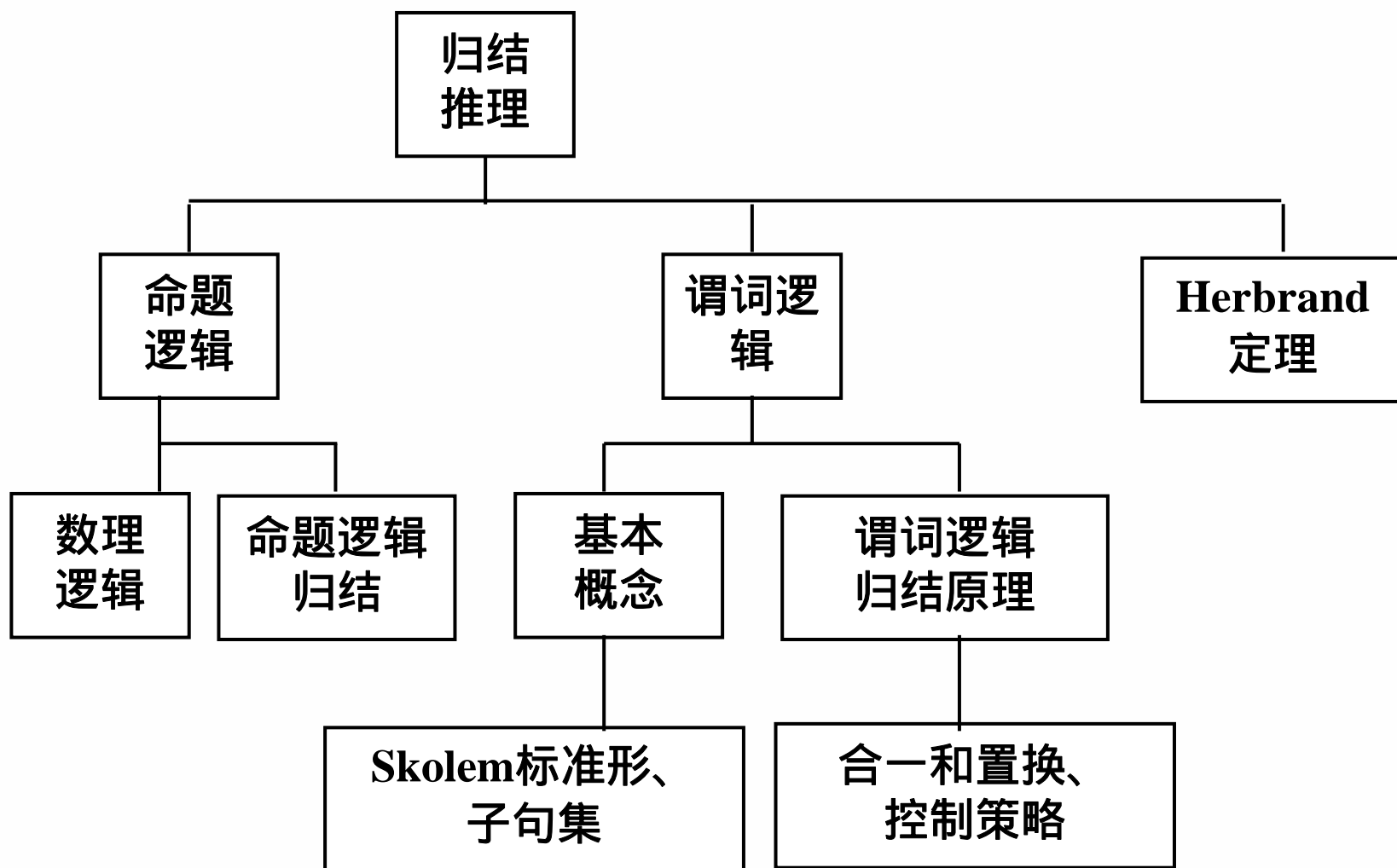


第三章 归结推理方法

- 概述
- 命题逻辑的归结法
- 谓词归结子句形
- 归结原理
- 归结过程的策略控制
- Herbrand定理



第三章 归结推理方法

✓ 概述

- 命题逻辑的归结法
- 谓词归结子句形
- 归结原理
- 归结过程的策略控制

概述

- 归结原理由J.A.Robinson由1965年提出。
 - 与演绎法(deductive inference)完全不同，新的逻辑演算(inductive inference)算法。
 - 迄今最有效的一阶逻辑半可判定的算法。即，一阶逻辑中任意恒真公式，归结原理总可在有限步内给以判定。
 - 语义网络、框架表示、产生式规则等都是推理方法为前提的。
 - 归结方法用于自动推理、自动证明。（“数学定理机器证明”）
- 本课程只讨论一阶谓词逻辑描述下的归结推理方法，不涉及高阶谓词逻辑问题。

第三章 归结推理方法

✓ 概述

- 命题逻辑的归结法
- 谓词归结子句形
- 归结原理
- 归结过程的策略控制
- Herbrand定理

第三章 归结推理方法

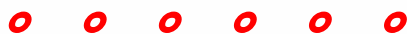
- ✓ 概述
- ✓ 命题逻辑的归结法
 - 谓词归结子句形
 - 归结原理
 - 归结过程的策略控制
 - Herbrand定理

命题逻辑的归结法

- 命题逻辑基础：

定义：

- 合取式： p 与 q ，记做 $p \wedge q$
- 析取式： p 或 q ，记做 $p \vee q$
- 蕴含式：如果 p 则 q ，记做 $p \rightarrow q$
- 等价式： p 当且仅当 q ，记做 $p \leftrightarrow q$



命题逻辑基础

- 定义：

- 若A无成假赋值，则称A为重言式或永真式；
- 若A无成真赋值，则称A为矛盾式或永假式；
- 若A至少有一个成真赋值，则称A为可满足的；
- 析取范式：仅由有限个简单合取式组成的析取式。
- 合取范式：仅由有限个简单析取式组成的合取式。

命题逻辑基础

- 基本等价式24个(1)

- 交换率： $p \quad q \Leftrightarrow q \quad p$;

$$p \quad q \Leftrightarrow q \quad p$$

- 结合率： $(p \quad q) \quad r \Leftrightarrow p \quad (q \quad r)$;

$$(p \quad q) \quad r \Leftrightarrow p \quad (q \quad r)$$

- 分配率： $p \quad (q \quad r) \Leftrightarrow (p \quad q) \quad (p \quad r)$;

$$p \quad (q \quad r) \Leftrightarrow (p \quad q) \quad (p \quad r)$$

命题逻辑基础

• 基本等值式 (1)

– 摩根率: $\neg (p \quad q) \Leftrightarrow \neg p \quad \neg q$;

$$\neg (p \quad q) \Leftrightarrow \neg p \quad \neg q$$

– 吸收率: $p \quad (p \quad q) \Leftrightarrow p$;

$$p \quad (p \quad q) \Leftrightarrow p$$

– 同一律: $p \quad 0 \Leftrightarrow p$;

$$p \quad 1 \Leftrightarrow p$$

– 蕴含等值式: $p \quad q \Leftrightarrow \neg p \quad q$

– 假言易位式: $p \quad q \Leftrightarrow \neg q \quad \neg p$

永真蕴涵式

化简式： $P \vee Q \Rightarrow P, P \vee Q \Rightarrow Q$

附加式： $P \Rightarrow P \vee Q, Q \Rightarrow P \vee Q$

析取三段论： $\neg P, P \vee Q \Rightarrow Q$

假言推理： $P, P \vee Q \Rightarrow Q$

拒取式： $\neg Q, P \vee Q \Rightarrow \neg P$

假言三段论： $P \vee Q, Q \vee R \Rightarrow P \vee R$

二难推论： $P \vee Q, P \vee R, Q \vee R \Rightarrow R$

全称固化： $(\forall x)P(x) \Rightarrow P(y)$

存在固化： $(\exists x) P(x) \Rightarrow P(a)$

命题例

- **命题**：能判断真假（不是既真又假）的陈述句。

简单陈述句描述事实、事物的状态、关系等性质。

例如： 1. $1+1=2$

- 2. 雪是黑色的。
- 3. 北京是中国的首都。
- 4. 到冥王星去渡假。

判断一个句子是否是命题，有先要看它是否是陈述句，而后看它的真值是否唯一。

而例如：1. 快点走吧！

2. 到那去？

3. $x+y>10$

等等句子，都不是命题。

命题表示公式 (1)

将陈述句转化成命题公式。

如：设“下雨”为 p ，“骑车上上班”为 q ，

1. “只要不下雨，我骑自行车上班”。 $\neg p$ 是 q 的充分条件，

因而，可得命题公式： $\neg p \rightarrow q$

2. “只有不下雨，我才骑自行车上班”。 $\neg p$ 是 q 的必要条件，

因而，可得命题公式： $q \rightarrow \neg p$

命题表示公式 (2)

例如：

- 1. “如果进城我就去看你，除非我很累。”

设： p ，我进城， q ，去看你， r ，我很累。

则有命题公式： $\neg r \rightarrow (p \rightarrow q)$ 。

- 2. “应届高中生，得过数学或物理竞赛的一等奖，保送上北京大学。”

设： p ，应届高中生， q ，保送上北京大学，

r ，是得过数学一等奖。 t ，是得过物理一等奖。

则有命题公式公式： $p \rightarrow ((r \vee t) \rightarrow q)$ 。

命题逻辑的归结法

- 归结原理
- $P \vee Q$ 为真, $\neg P \vee Q$ 为真, $P \vee \neg Q$ 为假
- $P \vee \neg Q$ 化为子句集 $S=\{C_1, C_2, \dots, C_n\}$, (合取范式)
- 即: $S=C_1 \vee C_2 \vee \dots \vee C_n$, C_i =命题的析取
- 例 $(\neg P \vee Q \vee R) \vee (\neg Q \vee S)$
- 子句集: $S=\{\neg P \vee Q \vee R, \neg Q \vee S\}$
- 如果归结出空子句: $C_i \vee C_j = \text{Nil}$ (如: $P \vee \neg P = \text{Nil}$)
S为不可满足的
(永假式)

命题逻辑的归结法

- 求合取范式基本步骤：
 - ✗ 消去连接词{ \neg , \vee , \wedge }
 - ✗ 内移或消去否定号 \neg
 - ✗ 利用分配率

命题逻辑的归结法

通过归结使得 $C_i \quad C_j$ 简化，寻找是否有空子句存在

定义1：P是命题，则P 与 $\neg P$ 为互补文字

定义2：C1，C2是S中任意两个子句，C1中的文字L1与 C2中的

文字L2互补，则 $C1 \quad C2$ 可以归结为：

$$(C1 - \{L1\}) \quad (C2 - \{L2\}) \implies C_{12}$$

定理：C₁₂为C1，C2的逻辑结论，i.e. $C1 \quad C2 \Rightarrow C_{12}$

命题逻辑的归结法

- 归结过程
 - 将命题写成合取范式
 - 求出子句集
 - 对子句集使用归结推理规则
 - 归结式作为新子句参加归结
 - 归结式出空子句Nil，S是不可满足的（矛盾），原命题成立。

（证明完毕）
- 谓词的归结：增加了量词、函数、变量，其余和命题归结过程一样。

命题逻辑归结例题 (1)

- 例题，证明公式： $(P \rightarrow Q) \rightarrow (\neg Q \rightarrow \neg P)$

- 证明：

(1) 将待证明公式转化成待归结命题公式：

$$(P \rightarrow Q) \rightarrow (\neg Q \rightarrow \neg P)$$

(2) 分别将公式前项化为合取范式：

$$P \rightarrow Q = \neg P \vee Q$$

结论求 \neg 后的后项化为合取范式：

$$\neg(\neg Q \rightarrow \neg P) = \neg(Q \vee \neg P) = \neg Q \wedge P$$

两项合并后化为合取范式：

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

(3) 则子句集为： $\{\neg P \vee Q, \neg Q, P\}$

命题逻辑归结例题 (2)

子句集为： $\{\neg P \vee Q, \neg Q, P\}$

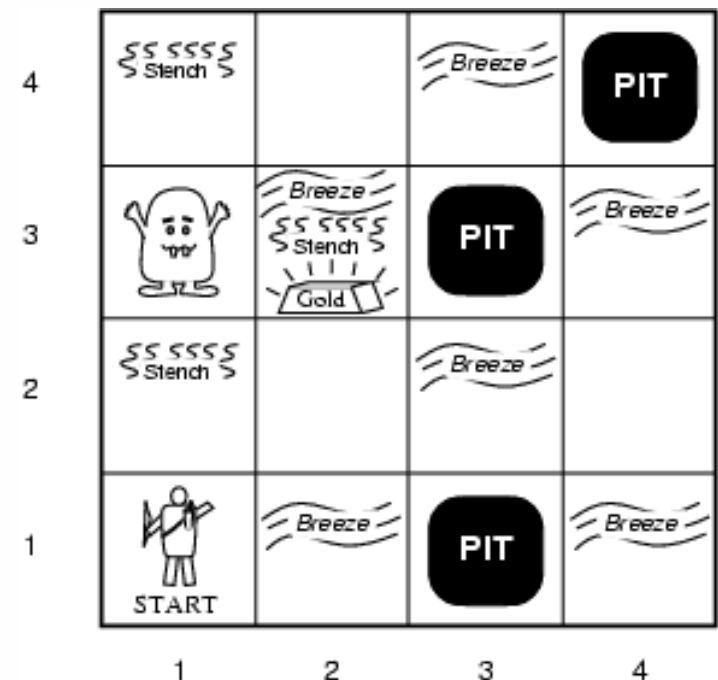
(4) 对子句集中的子句进行归结可得：

- 1. $\neg P \vee Q$
- 2. $\neg Q$
- 3. P
- 4. Q , (1, 3归结)
- 5. \square , (2, 4归结)

由上可得原公式成立。

Wumpus World PEAS description

- **Performance measure**
 - gold +1000, death -1000
 - -1 per step, -10 for using the arrow
- **Environment**
 - Squares adjacent to wumpus are smelly
 - Squares adjacent to pit are breezy
 - Glitter iff gold is in the same square
 - Shooting kills wumpus if you are facing it
 - Shooting uses up the only arrow
 - Grabbing picks up gold if in same square
 - Releasing drops the gold in same square



Pit:0.2

- **Sensors:** Stench, Breeze, Glitter, Bump, Scream
- **Actuators:** Left turn, Right turn, Forward, Grab, Release, Shoot

第三章 归结推理方法

- ✓ 概述
- ✓ 命题逻辑的归结法
 - 谓词归结子句形
 - 归结原理
 - 归结过程的策略控制
 - Herbrand定理

第三章 归结推理方法

- ✓ • 概述
- ✓ • 命题逻辑的归结法
- ✓ • 谓词归结子句形
 - 归结原理
 - 归结过程的策略控制
 - Herbrand定理

谓词归结原理基础

一阶逻辑

• 基本概念

- 个体词(项)：表作用对象，常量(a, b, c)，变元(x, y, z)，函数
函数：个体域 个体域 ($s(x)$, $father(x)$ 等)
- 谓词：表对象动作、性质、关系；大写字母打头表示
- 量词： \forall , \exists
- 联接词： \neg , \wedge , \vee , \leftrightarrow
- 一阶谓词：谓词中只含有个体词，不嵌套含有谓词

一阶谓词表示法 (1)

例1：刘欢比他父亲出名

Well-known(x,y)：x比y出名； father(x)：x的父亲

Well-known(Liuhuan, father(Liuhuan))

例2：高扬是计算机系的一名学生，但他不喜欢编程

Computer(x)：x是计算机系的学生； Like(x,y)：x喜欢y

Computer(Gao) \neg Like(Gao,programming)

Student(x,y)：x是y系的学生

Student(Gao,Computer) \neg Like(Gao,programming)

例3：偶数除以2是整数

E(x)：x是偶数； I(x)：x是整数； s(x)：x除以2

$(\forall x)(E(x) \rightarrow I(s(x)))$

例4：每个人都有父亲

Person(x)：x是一个人； Hasfather(x,y)：x的父亲是 y

$(\forall x \exists y)(\text{Person}(x) \rightarrow \text{Hasfather}(x,y))$

Isa(x,y)：x属于y类； $(\forall x \exists y)(\text{Isa}(x,\text{Human}) \rightarrow \text{Hasfather}(x,y))$

一阶谓词表示法 (3)

- 例如：(1) 所有的人都是要死的。
- (2) 有的人活到一百岁以上。

在个体域 D 为人类集合时：

(1) $\forall x P(x)$, $P(x)$: x 是要死的。

(2) $\exists x Q(x)$, $Q(x)$: x 活到一百岁以上。

在个体域 D 是全总个体域时：

(1) $\forall x (R(x) \rightarrow P(x))$, $R(x)$: x 是人

(2) $\exists x (R(x) \wedge Q(x))$,

一阶谓词表示法（4）

- 有的人喜欢梅花，有的人喜欢菊花，有的人既喜欢梅花又喜欢菊花

$$\exists x(Person(x) \rightarrow Like(x, 梅花))$$

$$\exists x(Person(x) \wedge Like(x, 梅花))$$

$$\exists y(Person(y) \wedge Like(y, 菊花))$$

$$\exists z(Person(z) \wedge Like(z, 梅花) \wedge Like(z, 菊花))$$

一阶谓词表示法 (5)

- 他每天下午都去打篮球

$Plays(x, y, z): x$ 在 z 时段plays y
 $Plays(\text{他}, \text{篮球}, \text{每天下午})$

$Plays(x, y, z): x$ 在 z 时段plays $y; s(x): x$ 的下午
 $Plays(\text{他}, \text{篮球}, s(\text{每天}))$

一阶谓词表示法（6）

- 西安市的夏天既炎热又干燥

$Summer(x): x$ 地的夏天 ; $Dry(x): x$ 干燥 ; $Hot(x): x$ 炎热
 $Dry(Summer(\text{西安})) \wedge Hot(Summer(\text{西安}))$

$Have(x, y): x$ 具有气候特征 y ; $Summer(x): x$ 地的夏天 ;
 $Have(Summer(\text{西安}), dry) \wedge Have(Summer(\text{西安}), hot)$

一阶谓词表示法 (7)

- 并不是人人都喜欢吃臭豆腐

$Person(x): x \text{是人} ; Like(x, y): x \text{喜欢吃} y$
 $\neg \forall x (Person(x) \rightarrow Like(x, \text{臭豆腐}))$

- 人人都喜欢吃冰淇淋；有的人喜欢可乐

$\forall x (Person(x) \rightarrow Like(x, \text{冰淇淋}))$
 $\exists x (Person(x) \wedge Like(x, \text{可乐}))$



一阶谓词表示法（8）

- 喜欢读《三国演义》的人必读《水浒》

$Person(x): x$ 是人 $Read(x, y): x$ 读 y ;

$\forall x(Person(x) \wedge Read(x, 三国演义) \rightarrow Read(x, 水浒))$

$Person(x): x$ 是人 ; $Enjoyreading(x, y): x$ 喜欢读 y $Read(x, y): x$ 读 y ;

$\left\{ \begin{array}{l} \forall x(Person(x) \wedge Enjoyreading(x, 三国演义) \rightarrow Read(x, 水浒)) \\ \forall x \forall y(Enjoyread(x, y) \rightarrow Read(x, y)) \end{array} \right.$

一阶谓词表示法（9）

➤ 关于一阶谓词逻辑的若干问题：

分清蕴涵 与合取 的区别：

- 1) 蕴涵的前提可以不发生，且前提不为真时蕴涵式仍能成立
- 2) 合取表示的是已经发生或存在的事实
- 3) 蕴含通常与全称量词相关联，是它的自然联接符；蕴涵符若与存在量词合用易导致过弱表示，未能充分表示有用的信息。
- 4) 合取符通常与存在量词相关联，是它的自然联接符；合取符若与全称量词合用易导致过强表示。
- 5) 逆否命题

一阶谓词表示法（10）

- 表达过程中能关联的谓词尽可能关联，便于后续的推理
- 推理时应根据需要补充默认的事实
- 适当设置谓词项的数目，不可太笼统，不便推理
- 在一阶谓词逻辑中，谓词本身不可作为另一谓词的项
- 区分函数与谓词的不同，谓词的值是“真”或“假”，函数是一个个体域到另一个个体域一种映射，其值是某个个体域的个体
- 充分利用函数进行表达，可简化。
- \forall 和 \exists 可相互转化， $\neg \forall$ 和 $\neg \exists$ 也可相互转化；尽管如此语句的可读性比精简更重要
- 语义是什么？语气，情态的谓词表达？

一阶谓词表示法 (11)

$F(x)$: x 是火车 ;

$G(x)$: x 是汽车;

$H(x,y)$: x 比 y 快 ,

命题“某些汽车比所有火车慢”的符号化公式是:

$$A . (\exists y) (G(y) \quad (\forall x) (F(x) \quad H(x,y)))$$

$$B . (\exists y) (G(y) \quad (\forall x) (F(x) \quad H(x,y)))$$

$$C . (\forall x) (\exists y) (G(y) \quad (F(x) \quad H(x,y)))$$

$$D . (\exists y) (G(y) \quad (\forall x) (F(x) \quad H(x,y)))$$

一阶谓词表示法（12）

用一阶谓词逻辑表示下列语句：

1. 有的实数不是有理数，但所有的有理数都是实数。
2. 任意两个实数 x 和 y 之间必可找到另一个实数 z 。
3. 除0之外，每个自然数有且仅有一个相继前元（论域已设定为自然数集）。
4. 条条大路通罗马
5. Rex is a dog from Fairfax.
6. Bill has been beaten by every dog from Fairfax.
7. Jill has not been beaten by every dog in Fairfax.
8. There is water in a refrigerator.

一阶谓词表示法 (13)

[6 points] Translate the following sentences in first-order logic:

- i. All citizens of Fredonia speak the same language.
- ii. The Fredonese language has two dialects
- iii. Each citizen of Fredonia speaks exactly one of the two dialects

一阶谓词表示法（14）

表示特点：

优：自然性，精确性，严密性，充分性，易实现，是后续三种表示法的基础，prolog

劣：不能表示不精确知识，组合爆炸，效率低，启发知识难利用

一阶谓词逻辑表达中的注意点：

- 1) 谓词与函数的区别
- 2) 与全称量词和存在量词关联的自然连接符
- 3) 表达的可读性与精简问题；
- 4) 适用的问题类型

谓词归结原理基础

量词否定等值式：

$$\neg(\forall x) M(x) \Leftrightarrow (\exists y) \neg M(y)$$

$$\neg(\exists x) M(x) \Leftrightarrow (\forall y) \neg M(y)$$

量词分配等值式：

$$(\forall x) (P(x) \wedge Q(x)) \Leftrightarrow (\forall x) P(x) \wedge (\forall x) Q(x)$$

$$(\exists x) (P(x) \wedge Q(x)) \Leftrightarrow (\exists x) P(x) \wedge (\exists x) Q(x)$$

消去量词等值式：设个体域为有穷集合 (a_1, a_2, \dots, a_n)

$$(\forall x) P(x) \Leftrightarrow P(a_1) \wedge P(a_2) \wedge \dots \wedge P(a_n)$$

$$(\exists x) P(x) \Leftrightarrow P(a_1) \vee P(a_2) \vee \dots \vee P(a_n)$$

谓词归结原理基础

量词辖域收缩与扩张等值式：

- $(\forall x) (P(x) \rightarrow Q) \Leftrightarrow (\forall x) P(x) \rightarrow Q$
- $(\forall x) (P(x) \wedge Q) \Leftrightarrow (\forall x) P(x) \wedge Q$
- $(\forall x) (P(x) \rightarrow Q) \Leftrightarrow (\exists x) P(x) \rightarrow Q$
- $(\forall x) (Q \rightarrow P(x)) \Leftrightarrow Q \rightarrow (\forall x) P(x)$
- $(\exists x) (P(x) \rightarrow Q) \Leftrightarrow (\exists x) P(x) \rightarrow Q$
- $(\exists x) (P(x) \wedge Q) \Leftrightarrow (\exists x) P(x) \wedge Q$
- $(\exists x) (P(x) \rightarrow Q) \Leftrightarrow (\forall x) P(x) \rightarrow Q$
- $(\exists x) (Q \rightarrow P(x)) \Leftrightarrow Q \rightarrow (\exists x) P(x)$

谓词归结子句形(Skolem 标准形)

SKOLEM标准形

– 前束范式

定义：说公式A是一个前束范式，如果A中的一切量词都位于该公式的最左边（不含否定词），且这些量词的辖域都延伸到公式的末端。

谓词归结子句形(Skolem 标准形)

- Skolem定理：

谓词逻辑的任意公式都可以化为与之等价的前束范式，但其前束范式不唯一。

- SKOLEM标准形定义：

带有任意量词的前束合取范式。

注意：谓词公式G的SKOLEM标准形同G并不等值。

谓词归结子句形(Skolem 标准形)

子句集的化约步骤

例1： $(\exists z) (\forall x)(\exists y)\{[(P(x) \vee Q(x)) \rightarrow R(y)] \vee U(z)\}$

1) 消蕴涵符 (\rightarrow 及 \Rightarrow)

$$(\exists z) (\forall x)(\exists y)\{[\neg(P(x) \vee Q(x)) \vee R(y)] \vee U(z)\}$$

2) 移动一至紧邻的谓词前

$$(\exists z) (\forall x)(\exists y)\{[(\neg P(x) \wedge \neg Q(x)) \vee R(y)] \vee U(z)\}$$

3) 变量标准化

即：不同的量词约束，对应于不同的变量

$$\text{如：} (\exists x)A(x) \vee (\exists x)B(x) \Rightarrow (\exists x)A(x) \vee (\exists y)B(y)$$

4) 量词左移

$$\text{如：} (\exists x)A(x) \vee (\exists y)B(y) \Rightarrow (\exists x) (\exists y) \{A(x) \vee B(y)\}$$

5) 消存在量词 \exists (skolem化)

受 \exists 约束但不受 \forall 约束的变量：用一个常量代替；

受 \exists 约束同时受 \forall 约束的变量：用一个函数代替；

$$(\exists z) (\forall x)(\exists y)\{[(\neg P(x) \wedge \neg Q(x)) \vee R(y)] \vee U(z)\}$$

$$\Rightarrow (\forall x) \{[(\neg P(x) \wedge \neg Q(x)) \vee R(f(x))] \vee U(a)\}$$

6) 化为合取范式

$$(\forall x)\{[(\neg P(x) \wedge \neg Q(x)) \vee R(f(x))] \vee U(a)\}$$

$$\Rightarrow (\forall x)\{(\neg P(x) \wedge \neg Q(x)) \vee R(f(x)) \vee U(a)\}$$

$$\Rightarrow (\forall x)\{[\neg P(x) \vee R(f(x)) \vee U(a)] \wedge [\neg Q(x) \vee R(f(x)) \vee U(a)]\}$$

谓词归结子句形(Skolem 标准形)

7) 隐去全称量词 \forall

$$\{[\neg P(x) \vee R(f(x)) \vee U(a)] \wedge [\neg Q(x) \vee R(f(x)) \vee U(a)]\}$$

8) 表示为子句集

$$\{\neg P(x) \vee R(f(x)) \vee U(a), \neg Q(x) \vee R(f(x)) \vee U(a)\}$$

9) 变量标准化 (变量换名)

$$\{\neg P(x) \vee R(f(x)) \vee U(a), \neg Q(y) \vee R(f(y)) \vee U(a)\}$$

谓词归结子句形

- 若 G 是给定的公式，而 S 是相应的子句集，
则 G 是不可满足的 $\Leftrightarrow S$ 是不可满足的。
- G 与 S 不等价，但在不可满足得意义下是一致的。

★ 注意： G 真不一定 S 真，而 S 真必有 G 真。

即： $S \Rightarrow G$

求取子句集例(1)

设已知

(1) 如果 x 是 y 的父亲， y 是 z 的父亲，则 x 是 z 的祖父

(2) 每个人都有一个父亲

试用归结演绎推理证明：对于某人 u ，一定存在一个人 v ， v 是 u 的祖父。

求取子句集例(2)

$$(1) \forall x \forall y \forall z (F(x, y) \wedge F(y, z) \rightarrow G(x, z))$$

$$\neg F(x, y) \vee \neg F(y, z) \vee G(x, z)$$

$$(2) \forall w \exists t F(t, w) \xrightarrow{\text{化为}} F(f(w), w)$$

$$(3) \neg \forall u \exists v G(v, u) \Leftrightarrow \exists u \forall v \neg G(v, u) \xrightarrow{\text{化为}} \neg G(v, a)$$

第三章 归结推理方法

- ✓ 概述
- ✓ 命题逻辑的归结法
- ✓ 谓词归结子句形
- 归结原理
- 归结过程的策略控制
- Herbrand定理

第三章 归结推理方法

- ✓ • 概述
- ✓ • 命题逻辑的归结法
- ✓ • 谓词归结子句形
- ✓ • 归结原理
 - 归结过程的策略控制
 - Herbrand定理

- 归结原理是一种逻辑推理方法
- 从理论和实际上解决了定理证明问题。
- 前提：记子句集 $S = \{C_1, C_2, \dots, C_n\}$,

S 的不可满足 \Leftrightarrow 至少有一个 $C_i = \text{“F”}$

- 思路：若存在 $C_i \quad C_j = \text{Nil}$ (如： $P \quad \neg P = \text{Nil}$) $\Rightarrow S$ 不可满足

归结原理

- 归结原理正确性的根本在于，找到矛盾可以肯定不真。
- 方法：
 - 和命题逻辑一样。
 - 但由于有函数，所以要考虑模式匹配：合一和置换。

模式匹配

对两个知识模式（两个谓词公式，框架片断，语义网片断）进行比较，是推理的前期工作

例：在知识库中查询Knows(John,x)：John认识谁？

解：知识库有

Knows(John,Jane) {Jane/x} Knows(John,Jane)

Knows(y,Bill) {John/y,Bill/x} Knows(John,Bill)

Knows(y,Mother(y)) {John/y, Mother(y)/x}

 {John/y,Mother(John)/x} Knows(John,Mother(John))

Knows(x,Elizabeth)=fail? x不能同时选择John和Elizabeth

Knows(y,Elizabeth) {John/y,Elizabeth/x} Knows(John,Elizabeth)

置换

- 置换：在一个谓词公式中用置换项去置换变量。

- 定义：

置换是形如 $\{t_1/x_1, t_2/x_2, \dots, t_n/x_n\}$ 的有限集合。

1) x_1, x_2, \dots, x_n ：互不相同的变量，

2) t_1, t_2, \dots, t_n ：不同于 x_i 的项（常量、变量、函数）；

t_i/x_i ：用 t_i 置换 x_i ，且 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。

$\{g(y)/x, f(x)/y\}$ 不是一个置换，

置换的合成

- 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 θ 与 λ 的合成也是一个置换 , 记作 $\theta \cdot \lambda$ 。它是从集合

$\{t_1 \cdot \lambda/x_1, t_2 \cdot \lambda/x_2, \dots, t_n \cdot \lambda/x_n, u_1/y_1, u_2/y_2, \dots, u_m/y_m\}$

中删去以下两种元素 :

- 当 $t_i \cdot \lambda = x_i$ 时 , 删去 $t_i \cdot \lambda/x_i$ ($i = 1, 2, \dots, n$);
- 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。

合成即是对 t_i 先做 λ 置换然后再做 θ 置换 , 置换 x_i

置换的合成

设： $\theta = \{f(y)/x, z/y\}$ ， $\lambda = \{a/x, b/y, y/z\}$ ，求 θ 与 λ 的合成。

解：先求出集合

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

其中， $f(b)/x$ 中的 $f(b)$ 是置换 λ 作用于 $f(y)$ 的结果； y/y 中的 y 是置换 λ 作用于 z 的结果。

在该集合中， y/y 满足定义中的条件i，需要删除； a/x ， b/y 满足定义中的条件ii，也需要删除。

最后得

$$\theta\lambda = \{f(b)/x, y/z\}$$

置换的合成

代
换
内
部
需
要
复
合
的
应
先
复
合

$$\theta = \{f(z)/x, z/y\}, \quad \lambda = \{a/x, b/y, y/z\}$$

$$\lambda = \{a/x, b/y, b/z\}$$

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

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

$$\theta \cdot \lambda = \{f(b)/x, b/y, b/z\}$$

合一

- 合一可以简单地理解为“寻找相对变量的置换，使两个谓词公式一致”。
- 定义：设有公式集 $F = \{F_1, F_2, \dots, F_n\}$ ，若存在一个置换 θ ，可使 $F_1\theta = F_2\theta = \dots = F_n\theta$ ，则称 θ 是 F 的一个合一。同时称 F_1, F_2, \dots, F_n 是可合一的。
- 例：
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。

合一

注意：一般说来，一个公式集的合一不是唯一的。

$$\{P(x, f(y), B), P(z, f(B), B)\}$$

$$P1 = \{A/x, B/y, A/z\} ,$$

$$P(A, f(B), B)$$

$$P2 = \{z/x, B/y\}$$

$$P(z, f(B), B)$$

合一

定义4：设 θ 是公式集 F 的一个合一，若对任意的合一 τ 都存在代换 λ 使 $\tau = \theta \circ \lambda$ ，则称 θ 是最一般的合一。

注：任一合一都可由最一般合一与另一代换复合而得。

不考虑变量的重新命名最一般合一是唯一的。

合一

最一般合一的求解步骤：

- 1) 令 $k \leftarrow 0$, $F_k \leftarrow F$, $\theta_k = \phi$ (空代换)
- 2) F_k 只含一个表达式 , stop 得 θ_k
- 3) 找出差异集 D_k
- 4) 若 $t_k, x_k \in D_k$ 且 x_k 在 t_k 中不出现 , 则 :
$$\theta_{k+1} \leftarrow \theta_k \circ \{t_k / x_k\}$$
$$F_{k+1} \leftarrow F_k \{t_k / x_k\} \quad \text{转2)}$$
$$k \leftarrow k+1$$
- 5) stop , 不存在

合一

例: $F = \{P(a, x, f(g(y))), P(z, f(z), f(u))\}$

$F0 = F, D0 = \{a, z\}, Pk = E$

$P1 = \{a/z\}, F1 = \{P(a, x, f(g(y))), P(a, f(a), f(u))\}$

$D1 = \{x, f(a)\}, P2 = P1 \circ \{f(a)/x\} = \{a/z, f(a)/x\}$

$F2 = \{P(a, f(a), f(g(y))), P(a, f(a), f(u))\}$

$D2 = \{g(y), u\}, P3 = P2 \circ \{g(y)/u\} = \{a/z, f(a)/x, g(y)/u\}$

$F3 = \{P(a, f(a), f(g(y))), P(a, f(a), f(g(y)))\}$

stop, $P = \{a/z, f(a)/x, g(y)/u\}$

归结原理

归结的注意事项：

- 谓词的一致性， $P()$ 与 $Q()$ ，不可
- 常量的一致性， $P(a, \dots)$ 与 $P(b, \dots)$ ，不可
- 变量， $P(a, \dots)$ 与 $P(x, \dots)$ ，可以
- 变量与函数， $P(a, x, \dots)$ 与 $P(x, f(x), \dots)$ 考虑
- 不能同时消去两个互补对， $P \quad Q$ 与 $\sim P \quad \sim Q$ ，不可
- 先进行内部简化（置换、合并）？？

例： $C1 = P(a) \quad \neg Q(x) \quad R(x)$, $C2 = \neg P(y) \quad Q(b)$

$\{P(a), \neg P(y)\}, P = \{a/y\}$ $\{\neg Q(x), Q(b)\}, P = \{b/x\}$

$C12 = \neg Q(x) \quad R(x) \quad Q(b)$ $C12 = P(a) \quad R(b) \quad \neg P(y)$

归结方式不唯一

例： $C1 = P(x) \quad Q(a)$, $C2 = \neg P(b) \quad R(x)$

换变元： $C2 = \neg P(b) \quad R(y)$ 变元相同要变换

$\{P(x), \neg P(b)\}, P = \{b/x\}$; $C12 = Q(a) \quad R(y)$

例： $C1 = P(x) \quad P(f(a)) \quad Q(x)$, $C2 = \neg P(y) \quad R(b)$

$C1: \{P(x), P(f(a))\}, P = \{f(a)/x\}$, $C1 P = P(f(a)) \quad Q(f(a))$

$\{P(f(a)), \neg P(y)\}, P' = \{f(a)/y\}$, $C12 = Q(f(a)) \quad R(b)$

?子句内部可合一的，先合一再与其它归结?

例： $C_1 = P(x) \quad P(a) \quad Q(x)$, $C_2 = \neg P(y) / C_2 = \neg P(f(b))$

直接归结

$C_{12} = P(a) \quad Q(x)$

$C_{122} = Q(x)$

内部简化

$C_1' = P(a) \quad Q(a)$ (先合一)

$C_{12}' = Q(a)$ (少了普遍性)

$C_{12} = P(a) \quad Q(f(b))$

$C_1' = P(a) \quad Q(a)$ (先合一)

无法归结

归结原理

- 归结的过程

写出谓词关系公式

用反演法写出谓词表达式

SKOLEM标准形

子句集S

对S中可归结的子句做归结

归结式仍放入S中，反复归结过程

得到空子句

□得证

归结原理

- 将归结原理用于自动定理证明
- 问题：证明 $F = \{P_1, P_2, \dots, P_n\} \vdash Q$,
- 只要证 $P_1 \quad P_2 \quad \dots, \quad P_n \quad \neg Q$ 是不可满足的
- 步骤：
 - 1) 写出 $\neg Q$
 - 2) 将 $\{F, \neg Q\}$ 化成子句集 S
 - 3) 对 S 进行归结，归结式并入 S
 - 4) 出现 Nil 停止，得证

例题“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的考试，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：先将问题用谓词表示如下：
- R1:“任何通过计算机考试并获奖的人都是快乐的”
 $(\forall x)((\text{Pass}(x, \text{computer}) \wedge \text{Win}(x, \text{prize})) \rightarrow \text{Happy}(x))$
- R2:“任何肯学习或幸运的人都可以通过所有考试”
 $(\forall x)(\forall y)(\text{Study}(x) \vee \text{Lucky}(x) \rightarrow \text{Pass}(x, y))$
- R3:“张不肯学习但他是幸运的”
 $\neg \text{Study}(\text{zhang}) \wedge \text{Lucky}(\text{zhang})$
- R4:“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
 $\neg \text{Happy}(\text{zhang})$

$(\forall x)((\text{Pass}(x, \text{computer}) \rightarrow \text{Win}(x, \text{prize})) \rightarrow \text{Happy}(x))$
 $(\forall x)(\forall y)(\text{Study}(x) \rightarrow \text{Lucky}(x) \rightarrow \text{Pass}(x, y))$
 $\neg \text{Study}(\text{zhang}) \rightarrow \text{Lucky}(\text{zhang})$
 $(\forall x)(\text{Luck}(x) \rightarrow \text{Win}(x, \text{prize}))$
 $\neg \text{Happy}(\text{zhang})$

(1) $\neg \text{Pass}(x, \text{computer}) \rightarrow \text{Win}(x, \text{prize}) \rightarrow \text{Happy}(x)$
 (2) $\neg \text{Study}(y) \rightarrow \text{Pass}(y, z)$
 (3) $\neg \text{Lucky}(u) \rightarrow \text{Pass}(u, v)$
 (4) $\neg \text{Study}(\text{zhang})$
 (5) $\text{Lucky}(\text{zhang})$
 (6) $\neg \text{Lucky}(w) \rightarrow \text{Win}(w, \text{prize})$
 (7) $\neg \text{Happy}(\text{zhang})$

(8) $\neg \text{Pass}(w, \text{computer}) \rightarrow \text{Happy}(w) \rightarrow \text{Luck}(w)$ (1)(6), {w/x}
 (9) $\neg \text{Pass}(\text{zhang}, \text{computer}) \rightarrow \text{Lucky}(\text{zhang})$ (8)(7), {zhang/w}
 (10) $\neg \text{Pass}(\text{zhang}, \text{computer})$ (9)(5)
 (11) $\neg \text{Lucky}(\text{zhang})$ (10)(3), {zhang/u, computer/v}
 (12) NIL (11)(5)

例1 : F: $(\forall x) ((\exists y)(A(x,y) \wedge B(y)) \rightarrow (\exists y)(C(y) \wedge D(x,y)))$

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

1) F化成子句集

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

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

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

$\neg A(u,v) \wedge \neg B(v) \wedge D(u,f(u))$

2) $\neg G$ 化成子句集

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

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

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

$$(\forall x) \neg C(x) \quad (\exists s) (\exists t) (A(s,t) \quad B(t))$$

$$(\exists s) (\exists t) (\forall x) (\neg C(x) \quad (A(s,t) \quad B(t)))$$

$$(\forall x) \neg C(x) \quad A(a,b) \quad B(b)$$

$$\neg C(x) \quad A(a,b) \quad B(b)$$

$$\neg A(x,y) \quad \neg B(y) \quad C(f(x)) \quad \left. \vphantom{\neg A(x,y)} \right\} F$$

$$\neg A(u,v) \quad \neg B(v) \quad D(u,f(u)) \quad \left. \vphantom{\neg A(u,v)} \right\}$$

$$\left. \begin{array}{l} \neg C(z) \\ A(a,b) \\ B(b) \end{array} \right\} \neg G$$

3)归结

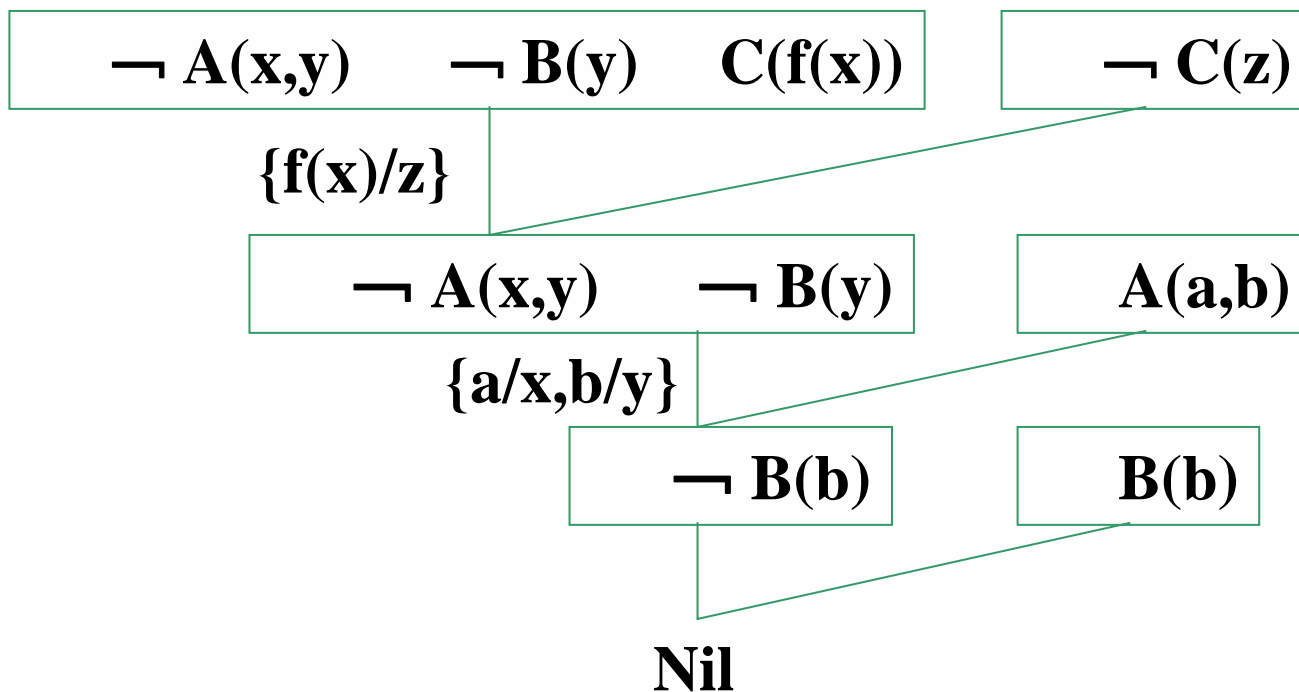
$$+ \quad : \quad \{f(x)/z\}$$

$$\neg A(x,y) \quad \neg B(y)$$

$$+ \quad : \quad \{a/x, b/y\}$$

$$\neg B(b)$$

$$+ \quad = Nil$$

$\neg A(x,y) \quad \neg B(y) \quad C(f(x))$ $\neg A(x,y) \quad \neg B(y) \quad D(x,f(x))$ $\neg C(z)$ $A(a,b) \quad \neg G$ $B(b)$ 

例2:自然数都是大于零的整数

所有整数不是偶数就是奇数

偶数除以2是整数

证: 所有自然数不是奇数就是
其一半为整数的数

$$(\forall x)(N(x) \rightarrow GZ(x) \quad I(x))$$

$$(\forall x)(I(x) \rightarrow E(x) \quad O(x))$$

$$(\forall x)(E(x) \rightarrow I(s(x)))$$

$$(\forall x)(N(x) \rightarrow (O(x) \quad I(s(x))))$$

$$\neg N(x) \quad GZ(x)$$

$$\neg N(u) \quad I(u)$$

$$\neg I(y) \quad E(y) \quad O(y)$$

$$\neg E(z) \quad I(s(z))$$

$$N(a)$$

$$\neg O(a)$$

$$\neg I(s(a))$$

$\neg N(x) \quad GZ(x)$ $\neg N(u) \quad I(u)$ $\neg I(y) \quad E(y) \quad O(y)$ $\neg E(z) \quad I(s(z))$ $N(a)$ $\neg O(a)$ $\neg I(s(a))$ $\neg I(y) \quad E(y) \quad O(y)$ $\{a/y\}$ $\neg O(a)$ $\neg I(a) \quad E(a)$ $\neg N(u) \quad I(u)$ $N(a)$ $\neg E(z) \quad I(s(z))$ $\{a/z\}$ $\neg I(s(a))$ $\neg E(a)$ $\neg I(a)$ $\{a/u\}$ $\neg N(a)$ Nil

例3：A，B，C三人中至少录取一人，若录A不录B，则一定录C，录B则一定录C

证：公司一定录C

$P(A) \quad P(B) \quad P(C)$

$P(A) \quad \neg P(B) \rightarrow P(C)$

$P(B) \rightarrow P(C)$

$\neg P(C)$

$P(A) \quad P(B) \quad P(C)$

$\neg P(A) \quad P(B) \quad P(C)$

$\neg P(B) \quad P(C)$

$\neg P(C)$

+ : $P(B) \quad P(C)$

+ : $P(C)$

+ : Nil

例：美国人(American)卖(Sells)武器(Weapon)给有敌意(Hostile)国家是犯法的(Crime)。美国的敌国(Enemy)Nono有一些导弹(Missile)，所有这些导弹都是韦斯特(West)上校卖给他们的，而韦斯特上校是一个美国人。证明：韦斯特上校是一个罪犯。

American(x):X是美国人

Weapon(y)：Y是武器

Sells(x,y,z)：X销售Y给Z

Hostile(z)：Z对美国有敌意

Criminal(x)：X是有罪的

Missile(x)：X是导弹

Owns(x,Y):X拥有Y

Enemy(x, Y):X是Y的敌国

归结推理 - 用归结原理求问题的答案 (1)

步骤：

- 1) 前提用谓词公式表示并化为子句集S
- 2) 构造： $\neg Q(\cdot)$ Answer (\cdot)，其中变元相同，Q为问题集
- 3) 对S $\{ \neg Q(\cdot)$ Answer (\cdot) }进行归结
- 4) 归结中取代Answer (\cdot) 中变元的项就是问题的答案

例1：已知： $(\forall x)[AT(John, x) \rightarrow AT(Fido, x)]$

$AT(John, School)$

求：Fido在何处？求证： $(\exists x)AT(Fido, x)$

子句集：

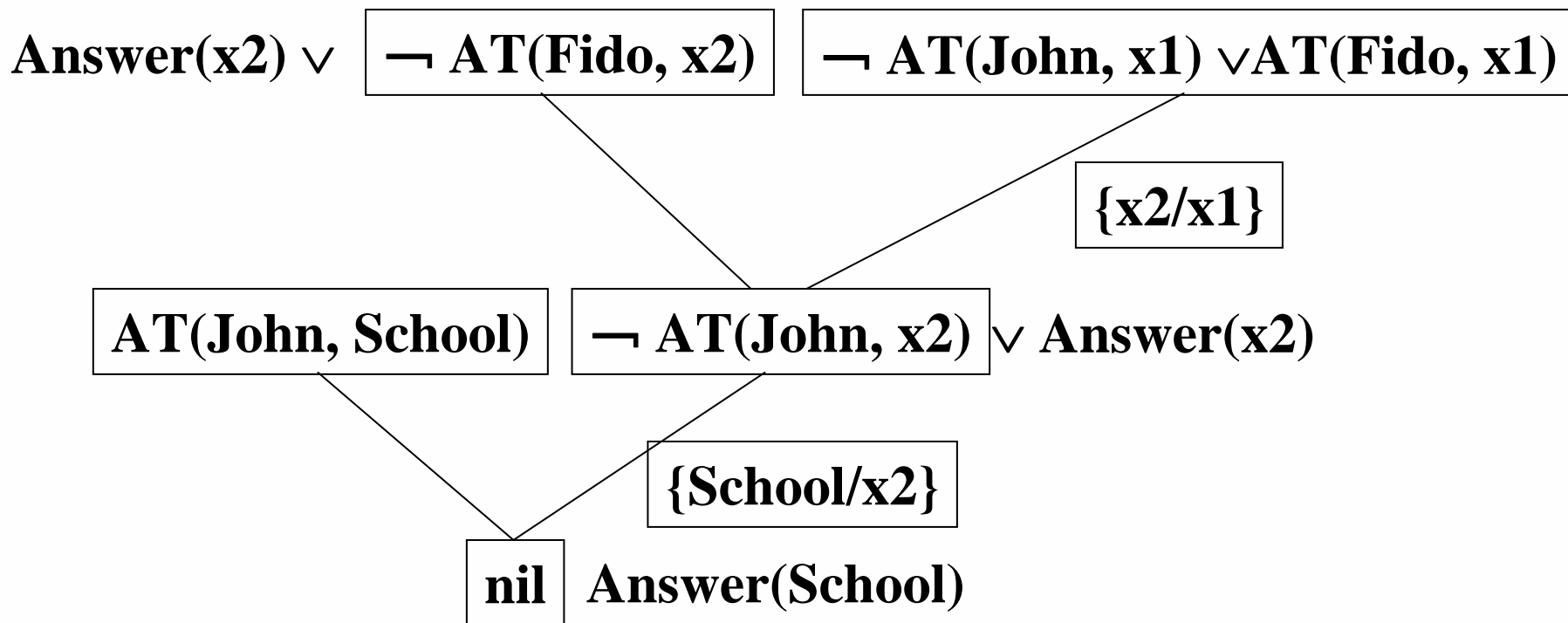
$\neg AT(John, x1) \vee AT(Fido, x1)$

$AT(John, School)$

$\neg AT(Fido, x2) \vee Answer(x2)$

归结演绎推理 - 用归结原理求问题的答案 (2)

子句集：
 $\neg AT(\text{John}, x1) \vee AT(\text{Fido}, x1)$
 $AT(\text{John}, \text{School})$
 $\neg AT(\text{Fido}, x2) \vee \text{Answer}(x2)$



归结演绎推理 - 用归结原理求问题的答案 (3)

例2 :

王是李的老师

$T(\text{Wang}, \text{Li})$

李与张是同学

$C(\text{Li}, \text{Zhang})$

1) $T(\text{Wang}, \text{Li})$ 子句集

2) $C(\text{Li}, \text{Zhang})$

3) $\neg C(x, y) \quad \neg T(z, x) \quad T(z, y)$

4) $\neg T(u, \text{Zhang}) \quad \text{Answer}(u)$

若x与y是同学，则x的老师也是y的老师

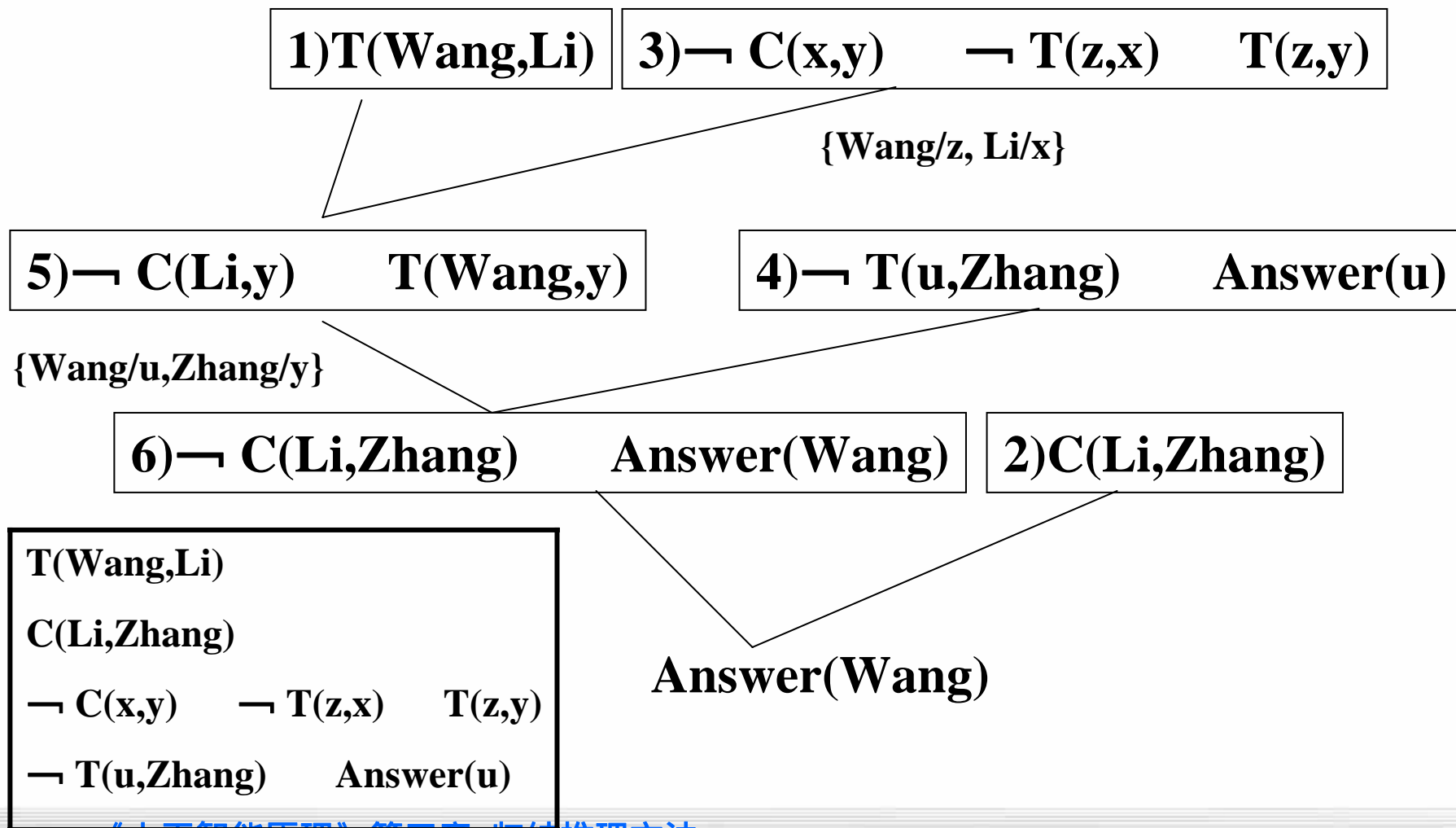
$(\forall x) (\forall y) ((C(x, y) \quad \neg T(z, x)) \rightarrow T(z, y))$

张的老师是谁？

$(\exists x) T(x, \text{Zhang}), \text{Answer}(x)$

归结演绎推理 - 用归结原理求问题的答案 (4)

例2的归结树



归结演绎推理 用归结原理求问题的答案(5)

例3：A，B，C三人有人从不说假话，有人从不说真话，某人问：谁是说谎者？A答：B和C都是说谎者；B答：A和C都是说谎者；C答：B和C中至少有一个是说谎者；问：谁是诚实者，谁是说谎者？

$T(A) \rightarrow \neg T(B) \quad \neg T(C) \qquad \neg T(A) \rightarrow T(B) \quad T(C)$

$T(B) \rightarrow \neg T(A) \quad \neg T(C) \qquad \neg T(B) \rightarrow T(A) \quad T(C)$

$T(C) \rightarrow \neg T(A) \quad \neg T(C) \qquad \neg T(C) \rightarrow T(A) \quad T(B)$

注：一般情况下， $A \rightarrow B$ 并不意味着 $\neg A \rightarrow \neg B$ ，此题由内容决定

1) $\neg T(A) \quad \neg T(B)$

5) $\neg T(A) \quad \neg T(B) \quad \neg T(C)$

2) $\neg T(A) \quad \neg T(C)$

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

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

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

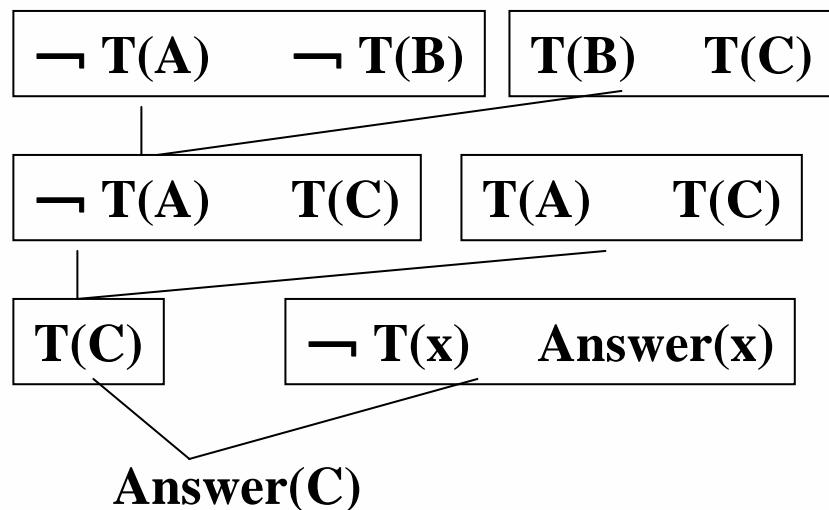
子句集

4) $\neg T(B) \quad \neg T(C)$

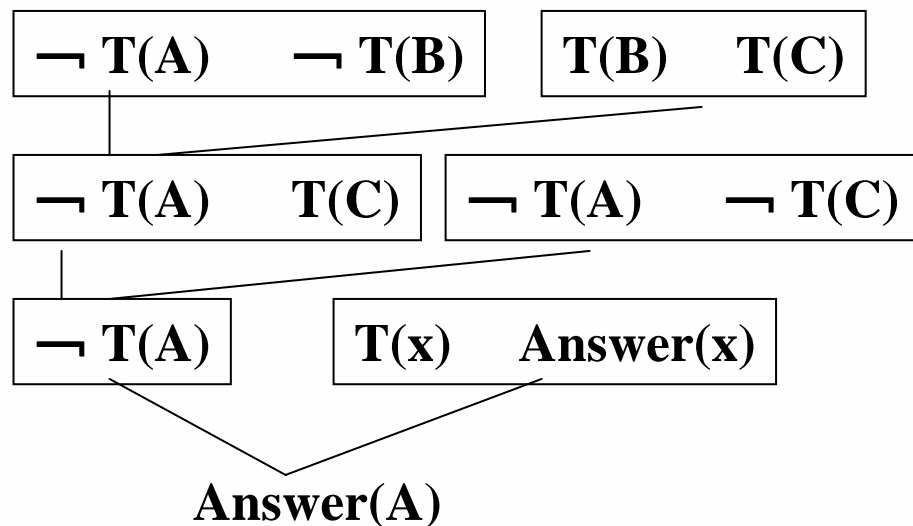
归结演绎推理

用归结原理求问题的答案(6)

查找诚实者： $(\exists x)T(x)$



查找说谎者： $(\exists x) \neg T(x)$



1) $\neg T(A) \quad \neg T(B)$

5) $\neg T(A) \quad \neg T(B) \quad \neg T(C)$

2) $\neg T(A) \quad \neg T(C)$

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

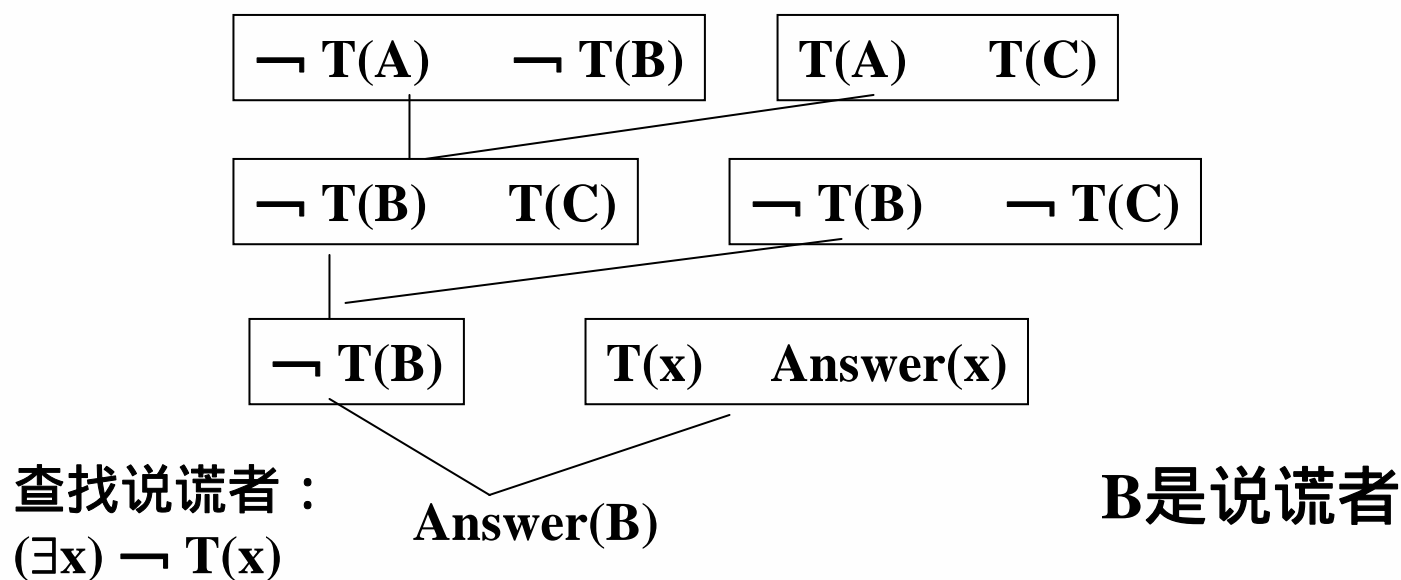
3) $T(A) \quad T(B) \quad T(C)$

7) $T(B) \quad T(C)$ 子句集

4) $\neg T(B) \quad \neg T(C)$

C是诚实者,A是说谎者

- | | |
|---------------------------------|--|
| 1) $\neg T(A) \quad \neg T(B)$ | 5) $\neg T(A) \quad \neg T(B) \quad \neg T(C)$ |
| 2) $\neg T(A) \quad \neg T(C)$ | 6) $T(A) \quad T(C)$ |
| 3) $T(A) \quad T(B) \quad T(C)$ | 7) $T(B) \quad T(C)$ |
| 4) $\neg T(B) \quad \neg T(C)$ | |
- 子句集



第三章 归结推理方法

- ✓ • 概述
- ✓ • 命题逻辑的归结法
- ✓ • 谓词归结子句形
- ✓ • 归结原理
 - 归结过程的策略控制
 - Herbrand定理

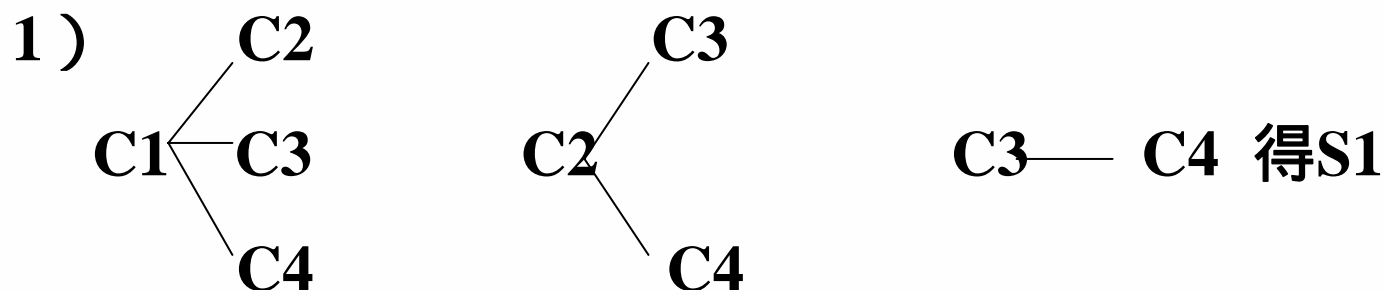
第三章 归结推理方法

- ✓ • 概述
- ✓ • 命题逻辑的归结法
- ✓ • 谓词归结子句形
- ✓ • 归结原理
- ✓ • 归结过程的策略控制
- Herbrand定理

归结过程的控制策略

归结的一般过程（广度优先）

例1：S={C1,C2,C3,C4}



2) S与S1进行归结得S2

3) S, S1与S2进行归结得S3

注：可能产生无用子句，重复子句，效率不高

例2：S={P, $\neg R$, $\neg P \vee Q$, $\neg Q \vee R$ }

1) S1={Q, $\neg Q$, $\neg P \vee R$ } ; 2) S2={R, $\neg P$ } ; 3) Nil

归结过程的控制策略

- 要解决的问题：
 - 归结方法的知识爆炸。
- 控制策略的目的
 - 归结点尽量少
- 控制策略的原则
 - 给出控制策略，仅对选择合适的子句间做归结。避免多余的的归结式出现。希望少做些归结仍能导出空子句。

控制策略的方法(1)

1) 删除策略 (完备)

- 纯文字删除：文字L在S中不存在 $\neg L$ ，删去含L的子句

$S = \{P \quad Q \quad R, \neg Q \quad R, Q, \neg R\}$ ；删去
 $P \quad Q \quad R$

- 重言删除：某一子句含L $\neg L$ ，删去
- 包蕴删除：若存在代换P，使得 $C1 \text{ P} \subseteq C2$ ，则删去C1

$P(x) \quad P = \{a/x\} \subseteq P(a) \quad Q(z)$, 删去 $P(x)$

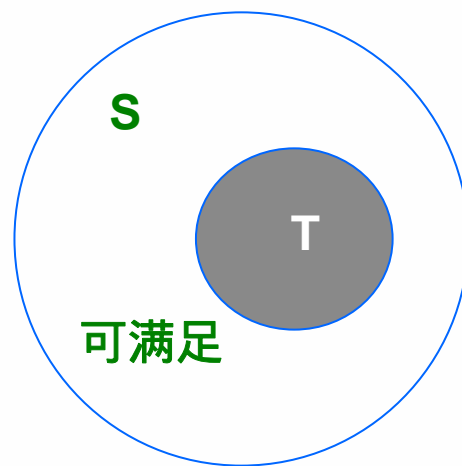
控制策略的方法(2)

2) 支撑集策略 (完备)

支撑集：设有不可满足的子句集S的子集T，如果S-T是可满足的，则称T是支撑集。

支撑集策略：归结过程的两个子句至少有一个来自支撑集T或其后裔。

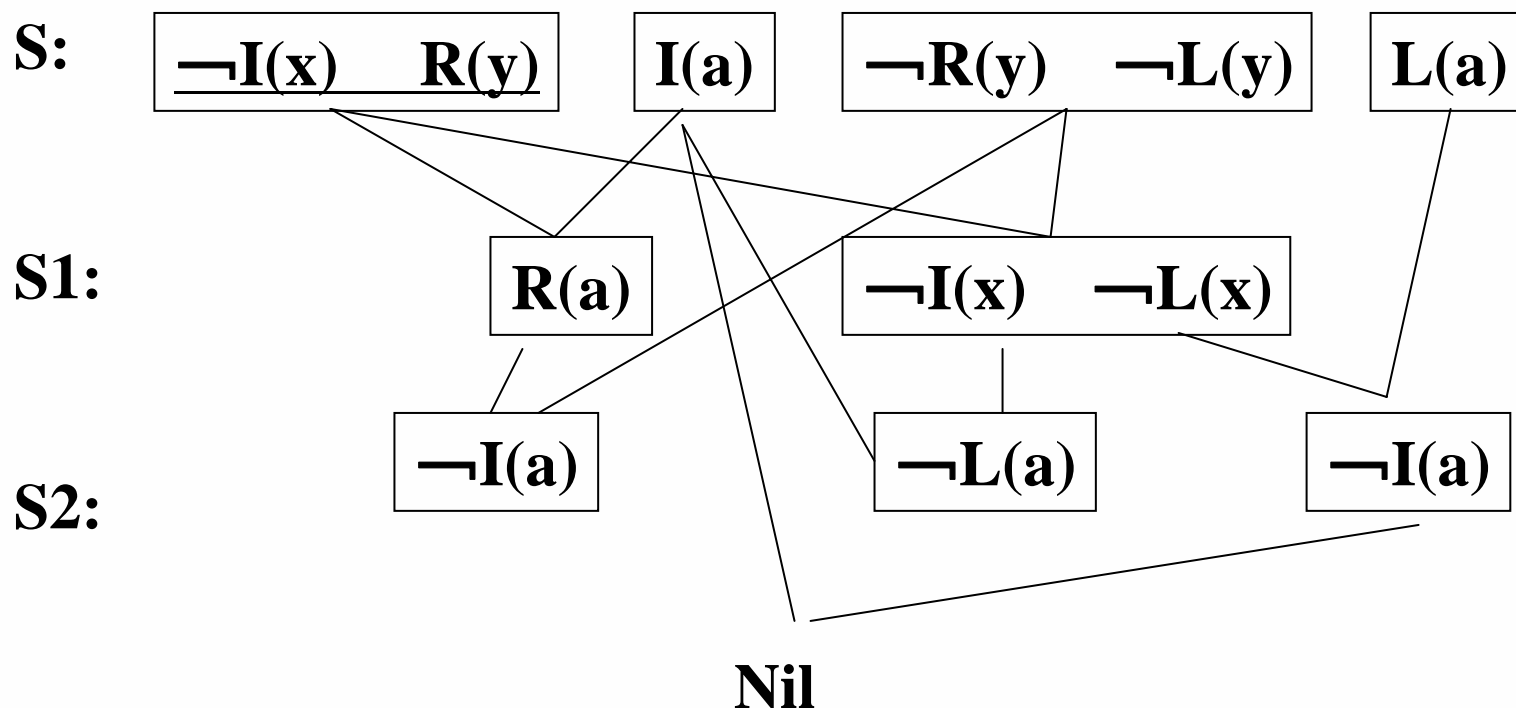
合适的支撑集：一个最容易找到的支撑集是目标子句的非，即 $\sim B$ 。



支撑集策略例

目标的否定

例3 : $S = \{\neg I(x) \quad R(y), I(a), \neg R(y) \quad \neg L(y), L(a)\}$



控制策略的方法(3)

3) 语义归结 (完备)

将子句集S按照一定的语义分成两部分，每部分内的子句间不允许作归结。同时还引入了文字次序，约定归结时其中的一个子句的被归结文字应是该子句集中“最大”的文字。

可加快归结速度的目的。问题是如何寻找合适的语义分类方法，并根据其含义将子句集两个部分中的子句进行排序。

语义归结例

- $S = \{\neg P \quad \neg Q \quad R, P \quad R, Q \quad R, \neg R\}$
- 语义划分规则：解释 $I = \{\neg P, \neg Q, \neg R\}$ 是否为真
- 文字顺序： $P > Q > R$
- $S1 = \{P \quad R, Q \quad R\}; S2 = \{\neg P \quad \neg Q \quad R, \neg R\}$

控制策略的方法(4)

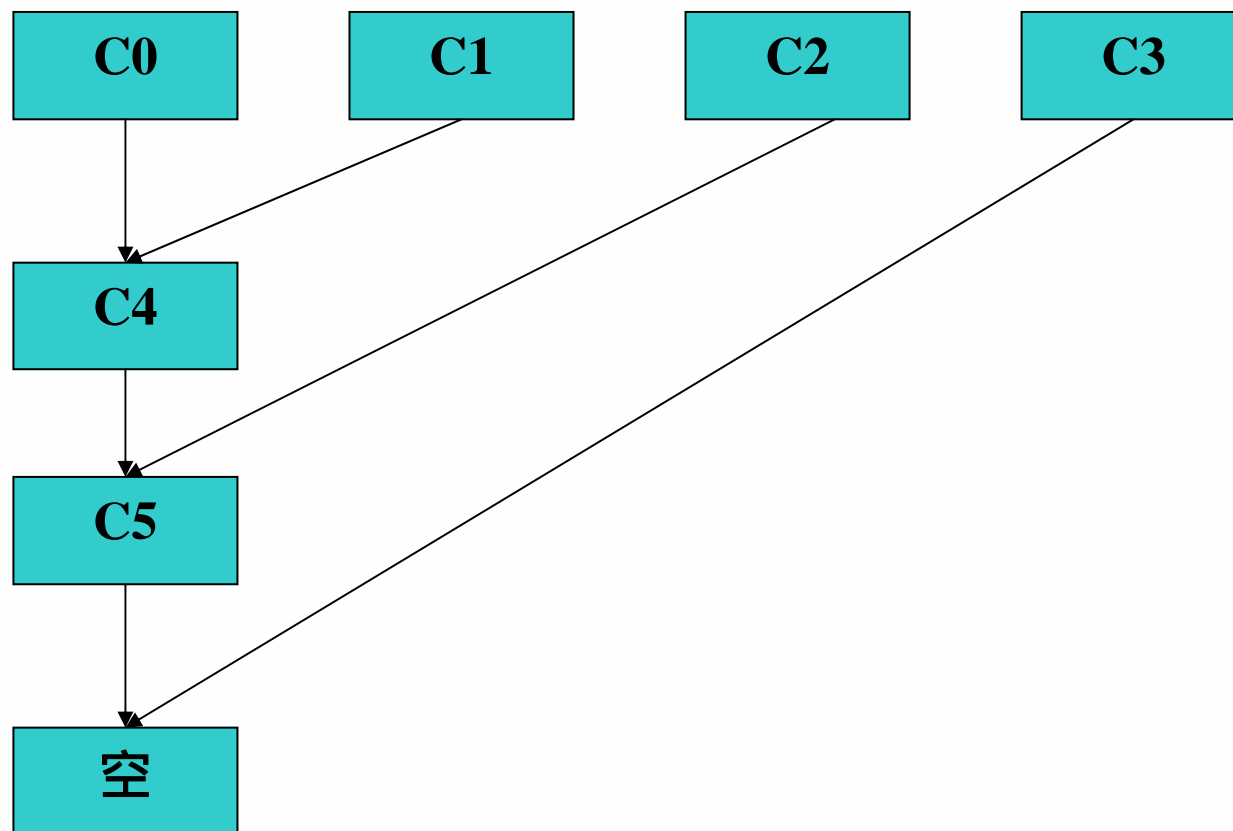
4) 线性归结 (完备)

首先从子句集中选取一个称作顶子句的子句 C_0 开始作归结。

归结过程中所得到的归结式 C_i 立即同另一子句 B_i 进行归结得归结式 C_{i+1} 。

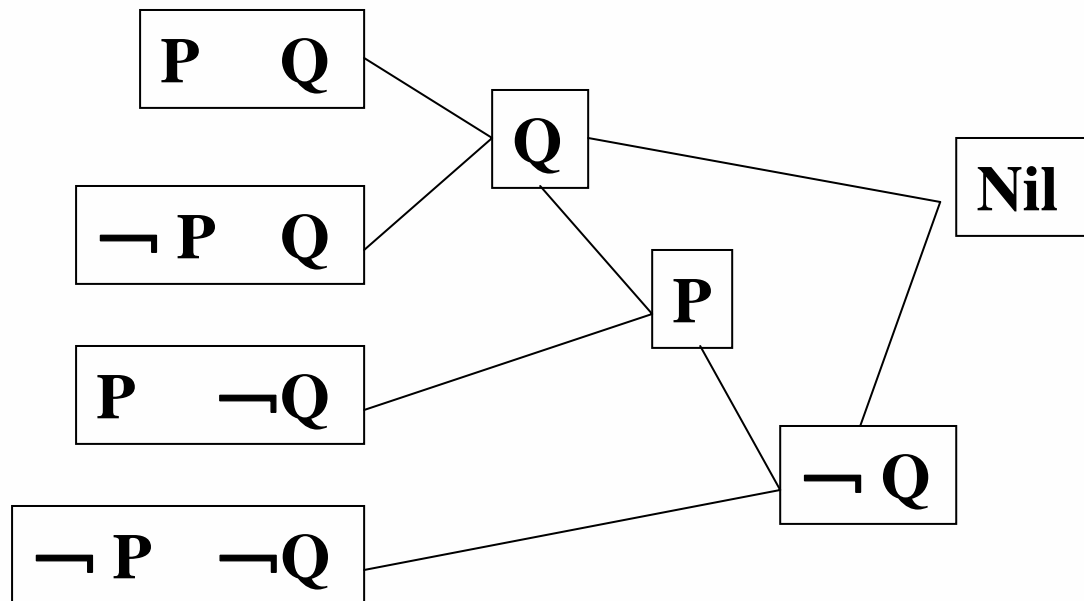
而 B_i 属于 S 或是已出现的归结式 $C_j (j < i)$ 。

。若能找到一个较好的顶子句，可加快归结速度。否则也可能事与愿违。



线性归结策略示意图

例： $S = \{P \quad Q, \neg P \quad Q, P \quad \neg Q, \neg P \quad \neg Q\}$



控制策略的方法(5)

5) 单元归结 (完备)

每次归结都有一个子句是单元子句 (只含一个文字的子句)。

初始子句集中没有单元子句时, 单元归结策略无效。即此问题不能采用单元归结策略。

控制策略的方法(6)

6) 输入归结 (完备)

归结过程的两个子句中必须有一个是S的原始子句。

如同单元归结策略，不是所有的可归结谓词公式的最后结论都是可以从原始子句集中的得到的。

如归结结束时，即最后一个归结式为空子句的条件是，归结的双方必须是两个单元子句。原子句集中无单元子句的谓词公式一定不能采用输入归结策略。

归结推理的不足

1) 不便于阅读理解

鸟能飞 $(\forall x)(\text{Bird}(x) \rightarrow \text{Canfly}(x))$ 等价于
 $\neg \text{Bird}(x) \vee \text{Canfly}(x)$

2) 容易丢失控制信息

$(\neg A \vee \neg B) \vee C, (\neg A \vee \neg C) \vee B,$
 $\neg A \vee (B \vee C)$

均可化成： $A \vee B \vee C$

第三章 归结推理方法

- ✓ • 概述
- ✓ • 命题逻辑的归结法
- ✓ • 谓词归结子句形
- ✓ • 归结原理
- ✓ • 归结过程的策略控制
- Herbrand定理

第三章 归结推理方法

- ✓ • 概述
- ✓ • 命题逻辑的归结法
- ✓ • 谓词归结子句形
- ✓ • 归结原理
- ✓ • 归结过程的策略控制
- ✓ • Herbrand定理

Herbrand定理

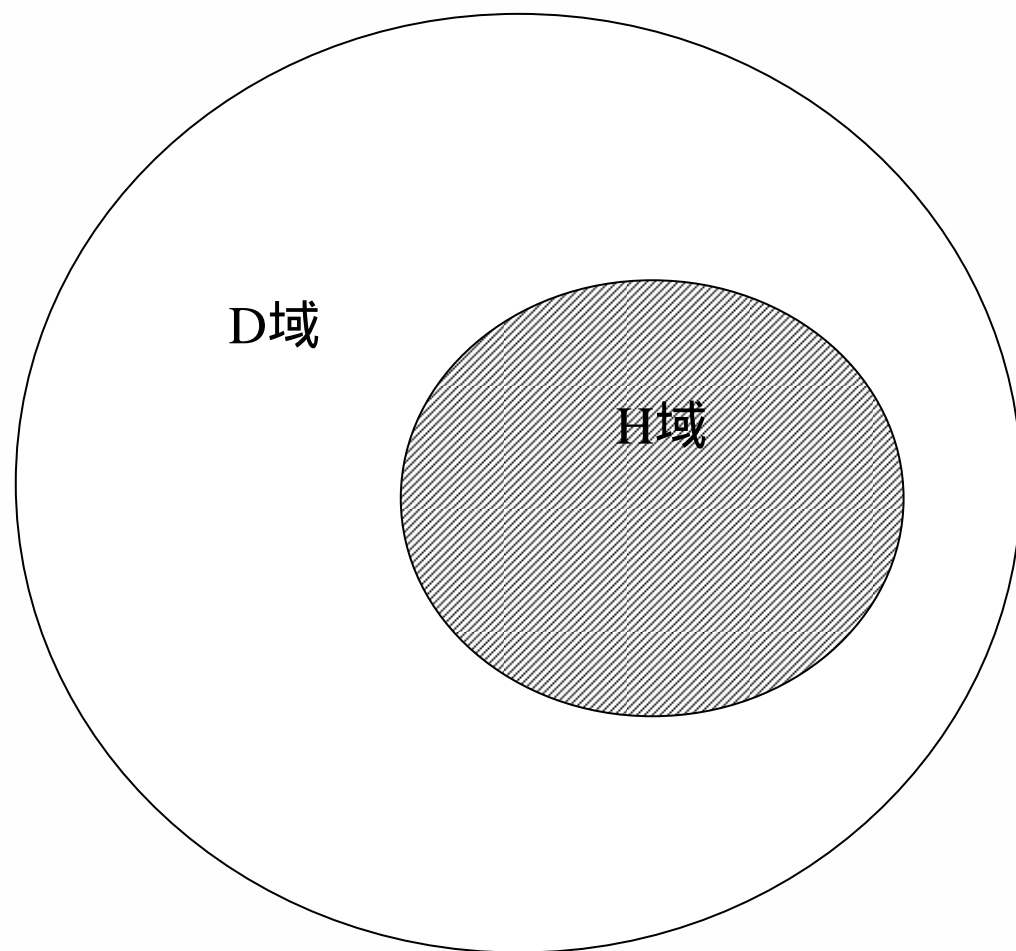
- 问题：
一阶逻辑公式的永真性（永假性）
的判定是否能在有限步内完成？

Herbrand定理

- 1936年图灵(Turing)和邱吉 Church) 互相独立地证明了：
 - “没有一般的方法使得在有限步内判定一阶逻辑的公式是否是永真（或永假）。但是如果公式本身是永真（或永假）的，那么就能在有限步内判定它是永真（或永假）。对于非永真（或永假）的公式就不一定能在有限步内得到结论。判定的过程将可能是不停止的。”

Herbrand定理 (H域)

- 基本方法：
 - 因为量词是任意的，所讨论的个体变量域D是任意的，所以解释的个数是无限、不可数的。
 - 简化讨论域。建立一个比较简单、特殊的域，使得只要在这个论域上，该公式是不可满足的。 $H_i = H_{i-1} \cup \{f(t_1, \dots, t_n)\}$
 - 此域称为H域。



H域与D域关系示意图

Herbrand定理 (H域定义)

定义：S为子句集，海伯伦域H 定义为

(1) 令 H_0 定义为S中所有个体常量的集合，若S中无个体常量，取 $H_0=\{a\}$, a为任意指定的一个个体常量

(2) 令 $H_{i+1} = H_i \cup \{S中所有n元函数f(x_1, x_2, \dots, x_n) \mid x_j \text{ 是 } H_i \text{ 中的元素}\}$, $i=0,1,2,\dots$

例2 : $S=\{P(x) \vee Q(x), R(f(y))\}$

$$H_0=\{a\} , H_1=\{a,f(a)\} , H_2=\{a,f(a),f(f(a))\},...$$

$$H = \{a,f(a),f(f(a)) , f(f(f(a))),...\}$$

例3 : $S=\{P(a) , Q(b), R(f(x))\}$

$$H_0=\{a,b\} , H_1=\{a,b,f(a),f(b)\} ,$$

$$H_2=\{a,b,f(a),f(b),f(f(a)),f(f(b))\},...$$

例4 : $S=\{P(a) , Q(f(x)), R(g(y))\}$

$$H_0=\{a\} , H_1=\{a,f(a),g(a)\} ,$$

$$H_2=\{a,f(a),g(a),f(f(a)),f(g(a)),g(f(a)),g(g(a))\},...$$

例5 : $S=\{P(x) , Q(y) \vee R(y)\}$

H域通常是无限的

$$H_0= H_1= H_2=...= H = \{a\}$$

H域例题

- 设子句集 $S = \{ P(x), Q(y, f(z, b)), R(a) \}$, 求H域

- 解 :

$H_0 = \{a, b\}$ 为子句集中出现的常量

$H_1 = \{a, b, f(a, b), f(a, a), f(b, a), f(b, b)\}$

$H_2 = \{ a, b, f(a, b), f(a, a), f(b, a), f(b, b),$
 $f(a, f(a, b)), f(a, f(a, a)), f(a, f(b, a)), f(a, f(b, b)),$
 $f(b, f(a, b)), f(b, f(a, a)), f(b, f(b, a)), f(b, f(b, b)),$
 $f(f(a, b), f(a, b)), f(f(a, b), f(a, a)), f(f(a, b), f(b, a)), f(f(a, b), f(b, b)),$
 $f(f(a, a), f(a, b)), f(f(a, a), f(a, a)), f(f(a, a), f(b, a)), f(f(a, a), f(b, b)),$
 $f(f(b, a), f(a, b)), f(f(b, a), f(a, a)), f(f(b, a), f(b, a)), f(f(b, a), f(b, b)),$
 $f(f(b, b), f(a, b)), f(f(b, b), f(a, a)), f(f(b, b), f(b, a)), f(f(b, b), f(b, b)) \}$

•

• $H = H_1 \cup H_2 \cup H_3 \cup \dots$

定义3：用H域上的元素代换子句集中的变元，所得的子句集称为基子句集；

定理1：子句集S不可满足 \Leftrightarrow S对H域上的一切解释均为假

定理2：子句集S不可满足 \Leftrightarrow 存在有限的不可满足的基子句集S'

定理的意义

- 将证明问题转化成了命题逻辑问题。
- 从理论上可用机器实现自动定理证明。

不足

- 只给出了半可判定方法
- 在可判定情况下，计算量大

Herbrand定理 (结论)

- 定理的意义
 - Herbrand定理已将证明问题转化成了命题逻辑问题。
 - 由此定理保证，可以放心的用机器来实现自动推理了。（归结原理）
- 注意
 - Herbrand定理给出了一阶逻辑的半可判定算法，即仅当被证明定理是成立时，使用该算法可以在有限步得证。而当被证定理并不成立时，使用该算法得不出任何结论。

但是

Herbrand定理 (结论)

- 仍存在的问题：
基子句集序列元素的数目随子句集的元素数目成指数地增加。
- 因此，Herbrand定理是30年代提出的，始终没有显著的成绩。

.....

直至1965年Robinson提出了归结原理。

第三章 归结推理方法

The End.