人工智能

Artificial Intelligence

第四章 归结推理

课 前 索 引

推理的理论和技术是专家系统、程序推导、程序正确性证明、智能机器人等研究领域的重要基础。

【学习目标】

本章主要讨论命题逻辑和一元谓词逻辑的归结推理方法。同学需要在熟练掌握一般逻辑知识的基础上，学习Skolem标准形和Herbrand定理，从而对归结原理有一个比较透彻的了解。

【课前思考】

◇在命题逻辑中，归结法的逻辑基础是什么？
◇什么样的命题可以由归结法来证明？
◇如何证明一个逻辑等式为真？
◇什么是合取范式和析取范式？
◇什么是子句集？
◇什么叫归结?
◇什么叫归结策略?
◇Herbrand定理和归结法之间的关系。

本章主要内容

【主要内容】

理解逻辑是人工智能的基本工具。认识归结方法是实现定理机器证明的一种方法，归结过程简单明了，而且归结法对于谓词逻辑描述的定理集来说是完备的。会用归结法证明定理。理解**Herbrand**定理给出了一阶逻辑的半可判定算法。

【知识点】

◇ 命题逻辑的归结

◇ 谓词逻辑的前束范式，约束变量换名规则

◇ **Skolem**(斯柯伦)标准形的定义，子句和子句集， **Skolem**定理及推广

◇ **H**域、**H**解释、语义树、 **Herbrand**(海伯伦)定理

◇ 合一和置换，归结过程

概述

什么是推理

所谓推理是指按照某种策略从已知事实出发去推出结论的过程。

推理所用的事实可分为两种情况：一种是与求解问题有关的初始证据，另一种是推理过程中所得到的中间结论。

智能系统的推理过程是通过推理机来完成的。推理机是智能系统中用来实现推理的那些程序。

智能系统的推理包括两个基本问题：

一个是推理的方法，另一个是推理的控制策略。

概述

推理方法及其分类

推理方法主要解决在推理过程中前提与结论之间的逻辑关系，以及在非精确性推理中不确定性的传递问题。

1．按推理的逻辑基础分类

常用的推理方法可分为演绎推理、归纳推理和默认推理。

(1)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。它是一种由一般到个别的推理方法，其核心是三段论。

常用的三段论是由一个大前提、一个小前提和一个结论三部分组成的。

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。

概述

例如，有如下三个判断：

① 计算机系的学生都会编程序；

② 程强是计算机系的一位学生；

③ 程强会编程序。

这是一个三段论推理。其中，①是大前提，②是小前提；③是经演绎推出来的结论。

演绎推理就是从已知的大前提中推导出适应于小前提的结论，即从已知的一般性知识中抽取所包含的特殊性知识。由此可见，只要大前提和小前提是正确的，则由它们推出的结论也必然是正确的。

概述

(2)归纳推理

从一类事物的大量特殊事例出发，去推出该类事物的一般性结论。它是一种由个别到一般的推理方法。

归纳推理的基本思想是：先从已知事实中猜测出一个结论，然后对这个结论的正确性加以证明确认，数学归纳法就是归纳推理的一种典型例子。

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归纳推理、类比归纳推理、统计归纳推理和差异归纳推理等。

概述

(3)默认推理

是在知识不完全的情况下假设某些条件已经具备所进行的推理，因此也称为缺省推理。

在推理过程中，如果发现原先的假设不正确，就撤消原来的假设以及由此假设所推出的所有结论，重新按新情况进行推理。

概述

(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示出来，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，当他刚开始从事这项工作时，只有书本知识，而无实际经验。但当他经过一段时间的工作实践后，就会通过大量实例积累起来一些经验，这些经验就是由一个个实例归纳出来的一般性知识，采用的是归纳推理方式。

当它有了这些一般性知识后，就可以运用这些知识去完成计算机的维修工作。而这种为某一台具体的计算机运用一般性知识进行维修的过程则是演绎推理。

概述

2．按所用知识的确定性分类

可分为确定性推理和不确定性推理。

确定性推理是指推理所使用的知识和推出的结论都是可以精确表示的，其真值要么为真，要么为假，不会有第三种情况出现。

不确定性推理是指推理时所用的知识不都是确定的，推出的结论也不完全是确定的，其真值会位于真与假之间。

由于现实世界中的大多数事物都具有一定程度的不确定性，并且这些事物是很难用精确的数学模型来进行表示与处理的，因此不确定性推理也就成了人工智能的一个重要研究课题。

概述

3．按推理过程的单调性

分为单调推理与非单调推理

单调推理指在推理过程中，每当使用新的知识后，所得到的结论会越来越接近于目标，而不会出现反复情况，即不会由于新知识的加入否定了前面推出的结论，从而使推理过程又退回到先前的某一步。

非单调推理是指在推理过程中，当某些新知识加入后，会否定原来推出的结论，使推理过程退回到先前的某一步。非单调推理往往是在知识不完全的情况下发生的。在这种情况下，为使推理能够进行下去，就需要先作某些假设，并在此假设的基础上进行推理。但是，当后来由于新的知识加入，发现原来的假设不正确时，就需要撤销原来的假设以及由此假设为基础推出的一切结论，再运用新知识重新进行推理。

概述

4. 推理的控制策略及其分类

智能系统的推理过程相当于人类的思维过程，它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。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。

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；

搜索策略主要解决推理线路、推理效果、推理效率等问题。

概述

4. 推理的控制策略及其分类

推理方向用来确定推理的控制方式，即推理过程是从初始证据开始到目标，还是从目标开始到初始证据。

按照对推理方向的控制，推理可分为正向推理、逆向推理、混合推理及双向推理四种情况。

无论哪一种推理方式，系统都需要有一个存放知识的知识库，一个存放初始证据及中间结果的综合数据库和一个用于推理的推理机。

概述

归结原理由J.A.Robinson于1965年提出，又称为消解原理。

其基本思路是：要证明在一个论域上一个事件是永真的，就要证明在该域中的每一个点上该事实都成立。很显然，论域是不可数时，该问题不可能解决。即使可数，如果该轮域是无限的，问题也无法简单地解决。

Herbrand采用了反证法的思想，将永真性的证明问题转化成为不可满足性的证明问题。

归结法的基本原理是采用反证法或者称为反演推理方法，将待证明的表达式(定理)转换成为逻辑公式(谓词公式)，然后再进行归结，归结能够顺利完成，则证明原公式(定理)是正确性的。

概述

- 定理证明的任务：

  由前提A1 ∧A2 ∧... ∧An

  推出结论B

  即证明：A1 ∧A2 ∧... ∧An→B 永真

- 转化为证明：

  A1 ∧A2 ∧... ∧An ∧ ~B为永假式

- 归结推理就是：从A1 ∧A2 ∧... ∧An ∧ ~B出发，使用归结推理规则来找出矛盾，最后证明定理A1 ∧A2 ∧... ∧An→B的成立

命题逻辑的归结

◇数理逻辑的基本定义

下面所列的是一些数理逻辑中重要的定义，在后面的分析中要用到：

- ·命题：真假确定的简单陈述句
- ·合取式：**p**与**q**，记做**p** ∧ **q**
- ·析取式：**p**或**q**，记做**p** ∨ **q**
- ·蕴含式：如果**p**则**q**，记做**p** → **q**
- ·等价式：**p**当且仅当**q**，记做**p**　**q**
- ·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

◇数理逻辑的基本等值式(略)

命题逻辑的归结

◇合取范式

范式：范式是公式的标准形式，公式往往需要变换为同它等价的范式，以便对它们作一般性的处理。

合取范式：单元子句、单元子句的析取(∨)的合取(∧)。

如：$P \wedge (P \vee Q) \wedge (\sim P \vee Q)$

文字：指任一原子公式或原子公式的非。

例如：$\neg P, Q, R$ 都是文字。

子句：文字的析取范式。

例如：$\neg P \vee Q \vee R$ 就是一个子句。

$= \sim P \vee (\sim\sim Q \wedge \sim R) \vee S$

$= \sim P \vee (Q \wedge \sim R) \vee S$

$= \sim P \vee S \vee (Q \wedge \sim R)$

$= (\sim P \vee S \vee Q) \wedge (\sim P \vee S \vee \sim R)$

命题逻辑的归结

◇子句集

命题公式的子句集S是合取范式形式下的子命题(元素，即子句)的集合。

子句集是合取范式中各个合取分量的集合，生成子句集的过程可以简单地理解为将命题公式的合取范式中的与符号“∧”，置换为逗号“，”。

上例转换的合取范式：

$$(\sim P \lor S \lor Q) \land (\sim P \lor S \lor \sim R)$$

其子句集为

$S = \{\sim P \lor S \lor Q, \sim P \lor S \lor \sim R\}$

又如，有命题公式：$P \land (P \lor Q) \land (\sim P \lor Q)$

其子句集 S：$S = \{P, P \lor Q, \sim P \lor Q\}$

命题逻辑的归结

归结推理的核心是求两个子句的归结式，因此需要先讨论归结式的定义和性质。

归结式的定义：

设C1和C2是子句集中的任意两个子句，如果C1中的文字L1与C2中的文字L2互补，那么可从C1和C2中分别消去L1和L2，并将C1和C2中余下的部分按析取关系构成一个新子句C12，则称这一个过程为归结，称C12为C1和C2的归结式，称C1和C2为C12的亲本子句。

例如：有子句：$C1 = P \vee C1'$，

$C2 = \sim P \vee C2'$

存在互补对 P和～P，

则可得归结式：$C12 = C1' \vee C2'$

注意：$C1 \wedge C2 \rightarrow C12$， 反之不一定成立。

命题逻辑的归结

下面证明归结式是原两子句的逻辑推论，或者说任一使C1、C2为真的解释I下必有归结式C12也为真。(C1ΛC2 → C12)

有子句：C1＝P∨C1‘，

C2＝～P∨C2’

存在互补对 P和～P，

证明：设I是使C1，C2为真的任一解释，若I下的P为真，从而～P为假，必有I下C2‘为真，故C12为真。若不然，在I下P为假，从而I下C1’为真，故I下C12为真。于是C1∧C2为真。于是C1∧C2→R(C1,C2)成立。

反之不一定成立，因为存在一个使C1‘∨C2’为真的解释I，不妨设C1‘为真，C2’为假。若P为真，则～P∨C2‘就为假了。因此反之不一定成立。由此可得归结式的性质。

归结式的性质：归结式C12 是亲本子句C1 和C2的逻辑结论。

命题逻辑的归结

命题逻辑的归结方法推理过程可以分为如下几个步骤：

1 建立待归结命题公式：首先根据反证法将所求证的问题转化成为命题公式，求证其是矛盾式(永假式)。

2 求取合取范式

3 建立子句集

4 归结

归结步骤：

1) 对子句集中的子句使用归结规则

2) 归结式作为新子句加入子句集参加归结

3) 归结式为空子句“□”为止。

(证明完毕)

得到空子句，表示S是不可满足的(矛盾)，故原命题成立。

命题逻辑的归结

- 一般过程:
  1) 建立子句集S
  2) 从子句集S出发,仅对S的子句间使用归结推理规则
  3) 如果得出空子句•, 则结束;否则转下一步
  4) 将所得归结式仍放入S中
  5) 对新的子句集使用归结推理规则
  6) 转(3)
- 空子句不含有文字,它不能被任何解释满足,所以空子句是永假的,不可满足的
- 归结过程出现空子句,说明出现互补子句对,说明S中有矛盾,因此S是不可满足的.

命题逻辑的归结

- 例子:证明$(P \rightarrow Q) \wedge \sim Q \Rightarrow \sim p$
- 首先建立子句集:

原式等价$(P \rightarrow Q) \wedge \sim Q \rightarrow \sim p$

  - $(P \rightarrow Q) \wedge \sim Q \wedge \sim(\sim P)$
  - $(\sim P \vee Q) \wedge \sim Q \wedge P$
  - $S=\{\sim P \vee Q,\ \sim Q\ ,\ P\}$

- 对S作归结:

(1) $\sim P \vee Q$

(2) $\sim Q$

(3) P

(4) $\sim P$ (1)(2)归结

(5) (3)(4)归结

命题逻辑的归结

例:证明公式$(P \rightarrow Q) \rightarrow (\sim Q \rightarrow \sim P)$

证明：根据归结原理

将待证明公式转化成待归结命题公式：$(P \rightarrow Q) \wedge \sim(\sim Q \rightarrow \sim P)$

前项化为合取范式：$P \rightarrow Q = \sim P \vee Q$

后项化为合取范式：$\sim(\sim Q \rightarrow \sim P) = \sim(Q \vee \sim P) = \sim Q \wedge P$

两项合并后化为合取范式：$(\sim P \vee Q) \wedge \sim Q \wedge P$

则子句集为：

$\{\sim P \vee Q，\sim Q，P\}$

对子句集中的子句进行归结可得：

1. $\sim P \vee Q$
2. $\sim Q$
3. P
4. Q，(1，3归结)
5. □，(2，4归结)

由上可得原公式成立。

◇谓词的归结：除了有量词和函数以外，其余和命题归结过程一样。

命题逻辑的归结

- 归结方法是一种机械化的，可在计算机上加以实现的推理方法
- 可认为是一种反向推理形式
- 提供了一种自动定理证明的方法

- 归结原理是合理的
- 归结原理是完备的

谓词逻辑归结法基础

由于谓词逻辑与命题逻辑不同，有量词、变量和函数，所以在生成子句集之前要对逻辑公式做处理，具体的说就是要将其转化为**Skolem**标准形，然后在子句集的基础上再进行归结，虽然基本的归结的基本方法都相同，但是其过程较之命题公式的归结过程要复杂得多。

谓词逻辑归结法基础

Skolem 标准形

前束范式中消去所有的存在量词，则称这种形式的谓词公式为**Skolem**标准形。

前束范式：**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。其形式为：$(Q_1x_1)(Q_2x_2)\ldots(Q_nx_n)M(x_1,x_2,\ldots,x_n)$, Q_i为量词符号。

前束形==（前缀）{母式}

量词串　无量词公式

定理：任何公式**G**都等价于一个前束范式

Skolem标准形的转化过程为：依据约束变量换名规则，首先把公式变型为前束范式，然后依照量词消去原则消去或者略去所有量词。

谓词逻辑归结法基础

◇量词消去原则：

1) 消去存在量词 "∃"，

如果存在量词左边没有任何全称量词，则只将其改写成为常量；如果是左边有全称量词的存在量词，消去时该变量改写成为全称量词的函数。

即，将该量词约束的变量用任意常量(a, b等)、或全称变量的函数(**Skolem函数**)(f(x), g(y)等)代替。

2) 略去全称量词 "∀"，简单地省略掉该量词。

Skolem函数：

存在量词位于一个或多个全称量词的辖域内.此时需要Skolem函数,该函数的变元就是由那些全称量词所约束的全称量词量化的变量.Skolem函数所使用的函数符号必须是新的.

子句形--SKOLEM标准型

- **Skolem**标准型：

  没有存在量词的公式。

- 设**G**是一阶逻辑中的公式，将其化为**Skolem**标准型，母式**M**给出的子句集**S**称为公式**G**的子句集

子句形—化子句集-1

- 谓词公式化为子句形的步骤

$\forall x\ [P(x) \rightarrow [\forall y\ [P(y) \rightarrow P(f(x,y))] \wedge \sim\forall y\ [Q(x,y) \rightarrow P(y)]]]$

(1)消去蕴含符号: $P\rightarrow Q \Leftrightarrow \sim P \vee Q$

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

(2)减少否定符号的辖域,把 "~" 移到紧靠谓词的位置上

$\sim(\sim P) \Leftrightarrow P$

$\sim(P \vee Q) \Leftrightarrow \sim P \wedge \sim Q \qquad \sim(P \wedge Q) \Leftrightarrow \sim P \vee \sim Q$

$\sim(\forall x)P \Leftrightarrow (\exists x) \sim P \qquad \sim(\exists x)\ P \Leftrightarrow (\forall x) \sim P$

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

子句形—化子句集-2

(3)变量标准化:重新命名变元名,使不同量词约束的变元有不同的名字.

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

(4)消去存在量词:

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

子句形—化子句集-3

(5)化为前束形:

- 把所有的全称量词移到公式的左边,并使每个量词的辖域包含这个量词后面公式的整个部分.即得前束形
- 上例变为:

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

(6)把母式化为合取范式:

- 上例变为:

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

子句形—化子句集-4

(7)消去全称量词:

$[[\sim P(x) \vee \sim P(y) \vee P(f(x,y))] \wedge$

$[\sim P(x) \vee Q(x,g(x))] \wedge [\sim P(x) \vee \sim P(g(x))]]$

(8)消去连结词符号∧

$\sim P(x) \vee \sim P(y) \vee P(f(x,y))$

$\sim P(x) \vee Q(x,g(x))$

$\sim P(x) \vee \sim P(g(x))$

(9)更换变量名称:对变元更名,使不同子句中的变元不同名.

$\sim P(x_1) \vee \sim P(y) \vee P(f(x_1,y))$

$\sim P(x_2) \vee Q(x_2,g(x_2))$

$\sim P(x_3) \vee \sim P(g(x_3))$

子句形—化子句集--6

- 一个子句内的文字可含有变量,但这些变量总是被理解为全称量词量化的变量
- **G**与其子句集**S**并不等值.但是在不可满足的意义下两者是等价的.而且**G**是**S**的逻辑推论,**S⇒G**.反过来不成立

谓词逻辑归结法基础

总结：

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。

归结推理所需的子句集S可由下面的步骤求取：

1. 谓词公式G转换成前束范式(离散数学)；
2. 消去前束范式中的存在量词，略去其中的任意量词，生成Skolem标准形
3. 将Skolem标准形中的各个子句提出，表示为集合形式

注意： Skolem标准形必须满足合取范式的条件。在生成子句集之前逻辑表达式必须是各“谓词表达式”或“谓词或表达式”的与 。

谓词逻辑的重要定理

- 定理:若G是给定的公式,而相应的子句集为S,则G是不可满足的当且仅当S是不可满足的

- 推论：设$G=G_1\wedge \ldots \wedge G_n$, S_i 是 G_i的Skolem标准型，令$S=S_i\cup\ldots\cup S_n$，则，G是不可满足的当且仅当S是不可满足的。

谓词逻辑归结法基础

有错

定理: 谓词表达式G是不可满足当且仅当其子句集S是不可满足。

对于形如$G = G1 \wedge G2 \wedge G3 \wedge \ldots \wedge Gn$的谓词公式，G的子句集的求取过程可以分解成几个部分单独处理。如果Gi的子句集为Si,

则$S_G = \cup Si$

例：对所有的 x, y, z 来说，如果 y 是 x 的父亲，z 又是 y 的父亲，则 z 是 x 的祖父。又知每个人都有父亲，试问对某个人来说谁是它的祖父？用谓词表示这个问题，并建立子句集。

解：引入谓词：

P(x, y) 表示 x 是 y 的父亲；Q(x, y) 表示 x 是 y 的祖父

ANS(x) 表示问题的解答

对于第一个条件，"如果 y 是 x 的父亲，z 又是 y 的父亲，则 z 是 x 的祖父"，

有 A_1：$(\forall x)(\forall y)(\forall z)(P(x, y) \wedge P(y, z) \rightarrow Q(x, z))$

则把 A_1 化为合取范式，进而化为 Skolem 标准形：

S_{A1}：$\sim P(x, y) \vee \sim P(y, z) \vee Q(x, z)$

对于第二个条件："每个人都有父亲"，有：A_2：$(\forall y)(\exists x)P(x, y)$

化为 Skolem 标准形：S_{A2}：$P(f(y), x)$

结论：某个人是它的祖父，有：B：$(\exists x)(\exists y)Q(x, y)$

否定后得到子句：$S_{\sim B}$：$\sim Q(x, y) \vee ANS(x)$

则得到的相应的子句集为：$\{S_{A1}, S_{A2}, S_{\sim B}\}$。解毕。

谓词逻辑归结法基础

例　设有：

F1：$(\forall x)(N(x) \rightarrow GZ(x) \wedge I(x))$　　自然数都是大于零的整数

F2：$(\forall x)(I(x) \rightarrow E(x) \vee O(x))$　　所有整数不是偶数就是奇数

F3：$(\forall x)(E(x) \rightarrow I(s(x)))$　　偶数除以 2 是整数

求证：所有自然数不是奇数就是其一半为整数的数。

证明：首先把要求的问题用谓词公式表示出来：

G：$(\forall x)(N(x) \rightarrow (O(x) \vee I(s(x))))$

把 F1、F2、F3、及¬ G 化成子句集：

(1) $\neg N(x) \vee GZ(x)$

(2) $\neg N(u) \vee I(u)$

(3) $\neg I(y) \vee E(y) \vee O(y)$

(4) $\neg E(z) \vee I(s(z))$

(5) $N(t)$

(6) $\neg O(t)$

(7) $\neg I(s(t))$

对上述子句进行归结：

(8) $\neg I(y) \vee E(y)$　　(3)与(6)归结

(9) $\neg E(z)$　　(4)与(7)归结

(10) $\neg I(y)$　　(8)与(9)归结

(11) $\neg N(u)$　　(2)与(10)归结

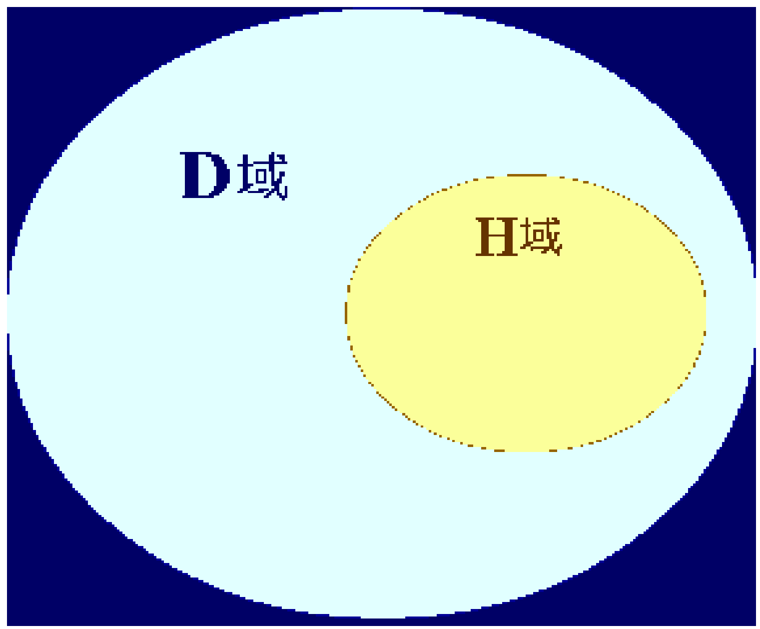
(12) ■　　(5)与(11)归结

∴ 所有自然数不是奇数就是其一半为整数的数。

Herbrand定理

Herbrand 定理思想：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，要找到所有的解释是不可能的。**Herbrand** 定理的基本思想是简化讨论域，建立一个比较简单、特殊的域，使得只要在这个论域上(此域称为**H**域)，原谓词公式仍是不可满足的，即保证不可满足的性质不变。

H域和**D**域关系的如下图表示：



Herbrand定理

H域(Herbrand域)的定义：

设G是已给的公式，定义在论域D上，令H_0是G中所出现的常量的集合。

若G中没有常量出现，就任取常量$a \in D$，而规定$H_0=\{a\}$。

$H_i= H_{i-1} \cup \{$所有形如$f(t_1,\ldots,t_n)$的元素$\}$

其中$f(t_1,\ldots,t_n)$是出现于G中的任一函数符号，而$t_1,\ldots,t_n$ 是H_{i-1}的元素，$i=1,2,\ldots$。

规定H_∞为G的H域(或说是相应的子句集S的H域)。

不难看出，H域是直接依赖于G的，最多只有可数个元素。

例: $S=\{P(a),\sim P(x) \vee P(f(x))\}$

依定义有

$H_0=\{a\}$

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

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

…

$H_\infty=\{a,f(a),f(f(a)),\ldots\}$

Herbrand定理

例： S={P(f(x),a,g(y),b)}

依定义有

$H_0=\{a,b\}$

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

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

…

$H_\infty=H_0 \cup H_1 \cup H_2 \ldots$

如果在S中出现函数形如f(x,a)仍视为f(X1 ,X2)的形式，这时若$H_0=\{a,b\}$,则H1中除有f(a,a), f(b,a)外，还出现f(a,b), f(b,b)

注意：一个函数中含有多个变量时，每个变量都要做到全部的组合。

Herbrand定理

◇原子集A：

为研究子句集S中的不可满足性，需要讨论H域上S中各谓词的真值。这里原子集A为公式中出现的谓词套上H域的元素组成的集合。

例： S={P(x)∨Q(x),R(f(y))}，有

H={a,f(a),f(f(a)), …}

A={P(a),Q(a),R(a),P(f(a)),Q(f(a)),R(f(a)),…}。

◇解释I:

谓词公式G在论域D上任何一组真值的指派称为一个解释。

◇H解释：

子句集S在的H域上的解释称为H解释。

I是H域下的一个指派。简单地说，原子集A中的各元素真/假组合都是H的解释(或真或假只取一个)。或者说凡对A中各元素真假值的一个具体设定，都是S的一个H解释。

Herbrand定理

例：$S=\{P(x)\vee Q(x), R(f(y))\}$，求其一个H解释I*

解：S的H域为：{a, f(a), f(f(a)), …}

S的原子集为：

{P(a), Q(a), R(a), P(f(a)), Q(f(a)), R(f(a)), …}

凡对A中各元素真假值的一个具体设定都为S的一个H解释。

I1*＝{P(a), Q(a), R(a), P(f(a)), Q(f(a)), R(f(a)), …}

I2*＝{～P(a), ～Q(a), ～R(a), ～P(f(a)), Q(f(a)), R(f(a)), …}

I3*＝{P(a), ～Q(a), ～R(a), P(f(a)), Q(f(a)), ～R(f(a)), …}

I1*， I2*， I3*中出现的P(a)表示P(a)的取值为T, 出现的～P(a)表示P(a)的取值为F。显然在H域上，这样的定义I*下，S的真值就确定了。

如：S| I1*＝T， S| I2*＝F， S| I3*＝F

这是因为子句集$S=\{P(x)\vee Q(x), R(f(y))\}$的逻辑含义为：

$(\forall x)(\forall y)((P(x)\vee Q(x))\wedge R(f(y)))$,

论域H为{a, f(a), f(f(a)), …}。

Herbrand定理

术语的定义：

没有变量出现的原子、文字、子句和子句集，分别称为基原子、基文字、基子句和基子句集。

若一个解释I*使得某个基子句为假，则此解释I*为假。

关键：

对于公式G的所有的解释，如果公式取值全为假，才可以判定G是不可满足的。因为所有解释代表了所有的情况，如果这些解释可以被穷举，我们就可以在有限的步数内判断公式G的不可满足性，本小节开始所提的问题便可解决 。

定理1：子句集S是不可满足的，当且仅当S的一切H解释都为假。

定理2(Herbrand定理)：子句集S是不可满足的，当且仅当对每一个解释I下，至少存在一个有限的不可满足的基子句集S'。

归结原理

虽然Herbrand 定理给出了推理算法，但这种方法需逐次生成基子句集S0',S1', …再检验Si'的不可满足性，这常常是难以实现。

例:S＝{P(x,g(x),y,h(x,y),z,k(x,y,z)),～P(u,v,e(v),w,f(v,w),x)}

有H0＝{a}

S0'＝{P(a,g(a),a,h(a,a),a,k(a,a,a)),～P(a,a,e(a),a,f(a,a),a)}

H1 = {a,g(a),h(a,a),k(a,a,a),e(a),f(a,a)}共含6个元素。

S1'：对S中文字P的变量x,y,z均可取值于H1的6个元素，从而对文字P可构成6^3种可能的形式。对文字～P的变量u,v,w,x也可取值于H1的6个元素。从而对文字～P可构成6^4种可能的形式。故S1' 有$6^3+6^4=1512$个元素。

H2：元素个数有6^3数量级

S2'：元素个数有$(6^3)^4$数量级

由于S 是可满足的基例集，还要继续这个过程，建立S3',S4',直到S5'才是不可满足。然而S5'元素个数已达$(10^{64})^4$数量级上，这是当今计算机无法处理的。

这样简单例子说明Herbrand定理所给的算法不能直接应用。

归结原理

1965年 **Robinson** 提出了归结原理。谓词逻辑的归结法是以命题逻辑的归结法为基础，在**Skolem**标准形的子句集上，通过置换和合一进行归结的。

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<u>合一</u>和<u>置换</u>。

置换

- 置换：可以简单的理解为是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, …, t_n/x_n}的有限集合。其中，x_1, x_2, …, x_n是互不相同的变量，t_1，t_2，…，t_n是不同于x_i的项（常量、变量、函数）；t_i/x_i表示用t_i置换x_i，并且要求t_i与x_i不能相同，而且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$\theta = \{t_1/x_1, t_2/x_2, \ldots, t_n/x_n\}$,

  $\lambda = \{u_1/y_1, u_2/y_2, \ldots, u_n/y_n\}$，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

  $\{t_1\cdot\lambda/x_1, t_2\cdot\lambda/x_2, \ldots, t_n\cdot\lambda/x_n, u_1/y_1, u_2/y_2, \ldots, u_n/y_n\}$

  中删去以下两种元素：

  - i. 当$t_i\lambda=x_i$时，删去$t_i\lambda/x_i$ $(i = 1, 2, \ldots, n)$;
  - ii. 当$y_i\in\{x_1,x_2, \ldots, x_n\}$时，删去$u_j/y_j$ $(j = 1, 2, \ldots, m)$

  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

- 例：

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

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

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， …， F_n}，若存在一个置换θ，可使F_1θ＝F_2θ＝…= F_nθ，则称θ是F的一个合一。同时称F_1， F_2， … ， 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λ＝{a/x, g(a)/y, f(g(a))/z}是它的一个合一。

注意：一般说来，一个公式集的合一不是唯一的。

归结原理

归结过程

谓词逻辑的归结过程与命题逻辑的归结过程相比，其基本步骤相同，但每步的处理对象不同。谓词逻辑需要把由谓词构成的公式集化为子句集，必要时在得到归结式前要进行置换和合一。

·具体的谓词逻辑归结过程如下：
·写出谓词关系公式
·用反演法写出谓词表达式
·化为**Skolem**标准形
·求取子句集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
·命题得证

反演法举例：A→B，其中A，B是谓词公式，（求非蕴含等值）

使用反演法后得：　$G = A \wedge \sim B$

归结原理

例："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"

$(\forall x)((Pass(x, computer) \wedge Win(x, prize)) \rightarrow Happy(x))$

R2:"任何肯学习或幸运的人都可以通过所有考试"

$(\forall x)(\forall y)(Study(x) \vee Lucky(x) \rightarrow Pass(x, y))$

R3:"张不肯学习但他是幸运的"

$\sim Study(zhang) \wedge Lucky(zhang)$

R4:"任何幸运的人都能获奖"

$(\forall x)(Luck(x) \rightarrow Win(x, prize))$

结论"张是快乐的"的否定

$\sim Happy(zhang)$

归结原理

将上述谓词公式转化为子句集并进行归结如下：

首先将每一个表示逻辑条件的谓词子句转换为子句集可以接受的Skolem标准形。

由R1及逻辑转换公式:$P \wedge W \rightarrow H = \sim(P \wedge W) \vee H$，可得

(1) ～Pass(x, computer)∨～Win(x, prize)∨Happy(x)

由R2可得

(2) ～Study(y)∨Pass(y,z)

(3) ～Lucky(u)∨Pass(u,v)

由R3可得

(4) ～Study(zhang)

(5) Lucky(zhang)

由R4可得

(6) ～Lucky(w)∨Win(w，prize)

由结论可得

(7) ～Happy(zhang) 结论的否定

归结原理

根据以上7条子句，归结如下：

(8) ～Pass(w, computer)∨Happy(w)∨～Luck(w)

(1), (6)归结，{w/x}

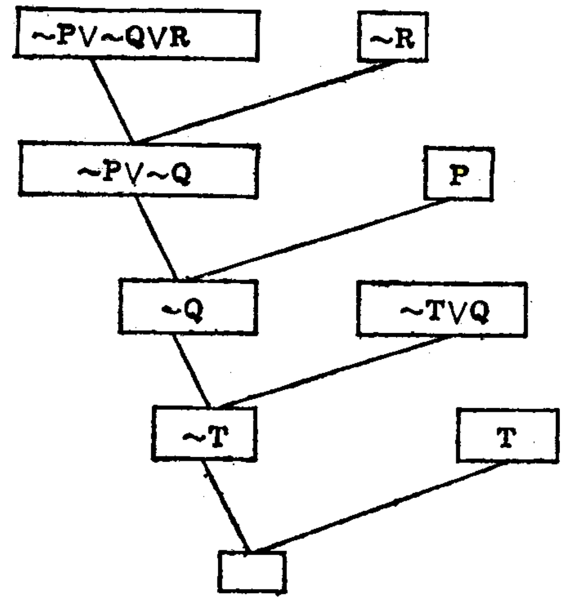
(9) ～Pass(zhang, computer)∨～Lucky(zhang)

(8), (7)归结，{zhang/w}

(10) ～Pass(zhang, computer) (9), (5)归结

(11) ～Lucky(zhang) (10),(3)归结,{zhang/u, computer/v}

(12) •□ (11),(5)归结



结论

1．归结法的实质：

·归结法是仅有一条推理规则的推理方法。

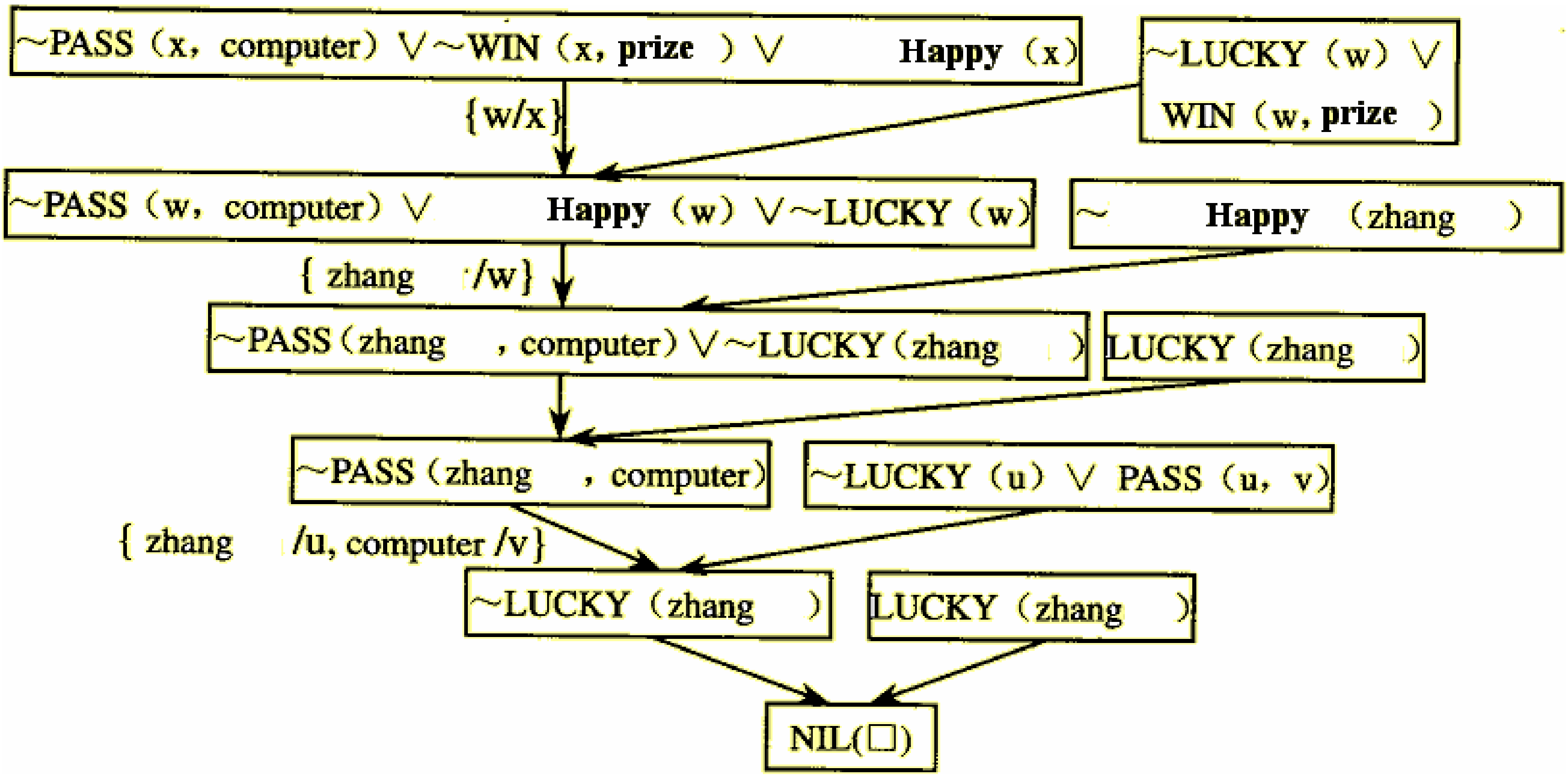
·归结的过程是一个归结树的过程。

2．归结法的问题

·传统的归结法不能处理相等的关系。

归结原理

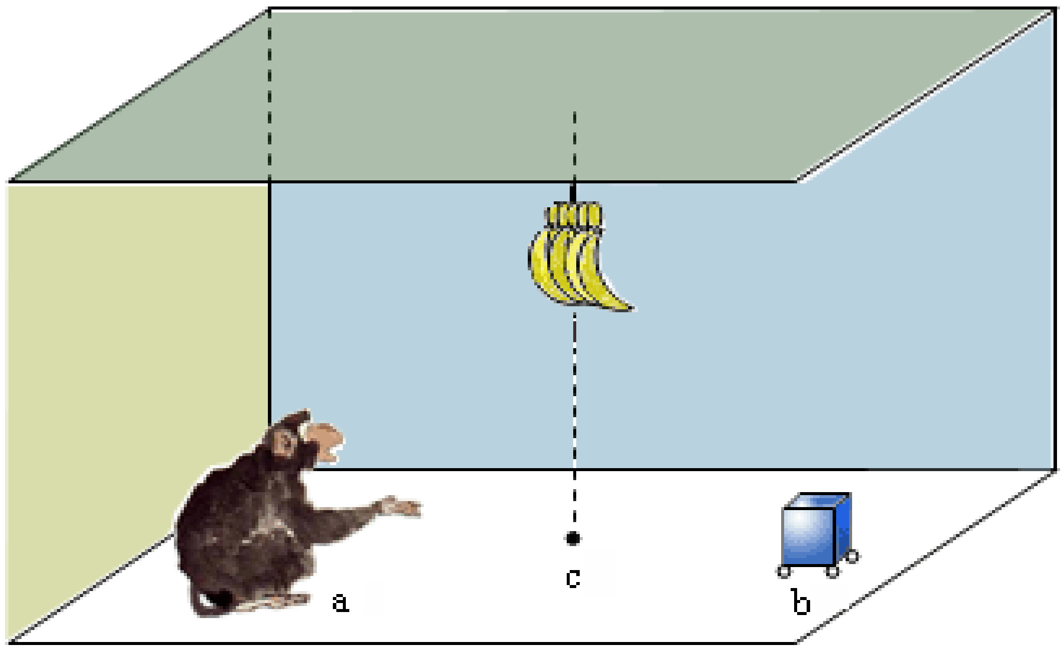
上述例题的归结树的示例



归结原理

例：猴子香蕉问题

已知一串香蕉挂在天花板上，猴子直接去拿是够不到的，但猴子可以走动且可以搬着梯子走动，也可以爬上梯子来达到吃香蕉的目的。对这个问题的描述，不可忽视动作的先后次序，需体现出时间概念。要引入状态S来区分动作的先后，以不同的状态表现不同的时间，而状态间的转换由一些算子(函数)来实现。



归结原理

例：猴子香蕉问题

已知一串香蕉挂在天花板上，猴子直接去拿是够不到的，但猴子可以走动且可以搬着梯子走动，也可以爬上梯子来达到吃香蕉的目的。对这个问题的描述，不可忽视动作的先后次序，需体现出时间概念。要引入状态S来区分动作的先后，以不同的状态表现不同的时间，而状态间的转换由一些算子(函数)来实现。

首先引入谓词：

P(x,y,z,s)表示猴子位于**x**处，香蕉位于**y** 处，梯子位于**z**处，相应的状态为**s**。此时，谓词**P(x,y,z,s)**方为真。

R(s)表示**s**状态下猴子吃到香蕉。

ANS(s)表示形式谓词，只是为求得回答的动作序列而虚设的。

其次引入状态转移函数：

Walk (y,z,s)表示原状态**s**下，在**walk**作用下猴子从**y**走到**z** 处所建立的一个新状态。

Carry(y,z,s)表示原状态**s** 下，在**Carry** 作用下猴子搬着梯子从**y**走到**z** 处建立的一个新状态。

Climb(s) 表示原状态**s**下，在**Climb**作用下猴子爬上梯子所建立的一个新状态。

归结原理

设初始状态为S0,猴子位于a,香蕉位于b ,梯子位于c。

问题可描述如下：

A_1: $(\forall x)(\forall y)(\forall z)(\forall s)(P(x,y,z,s)\rightarrow P(z,y,z,walk(x,z,s)))$ (猴子走到梯子处)

SA_1: $\sim P(x,y,z,s)\vee(P(z,y,z,walk(x,z,s))$

A_2: $(\forall x)(\forall y)(\forall s)(P(x,y,x,s)\rightarrow P(y,y,y,carry(x,y,s)))$ (猴子搬着梯子到 y)

SA_2: $\sim P(x,y,x,s)\vee P(y,y,y,carry(x,y,s))$

A_3: $(\forall s)(P(b,b,b,s)\rightarrow R(climb(s)))$ (猴子爬上梯子吃到香蕉)

SA_3: $\sim P(b,b,b,s)\vee R(climb(x)))$

A_4: $P(a,b,c,s_0)$

SA_4: $P(a,b,c,s_0)$

B: $(\exists s)R(s)$

$S_{\sim B}$: $\sim R(s)\vee ANS(s)$

其中 ANS(s)是人为附加的，在推理过程中 ANS(s)的变量 s 同 R(s)的变量将作同样的变换，当证明结束时，ANS(s)中变量 s 便给出所要求的整个动作序列。

子句集 $S=\{SA_1,SA_2,SA_3,SA_4,S_{\sim B}\}$

课后作业

1. 将下面的公式化成子句集：$\sim(((P \vee \sim Q) \rightarrow R) \rightarrow (P \wedge R))$
2. 命题是数理逻辑中常用的公式，试使用归结法证明它们的正确性：
   a) $(P \rightarrow (Q \rightarrow R)) \rightarrow ((P \rightarrow Q) \rightarrow (P \rightarrow R))$
   b) $(Q \rightarrow \sim P) \rightarrow ((Q \rightarrow P) \rightarrow \sim Q)$
3. 下列子句是否可以合一，如果可以，写出最一般合一置换
   a) P(x, B, B) 和 P(A, y, z)
   b) P(g(f (v)) , g(u)) 和 P(x , x)
4. 将下列公式化为skolem子句形:
   $((\exists x) P(x) \vee (\exists x) Q(x)) \rightarrow (\exists x)(P(x) \vee Q(x))$
5. 使用归结法证明: $(\forall x)(A(x) \rightarrow (B(x) \wedge C(x))) \wedge (\forall x)(A(x) \wedge D(x)) \rightarrow (\forall x)(D(x) \wedge C(x))$
6. 使用归结法证明: 梯形的对角线与上下底构成的内错角相等。
7. 什么是推理？推理如何分类？

练习

将这些句子表达成一组逻辑公式，即为：

(1) 马科斯是男人。

Man(Marcus)

(2) 马科斯是庞贝人。

Pompeian(Marcus)

(3) 所有庞贝人都是罗马人。

$\forall x\ \mathrm{Pompeian}(x) \rightarrow \mathrm{Roman}(x)$

(4) 恺撒是一位统治者。

Ruler(Caesar)

(5) 所有罗马人或忠于或仇恨恺撒。

$\forall x\ \mathrm{Roman}(x) \rightarrow \mathrm{Loyalto}(x, \mathrm{Caesar}) \vee \mathrm{Hate}(x, \mathrm{Caesar})$

(6) 每个人都忠于某个人。

$\forall x \exists y\ \mathrm{Loyalto}(x, y)$

(7) 人们只想暗杀他们不忠于的统治者。

$\forall x \forall y\ \mathrm{Person}(x) \wedge \mathrm{Ruler}(y) \wedge \mathrm{Tryassassinate}(x, y) \rightarrow \neg\ \mathrm{Loyalto}(x, y)$

归结证明:

(8) 马科斯试图谋杀恺撒。

Tryassassinate(Marcus,Caesar)