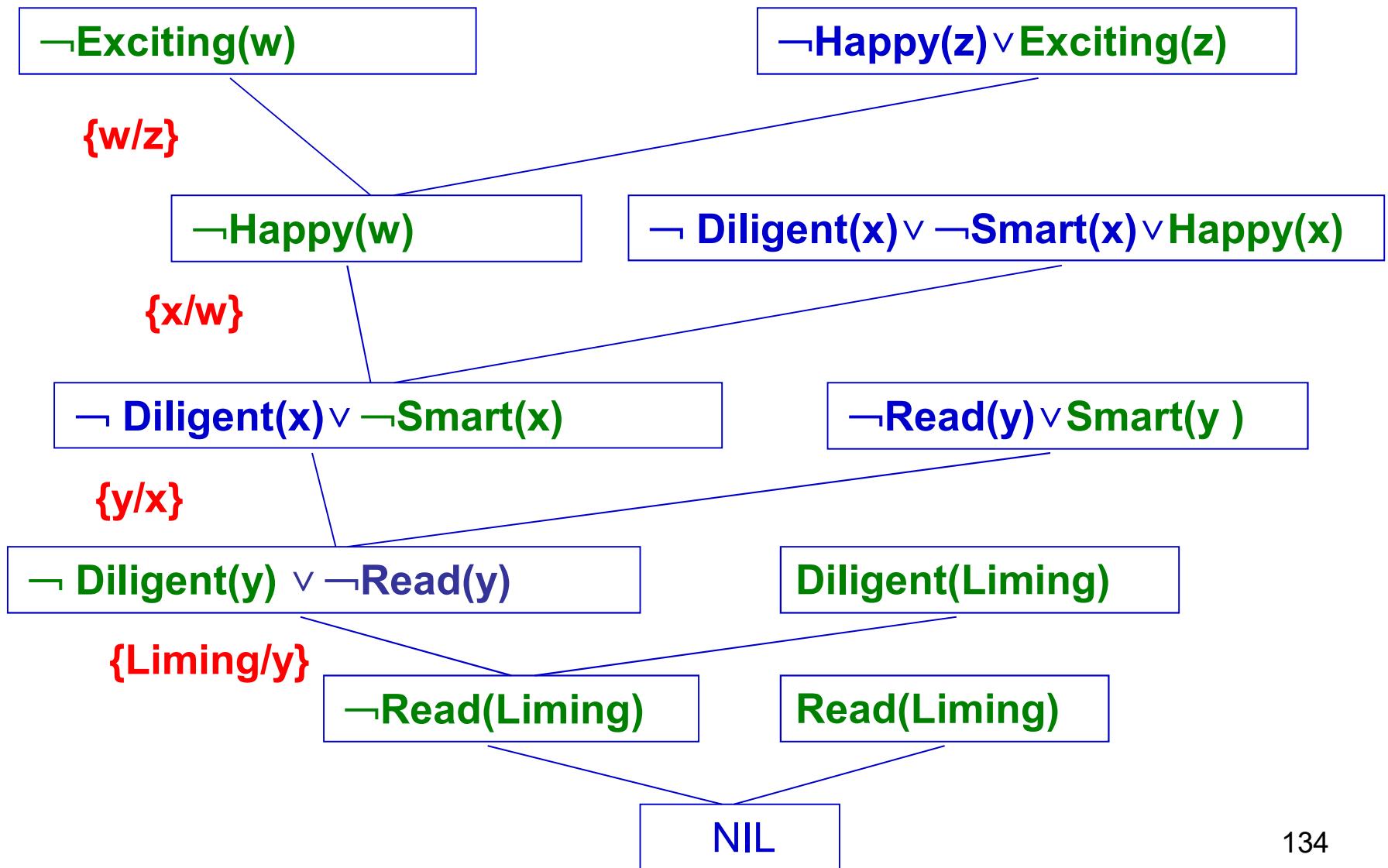
### 3.4.4 用归结反演求取问题的答案

- 再将问题用谓词表示如下：
- $(\forall x)((\text{Diligent}(x) \wedge \text{Smart}(x)) \rightarrow \text{Happy}(x))$
- $(\forall y) (\text{Read}(y) \rightarrow \text{Smart}(y))$
- $\text{Read}(\text{Liming}) \wedge \text{Diligent}(\text{Liming})$
- $(\forall z) (\text{Happy}(z) \rightarrow \text{Exciting}(z))$
- 目标“存在过着兴奋生活的人”的否定
  - $\neg (\exists w) \text{ Exciting}(w)$

### 3.4.4 用归结反演求取问题的答案

- 将上述谓词公式转化为子句集如下：
- (1)  $\neg \text{Diligent}(x) \vee \neg \text{Smart}(x) \vee \text{Happy}(x)$
- (2)  $\neg \text{Read}(y) \vee \text{Smart}(y)$
- (3)  $\text{Read}(\text{Liming})$
- (4)  $\neg \text{Diligent}(\text{Liming})$
- (5)  $\neg \text{Happy}(z) \vee \text{Exciting}(z)$
- (6)  $\neg \text{Exciting}(w)$  (结论的否定)

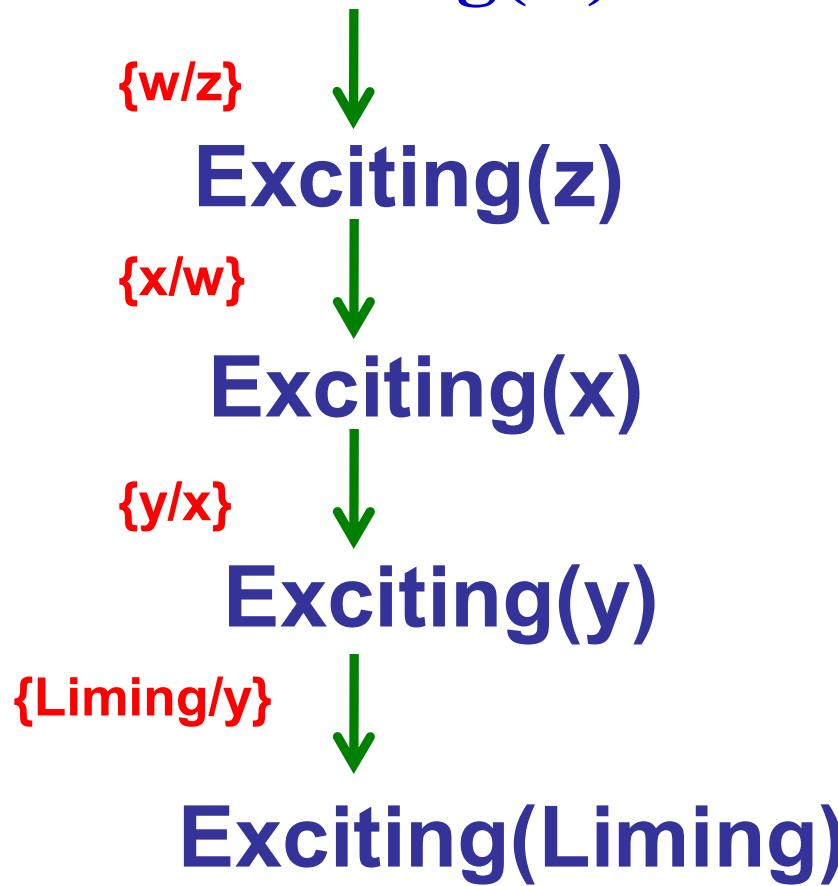
### 3.4.4 用归结反演求取问题的答案



### 3.4.4 用归结反演求取问题的答案

- 目标“存在过着兴奋生活的人”？

(  $\exists w$  ) Exciting( $w$ )



### 3.4.4 用归结反演求取问题的答案

- 一般步骤为：
- (1) 把问题的已知条件用谓词公式表示出来，化为子句集；
- (2) 把问题的目标的否定用谓词公式表示出来，并化为子句集；
- (3) 对目标否定子句集中的每个子句，构造该子句的重言式（即把该目标否定子句和此目标否定子句的否定之间再进行析取所得到的子句），用这些重言式代替相应的目标否定子句式，并把这些重言式加入到前提子句集中，得到一个新的子句集；
- (4) 对这个新的子句集，应用归结原理求出其证明树，这时证明树的根子句不为空，称这个证明树为修改的证明树；
- (5) 用修改证明树的根子句作为答案。

### 3.4.4 用归结反演求取问题的答案

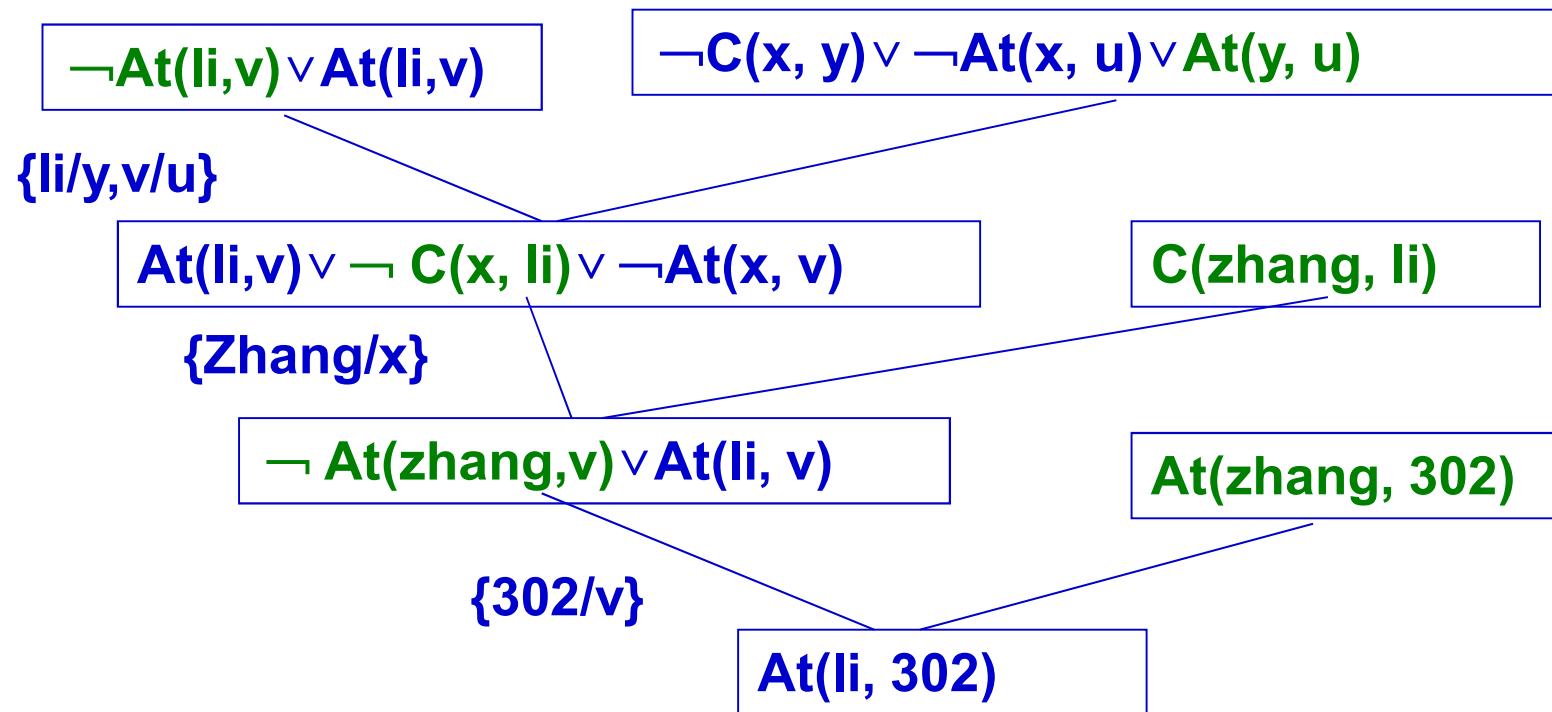
- 例3.24 已知：“张和李是同班同学，如果x和y是同班同学，则x的教室也是y的教室，现在张在302教室上课。”
- 问：“现在李在哪个教室上课？”
- 解：首先定义谓词：
  - $C(x, y)$  x和y是同班同学；
  - $At(x, u)$  x在u教室上课。
- 把已知前提用谓词公式表示如下：
  - $C(zhang, li)$
  - $(\forall x)(\forall y)(\forall u)(C(x, y) \wedge At(x, u) \rightarrow At(y, u))$
  - $At(zhang, 302)$
- 把目标的否定用谓词公式表示如下：
  - $\neg(\exists v)At(li, v)$

### 3.4.4 用归结反演求取问题的答案

- 把上述公式化为子句集:
- $C(zhang, li)$
- $\neg C(x, y) \vee \neg At(x, u) \vee At(y, u)$
- $At(zhang, 302)$
- 把目标的否定化成子句式，并用重言式
- $\neg At(li, v) \vee At(li, v)$
- 代替之。

## 3.4.4 用归结反演求取问题的答案

根子句就是所求的答案



某人被盗，公安局派出所派出**5**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**5**个侦察员的话都是可信的，试问谁是盗窃犯呢？

解：将**5**位侦察员的话表示成谓词公式，为此先定义谓词。

**第一步：**设谓词**P(x)**表示x是作案者，所以根据题意：

- A:  $P(zhao) \vee P(qian)$     B:  $P(qian) \vee P(sun)$
- C:  $P(sun) \vee P(li)$         D:  $\neg P(zhao) \vee \neg P(sun)$
- E:  $\neg P(qian) \vee \neg P(li)$

以上每个侦查员的话都是一个子句。

**第二步：**将待求解的问题表示成谓词。设y是盗窃犯，则问题的谓词公式为**P(y)**,将其否定并与**ANS(y)**做析取得  
 $\neg P(y) \vee \text{ANS}(y)$

**第三步：**求前提条件及 $\neg P(y) \vee \text{ANS}(y)$ 的子句集，并将各子句列表如下：

(1)  $P(zhao) \vee P(qian)$

(2)  $P(qian) \vee P(sun)$

(3)  $P(sun) \vee P(li)$

(4)  $\sim P(zhao) \vee \sim P(sun)$

(5)  $\sim P(qian) \vee \sim P(li)$

(6)  $\sim P(y) \vee ANS(y)$

第四步：应用归结原理进行推理。

(7)  $P(qian) \vee \sim P(sun)$  (1) 与 (4) 归结

(8)  $P(zhao) \vee \sim P(li)$  (1) 与 (5) 归结

- (9)  $P(qian) \vee \neg P(zhao)$  (2) 与 (4) 归结  
(10)  $P(sun) \vee \neg P(li)$  (2) 与 (5) 归结  
(11)  $\neg P(zhao) \vee P(li)$  (3) 与 (4) 归结  
(12)  $P(sun) \vee \neg P(qian)$  (3) 与 (5) 归结  
(13)  $P(qian)$  (2) 与 (7) 归结  
(14)  $P(sun)$  (2) 与 (12) 归结  
(15) **ANS(qian)** (6) 与 (13) 归结,  
(16) **ANS(sun)** (6) 与 (14) 归结,

所以，本题的盗窃犯是两个人：钱和孙。

## ➤ 爱因斯坦的一道题：

一个土耳其商人，想找一个十分聪明的助手协助他经商，有两个人前来应聘。这个商人问：“现在我们房间里有五顶黑色的帽子，然后灯一开，什么颜色的人都会看到。请你们试一下，我将三个人的帽子弄乱，然后关掉灯，你们各自戴一顶帽子。你们每个人摸一下自己的帽子，然后把灯打开，同时说出自己戴的是什么颜色的帽子。请你们尽快地说出。”说完之后，商人将三顶帽子戴在头上，同时打开灯。这时，商人摸了摸帽子，发现一顶是红色的，一顶是黑色的，另一顶藏到了商人头上。接着把灯打开，商人便喊到：“我戴的是黑色的帽子。”

请问这个人猜得对吗？是怎么推导出来的？

**分析1：**如果我戴的也是红帽子，那么，B就马上可以猜到自己是戴黑帽子(因为红帽子只有两顶)；而现在B并没有立刻猜到，可见，我戴的不是红帽子。

可见，B的反应太慢了。

**分析2：**设 $P_1$ 表示“猜对的人戴红帽子”； $P_2$ 表示“猜对的人戴黑帽子”； $Q_1$ 表示“另一个人戴红帽子”； $Q_2$ 表示“另一个人戴黑帽子”； $R_1$ 表示“商人戴红帽子”。

现在知道，商人头上戴的是红帽子，即 $R_1$ 为真，又知道另一个人没有作出断定，即既不能断定 $Q_1$ 为真，也不能断定 $Q_2$ 为真。

根据题设条件，可得如下公式：

$R1 \wedge P1 \rightarrow Q2$ : 如果商人和猜对的人戴的都是红帽子，那么另一个戴的就是黑帽子，因为红帽子只有两顶。

$R1 \wedge Q1 \rightarrow P2$ : 如果商人和另一个戴的都是红帽子，那么猜对的人戴的就是黑帽子。

$\neg P1 \rightarrow P2$ : 如果猜对的人戴的不是红帽子，那么他戴的就是黑帽子。

$\neg Q1 \rightarrow Q2$ : 如果另一个人戴的不是红帽子，那么他戴的就是黑帽子。

**推演步骤如下：**

- (1)  $P_1$  (假设) ; (2)  $R_1$  (根据题设)
- (3)  $R_1 \wedge P_1$  (合取构成) ; (4)  $R_1 \wedge P_1 \rightarrow Q_2$  (根据题设)
- (5)  $Q_2$  ((3) (4) 分离)。

这就是说，“另一个人戴黑帽子”这个判定是必然可以作出的，但是这与题设条件（即“另一个没有作出判定”）相矛盾，因此， $P_1$ 为假，即 $\neg P_1$ 为真，故可得：  
：

- (6)  $\neg P_1$ ;
- (7)  $\neg P_1 \rightarrow P_2$  (根据题设) ;
- (8)  $P_2$  ((6) (7) 分离)。

这就是说，“猜对的人戴着黑帽子”是真的，所以猜对的人肯定的说：“我戴的是黑帽子”。

- 理发师问题：“理发师悖论”
  - 一个理发师声称他只给不为自己理发的人理发。
  - 问题：这个理发师是否给自己理发？
    - 如果他不给自己理发，那么按照他的声称，他应该给自己理发；如果他给自己理发，那么他便具有“不为自己理发”性质，也就是他不为自己理发。
- 悖论
  - “我现在在说谎”；“矛盾悖论”；“先有鸡先有蛋悖论”
  - 我爱自己，当且仅当，我不爱自己。
  - 我  $R$  也只  $R$  那样的人，他们不  $R$  自己。我  $R$  自己吗?
    - $R$  是任何二元关系，诸如：爱，恨，崇拜，新生，拥护，欺骗，打击，推荐，介绍……

- 理发师问题：罗素悖论的通俗形式
  - 逻辑错误：“非A即B”（潜前提： $A \neq B$ ）
  - “给 $x$ 理发的人，非 $x$ 自己即我”
- “罗素悖论”：引发“第三次数学危机”
  - 如果集合具有自己属于自己的性质，那么我们称这个集合是“自吞的”，比如所有集合的集合。
  - 假设 $T$ 是所有不自吞集合的集合。那么请问 $T$ 是否是自吞的？
  - 如果说 $T$ 不是自吞的，那么 $T$ 将属于自己，那么 $T$ 就是自吞的。
  - 如果说 $T$ 是自吞的，那么 $T$ 便具有 $T$ 内元素的性质“不自吞”，即 $T$ 是不自吞的。

## 3.5 基于规则的演绎推理

- 上一节讨论了归结演绎推理，它需要把谓词公式化为子句形，尽管这种转化在逻辑上是等价的，但是原来蕴含在谓词公式中的一些重要信息却会在求取子句形的过程中被丢失。例如，下面的几个蕴含式
  - $\neg A \wedge \neg B \rightarrow C, \neg A \wedge \neg C \rightarrow B, \neg A \rightarrow B \vee C,$
  - $\neg B \rightarrow A \vee C,$
  - 都与子句
  - $A \vee B \vee C$
  - 等价。但在 $A \vee B \vee C$ 中，得不到原逻辑公式中所蕴含的逻辑含义。况且，在不少情况下人们多希望使用接近于问题原始描述的形式来进行求解，而不希望把问题描述化为子句集。

## 3.5 基于规则的演绎推理

- 规则是一种比较接近于人们习惯的问题描述方式，按照这种问题描述方式进行求解的系统称为基于规则的系统，或者叫做基于规则的演绎系统。
  - 保留蕴涵式，将其作为推理规则（启发式知识），用于直接推导目标公式，不仅符合人的自然思维方式，也能通过规则更有效地引导演绎推理过程。
- 规则演绎系统的推理可分为正向演绎推理、逆向演绎推理和双向演绎推理三种形式。

### 3.5.1 规则正向演绎推理

- 规则正向演绎和前面所讨论过的正向推理相对应。
- 它是从已知事实出发，正向使用规则，直接进行演绎，直至到达目标为止的一种证明方法。其演绎过程容易被人们所理解。

## 3.5.1 规则正向演绎推理

### 1. 事实表达式的与/或形变换 (1/2)

- 在一个基于规则的正向演绎系统中，其前提条件中的事实表达式是作为系统的初始综合数据库来描述的。对事实的化简，只须转换成不含蕴含符号“ $\rightarrow$ ”的与/或形表示即可，而不必化成子句集。把事实表达式化为非蕴含形式的与/或形的主要步骤如下：
  - (1) 消去蕴含符号。
    - 在事实表达式中很少有含蕴含符号“ $\rightarrow$ ”出现，因为可把蕴含式表示成规则。
  - (2) 把“ $\neg$ ”移到紧靠谓词的位置，直到每个否定符号的辖域最多只含一个谓词为止。
  - (3) 前束化。
  - (4) 对全称量词辖域内的变量进行改名和标准化，对存在量词量化的变量用**skolem**函数代替，使不同量词约束的变元有不同的名字。
  - (5) 消去全称量词，而余下的变量都被认为是全程量词量化的变量。
  - (6) 对变量进行换名，使主合取元具有不同的变量名。

## 3.5.1 规则正向演绎推理

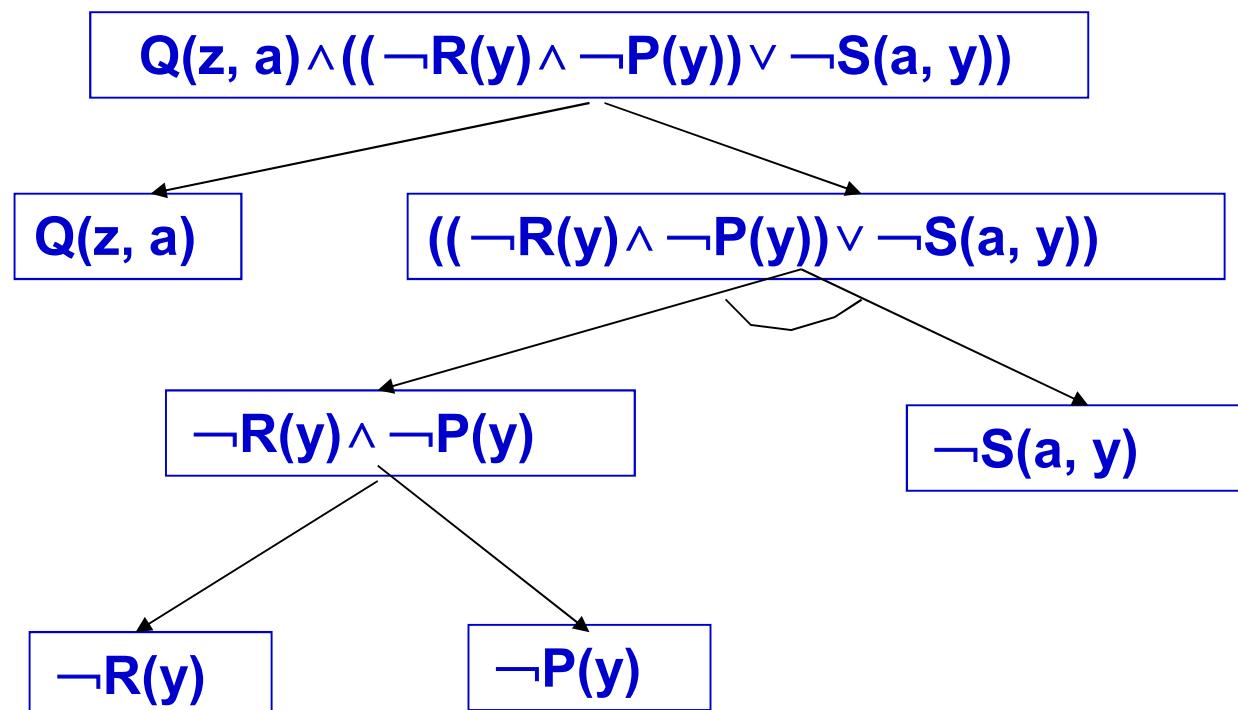
### 1. 事实表达式的与/或形变换 (2/2)

- 例如，有如下表达式
- $(\exists x) (\forall y)(Q(y, x) \wedge \neg((R(y) \vee P(y)) \wedge S(x, y)))$
- 可把它转化为：
- $Q(z, a) \wedge ((\neg R(y) \wedge \neg P(y)) \vee \neg S(a, y))$
- 这就是与/或形表示。

## 3.5.1 规则正向演绎推理

### 2. 事实表达式的与/或树表示 (1/3)

- 事实表达式的与/或形可用一棵与/或树表示出来。例如，对上例所给出的与/或式，可用如下与/或树来表示。



## 3.5.1 规则正向演绎推理

### 2. 事实表达式的与/或树表示 (2/3)

- 在该图中，每个节点表示该事实表达式的一个子表达式，子表达式之间的与、或关系规定如下：
- 当某个表达式为 $k$ 个子表达式的析取时，例如 $E_1 \vee E_2 \vee \dots \vee E_k$ ，其中的每个子表达式 $E_i$ 均被表示为 $E_1 \vee E_2 \vee \dots \vee E_k$ 的后继节点，并由一个 $k$ 线连接符将这些后继节点都连接到其父节点，即表示成与的关系。
- 当某个表达式为 $k$ 个子表达式的合取时，例如 $E_1 \wedge E_2 \wedge \dots \wedge E_k$ ，其中的每个子表达式 $E_i$ 均被表示成 $E_1 \wedge E_2 \wedge \dots \wedge E_k$ 的一个单一的后继节点，并由一个单一连接符连接到其父节点，即表示成或的关系。
- 这样，与/或树的根节点就是整个事实表达式，端节点均为事实表达式中的一个文字。有了与/或树的表示，就可以求出其解树（结束于文字节点上的子树）集。并且可以发现，事实表达式的子句集与解树集之间存在着一一对应关系，即解树集中的每个解树都对应着子句集中的一子句。其对应方式为：解树集中每个解树的端节点上的文字的析取就是子句集中的一子句。
- 例如，上图所示的与/或树有3个解树，分别对应这以下3个子句：
  - $Q(z, a)$
  - $\neg R(y) \vee \neg S(a, y)$
  - $\neg P(y) \vee \neg S(a, y)$

## 3.5.1 规则正向演绎推理

### 2. 事实表达式的与/或树表示 (3/3)

- 可以把与/或树看作是对子句集的简洁表示。
- 表示的通用性差一些，原因是与/或树中的合取元没有进一步展开，因此不能象在子句形中那样对某些变量进行改名，这就限制了与/或树表示的灵活性。例如，上面的最后一个子句，在子句集中其变量y可全部改名为x，但却无法在与/或树中加以表示，因而失去了通用性，并且可能带来一些困难。
- 还需要指出，这里的与/或树是作为综合数据库的一种表示，其中的变量受全称量词的约束。而在第2章可分解产生式系统中，所描述的与/或树则是搜索过程的一种表示，两者有着不同的目的和含义，因此应用时应加以区分。

## 3.5.1 规则正向演绎推理

### 3. 规则的与/或形变换

- 在与/或形正向演绎系统中，是以正向方式使用规则(**F**规则)对综合数据库中的与/或树结构进行变换的。为简化这种演绎过程，通常要求**F**规则应具有如下形式：
  - $L \rightarrow W$
  - 其中， $L$ 为单文字， $W$ 为与/或形公式。
  - 如果领域知识的规则表示形式与上述要求不同，则应将它转换成要求的形式。其变换步骤如下：

## 3.5.1 规则正向演绎推理

### 3. 规则的与/或形变换

- (1) 暂时消去蕴含符号 “ $\rightarrow$ ” 。  
$$(\forall x)((\exists y)(\forall z)P(x, y, z)) \rightarrow (\forall u)Q(x, u)$$
- 变为：  
$$(\forall x)(\neg((\exists y)(\forall z)P(x, y, z)) \vee (\forall u)Q(x, u))$$
- (2) 把否定符号 “ $\neg$ ” 移到紧靠谓词的位置上：  
$$(\forall x)((\forall y)(\exists z)\neg P(x, y, z)) \vee (\forall u)Q(x, u)$$
- (3) 引入Skolem函数，消去存在量词：  
$$(\forall x)((\forall y)(\neg P(x, y, f(x, y))) \vee (\forall u)Q(x, u))$$
- (4) 化成前束式，消去全部全称量词：  
$$\neg P(x, y, f(x, y)) \vee Q(x, u)$$
- 此公式中的变元都被视为受全称量词约束的变元。
- (5) 恢复蕴含式表示。

利用等价关系 “ $\neg P \vee Q \Leftrightarrow P \rightarrow Q$ ” 将上式变为：

$$P(x, y, f(x, y)) \rightarrow Q(x, u)$$

## 3.5.1 规则正向演绎推理

### 3. 规则的与/或形变换

- 在上述对F规则的要求中，之所以限制其前件为单文字，是因为表示事实的与/或树的叶节点都是单文字，这样就可用F规则的前件与叶节点进行简单匹配。
- 对非单文字情况，若形式为 $L_1 \vee L_2 \rightarrow W$ ，则可将其转换成与之等价的两个规则 $L_1 \rightarrow W$ 与  $L_2 \rightarrow W$ 进行处理。
- 单文字的限制对知识的表示有所限定，例如，

$$L_1 \wedge L_2 \Rightarrow W$$

这样的规则就不允许出现，只有转变为

$$L_1 \Rightarrow \neg L_2 \vee W$$

或  $L_2 \Rightarrow \neg L_1 \vee W$

才能使用

丢失了启发式知识（ $L_1$ 或 $L_2$ 同时为真导致 $W$ 为真）。

## 3.5.1 规则正向演绎推理

### 4. 目标公式的表示形式

- 与/或树正向演绎系统要求目标公式用子句形表示。如果目标公式不是子句形，则需要化成子句形。把一个目标公式转化为子句形的步骤与第3节所述的化简子句形的步骤类似，但**Skolem**化的过程不同。目标公式的**Skolem**化过程是化简子句形的**Skolem**过程的对偶形式，即把目标公式中属于存在量词辖域内的全称变量用存在量词量化变量的**Skolem**函数来代替，使经**Skolem**化目标公式只剩下存在量词，然后再对析取元作变量替换，最后把存在量词消掉。
- 例如：设目标公式为
- $(\exists y) (\forall x) (P(x, y) \vee Q(x, y))$
- 用**Skolem**函数消去全称量词后有
- $(\exists y)(P(f(y), y) \vee Q(f(y), y))$
- 进行变量换名，使每个析取元具有不同的变量符号，于是有
- $(\exists y)P(f(y), y) \vee (\exists z)Q(f(z), z)$
- 最后，消去存在量词得
- $P(f(y), y) \vee Q(f(z), z)$
- 这样，目标公式中的变量都假定受存在量词的约束

## 3.5.1 规则正向演绎推理

### 5. 规则正向演绎系统（1/5）

- 规则正向演绎推理过程是从已知事实出发，不断运用F规则，推出欲证明目标公式的过程。即先用与/或树把已知事实表示出来，然后再用F规则的前件和与或树的叶节点进行匹配，并通过一个匹配弧把匹配成功的F规则加入到与/或树中，依此使用F规则，直到产生一个含有以目标节点为终止节点的解图为止。

## 3.5.1 规则正向演绎推理

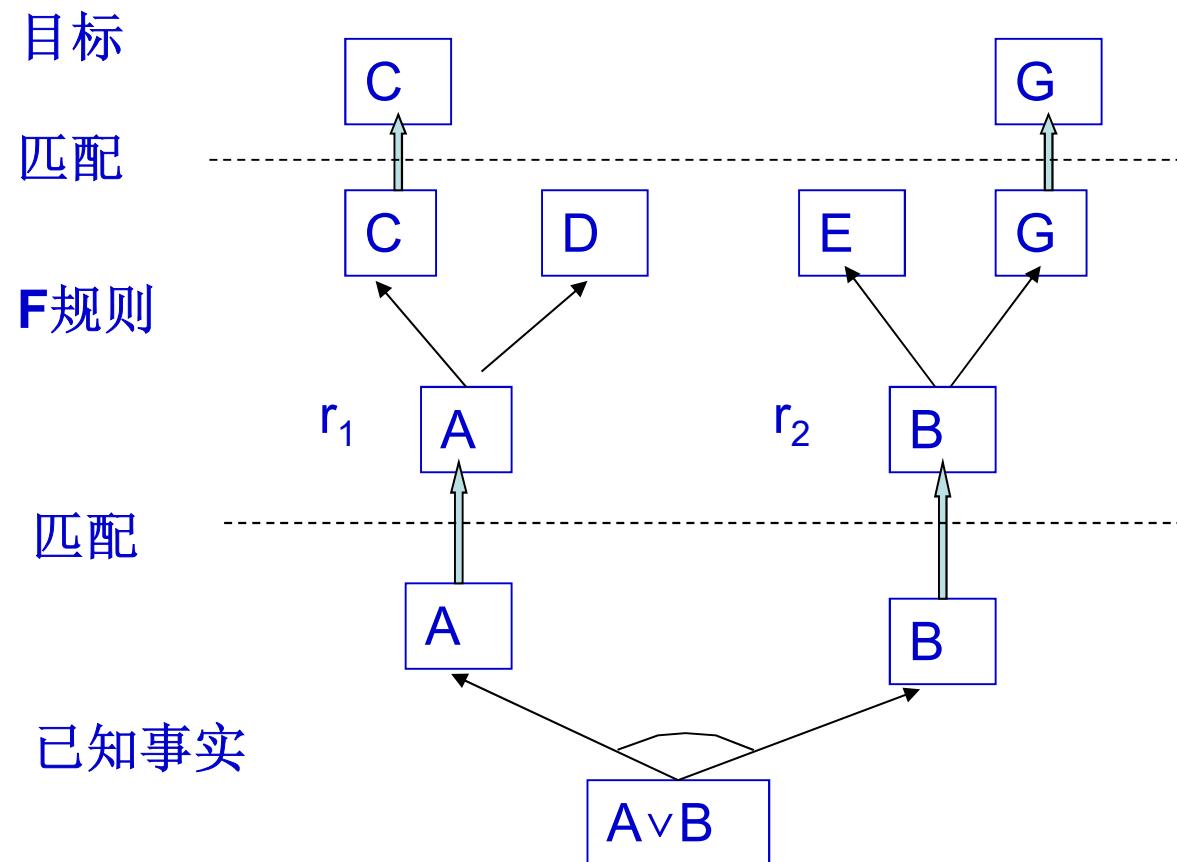
### 5. 规则正向演绎系统 (2/5)

- 命题逻辑的规则演绎过程
- 例3.25 设已知事实为:  $A \vee B$
- F规则为:  $r_1: A \rightarrow C \wedge D$
- $r_2: B \rightarrow E \wedge G$
- 目标公式为:  $C \vee G$
- 证明: 先将已知事实用与/或树表示出来, 然后再用匹配弧把  $r_1$  和  $r_2$  分别连接到事实与/或树中与  $r_1$  和  $r_2$  前件匹配的两个不同端节点, 由于出现了以目标节点为终节点的解树, 故推理过程结束。

## 3.5.1 规则正向演绎推理

### 5. 规则正向演绎系统 (3/5)

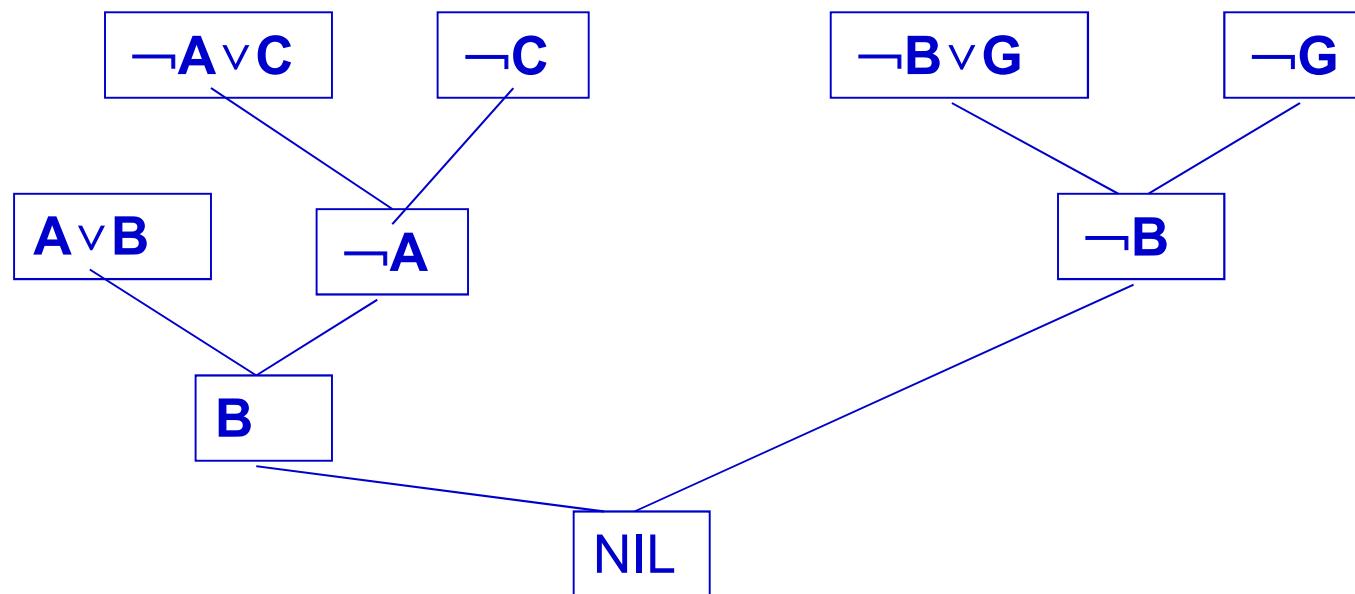
- 在该图中，双箭头表示匹配弧，它相当于一个单线连接符。



## 3.5.1 规则正向演绎推理

### 5. 规则正向演绎系统 (4/5)

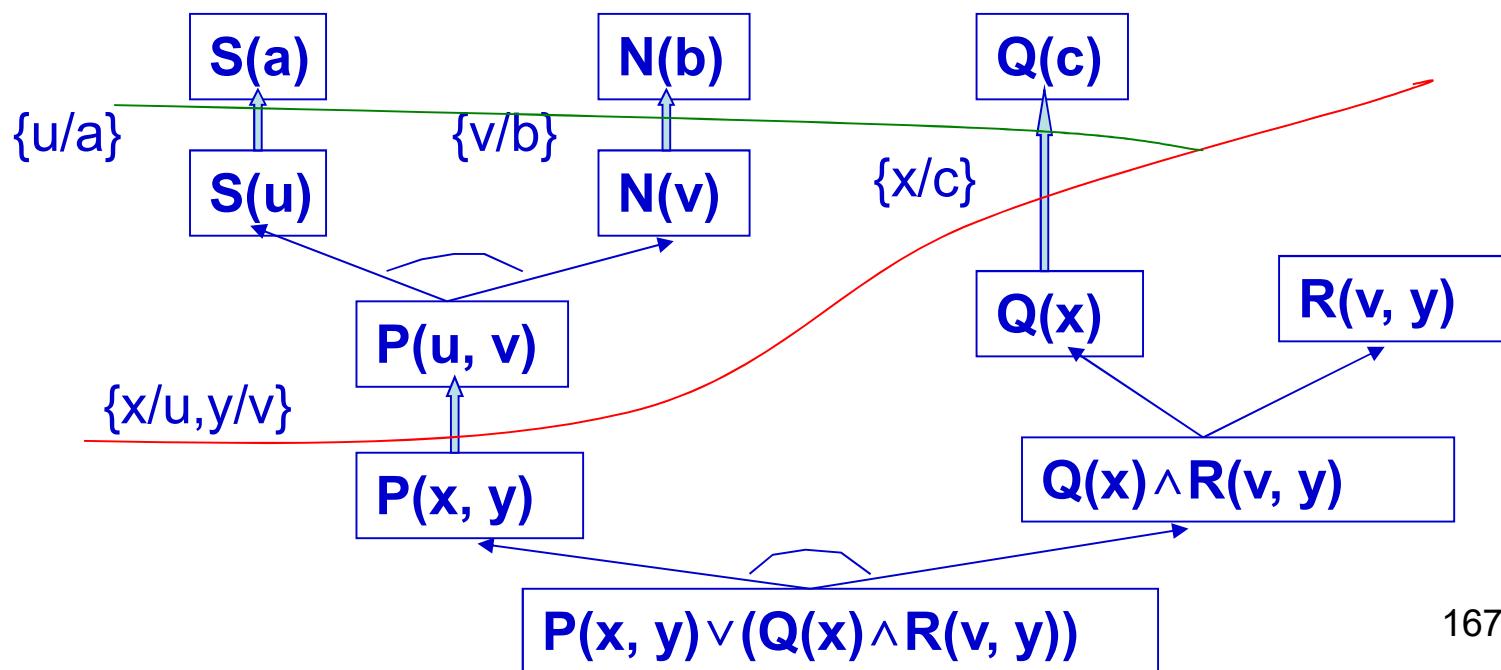
- 为了验证上述推理的正确性，用归结演绎推理证明。由已知事实、规则及目标的否定可得到如下子句集：
- $\{A \vee B, \neg A \vee C, \neg A \vee D, \neg B \vee E, \neg B \vee G, \neg C, \neg G\}$
- 与正向演绎推理所得到的结果是一致的。



## 3.5.1 规则正向演绎推理

### 5. 规则正向演绎系统 (5/5)

- 在谓词逻辑情况下，由于事实、F规则及目标中均含有变元，因此，其规则演绎过程还需要用最一般合一进行置换。
- 例3.26 设已知事实的与/或形表示为： $P(x, y) \vee (Q(x) \wedge R(v, y))$
- F规则为： $P(u, v) \rightarrow (S(u) \vee N(v))$
- 目标公式为： $S(a) \vee N(b) \vee Q(c)$
- 其推理过程如下图所示。



- 演绎推理过程可能会对同一变量进行不一致的置换，从而导致不一致解图的生成。
- 需检查所有置换的一致性，即：不能有互相矛盾的置换，称为：**一致解图**
- 例：**事实：**Fido会吠叫（Barks）和咬人（Bites），否则Fido就不是狗（Dog）。

**规则：**1. 所有猎狐狗（Terrier,一种小猎狗）都是狗。

2. 所有会吠叫的东西都是吵人的（Noisy）。

**目标：**

存在某个东西，除非它不是猎狐狗，否则是吵人的。

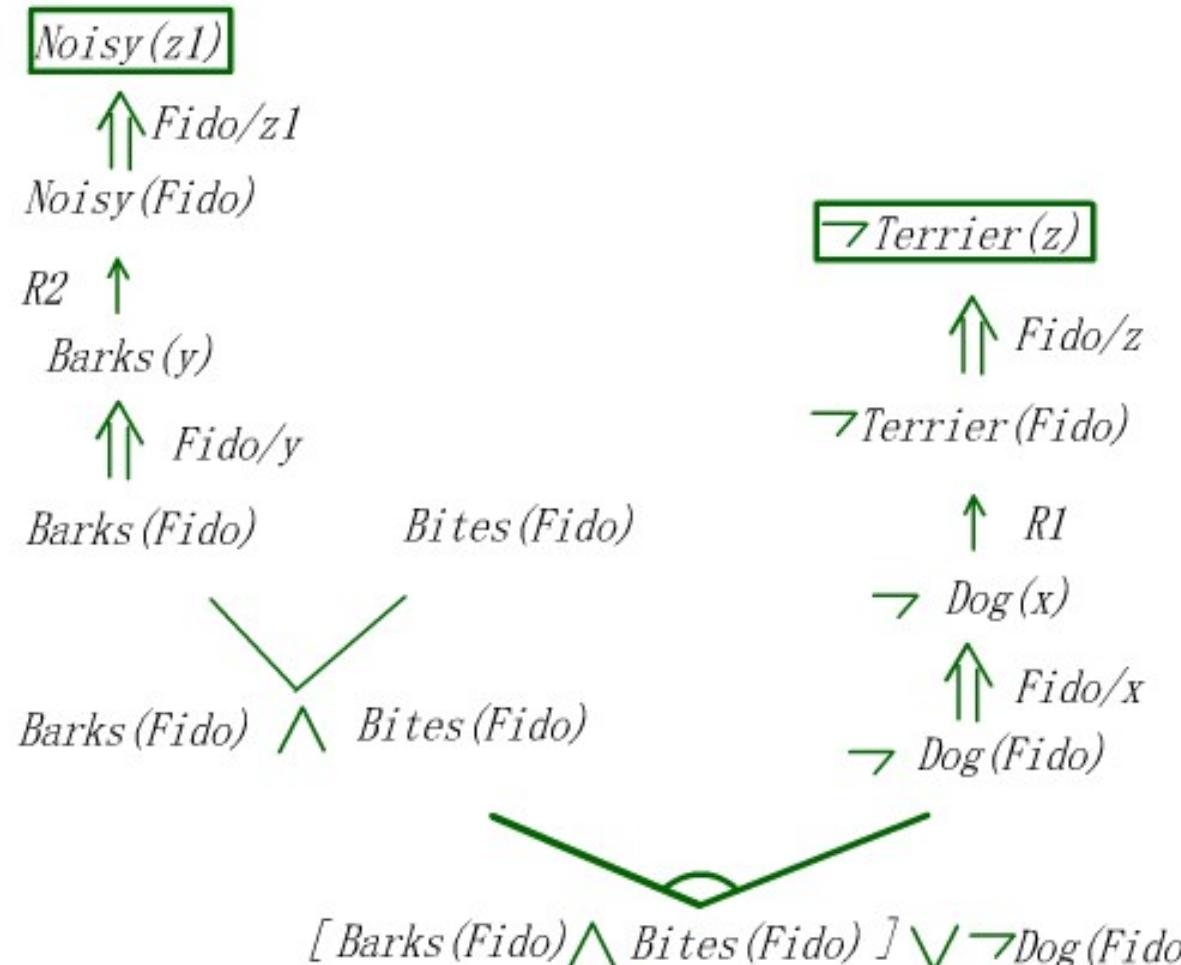
- 首先作形式化表示:
- 事实表达式:  
 $[Barks(Fido) \wedge Bites(Fido)] \vee \neg Dog(Fido)$
- 规则:  
R1:  $(\forall x) [Terrier(x) \Rightarrow Dog(x)]$   
R2:  $(\forall y) [Barks(y) \Rightarrow Noisy(y)]$
- 目标公式:  
 $(\exists z) [\neg Terrier(z) \vee Noisy(z)]$

- 对规则和目标公式标准化：

规则:  $\neg \text{Dog}(x) \Rightarrow \neg \text{Terrier}(x)$  逆否律

$\text{R2: } \text{Barks}(y) \Rightarrow \text{Noisy}(y)$

目标公式:  $\neg \text{Terrier}(z) \vee \text{Noisy}(z1)$



- 将解图中所有变量的置换都作为置换元素，收集于一个置换中，称为 **解图置换**，并对其作合一复合处理。
- 1. 设解图置换中元素形如  $t_i/v_i (i = 1, 2, \dots, n)$ ,  $t_i$  指示置换项，而  $v_i$  指示变量。  
2. 建立两个分别由  $t_i$  和  $v_i$  构成的表达式：

$$U_1 = (v_1, v_2, \dots, v_n)$$

$$U_2 = (t_1, t_2, \dots, t_n)$$

- 3. 检查  $U_1$  和  $U_2$  能否合一，若不能合一，则该置换是不一致的，进而解图是不一致的；若能合一，则建立起使  $U_1$  和  $U_2$  合一的置换，称为解图置换  $S$  的合一复合，进而解图是一致的。

# 一致置换举例

当置换集没有矛盾存在时，称该置换集是一致的，否则就是不一致的。

u1	u2	U1 和 U2	合一复合
$\{A/x\}$	$\{B/x\}$	$U1=(x, x)$ $U2=(A, B)$	不一致
$\{x/y\}$	$\{y/z\}$	$U1=(y, z)$ $U2=(x, y)$	$\{x/y, x/z\}$
$\{f(z)/x\}$	$\{f(A)/x\}$	$U1=(x, x)$ $U2=(f(z), f(A))$	$\{f(A)/x, A/z\}$
$\{x/y, x/z\}$	$\{A/z\}$	$U1=(y, z, z)$ $U2=(x, x, A)$	$\{A/x, A/y, A/z\}$

- 例子中的解图置换是：

$$S = \{Fido/z_1, Fido/y, Fido/z, Fido/x\}$$

显然，其合一复合就是其自身。至此，已得到一致解图，演绎推理成功结束，只要将目标公式中的变量z、z1分别以置换项Fido取代，就可得到解答：

$$\neg \text{Terrier}(Fido) \vee \text{Noisy}(Fido)$$

- 在解图不一致的情况下，并不意味着演绎推理失败，因为可能存在另一解图且其是一致的。
- 只有在找不到一致解图的情况下，演绎推理失败。

# 举例

事实：

$$\neg D(F) \vee (B(F) \wedge I(F))$$

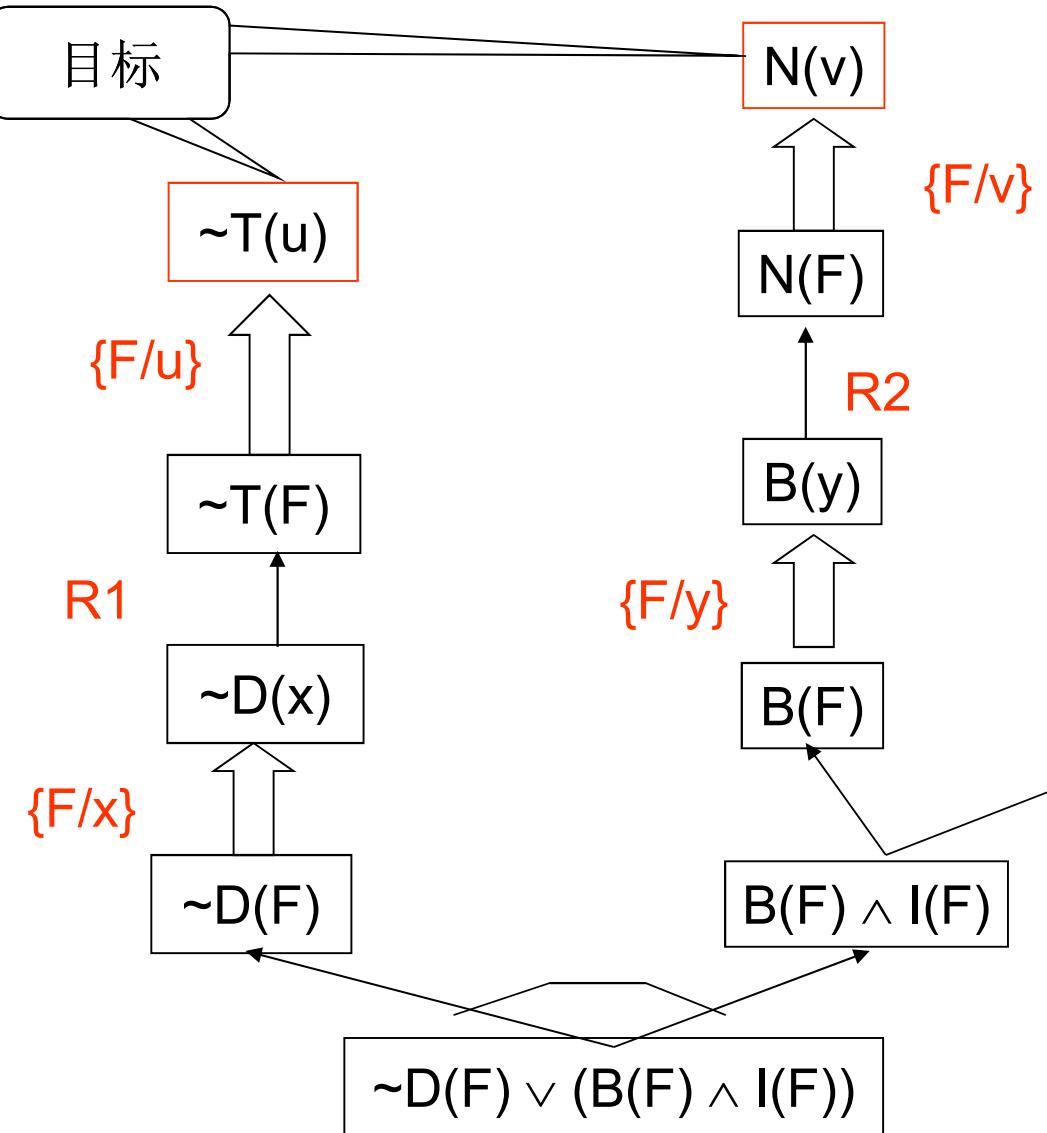
规则：

$$R1: \neg D(x) \rightarrow \neg T(x)$$

$$R2: B(y) \rightarrow N(y)$$

目标：

$$\neg T(u) \vee N(v)$$



$U_1 = (x, u, y, v)$   
 $U_2 = (F, F, F, F)$   
 合一复合  $u$ :  
 $\{F/x, F/u, F/y, F/v\}$   
 作用于目标:  
 $[\sim T(u) \vee N(v)]_u$   
 $= \sim T(F) \vee N(F)$

### 3.5.2 规则逆向演绎推理

- 规则逆向演绎推理过程是从目标公式的与/或树出发，反向使用规则（**B**规则），对目标公式的与/或树进行变换，直到得出含有事实节点一致解图为止。包括：
  - 目标公式的与/或形变换
  - 目标公式的与/或树表示
  - B规则的表示形式
  - 规则逆向演绎推理过程

## 3.5.2 规则逆向演绎推理

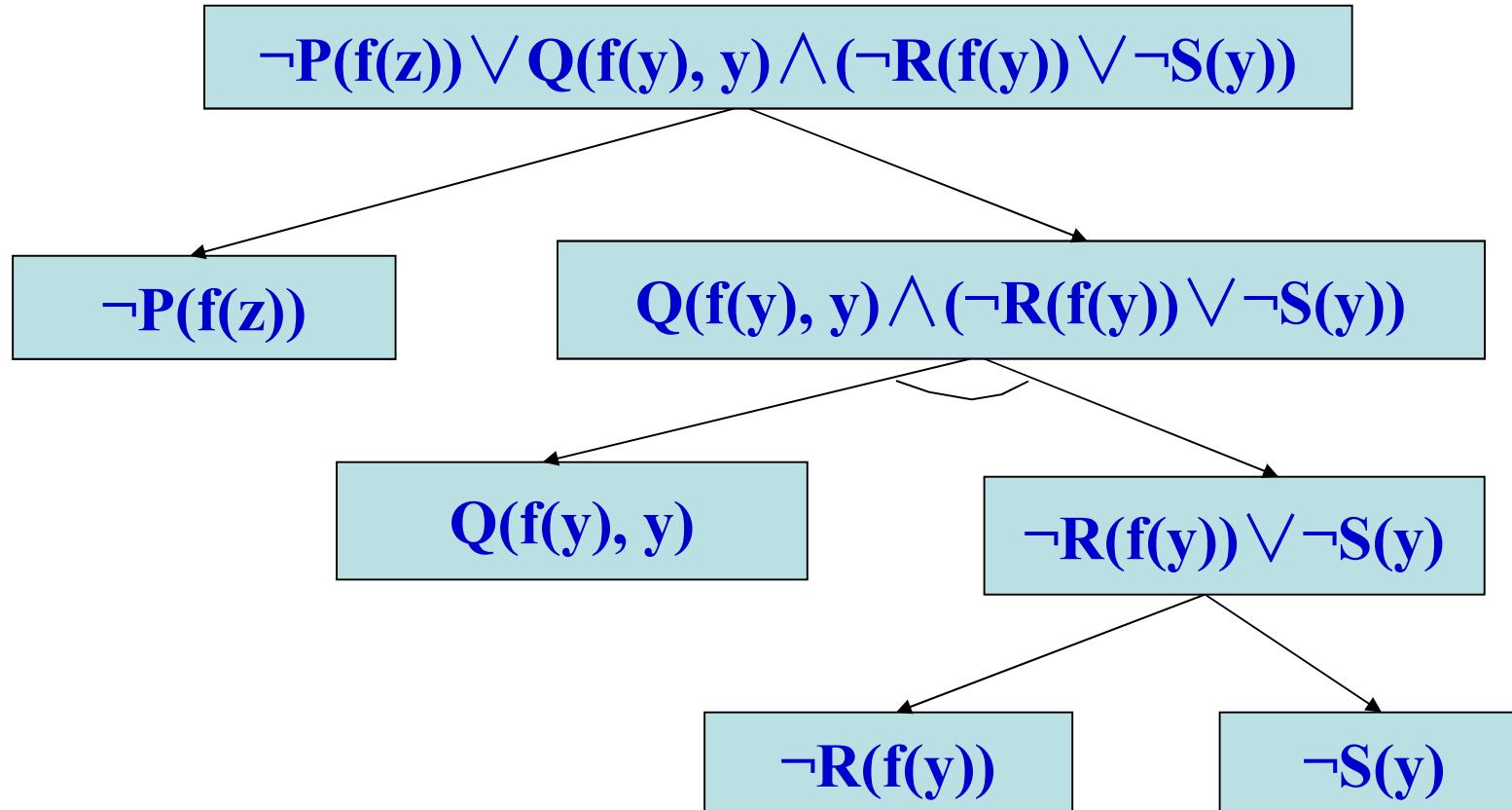
### 1. 目标公式的与/或形变换

- 逆向系统中的目标公式采用与/或形表示，其化简采用与正向系统中对事实表达式处理的对偶形式。
- 即要用存在量词约束变元的Skolem函数来替换由全称量词约束的相应变元，消去全称量词，再消去存在量词，并进行变元换名，使主析取元之间具有不同的变元名。
- 例如，有如下目标公式：
- $(\exists y) (\forall x)(P(x) \rightarrow (Q(x) \wedge \neg(R(x) \wedge S(y))))$
- Skolem化后为
- $\neg P(f(y)) \vee (Q(f(y), y) \wedge (\neg R(f(y)) \vee \neg S(y)))$
- 变元换名后为
- $\neg P(f(z)) \vee (Q(f(y), y) \wedge (\neg R(f(y)) \vee \neg S(y)))$

## 3.5.2 规则逆向演绎推理

### 2. 目标公式的与/或树表示

- 目标公式的与/或形:
- 子表达式之间的析取关系用单一连接符连接，表示称或的关系；
- 子表达式之间的合取关系则用**k**线连接符连接，表示为与的关系。
- 例如，对上述目标公式的与/或形，可用如下的与/或树表示。



在目标公式的与/或树中，若把叶节点用它们之间的合取及析取关系连接起来，就可得到原目标公式的三个子目标：

$\neg P(f(z))$   
 $Q(f(y), y) \wedge \neg R(f(y))$   
 $Q(f(y), y) \wedge \neg S(y)$

即：子目标是文字的合取式

## 3.5.2 规则逆向演绎推理

### 3. B规则和已知事实的表示形式

- **B规则的表示形式**
  - $W \rightarrow L$
  - 其中，前项W为任一与/或形公式，后项L为一单文字。
  - 当B规则为 $W \rightarrow L_1 \wedge L_2$ 时，则可化为两条规则 $W \rightarrow L_1$ 和 $W \rightarrow L_2$ 进行处理。
- **已知事实的表示形式**
  - 反向演绎系统的事表达式限制为文字合取形式，如
  - $F_1 \wedge F_2 \wedge \dots \wedge F_n$
  - 其中，每个 $F_i$  ( $i=1,2,\dots,n$ ) 都为单文字，且都可单独起作用，因此可表示为如下集合形式
  - $\{ F_1, F_2, \dots, F_n \}$

## 3.5.2 规则逆向演绎推理

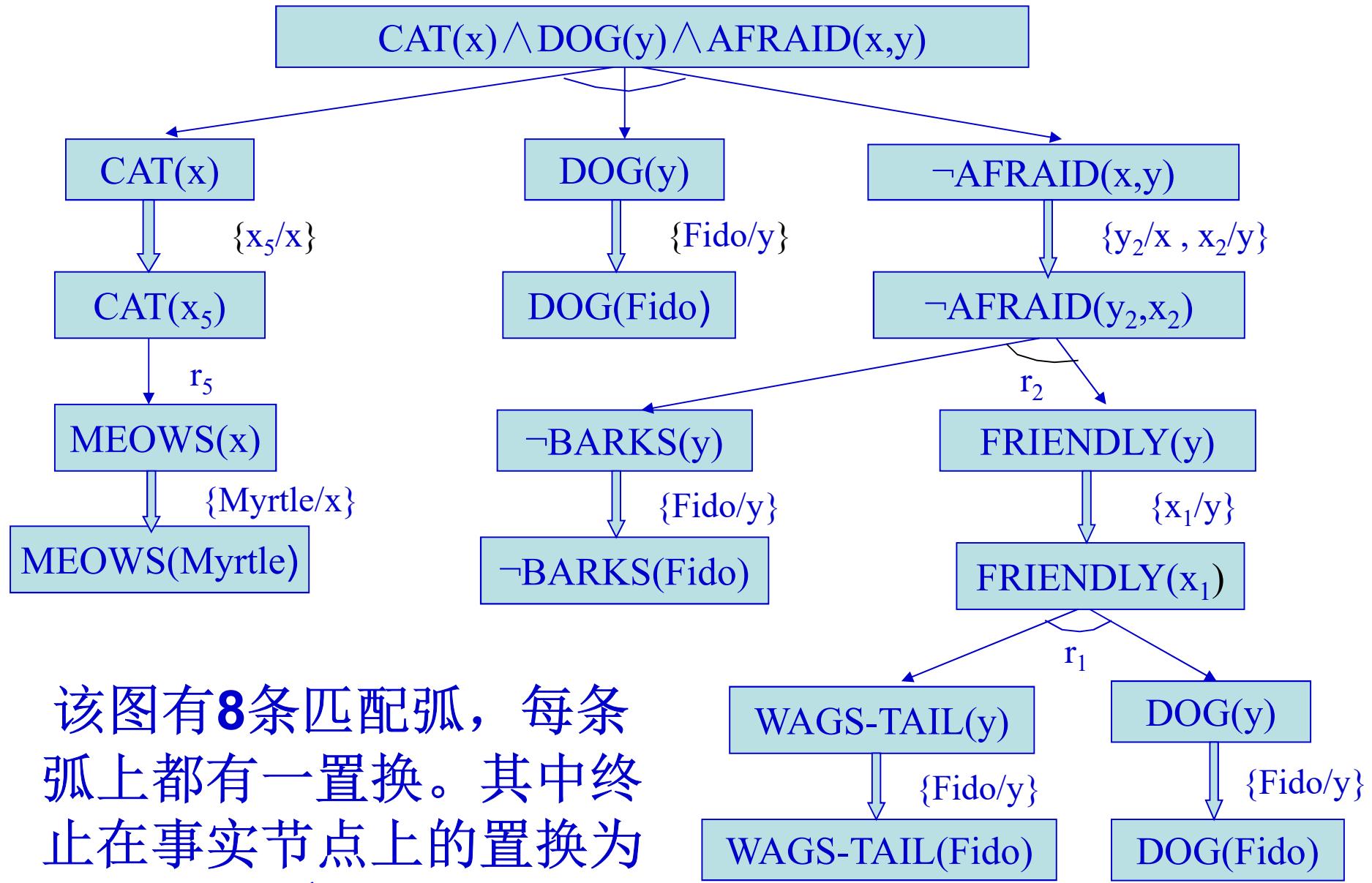
### 5. 规则逆向演绎推理过程(一)

- 规则逆向演绎推理过程
- 从目标公式的与/或树出发，不断用**B**规则的右部和与/或树的叶节点进行匹配，并将匹配成功的**B**规则用表有最一般合一匹配弧加入到与或树中，直到产生某个终止在事实节点上的一致解图为止。
- 一致解图是指在推理过程中所用到的置换应该是一致的。
- 例3.32 设有如下事实：
  - $f_1: \text{DOG}(\text{Fido})$  Fido是一只狗
  - $f_2: \text{BARKS}(\text{Fido})$  Fido是不叫的
  - $f_3: \text{WAGS-TAIL}(\text{Fido})$  Fido摇尾巴
  - $f_4: \text{MEOWS}(\text{Myrtle})$  猫咪的名字叫Myrtle

## 3.5.2 规则逆向演绎推理

### 5. 规则逆向演绎推理过程(二)

- 规则:  $r_1: (\text{WAGS-TAIL}(x_1) \wedge \text{DOG}(x_1)) \rightarrow \text{FRIENDLY}(x_1)$   
• 摆尾巴的狗是温顺的狗
- $r_2: (\text{FRIENDLY}(x_2) \wedge \text{BARKS}(x_2)) \rightarrow \text{AFRAID}(y_2, x_2)$   
• 温顺又不叫的东西是不值得害怕的
- $r_3: \text{DOG}(x_3) \rightarrow \text{ANIMAL}(x_3)$  狗为动物
- $r_4: \text{CAT}(x_4) \rightarrow \text{ANIMAL}(x_4)$  猫为动物
- $r_5: \text{MEOWS}(x_5) \rightarrow \text{CAT}(x_5)$  猫咪是猫
- 问题: 是否存在这样的一只猫和一条狗, 使得这只猫不害怕这只狗?
- 该问题的目标公式为
  - $(\exists x)(\exists y)(\text{CAT}(x) \wedge \text{DOG}(y) \wedge \neg \text{AFRAID}(x, y))$
  - 目标公式经变换(消去存在量词)后得到
  - $\text{CAT}(x) \wedge \text{DOG}(y) \wedge \text{AFRAID}(x, y)$



该图有**8**条匹配弧，每条弧上都有一置换。其中终止在事实节点上的置换为  
 $\{Myrtle/x\}$ 和 $\{Fido/y\}$ 。

## 一致解树

- 所有匹配弧上的置换集：  
 $\{x5/x\}, \{MYRTLE/x\}, \{FIDO/y\}, \{y2/x, x2/y\},$   
 $\{FIDO/y\}, \{x1/y\}, \{FIDO/y\}, \{FIDO/y\}$   
没有矛盾，是一致置换集。
- 把它们应用到目标公式，就得到该问题的解，  
得到问题的解：  
 $CAT(MYRTLE) \wedge DOG(FIDO) \wedge \neg AFRAID(MYRTLE, FIDO)$

- 例：应用逆向演绎推理来解决不等式问题
- 令谓词G表示大于 ( $>$ ) 关系， plus, times和divides 分别表示算术操作+、\*、/
  - $G(x, y)$ :  $x > y$ ;
  - $\text{plus}(A, B, C)$ :  $A + B + C$ ;
  - $\text{times}(A, B, C)$ :  $A * B * C$ ;
  - $\text{divides}(A, B)$ :  $A/B$
- 已知有事实：  $(E > 0) \wedge (B > 0) \wedge (A > 0) \wedge (C > 0) \wedge (C > E)$
- B规则:  $(x > 0) \wedge (y > z) \Rightarrow (x + y) > z$   
 $(x > 0) \wedge (y > z) \Rightarrow (xy > xz)$   
 $(x > wy) \wedge (y > 0) \Rightarrow (x/y > w)$
- 求证:  $[B(A+C)/E] > B$

- 首先作形式化表示：

事实表达式：

$$G(E, 0) \wedge G(B, 0) \wedge G(A, 0) \wedge G(C, 0) \wedge G(C, E)$$

规则：

$$R1: G(x, 0) \wedge G(y, z) \Rightarrow G(\text{plus}(x, y), z)$$

$$R2: G(x, 0) \wedge G(y, z) \Rightarrow G(\text{times}(x, y), \text{times}(x, z))$$

$$R3: G(x, \text{times}(w, y)) \wedge G(y, 0) \Rightarrow$$

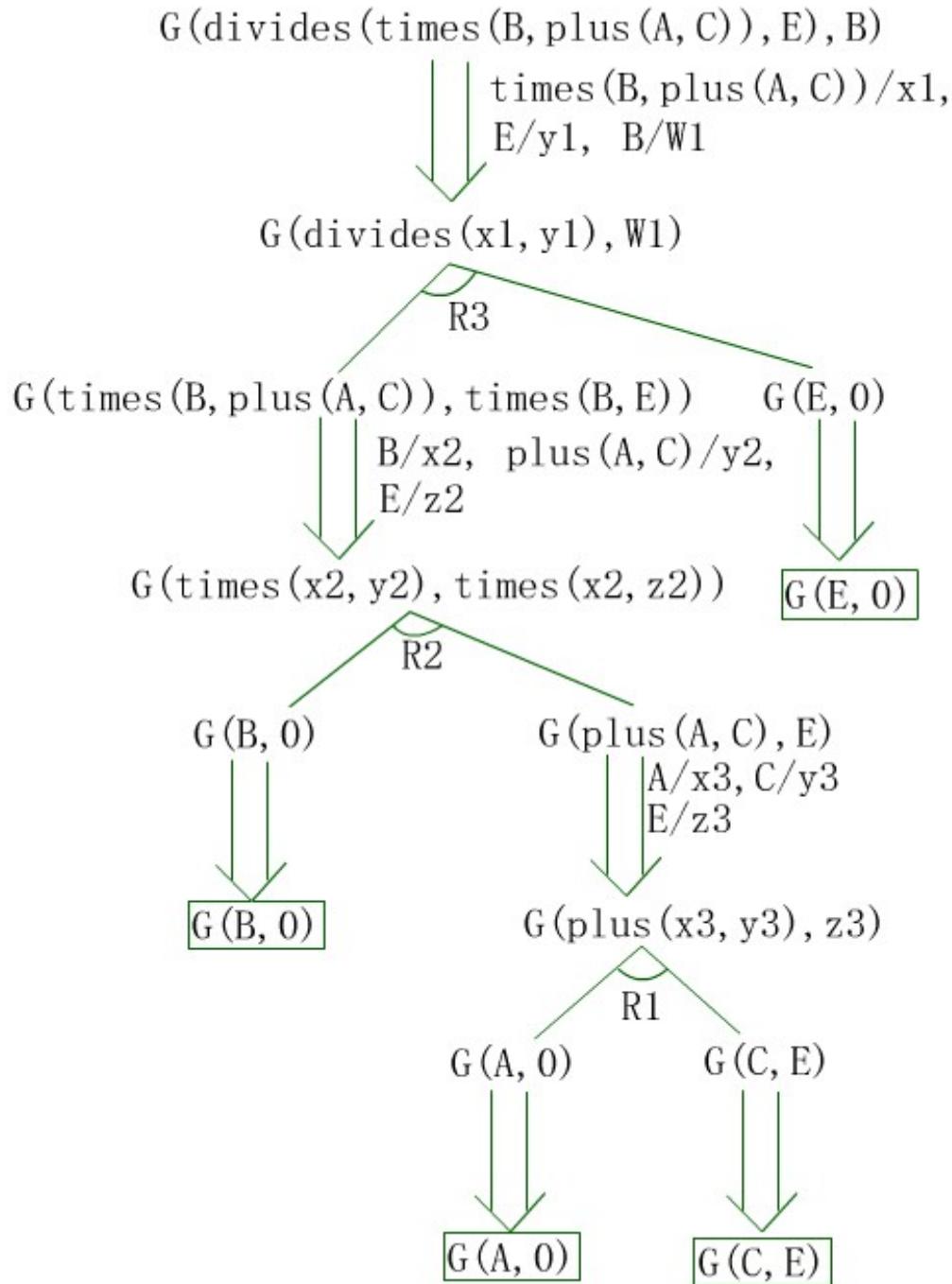
$$G(\text{divides}(x, y), w)$$

目标公式：

$$G(\text{divides}(\text{times}(B, \text{plus}(A, C)), E), B)$$

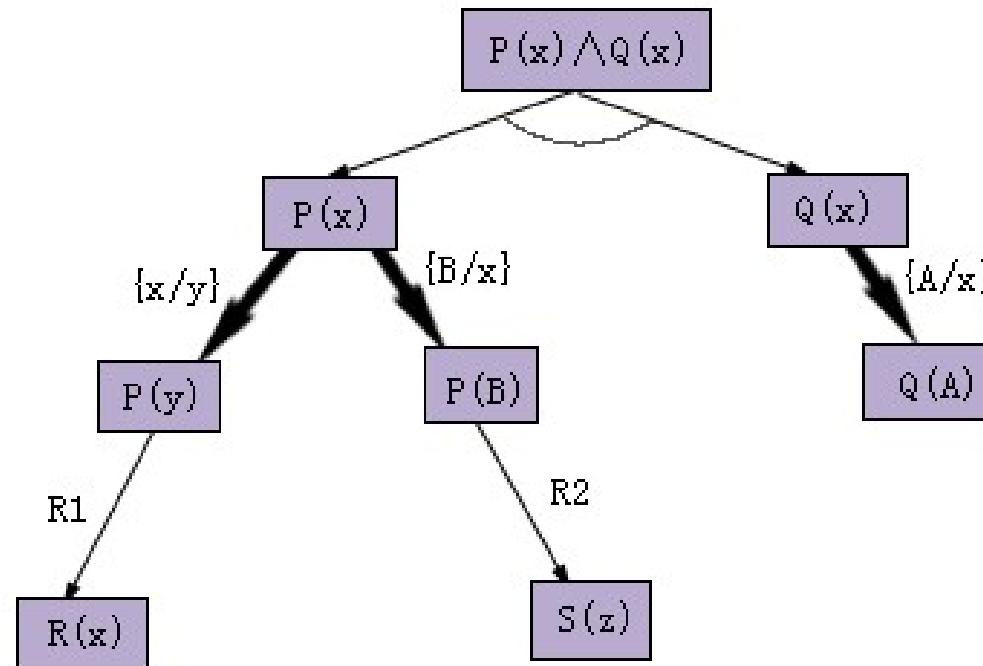
已经是标准化形式，可直接用于演绎推理

• 与或图：



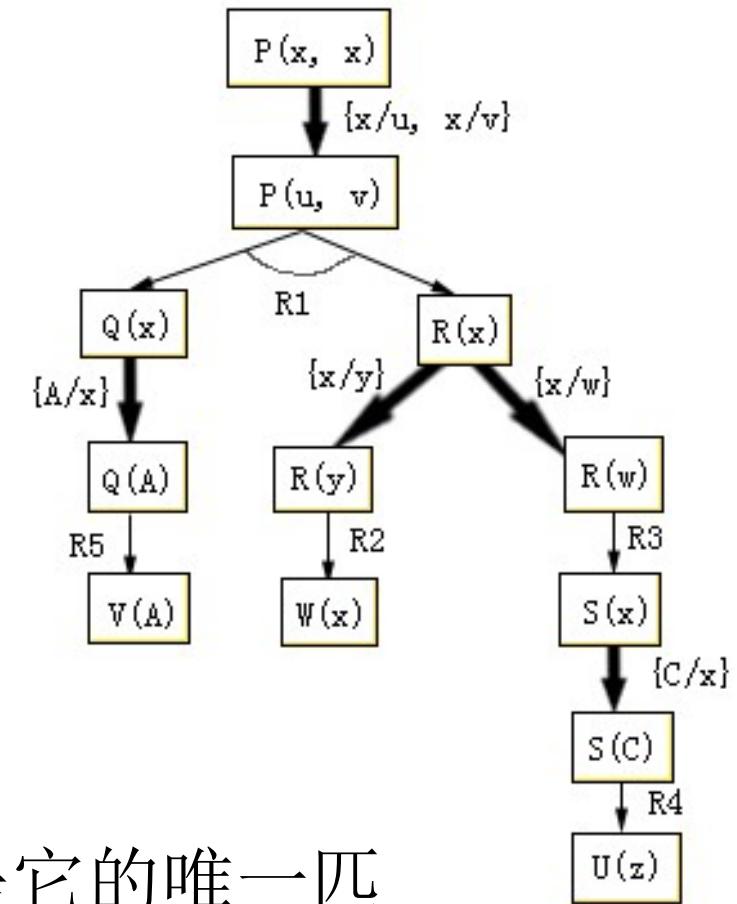
- 上图实际上就是一个解图，其所有叶节点都是事实节点。
- 事实表达式中的文字**G(C,0)**未出现在解图中，但并不影响推理的成功结束，因为推理的各个有"与"关系的分支都得到了原始证据（事实文字）的支持。
- 把演绎推理过程中建立的置换收集于解图置换**S**中：  
$$S = \{\text{times}(B, \text{plus}(A, C))/x_1, E_1/y_1, B/w_1, B/x_2, \text{plus}(A, C)/y_2, E/z_2, A/x_3, C/y_3, E/z_3\}$$

- 例：设要证明的目标： $P(x) \wedge Q(x)$   
 规则：R1:  $R(y) \rightarrow P(y)$   
 R2:  $S(z) \rightarrow P(B)$   
 事实： $R(A) \wedge Q(A)$
- 置换 $\{A/x\}$ 与 $\{B/x\}$ 是不一致的，只能保留一个。如果子目标 $Q(x)$ 只有一个唯一的匹配 $Q(A)$ ，则保留该匹配，而舍弃 $P(x)$ 与规则R2的匹配。



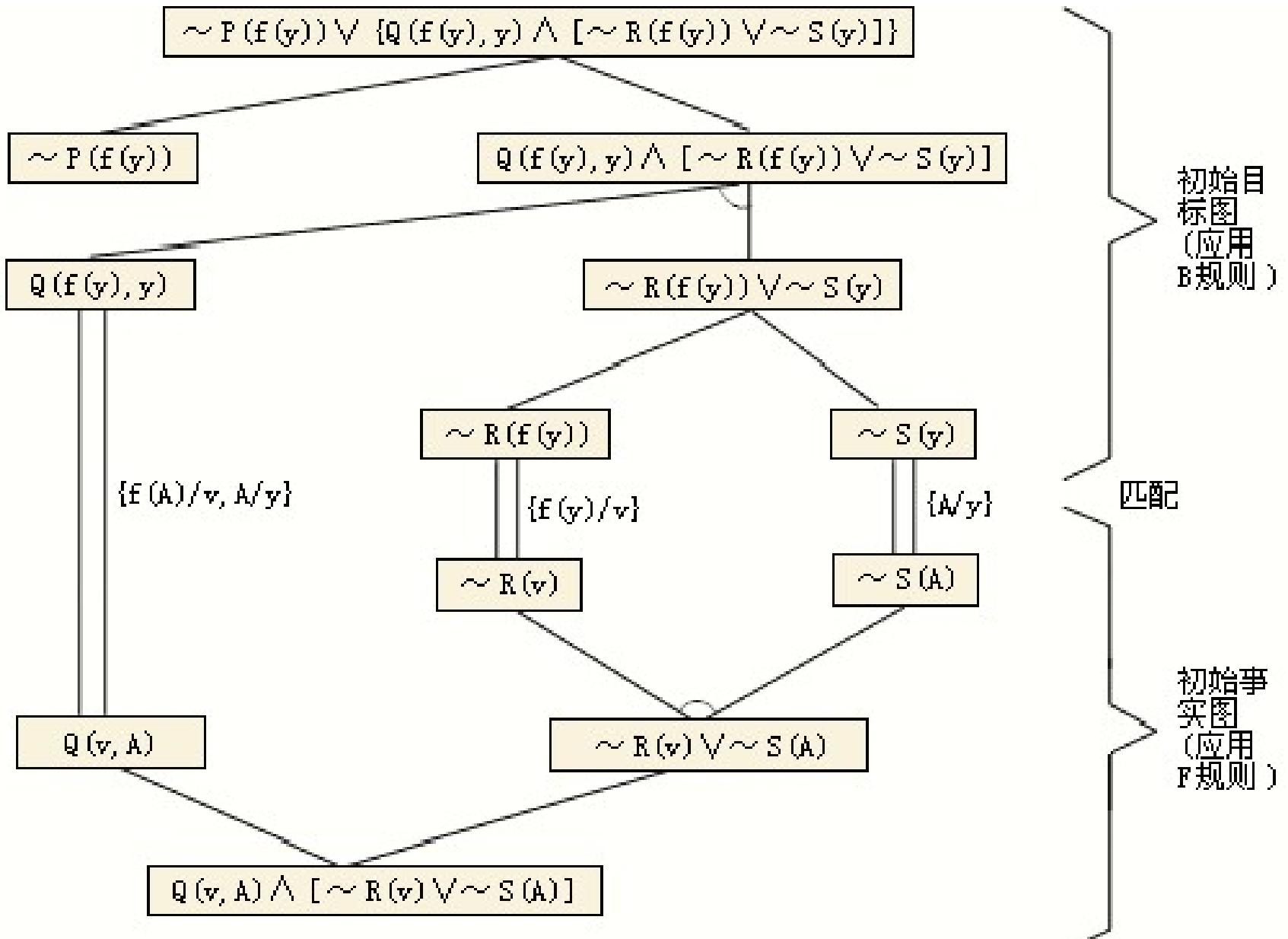
- 目标:  $P(x, x)$   
规则: R1:  $(Q(u) \wedge R(v)) \rightarrow P(u, v)$   
R2:  $W(y) \rightarrow R(y)$   
R3:  $S(w) \rightarrow R(w)$   
R4:  $U(z) \rightarrow S(C)$   
R5:  $V(A) \rightarrow Q(A)$

- 规则R4和R5的应用产生了两个不一致的置换 $\{A/x\}$ 和 $\{C/x\}$ 。
  - 对于 $S(x)$ 来说, 虽然规则R4是它的唯一匹配, 但由于前面 $R(x)$ 存在R2和R3两个匹配, 而R5是 $Q(x)$ 的唯一匹配, 所以只能舍弃规则R4这一边的匹配, 而保留规则R5这边的匹配。
  - 应修剪掉子目标 $U(z)$ , 而子目标 $S(x)$ 仍保留并允许作置换 $\{A/x\}$ 。



### 3.5.3 双向演绎推理

- 可以把正向和逆向演绎推理结合起来加以应用，建立基于规则的双向演绎推理系统。
  - **Pros:** 可以避免因单向推理而引入的限制，并实现优势互补。
  - **Cons:** 处理结束条件的复杂性，需分别支持正、逆向推理，增加了系统设计和知识获取的困难。
  - **折衷:** 更实用化的方式
    - 将复杂的问题求解任务划分为相对简单的若干子任务，然后根据子任务的特点选用正向或逆向演绎推理方式。



Cancel图

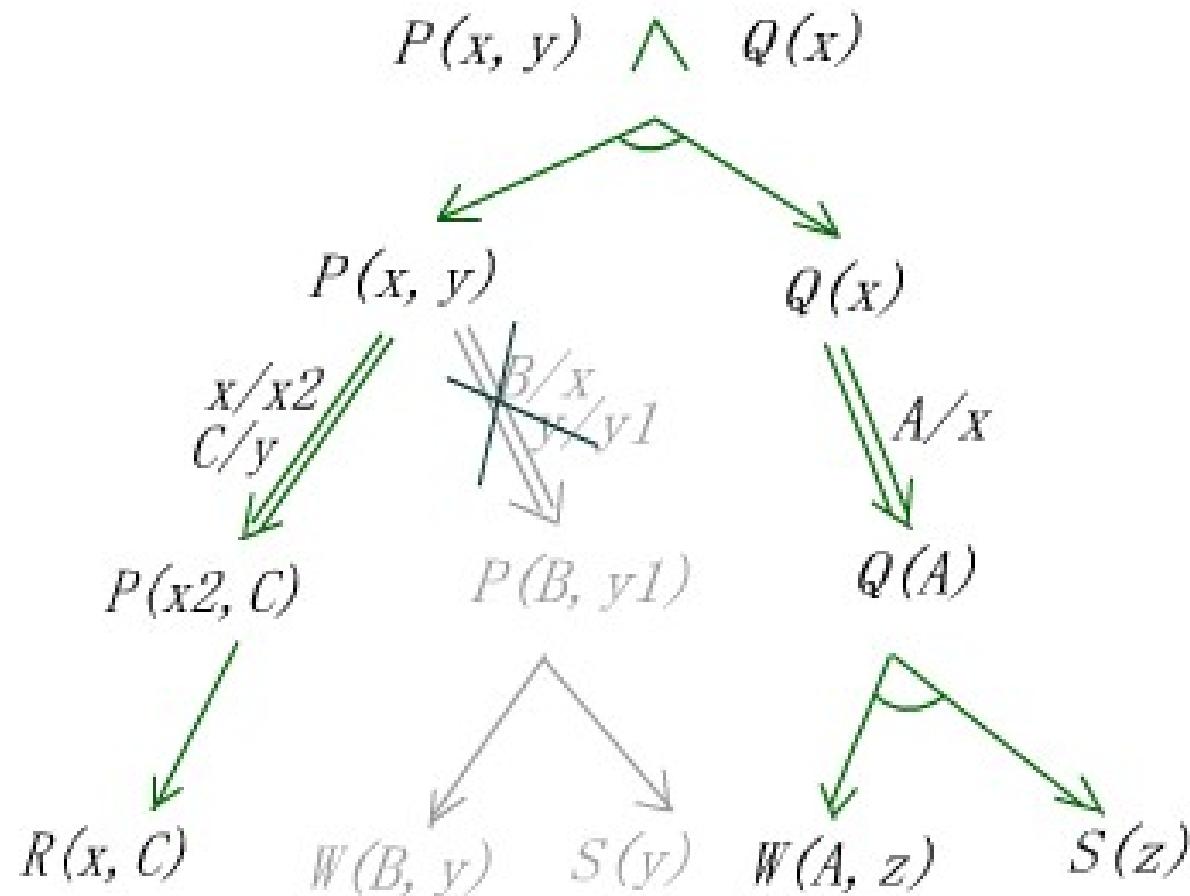
- 解图搜索修剪策略

在基于规则的演绎推理过程中，基本的解图搜索策略是先找到（生成）一个任意解图，检索其一致性；若否，则搜索（生成）另一个，直至找到一致解图为止。

- 由与或图中同一叶节点激活的多条规则隐含“或”关系，正是由于这种“或”关系，才导致了多个解图存在的可能性。
- 直到解图生成后才检查其一致性，往往太晚了；实际上，许多不一致的解图在其构建的早期就可发现其不一致。

所以建立**解图搜索修剪**策略，及早修剪掉不一致的局部解图是有重要意义的。

- 修剪策略，就是每当以规则扩展局部解图时，就及时检查解图置换的一致性，并随即修剪掉不一致的局部解图，以免除对这种局部解图的进一步徒劳搜索。



- 特点比较：

	正向演绎推理		逆向演绎推理	
问题求解的描述	事实	文字与或形	事实	文字合取式
	规则	$L \Rightarrow W$	规则	$W \Rightarrow L$
	目标	文字析取形	目标	文字与或形
初始与或图	相应于事实表达式 $\vee$ -K-连接		相应于目标公式 $\wedge$ -K-连接	
	F规则 事实-->目标		B规则 目标-->事实	
演绎推理				
结束条件	包含所有目标节点的一致解图		以事实节点作为所有终节点的一致解图	

# 小结

- 推理的相关概念和知识
- 谓词演算知识
- 自然演绎推理
- 归结演绎推理

~~基于规则的演绎推理~~