一阶逻辑推理

第8章

提纲

1. 命题逻辑的局限性
2. 一阶逻辑中的基本概念
3. 知识的一阶逻辑表达方法
4. 一阶逻辑化为子句
5. 归结原理
6. 归结原理的应用

一、命题逻辑的局限性

在怪兽世界中基于推理的智能体

使用命题逻辑来描述一个怪兽世界智能体:

$$\neg P_{1,1}$$
$$\neg W_{1,1}$$
$$B_{x,y} \Leftrightarrow (P_{x,y+1} \vee P_{x,y-1} \vee P_{x+1,y} \vee P_{x-1,y})$$
$$S_{x,y} \Leftrightarrow (W_{x,y+1} \vee W_{x,y-1} \vee W_{x+1,y} \vee W_{x-1,y})$$
$$W_{1,1} \vee W_{1,2} \vee \ldots \vee W_{4,4}$$
$$\neg W_{1,1} \vee \neg W_{1,2}$$
$$\neg W_{1,1} \vee \neg W_{1,3}$$
$$\ldots$$

$\Rightarrow$ 64 个不同的命题符号, 155 个语句

命题逻辑的局限性

- KB需要对每一个方块都有“物理”的语句描述
- 对任意时间t 和任意位置 $[x^t, y^t]$,

$$L^{t}_{x,y} \wedge FacingRight^{t} \wedge Forward^{t} \Rightarrow L^{t+1}_{x+1,y}$$

- 词的激增

命题逻辑的局限性

- 目前为止，我们学习了命题逻辑
- 一些英语状态很难在命题逻辑中建模:
- “If your roommate is wet because of rain, your roommate must not be carrying **any** umbrella”
- 命题逻辑会像这样不成功的建模:
- RoommateWetBecauseOfRain => (NOT(RoommateCarryingUmbrella0) AND NOT(RoommateCarryingUmbrella1) AND NOT(RoommateCarryingUmbrella2) AND …)

命题逻辑的局限性

- 尽管命题逻辑假设了世界是由事实组成的
- 但是对对象没有标记
- 对对象之间的关系也没有标记
- RoommateCarryingUmbrella0 **对我们来说**是有意义的，它表达了：
  - 有一个对象被我们称作室友,
  - 有一个对象被我们称作Umbrella0,
  - 这两个对象之间的关系是Carrying
  - 形式上, 这些含义都不存在
  - 我们是否应该用P取代RoommateCarryingUmbrella0？

命题逻辑的局限性

原因：命题逻辑不考虑命题之间的内在联系和数量关系。

要反映这种内在联系，就要对命题逻辑进行分析，分析出其中的个体词、谓词和量词，再研究它们之间的逻辑关系，总结出正确的推理形式和规则，这就是一阶(谓词)逻辑的研究内容。

办法：将命题再次细分。

一阶逻辑中的元素

- 对象: can give these names such as Umbrella0, Person0, John, Earth, wheel, door, body …
- 关系: Carrying(., .), IsAnUmbrella(.)
  - Carrying(Person0, Umbrella0), IsUmbrella(Umbrella0)
  - Relations with one object = unary relations = properties such as red, round, prime,
- 函数: Roommate(.), ColorOf(.)
  - Roommate(Person0), ColorOf(car)
- 等价性: Roommate(Person0) = Person1

思考题

为什么要使用一阶逻辑?

二、一阶逻辑中的基本概念

一阶逻辑中的基本概念

个体、谓词和命题函数

在谓词逻辑中，将原子命题分解为谓词和个体两部分。

主语	谓语	宾语
↑	↑	↑
讨论对象	对象的性质或关系	讨论对象
↓	↓	↓
个体词（组）	谓词	个体词（组）

1、定义：在原子命题中，所描述的对象称为个体；用以描述个体的性质或个体间关系的部分，称为**谓词**。

一阶逻辑中的基本概念

这两者之间存在对应关系

- 函数, 返回值
- 谓词, 要么为真要么为假

函数: father_of(Mary) = Bill

谓词: father_of(Mary, Bill)

函数是对于每个对象都有一个作为关系的值

一阶逻辑中的基本概念

FOL的语法: 基本元素

- 常量　KingJohn, 2, NUS,...
- 谓词　Brother, >, Person(John)...
- 函数　Sqrt, LeftLegOf,...
- 变量　x, y, a, b,...
- 连接符　$\neg$, $\Rightarrow$, $\wedge$, $\vee$, $\Leftrightarrow$
- 等词　=
- 量词　$\forall$, $\exists$

一阶逻辑中的基本概念

原子语句：

原子语句= $predicate\,(term_1,\ldots,term_n)$
or $term_1 = term_2$

术语 = $function\,(term_1,\ldots,term_n)$
or *constant* or *variable*

- 例如, *Brother(KingJohn,RichardTheLionheart)*
Length(LeftLegOf(Richard))

一阶逻辑中的基本概念

复杂语句

- 复杂语句是由原子语句通过连接词和量词连接组成的

$$\neg S,\ S_1 \wedge S_2,\ S_1 \vee S_2,\ S_1 \Rightarrow S_2,\ S_1 \Leftrightarrow S_2,$$

例如. 兄弟姐妹*(KingJohn,Richard)* ⇒兄弟姐妹*(Richard,KingJohn)*

$$>(1,2) \vee \leq (1,2)$$

$$>(1,2) \wedge \neg >(1,2)$$

一阶逻辑中的基本概念

FOL的语法

$$
\begin{array}{rcl}
Sentence & \rightarrow & AtomicSentence \\
& | & (\ Sentence\ Connective\ Sentence\ ) \\
& | & Quantifier\ Variable, \ldots\ Sentence \\
& | & \neg\ Sentence \\
\\
AtomicSentence & \rightarrow & Predicate(Term, \ldots)\ |\ Term = Term \\
\\
Term & \rightarrow & Function(Term, \ldots) \\
& | & Constant \\
& | & Variable \\
\\
Connective & \rightarrow & \Rightarrow\ |\ \wedge\ |\ \vee\ |\ \Leftrightarrow \\
Quantifier & \rightarrow & \forall\ |\ \exists \\
Constant & \rightarrow & A\ |\ X_1\ |\ John\ |\ \ldots \\
Variable & \rightarrow & a\ |\ x\ |\ s\ |\ \ldots \\
Predicate & \rightarrow & Before\ |\ HasColor\ |\ Raining\ |\ \ldots \\
Function & \rightarrow & Mother\ |\ LeftLeg\ |\ \ldots
\end{array}
$$

Syntax of Propositional Logic

$$
\begin{array}{rcl}
Sentence & \rightarrow & AtomicSentence\ |\ ComplexSentence \\
AtomicSentence & \rightarrow & \mathbf{True}\ |\ \mathbf{False}\ |\ \mathrm{Symbol} \\
Symbol & \rightarrow & P\ |\ \mathbf{Q}\ |\ \mathbf{R}\ |\ \ldots \\
ComplexSentence & \rightarrow & \neg\ Sentence \\
& | & (\ Sentence \wedge Sentence\ ) \\
& | & (\ Sentence \vee Sentence\ ) \\
& | & (\ Sentence \Rightarrow Sentence\ ) \\
& | & (\ Sentence \Leftrightarrow Sentence\ )
\end{array}
$$

一阶逻辑中的基本概念

FOL的语义

- 根据模型和解释语句为真
- 模型包含对象(域元素)和关系
- 具体参考如下对应关系来解释

常量符号 → 对象

谓词符号 → 关系

函数符号 → 函数关系

- 当对象$term_1,\ldots,term_n$之间的关系是*predicate*时，这个原子语句为真：$predicate(term_1,\ldots,term_n)$

思考题

1. 一阶逻辑中包含哪些基本概念？

2. 用一阶逻辑表示下列语句

（1） 丘华和李兵都是学生；

（2） 如果张华比黎明高，黎明比王宏高，则张华比王宏高。

一阶逻辑中的基本概念

全称量词

- $\forall$<变量> <语句>　　　$\forall$**x P(x)**
- 在NUS中的每个人都是聪明的：　$\forall x\ At(x,NUS) \Rightarrow Smart(x)$
- 　　思考一下为什么不是 **$\forall x\ At(x,UNC) \wedge Smart(x)$**?
- $\forall x\ P$ 在模型m中都为真，当且仅当 P 对于模型中每一个可能的对象x为真
- 大致来说, 等价于合取P中的实例

$$
\begin{aligned}
& At(KingJohn,NUS) \Rightarrow Smart(KingJohn) \\
& \wedge \quad At(Richard,NUS) \Rightarrow Smart(Richard) \\
& \wedge \quad At(NUS,NUS) \Rightarrow Smart(NUS) \\
& \wedge \ldots
\end{aligned}
$$

一阶逻辑中的基本概念

避免犯一个常见的错

- 特定的, $\Rightarrow$ 是全称量词$\forall$里的主要连接
- 常见的错误: 误把$\wedge$ 当作$\forall$里的主要连接:

  $\forall x\ At(x,NUS) \wedge Smart(x)$

  意思是“Everyone is at NUS and everyone is smart”

一阶逻辑中的基本概念

存在量词

- ∃<变量> <语句>　　**∃x P(x)**
- NUS中有人是聪明的:　$\exists x\, At(x,NUS) \wedge Smart(x)$$
- 　思考为什么不是 $\exists x\, At(x,UNC) \Rightarrow Smart(x)$?
- $\exists x\, P$ 在模型m中为真当且仅当P对于模型中一些可能的对象x为真
- 大致来说,等价于析取P中的实例

$$
\begin{aligned}
& At(KingJohn,NUS) \wedge Smart(KingJohn) \\
\vee\ & At(Richard,NUS) \wedge Smart(Richard) \\
\vee\ & At(NUS,NUS) \wedge Smart(NUS) \\
\vee\ & \ldots
\end{aligned}
$$

一阶逻辑中的基本概念

避免犯另一个常见的错

- 特定的, $\wedge$是存在量词$\exists$里主要的连接
- Common mistake: using $\Rightarrow$ as the main connective with $\exists$:
- 常见的错误: 误把$\Rightarrow$当作$\exists$里的主要连接:

$$\exists x\ At(x,NUS) \Rightarrow Smart(x)$$

当有人不在NUS时为真!

一阶逻辑中的基本概念

量词运算

- $\forall x\ \forall y$ 等价于 $\forall y\ \forall x$
- $\exists x\ \exists y$ 等价于 $\exists y\ \exists x$
- $\exists x\ \forall y$ 不等价于 $\forall y\ \exists x$
- $\exists x\ \forall y\ Loves(x,y)$：“有人爱世界上的每个人”
- $\forall y\ \exists x\ Loves(x,y)$： “世界上每个人都至少有一个人爱着”
- 量词二元性: 可以互相表示
- $\forall x\ Likes(x,IceCream)$ $\quad$ $\neg\exists x\ \neg Likes(x,IceCream)$
- $\exists x\ Likes(x,Broccoli)$ $\quad$ $\neg\forall x\ \neg Likes(x,Broccoli)$

一阶逻辑中的基本概念

等词

- $term_1 = term_2$ 被解释为真当且仅当 $term_1$和$term_2$ 指的是同一个对象
- 例如, 任何同胞的双亲都是指同一对父母:

$$\forall x,y\ Sibling(x,y) \Leftrightarrow [\neg(\mathrm{x = y}) \wedge\ \exists \mathrm{m,f} \neg\ (\mathrm{m = f}) \wedge \mathrm{Parent(m,x)} \wedge \mathrm{Parent(f,x)} \wedge \mathrm{Parent(m,y)} \wedge\ \mathrm{Parent(f,y)}]$$

思考题

1. 一阶逻辑量词的使用中常见的错误有哪些?

2. 用带量词的一阶逻辑表示下列语句

（1）所有大学生都热爱祖国;

（2）有些人是聪明的;

三、知识的一阶逻辑表达方法

知识的一阶逻辑表达方法

例“每个人都有一个父亲”

- 定义谓词:

Person(x)：表示x是人

HasFather(x,y)：表示x有父亲y

- 谓词公式

$(\forall x)(\exists y)(\mathrm{Person}(x) \rightarrow \mathrm{HasFather}(x,y))$

知识的一阶逻辑表达方法

表达下列知识：

- 每个有理数是实数;
- 存在一个数, 它是素数;
- 对每个数x, 存在一个数y, 使得x<y;

- 令:
  - Q(x): x是有理数; P(x): x是素数;
  - R(x): x是实数; LESS(x, y): x<y;
  - $(\forall x)(Q(x) \rightarrow R(x))$
  - $(\exists x)P(x)$
  - $(\forall x)(\exists y)LESS(x, y)$

知识的一阶逻辑表达方法

例子: (自然数公理)

- 自然数公理
  - 对每个数, 存在一个且仅仅一个直接后继;
  - 没有一个数, 它的直接后继是0;
- 令f(x), g(x)表示x的后继与前续; E(x, y)表示 “x=y”
- 逻辑表示
  - $(\forall x)(\exists y)(\mathbf{E(f(x), y)} \wedge \mathbf{(\forall z)(E(z, f(x)) \rightarrow E(y, z)))}$
  - $\neg((\exists x)E(f(x), 0))$

知识的一阶逻辑表达方法

怪兽世界的知识库

- 感知
  - $\forall t,s,b\ Percept([s,b,Glitter],t) \Rightarrow Glitter(t)$
- 反射
  - $\forall t\ Glitter(t) \Rightarrow BestAction(Grab,t)$

知识的一阶逻辑表达方法

怪兽世界的知识库

$\forall x,y,a,b\ \mathit{Adjacent}([x,y],[a,b]) \Leftrightarrow$

$[a,b] \in \{[x+1,y], [x-1,y],[x,y+1],[x,y-1]\}$

方块的属性:

- $\forall s,t\ \mathit{At}(\mathrm{Agent},s,t) \wedge \mathrm{Breeze}(t) \Rightarrow \mathrm{Breezy}(s)$

在无底黑洞旁边的方块有风:

- Diagnostic 规则—由现象推测原因

  ∀s Breezy(s) ⇒ \Exi{r} Adjacent(r,s) ∧ Pit(r)$

- Causal 规则—由原因推测现象

  ∀r Pit(r) ⇒ [∀s Adjacent(r,s) ⇒ Breezy(s)$]

思考题

如何用一阶逻辑表示下面语句：

1. 上法语课的每个学生都通过了考试

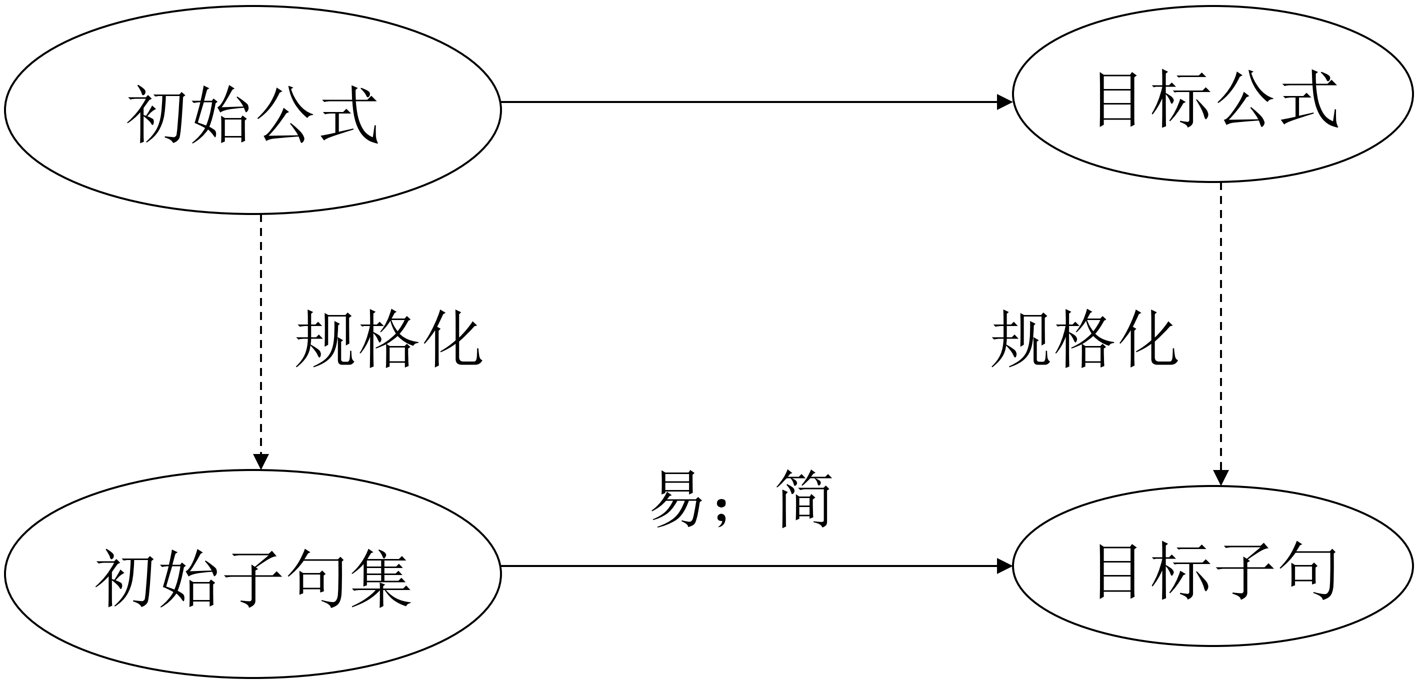
2. 猫必捕鼠

四、一阶逻辑化为子句

一阶逻辑化为子句

1. 为什么要化为子句

$(\forall x)\{[\neg P(x) \vee \neg Q(x)] \rightarrow (\exists y)[S(x, y) \wedge Q(x)] \wedge (\forall x)[P(x) \vee B(x)]$



一阶逻辑化为子句

- **子句概念**

谓词逻辑中，把原子公式及原子公式的否定统称为文字

【定义4．1】任何组成合取范式的析取式称为子句。

例如$P \vee Q$、$\neg P（x，f(x)，y）\vee Q(y) \vee R(f(x))$都是子句

【定义4．2】不包含任何文字的子句称为空子句，表示为NIL

由于空子句不包含有文字，它不能被任何解释满足，所以空子句是永假的，不可满足的

由子句构成的集合称为子句集

一阶逻辑中，任何一个一阶逻辑公式都可以化成一个子句集

一阶逻辑化为子句

将一阶逻辑公式化为子句集的步骤：

（1） **消蕴含和等价**：利用$P \to Q = \neg P \vee Q$；$P \leftrightarrow Q = (P \wedge Q) \vee (\neg P \wedge \neg Q)$等价关系消去蕴含符“$\to$”和双条件符“$\leftrightarrow$”

(2) **否定内移**：利用$\neg\neg P = P$；$\neg(P \vee Q) = \neg P \wedge \neg Q$；$\neg(P \wedge Q) = \neg P \vee \neg Q$；$\neg(\exists x)P = (\forall x)(\neg P)$；$\neg(\forall x)P = (\exists x)(\neg P)$等价关系把否定符号“$\neg$”移到紧靠谓词位置上

(3) **变量标准化**：　利用$(\forall x)P(x) = (\forall y)P(y)$；$(\exists x)P(x) = (\exists y)P(y)$等价关系将变量标准化，即使每个量词采用不同的变量

一阶逻辑化为子句

将一阶逻辑公式化为子句集的步骤：

（4）**消去存在量词∃：**

如果存在量词不在任何一个全称量词的辖域内，则该存在量词不依赖于任何其它的变量，因此可用一个新的个体常量代替 *如将(∃x)P（x）化为P(A)*

如果存在量词是在全称量词的辖域内（*如在公式(“∀y)（(∃x)P（x，y））中，变量x的取值依赖于变量y的取值）*

由**Skolem函数**（f(y)）表示依赖关系 注意，函数名应是原合式公式中没有的。

一阶逻辑化为子句

将一阶逻辑公式化为子句集的步骤：

(5) **将公式化为前束形**：把所有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的整个部分，所得的公式称为前束形

(6) **化为合取范式**:利用P∨（Q∧R）=（P∨Q）∧（P∨R）；（P∧Q）∨（P∧R）= P∧（Q∨R）等价关系将母式化为合取范式（**子句的合取式**）

(7) **略去全称量词**：母式中的变量都是全称量词量化的变量

(8) **消去合取符号∧**，把母式用**子句集**表示

如：P∧Q可表示为成子句集： P Q

(9) **子句变量标准化**：重新命名变量，使每个子句中的变量符号不同

思考题

将一阶逻辑公式化为子句集的步骤有哪些？

一阶逻辑化为子句

讲下列一阶逻辑表示化为子句：

【例】将(∀x){[¬P(x)∨¬Q(x)]→(∃y)[S(x, y)∧Q(x)]∧(∀x)[P(x)∨B(x)]}化成子句集

转换过程遵照上述9个步骤：

(1) (∀x){¬[¬P（x）∨¬Q(x)]∨(∃y)[S(x, y)∧Q(x)]}∧(∀x)[P（x）∨B（x）]

(2) (∀x){[P（x）∧Q(x)]∨(∃y)[S(x，y)∧Q(x)]}∧(∀x)[P（x）∨B（x）]

(3) (∀x){[P（x）∧Q(x)]∨(∃y)[S(x，y)∧Q(x)]}∧(∀w)[P（w）∨B（w）]

(4) (∀x){[P（x）∧Q(x)]∨[S(x，f（x））∧Q(x)]}∧(∀ w)[P（w）∨B（w）]

一阶逻辑化为子句

(5) $(\forall x)(\forall w)\{[P(x)\wedge Q(x)]\vee[S(x, f(x))\wedge Q(x)]\}\wedge[P(w)\vee B(w)]$

(6) $(\forall x)(\forall w)\{[P(x)\vee S(x, f(x))]\wedge Q(x)\wedge[P(w)\vee B(w)]\}$

(7) $[P(x)\vee S(x, f(x))]\wedge Q(x)\wedge[P(w)\vee B(w)]$

(8) 子句集为： $P(x)\vee S(x, f(x))$； $Q(x)$； $P(w)\vee B(w)$

(9) 子句变量标准化后，最终的子句集为：

$P(x)\vee S(x, f(x))$ ； $Q(y)$ ； $P(w)\vee B(w)$

思考题

将下列一阶逻辑转换成子句

$$(\forall x)((\forall y)P(x,y) \rightarrow \neg(\forall y)(Q(x,y) \rightarrow R(x,y)))$$

五、归结原理

归结原理

置换和合一

为了使用推理规则，如假言推理、假言三段论等，一个推理系统必须决定两个表达式是否相同或匹配：

两个表达式匹配当且仅当其语法是等价的

一个表达式的项可以是常量、变量或函数，**合一**就是寻找项对变量的**置换**而使表达式一致的过程，合一是人工智能中很重要的过程

如，为了使公式P(x,f(y),B)与P(x,f(B),B ）匹配，可以用常量B代替变量y，从而使两个公式一致。 称为通个置换{B/y}就可使上述公式集合一

归结原理

置换和合一

置换可用有序对的集合s={t_1/v_1， t_2/v_2， …， t_n/v_n}表示，其中t_i/v_i表示将表达式中所有的变量v_i都用项t_i代替，t_i可以是变量、常量或函数

注意，一个变量不能用含有同一变量的项来代替，被置换的一定是变量

一般可用Es表示一个表达式E用一个置换s所得到的表达式的置换

如：P（z，f（w），A）＝（P（x，f（y），A））$_s$

归结原理

置换和合一

例如：有表达式P（x，f（y），A），通过不同的置换，可分别得到：

P（z，f（w），A）

相应的置换为 $s_1 = \{ z/x , w/y \}$

P（x，f（B），A） 相应的置换为 $s_2 = \{ B/ y \}$

P（g(z)，f（B），A）

相应的置换为 $s_3 = \{ g(z)/x , B /y \}$

P（C，f（B），A）

相应的置换为 $s_4 = \{ C/x , B/y \}$

思考题

置换过程中要注意什么？

归结原理

归结原理

归结原理又称为消解原理，它是定理证明基础

由谓词公式转化为子句集的过程中可以看出，在子句集中子句之间是合取关系，其中只要有一个子句不可满足，则子句集就不可满足。若一个子句集中包含空子句，则这个子句集一定是不可满足的

归结原理就是基于这一认识提出来的

【定义5.1】若P是原子谓词公式，则称P与¬P为互补文字

归结原理

- 一阶逻辑归结

在一阶逻辑中，子句中含有变量

为将归结原理应用于含有变量的子句，应找出一个置换，作用于给定的两个子句，使它们包括互补的文字，然后才能进行归结

例1:子句集S={P(x)∨Q(x)，¬P(A)∨R(y)}，子句集中的两个子句不能直接归结，但若对子句集先进行置换s={A/x}，则两个子句分别为P(A)∨Q(A)和¬P(A)∨R(y)，这时再进行归结，归结结果为Q(A)∨R(y)

归结原理

【定义5.2】设C_1和C_2是两个没有相同变量的子句，并分别表示成两个文字集合$\{L_i\}$和$\{M_i\}$，$\{l_i\}$是$\{L_i\}$的一个子集，$\{m_i\}$是$\{M_i\}$的一个子集，若s是集合$\{l_i\}$和$\{\neg m_i\}$的并集的最简合一者，则称

$$C_{12}=\{\{L_i\}-\{l_i\}\}_s \vee \{\{M_i\}-\{m_i\}\}_s$$

为C_1和C_2的归结式。

当两个子句作归结时，子集$\{l_i\}$和$\{m_i\}$的选取可能有多种形式，所以得到的归结式不是唯一的

归结原理

例2： 设有两个子句$P(A)\vee\neg Q(x)\vee R(x)$和$\neg P(y)\vee Q(B)$，则由如下两种归结方法：

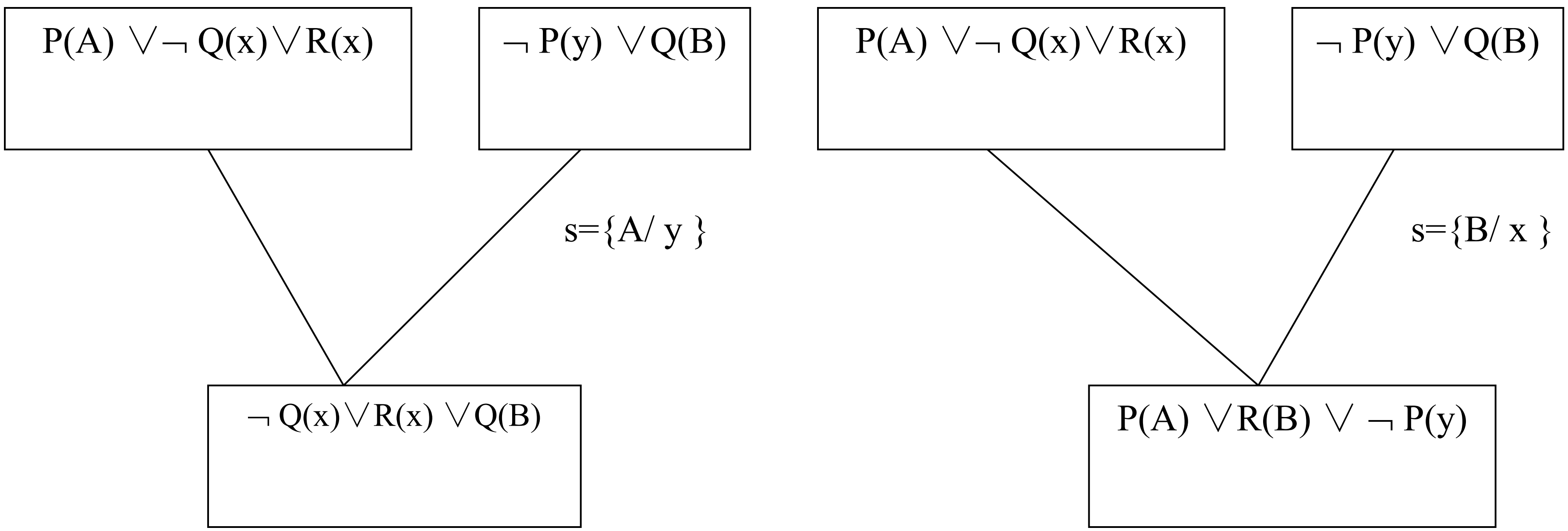
① 取$\{l_i\}=\{P(A)\}$，$\{m_i\}=\{\neg P(y)\}$，$\{l_i\}$和$\{\neg m_i\}$的最简合一者为$s=\{A/y\}$，此时归结结果为$\neg Q(x)\vee R(x)\vee Q(B)$

② 取$\{l_i\}=\{\neg Q(x)\}$，$\{m_i\}=\{Q(B)\}$，$\{l_i\}$和$\{\neg m_i\}$的最简合一者为$s=\{B/x\}$，此时归结结果为$P(A)\vee R(B)\vee\neg P(y)$

- **注意：** 在求归结式时，**不能同时消去两个互补文字对**，消去两个互补文字对所得的结果不是两个亲本子句的逻辑推论。

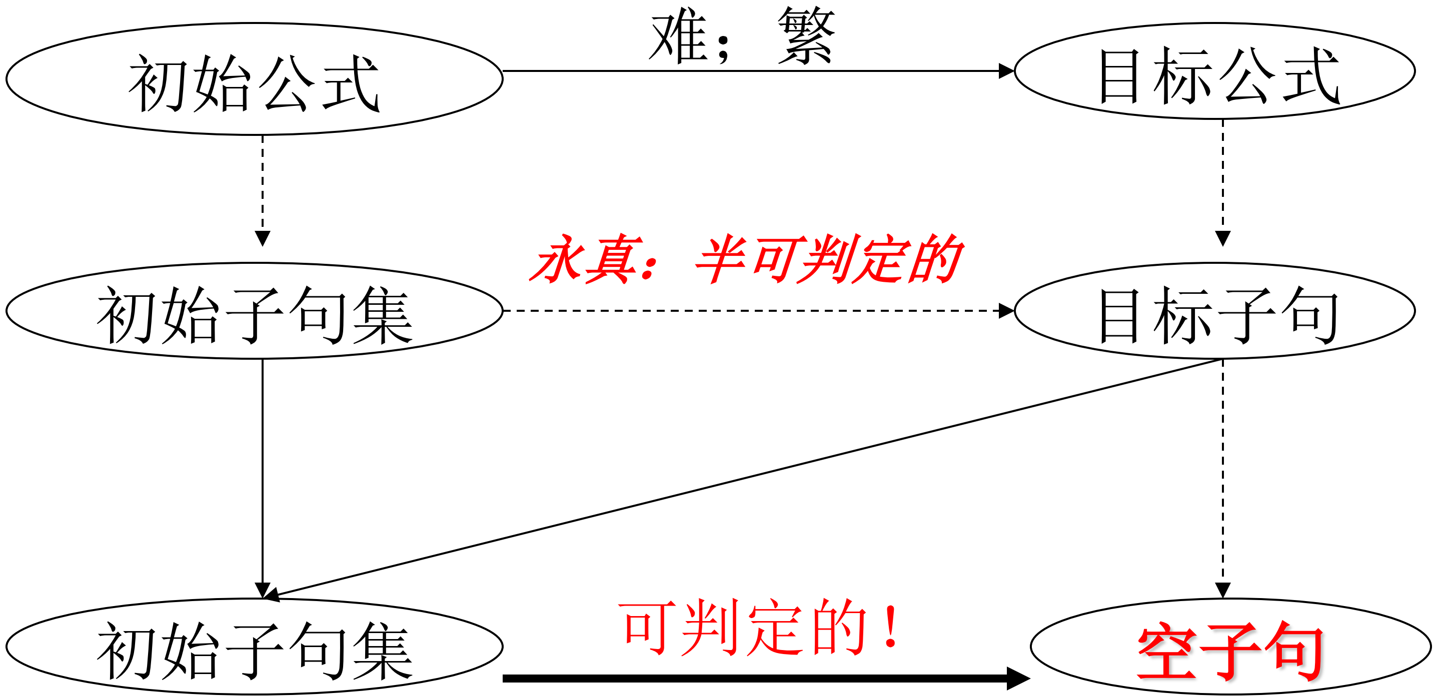
归结原理

例2的归结过程：



归结原理

- 归结反演



归结原理

- 归结反演

  归结反演就是利用归结和反演实现定理的证明

  具体过程：

(1) 将定理证明的前提谓词公式转化为子句集F

(2) 将求证的目标表示成合适的谓词公式G（目标公式）

(3) 将目标公式的否定式¬G转化成子句的形式，并加入到子句集F中，得到子句集S

(4) 应用归结原理对子句集中的子句进行归结，并把每次归结得到的归结式都并入S中。如此反复进行，若归结得到一个空子句NIL，则停止归结→ 证明了G为真

思考题

1. 归结原理的思想和过程是什么？

2. 空子句表示永真还是永假?

3. 归结过程中要注意什么？

六、归结原理的应用

归结原理的应用

- 应用归结原理进行定理证明

步骤：

设要被证明的定理可用谓词公式表示为如下的形式：

$$A_1 \wedge A_2 \wedge \ldots \wedge A_n \rightarrow B$$

(1)首先否定结论B，并将否定后的公式～B与前提公式集组成如下形式的谓词公式： $G= A_1 \wedge A_2 \wedge \ldots \wedge A_n \wedge \sim B$

(2) 求谓词公式G的子句集S。

(3) 应用归结原理，证明子句集S的不可满足性。

归结原理的应用

【例6.1】

- 已知前提为
  F: $(\forall x)\{[P(x,y)\wedge Q(y)]\rightarrow(\exists y)[R(y)\wedge S(x,y)]\}$
- 求证结论G: $\neg(\exists x)R(x)\rightarrow(\forall x)(\forall y)[P(x,y)\rightarrow\neg Q(y)]$成立
- 证明：先按前面所讲的方法将前提和结论化为子句集：

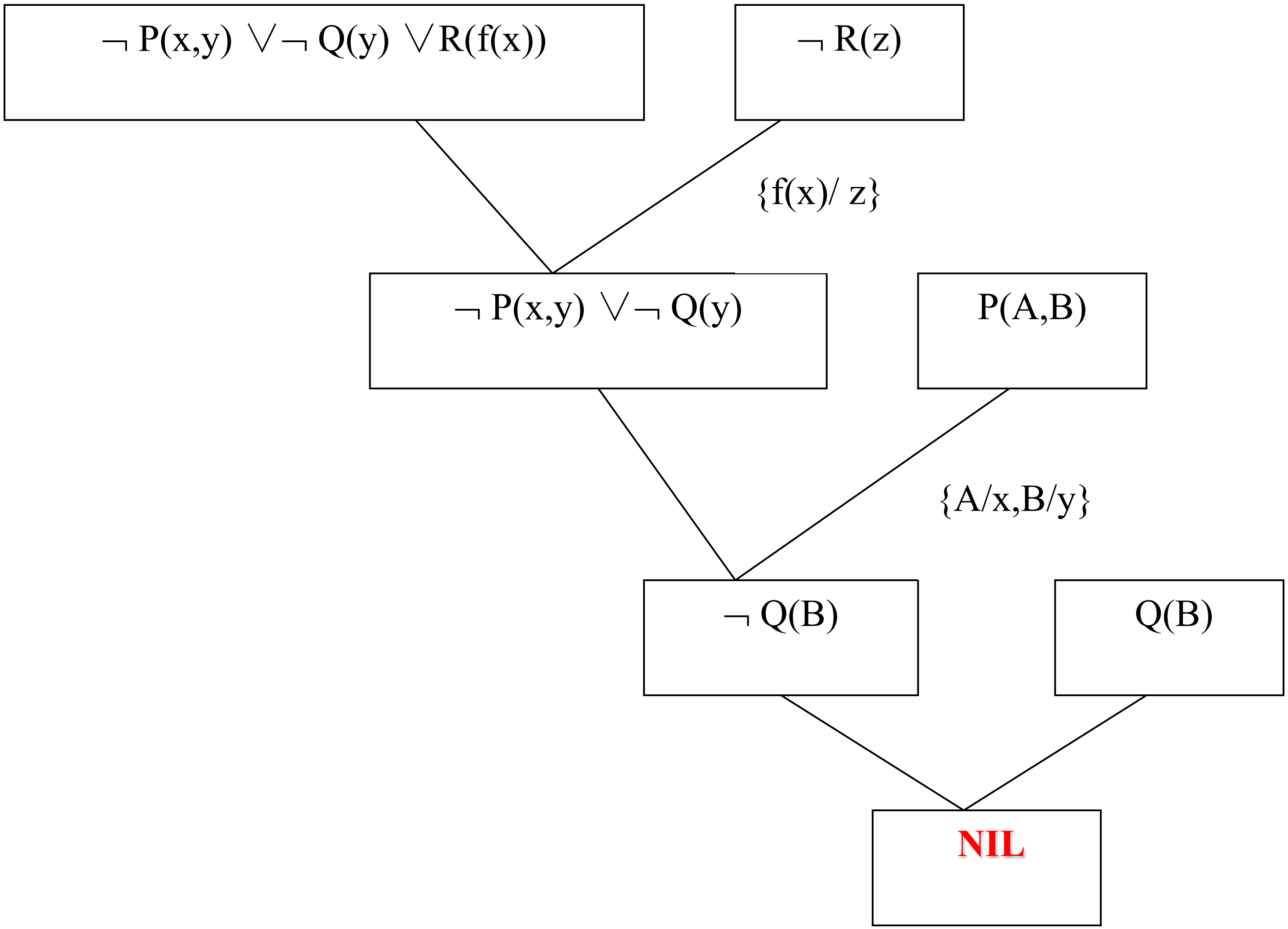
前提F所对应的子句集为：$\neg P(x,y)\vee\neg Q(y)\vee R(f(x))$

$\neg P(x,y)\vee\neg Q(y)\vee S(x,f(x))$

结论G否定对应子句集为：$\neg R(z)$； $P(A,B)$； $Q(B)$

归结原理的应用

- 归结过程如下：



归结原理的应用

例6.2.已知:某些病人喜欢所有的医生，

没有一个病人喜欢任意一个骗子。

证明: 任意一个医生都不是骗子。

证明：知识表示：令

P(x)：x是病人　　D(x)：x是医生

Q(x)：x是骗子　　L(x, y)：x喜欢y

A1: $\exists x\, (P(x) \wedge \forall y(D(y) \rightarrow L(x, y)))$

A2: $\forall x(P(x) \rightarrow \forall y(Q(y) \rightarrow \sim L(x, y)))$

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

我们要证明B是A1和A2的逻辑结果，即公式$A1 \wedge A2 \wedge \sim B$是不可满足的。

归结原理的应用

$A1 = \exists x\,(P(x) \wedge \forall y(\sim D(y) \vee L(x, y)))$

$\quad = \exists x\, \forall y\, (P(x) \wedge (\sim D(y) \vee L(x, y)))$

$\dashrightarrow \forall y\, (P(a) \wedge (\sim D(y) \vee L(a, y)))$

$A2 = \forall x(P(x) \rightarrow \forall y(\sim Q(y) \vee \sim L(x, y)))$

$\quad = \forall x(\sim P(x) \vee \forall y(\sim Q(y) \vee \sim L(x, y)))$

$\quad = \forall x \forall y\, (\sim P(x) \vee \sim Q(y) \vee \sim L(x, y))$

$\sim B = \sim(\forall x(D(x) \rightarrow \sim Q(x)))$

$\quad = \exists x\, (D(x) \wedge Q(x))$

$\dashrightarrow D(b) \wedge Q(b)$

因此，公式$A1 \wedge A2 \wedge \sim B$的子句集为

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

归结原理的应用

S不可满足的归结演绎序列为：

(1) $P(a)$

(2) $\sim D(y) \vee L(a, y)$

(3) $\sim P(x) \vee \sim Q(y) \vee \sim L(x, y)$

(4) $D(b)$

(5) $Q(b)$

(6) $L(a, b)$　　由(2)、(4) mgu:$\{b/y\}$

(7) $\sim Q(y) \vee \sim L(a, y)$　　由(1)、(3) mgu:$\{a/x\}$

(8) $\sim L(a, b)$　　由(5)、(7) mgu:$\{b/y\}$

(9) $\square$　　由(6)、(8)

思考题

练习：“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。

证明：张是快乐的。

归结原理的应用

- 利用归结原理求取问题答案

步骤：

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

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

（3）把（2）中的析取式化为子句集，并把该子句集与S_1合并构成子句集S。

归结原理的应用

（4）对子句集S应用归结原理进行归结，在归结的过程中，通过合一，改变ANSWER中的变元。

（5）如果得到归结式ANSWER，则问题的答案即在ANSWER谓词中。

归结原理的应用

例6.3. 任何兄弟都有同一个父亲，

John和Peter是兄弟，且John的父亲是David，

问:Peter的父亲是谁？

解 第一步：将已知条件用谓词公式表示出来，并化成子句集，那么要先定义谓词。

（1） 定义谓词：

设Father(x,y)表示x是y的父亲。

Brother(x,y)表示x和y是兄弟。

归结原理的应用

（2） 将已知事实用谓词公式表示出来。

F_1： 任何兄弟都有同一个父亲。

$\forall x \forall y \forall z\ (Brother(x,y) \land Father(z,x) \rightarrow Father(z,y))$

F_2： John和Peter是兄弟。

Brother(John,Peter)

F_3： John的父亲是David。

Father(David, John)

（3） 将它们化成子句集得：

$S_1=\{\sim Brother(x,y) \lor \sim Father(z,x) \lor Father(z,y),$

$Brother(John,Peter), Father(David,John)\}$

归结原理的应用

第二步：把问题用谓词公式表示出来，

并将其否定与谓词ANSWER作析取。

设Peter的父亲是u，则有：Father(u,Peter)。

将其否定与ANSWER作析取，得：

G：$\sim Father(u,Peter) \vee ANSWER(u)$

第三步：将上述公式G化为子句集S_2,并将S_1和S_2合并到S。

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

$S = S_1 \cup S_2$

归结原理的应用

第四步：应用归结原理进行归结

（5）$\sim$Brother(John,y)$\vee$Father(David,y)

（1）与（3）归结 $\sigma=\{David/z,John/x\}$

（6）$\sim$Brother(John,Peter)$\vee$ANSWER(David)

（4）与（5）归结$\sigma=\{David/u,Peter/y\}$

（7）ANSWER(David)　　　（2）与（6）归结

第五步：得到了归结式ANSWER(David)，答案即在其中，所以u=David。即Peter的父亲是David。

思考题

破案问题：在一栋房子里发生了一件神秘的谋杀案，现在可以肯定以下几点事实：

(1)在这栋房子里仅住有A,B,C三人；

(2)是住在这栋房子里的人杀了A；

(3)谋杀者非常恨受害者； (4)A所恨的人，C一定不恨；

(5)除了B以外，A恨所有的人； (6)B恨所有不比A富有的人；

(7)A所恨的人，B也恨； (8)没有一个人恨所有的人；

(9)杀人嫌疑犯一定不会比受害者富有。

为了推理需要，增加如下常识：(10)A不等于B。

问：谋杀者是谁？