数学思维与人工智能

第四章：确定性推理方法

中国矿业大学数学学院

祁永强

15862179376

qiyongqiang3@164.com

MATLAB

- 前面讨论了把知识用某种模式表示出来存储到计算机中去。但是，为使计算机具有智能，还必须使它具有思维能力。推理是求解问题的一种重要方法。因此，推理方法成为人工智能的一个重要研究课题。
- 下面首先讨论关于推理的基本概念，然后着重介绍鲁宾逊归结原理及其在机器定理证明和问题求解中的应用。鲁宾逊归结原理使定理证明能够在计算机上实现。

MATLAB

知识 → 智能！
推理 → 智能！

- 推理
  - 经典逻辑推理（确定性推理）
    - 自然演绎推理
    - 归结演绎推理
    - 与/或形演绎推理
  - 不确定性推理

MATLAB

4.1.2 推理方式及其分类

1. 演绎推理、归纳推理、默认推理

（1）演绎推理 (deductive reasoning) : 一般 → 个别

- 三段论式（三段论法）

① 足球运动员的身体都是强壮的；（大前提）

② 高波是一名足球运动员；（小前提）

---

③ 所以，高波的身体是强壮的。（结 论）

MATLAB

4.1.2 推理方式及其分类

1. 演绎推理、归纳推理、默认推理

（2）归纳推理 (inductive reasoning): 个别 → 一般

- 完全归纳推理（必然性推理）
- 不完全归纳推理（非必然性推理）

检查全部产品合格 —完全归纳推理→ 该厂产品合格

检查全部样品合格 —不完全归纳推理→ 该厂产品合格

MATLAB

4.1.2 推理方式及其分类

1. 演绎推理、归纳推理、默认推理

（3）默认推理（default reasoning，缺省推理）

- 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。

A 成立

B 成立？ → 结 论

（默认*B*成立）

制造鸟笼

鸟会飞？ → 鸟笼要有盖子

（默认成立）

MATLAB

4.1.2 推理方式及其分类

2. 确定性推理、不确定性推理

（1）**确定性推理**：推理时所用的知识与证据都是确定的，推出的结论也是确定的，其真值或者为真或者为假。

（2）**不确定性推理**：推理时所用的知识与证据不都是确定的，推出的结论也是不确定的。

- 不确定性推理
  - 似然推理（概率论）
  - 近似推理或模糊推理（模糊逻辑）

4.1.2 推理方式及其分类

4. 单调推理、非单调推理

（1）单调推理：随着推理向前推进及新知识的加入，推出的结论越来越接近最终目标。

基于经典逻辑的演绎推理

（2）非单调推理：由于新知识的加入，不仅没有加强已推出的结论，反而要否定它，使推理退回到前面的某一步，重新开始。

默认推理是非单调推理

MATLAB

4.1.2 推理方式及其分类

MATLAB

4. 启发式推理、非启发式推理

- **启发性知识**：与问题有关且能加快推理过程、提高搜索效率的知识。

- 目标：在脑膜炎、肺炎、流感中选择一个
- 产生式规则

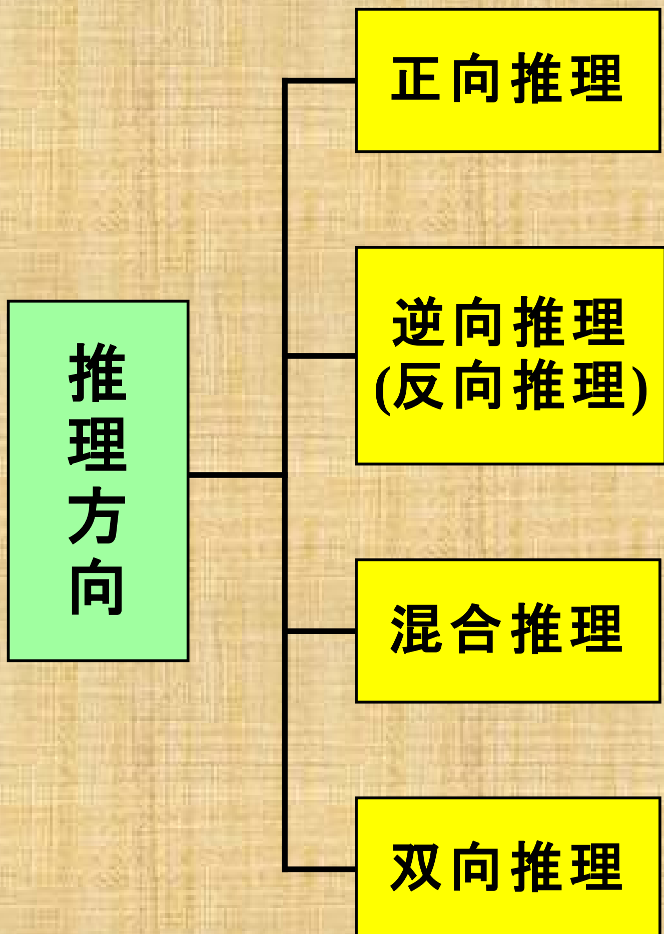
  r_1：脑膜炎

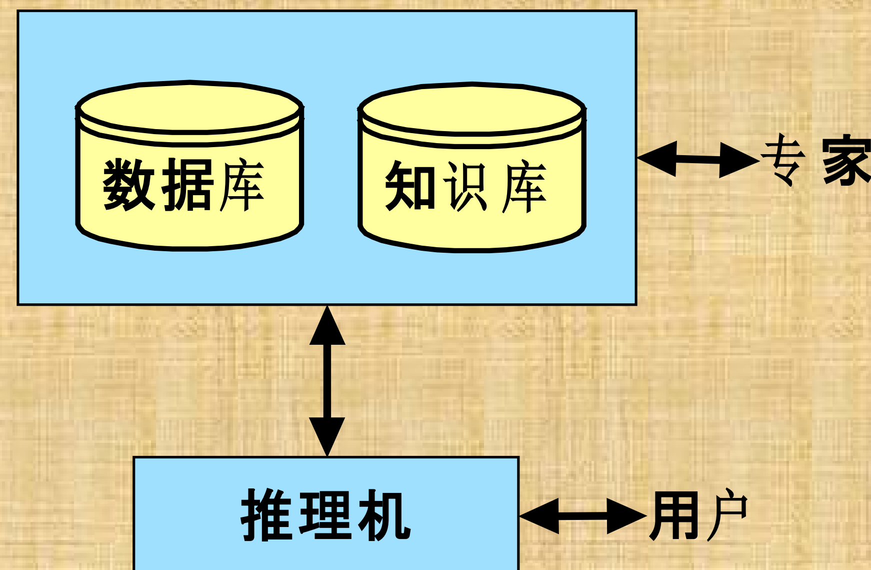
  r_2：肺 炎

  r_3：流 感

- 启发式知识："脑膜炎危险"、"目前正在盛行流感"。

4.1.3 推理的方向





MATLAB

4.1.3 推理的方向

1. 正向推理

- 正向推理（事实驱动推理）：已知事实 → 结论
- 基本思想

（1）从初始已知事实出发，在知识库*KB*中找出当前可适用的知识，构成可适用知识集*KS*。

（2）按某种冲突消解策略从*KS*中选出一条知识进行推理，并将推出的新事实加入到数据库*DB*中作为下一步推理的已知事实，再在*KB*中选取可适用知识构成*KS* 。

（3）重复（2），直到求得问题的解或*KB*中再无可适用的知识。

MATLAB

4.1.3 推理的方向

1. 正向推理

- 实现正向推理需要解决的问题：
  - 确定匹配（知识与已知事实）的方法。
  - 按什么策略搜索知识库。
  - 冲突消解策略。
- 正向推理简单，易实现，但目的性不强，效率低。

MATLAB

4.1.3 推理的方向

2. 逆向推理

- 逆向推理（目标驱动推理）：以某个假设目标作为出发点。
- 基本思想：
  - 选定一个假设目标。
  - 寻找支持该假设的证据，若所需的证据都能找到，则原假设成立；若无论如何都找不到所需要的证据，说明原假设不成立的；为此需要另作新的假设。
- 主要优点：不必使用与目标无关的知识，目的性强，同时它还有利于向用户提供解释。
- 主要缺点：起始目标的选择有盲目性。

MATLAB

4.1.3 推理的方向

2. 逆向推理

- 逆向推理需要解决的问题：
  - 如何判断一个假设是否是证据？
  - 当导出假设的知识有多条时，如何确定先选哪一条？
  - 一条知识的运用条件一般都有多个，当其中的一个经验证成立后，如何自动地换为对另一个的验证？
  - ……..
- **逆向推理：** 目的性强，利于向用户提供解释，但选择初始目标时具有盲目性，比正向推理复杂。

MATLAB

4.1.3 推理的方向

4. 混合推理

- 正向推理：盲目、效率低。
- 逆向推理：若提出的假设目标不符合实际，会降低效率。
- 正反向混合推理：

（1）先正向后逆向：先进行正向推理，帮助选择某个目标，即从已知事实演绎出部分结果，然后再用逆向推理证实该目标或提高其可信度；

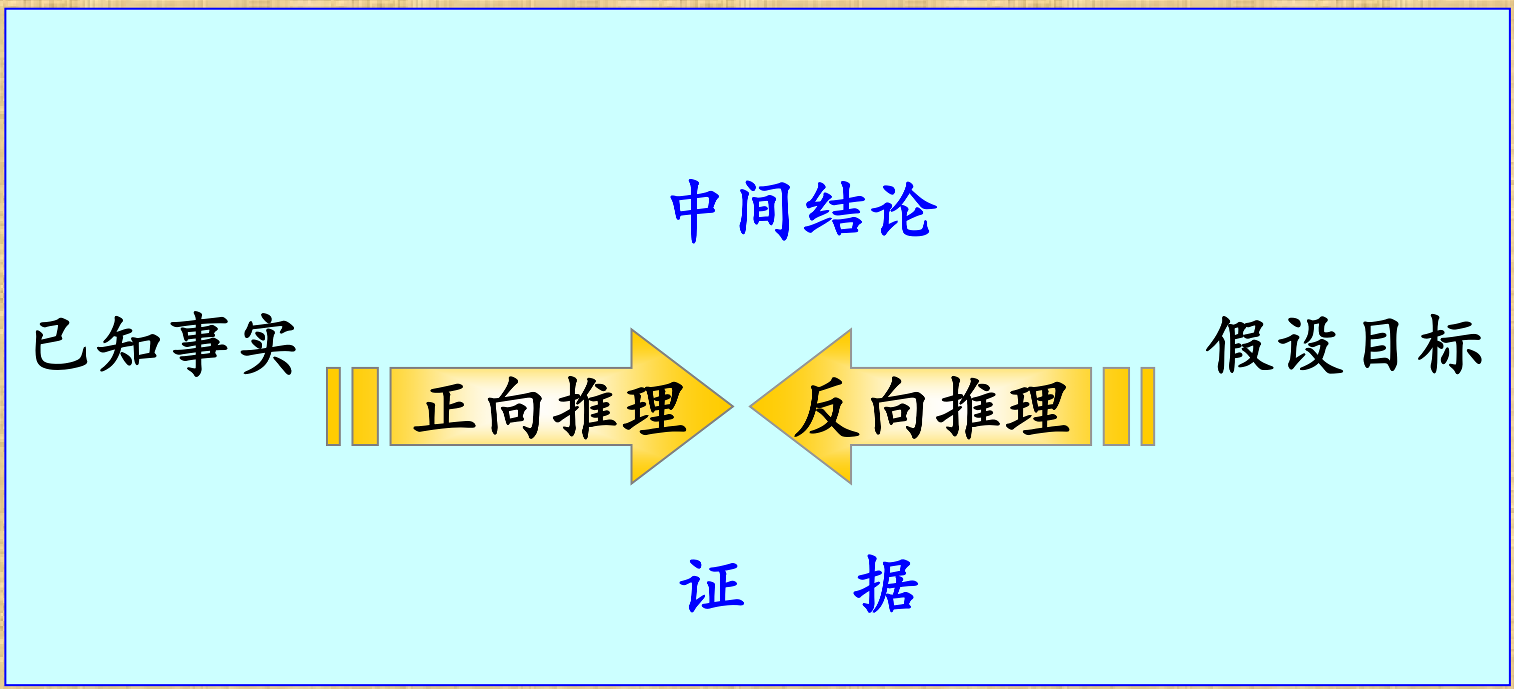
（2）先逆向后正向：先假设一个目标进行逆向推理，然后再利用逆向推理中得到的信息进行正向推理，以推出更多结论。

MATLAB

4.1.3 推理的方向

4. 双向推理

- **双向推理：** 正向推理与逆向推理同时进行，且在推理过程中的某一步骤上“碰头”的一种推理。



MATLAB

4.1.4 冲突消解策略

- 已知事实与知识的三种匹配情况:

（1）恰好匹配成功（一对一）；

（2）不能匹配成功;

（3）多种匹配成功（一对多、多对一、多对多）

MATLAB

4.1.4 冲突消解策略

- 多种冲突消解策略：

（1）按针对性排序

（2）按已知事实的新鲜性排序

（3）按匹配度排序

（4）按条件个数排序

r1: IF A1 AND A2
THEN H1
r2: IF A1 AND A2 AND A3 AND
A4 THEN H2

MATLAB

4.2 自然演绎推理

- 自然演绎推理：从一组已知为真的事实出发，运用经典逻辑的推理规则推出结论的过程。
- 推理规则：*P*规则、*T*规则、假言推理、拒取式推理

■ 假言推理： $P,\ P\rightarrow Q \ \Rightarrow\ Q$

■ "如果*x*是金属，则*x*能导电"，"铜是金属" 推出 铜能导电

拒取式推理： $P\rightarrow Q,\ \neg Q \ \Rightarrow\ \neg P$

■ "如果下雨，则地下就湿"，"地上不湿" 推出 "没有下雨"

MATLAB

4.2 自然演绎推理

错误1——否定前件： $P\rightarrow Q,\quad \neg P \quad \cancel{\Rightarrow} \quad \neg Q$

（1） 如果下雨，则地上是湿的（$P\rightarrow Q$）；

（2）没有下雨（$\neg P$）；

（3）所以，地上不湿（$\neg Q$）。

错误2——肯定后件： $P\rightarrow Q,\quad Q \quad \cancel{\Rightarrow} \quad P$

（1）如果行星系统是以太阳为中心的，则金星会显示出位相变化（$P\rightarrow Q$）；

（2）金星显示出位相变化（Q）；

（3） 所以，行星系统是以太阳为中心（P）。

MATLAB

4.2 自然演绎推理

- 例4.1 已知事实：

  （1）凡是容易的课程小王(Wang)都喜欢；

  （2）C 班的课程都是容易的；

  （3）ds 是 C 班的一门课程。

- 求证：小王喜欢 ds 这门课程。

MATLAB

4.2 自然演绎推理

- 证明：
- 定义谓词：

  $EASY(x)$：x 是容易的

  $LIKE(y, x)$：y 喜欢 x

  $C(x)$：x 是 C 班的一门课程

- 已知事实和结论用谓词公式表示：

$(\forall x)(EASY(x) \rightarrow LIKE(Wang, x))$

$(\forall x)(C(x) \rightarrow EASY(x))$

$C(ds)$

$LIKE(Wang, ds)$

MATLAB

4.2 自然演绎推理

- 应用推理规则进行推理：

$(\forall x)(EASY(x) \rightarrow LIKE(Wang, x))$

$EASY(z) \rightarrow LIKE(Wang, z)$　　全称固化

$(\forall x)(C(x) \rightarrow EASY(x))$

$C(y) \rightarrow EASY(y)$　　全称固化

所以 $C(ds), C(y) \rightarrow EASY(y)$

$\Longrightarrow EASY(ds)$　　P规则及假言推理

所以 $EASY(ds), EASY(z) \rightarrow LIKE(Wang, z)$

MATLAB

$\Rightarrow LIKE(Wang, ds)$　　T规则及假言推理

4.2 自然演绎推理

优点：

- 表达定理证明过程自然，易理解。
- 拥有丰富的推理规则，推理过程灵活。
- 便于嵌入领域启发式知识。

缺点： 易产生组合爆炸，得到的中间结论一般呈指数形式递增。

MATLAB

4.3 谓词公式化为子句集的方法

- 反证法：$P \Rightarrow Q$，当且仅当 $P \wedge \neg Q \Leftrightarrow F$，即 **Q**为 **P** 的逻辑结论，当且仅当$P \wedge \neg Q$ 是不可满足的。

- 定理：**Q** 为 P_1，P_2，…，P_n 的逻辑结论，当且仅当 $(P_1 \wedge P_2 \wedge \cdots \wedge P_n) \wedge \neg Q$ 是不可满足的。

- 思路：定理 $P \Rightarrow Q$ ⟶ $P \wedge \neg Q$ 不可满足 ⟶ 子句集不可满足 ⟶ 海伯伦定理

  鲁宾逊归结原理

MATLAB

4.3 谓词公式化为子句集的方法

- 原子（atom）谓词公式：一个不能再分解的命题。
- 文字（literal）：原子谓词公式及其否定。

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

- P：正文字，$\neg P$：负文字。
- 子句（clause）：任何文字的析取式。任何文字本身也都是子句。
- 空子句（NIL）：不包含任何文字的子句。
- 子句集：由子句构成的集合。

空子句是永假的，不可满足的。

MATLAB

4.3 谓词公式化为子句集的方法

- 例4.2 将下列谓词公式化为子句集。

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

- 解：（1）消去谓词公式中的“ $\rightarrow$ ”和“ $\leftrightarrow$ ”符号

$$P \rightarrow Q \Leftrightarrow \neg P \vee Q, \quad P \leftrightarrow Q \Leftrightarrow (P \wedge Q) \vee (\neg P \wedge \neg Q)$$

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

（2）把否定符号 $\neg$ 移到紧靠谓词的位置上

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

MATLAB

4.3 谓词公式化为子句集的方法

（3）变量标准化

双重否定律 $\neg(\neg P) \Leftrightarrow P$

德.摩根律 $\neg(P \wedge Q) \Leftrightarrow \neg P \vee \neg Q, \quad \neg(P \vee Q) \Leftrightarrow \neg P \wedge \neg Q$

量词转换律 $\neg(\exists x)P \Leftrightarrow (\forall x)\neg P, \quad \neg(\forall x)P \Leftrightarrow (\exists x)\neg P$

$$(\exists x)P(x) \equiv (\exists y)P(y), \quad (\forall x)P(x) \equiv (\forall y)P(y)$$

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

MATLAB

4.3 谓词公式化为子句集的方法

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

（4）消去存在量词

a. 存在量词不出现在全称量词的辖域内。

b. 存在量词出现在一个或者多个全称量词的辖域内。

$$\overset{\substack{y=f(x),\\ z=g(x)}}{\Longleftrightarrow} (\forall x)(\neg P(x, f(x)) \vee (Q(x, g(x)) \wedge \neg R(x, g(x))))$$

MATLAB

4.3 谓词公式化为子句集的方法

对于一般情况

$$(\forall x_1)((\forall x_2)\cdots((\forall x_n)((\exists y)P(x_1,x_2,\cdots,x_n,y)))\ldots)$$

存在量词y的Skolem函数为$y=f(x_1,x_2,\cdots,x_n)$

Skolem化：用Skolem函数代替每个存在量词量化的变量的过程。

（5）化为前束形

前束形=（前缀）{母式}

（前缀）：全称量词串。

{母式}：不含量词的谓词公式。

MATLAB

4.3 谓词公式化为子句集的方法

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

（6）化为 **Skolem 标准形**

Skolem 标准形：$(\forall x_1)(\forall x_2)\cdots(\forall x_n)M$

M：子句的合取式，称为Skolem标准形的母式。

$$\Longleftrightarrow (\forall x)((\neg P(x, f(x)) \vee Q(x, g(x))) \wedge (\neg P(x, f(x)) \vee \neg R(x, g(x))))$$

（7）略去全称量词

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

（8）消去合取词

$$\Longleftrightarrow \{\neg P(x, f(x)) \vee Q(x, g(x)), \quad \neg P(x, f(x)) \vee \neg R(x, g(x))\}$$

MATLAB **（9）子句变量标准化**

$$\Longleftrightarrow \{\neg P(x, f(x)) \vee Q(x, g(x)), \quad \neg P(y, f(y)) \vee \neg R(y, g(y))\}$$

4.3 谓词公式化为子句集的方法

例4.3 将下列谓词公式化为子句集。

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

- **（1）消去蕴含符号**

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

- **（2）把否定符号移到每个谓词前面**

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

- **（3）变量标准化**

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

- **（4）消去存在量词，设*y*的函数是*f*(*x*)，则**

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

MATLAB

例4.3 将下列谓词公式化为子句集。（续）

（5）化为前束形

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

（6）化为标准形

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

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

（7）略去全称量词

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

（8）消去合取词，把母式用子句集表示

$$\{Q(x),\ P(x) \vee S(x, f(x)),\ P(w) \vee B(w)\}$$

（9）子句变量标准化

MATLAB $\{Q(x),\ P(y) \vee S(y, f(y)),\ P(w) \vee B(w)\}$

4.3 谓词公式化为子句集的方法

- **例4.4 将下列谓词公式化为不含存在量词的前束形。**

$$(\exists x)(\forall y)((\forall z)(P(z) \wedge \neg Q(x,z)) \rightarrow R(x,y,f(a)))$$

- **（1）消去存在量词**

$$(\forall y)((\forall z)(P(z) \wedge \neg Q(b,z)) \rightarrow R(b,y,f(a)))$$

- **（2）消去蕴含符号**

$$(\forall y)(\neg(\forall z)(P(z) \wedge \neg Q(b,z)) \vee R(b,y,f(a)))$$

$$(\forall y)((\exists z)(\neg P(z) \vee Q(b,z)) \vee R(b,y,f(a)))$$

- **（3）设z的函数是g(y)，则**

$$(\forall y)(\neg P(g(y)) \vee Q(b,g(y)) \vee R(b,y,f(a)))$$

MATLAB

4.3 谓词公式化为子句集的方法

谓词公式 不可满足性 ⟷ 子句集 不可满足性 ?

定理 4.1:

谓词公式不可满足的充要条件是其子句集不可满足。

◆ 子句集中子句之间是合取关系，只要有一个子句不可满足，则子句集就不可满足。

◆ 鲁宾逊归结原理（消解原理）的基本思想：

- 检查子句集 S 中是否包含空子句，若包含，则 S 不可满足。
- 若不包含，在 S 中选择合适的子句进行归结，一旦归结出空子句，就说明 S 是不可满足的。

MATLAB

1. 命题逻辑中的归结原理（基子句的归结）

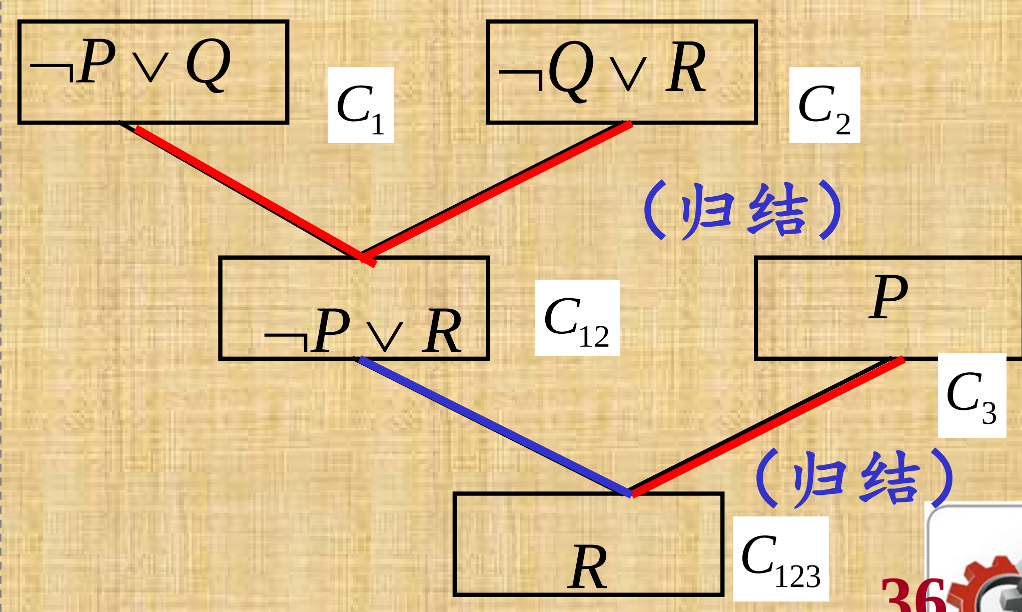
定义4.1（归结）：设C_1与C_2是子句集中的任意两个子句，如果 C_1中的文字L_1与 C_2中的文字L_2互补，那么从C_1和C_2中分别消去L_1和L_2，并将二个子句中余下的部分析取，构成一个新子句C_{12}。

C_{12}：C_1和C_2的归结式

C_1、C_2：C_{12}的亲本子句

例，设$C_1=\neg P\vee Q$，

$C_2=\neg Q\vee R$，$C_3=P$



MATLAB

4.4 鲁宾逊归结原理

◆ 定理4.2：归结式C_{12}是其亲本子句C_1与C_2的逻辑结论。即如果C_1与C_2为真，则C_{12}为真。

◆ 推论1：设C_1与C_2是子句集S中的两个子句，C_{12}是它们的归结式，若用C_{12}代替C_1与C_2后得到新子句集S_1，则由S_1不可满足性可推出原子句集S的不可满足性，即：

S_1 的不可满足性 $\Rightarrow$ S 的不可满足性

◆ 推论2：设C_1与C_2是子句集S中的两个子句，C_{12}是它们的归结式，若C_{12}加入原子句集S，得到新子句集S_1，则S与S_1在不可满足的意义上是等价的，即：

S_1的不可满足性 $\Leftrightarrow$ S的不可满足性

MATLAB

4.4 鲁宾逊归结原理

2. 谓词逻辑中的归结原理（含有变量的子句的归结）

?

$C_1 = P(x) \vee Q(x)$

$C_2 = \neg P(a) \vee R(y)$

最一般合一

$\sigma = \{a/x\}$

$C_1\sigma = P(a) \vee Q(a)$

$C_2\sigma = \neg P(a) \vee R(y)$

$C_{12} = Q(a) \vee R(y)$

定义 4.2： 设是C_1与C_2两个没有相同变元的子句，L_1和L_2分别是C_1和C_2中的文字，若σ是L_1和$\neg L_2$的**最一般合一**，则称

$$C_{12} = (C_1\sigma - \{L_1\sigma\}) \vee (C_2\sigma - \{L_2\sigma\})$$

为C_1和C_2的二元归结式。

MATLAB

4.4 鲁宾逊归结原理

- 例4.5 设：

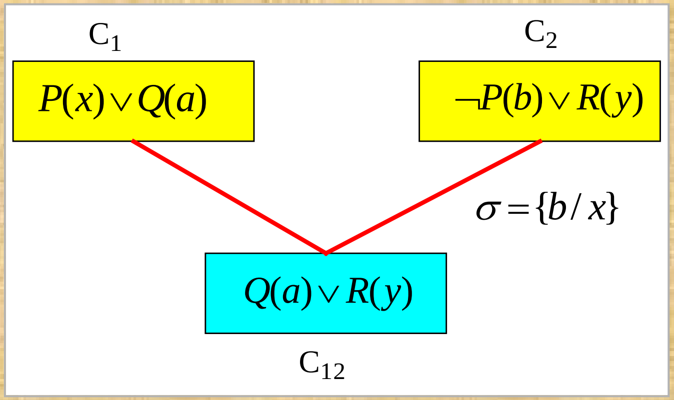
$$C_1 = P(x) \vee Q(a), \quad C_2 = \neg P(b) \vee R(x)$$

求其二元归结式。

- 解：令 $C_2 = \neg P(b) \vee R(y)$

选 $L_1 = P(x), L_2 = \neg P(b)$ 则 $\sigma = \{b/x\}$

得：



$$C_{12} = (\{P(b), Q(a)\} - \{P(b)\}) \vee (\{\neg P(b), R(y)\} - \{\neg P(b)\})$$

$$= \{Q(a), R(y)\} = Q(a) \vee R(y)$$

MATLAB

4.4 鲁宾逊归结原理

- 例4. 6 设：

$$C_1 = P(x) \vee P(f(a)) \vee Q(x), \quad C_2 = \neg P(y) \vee R(b)$$

求其二元归结式。

- **解：** $\sigma = \{f(a)/x\}$ $\qquad C_1\sigma = P(f(a)) \vee Q(f(a)),$

选 $L_1 = P(f(a)), L_2 = \neg P(y)$ $\qquad \sigma = \{f(a)/y\}$

则得： $C_{12} = R(b) \vee Q(f(a))$

MATLAB

4.4 鲁宾逊归结原理

- 对于谓词逻辑，归结式是其亲本子句的逻辑结论。
- 对于一阶谓词逻辑，即若子句集是不可满足的，则必存在一个从该子句集到空子句的归结演绎；若从子句集存在一个到空子句的演绎，则该子句集是不可满足的。
- 如果没有归结出空子句，则既不能说 S 不可满足，也不能说 S 是可满足的。

MATLAB

4.5 归结反演

- 应用归结原理证明定理的过程称为归结反演。
- 用归结反演证明的步骤是：

（1）将已知前提表示为谓词公式F。

（2）将待证明的结论表示为谓词公式Q，并否定得到$\neg Q$。

（3）把谓词公式集$\{F, \neg Q\}$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，并把每次归结得到的归结式都并入到S中。如此反复进行，若出现了空子句，则停止归结，此时就证明了Q为真。

4.5 归结反演

• 例4.7 某公司招聘工作人员，*A*，*B*，*C* 三人应试，经面试后公司表示如下想法：

（1） 三人中至少录取一人。

（2） 如果录取 *A* 而不录取 *B*，则一定录取 *C*。

（3） 如果录取 *B*，则一定录取 *C*。

■ 求证：公司一定录取 *C*。

MATLAB

4.5 归结反演

- 证明：公司的想法用谓词公式表示： $P(x)$：录取x

（1）$P(A) \vee P(B) \vee P(C)$

（2）$P(A) \wedge \neg P(B) \to P(C)$

（3）$P(B) \to P(C)$

- 把要求证的结论用谓词公式表示出来并否定，得：

（4）$\neg P(C)$

- 把上述公式化成子句集：

（1）$P(A) \vee P(B) \vee P(C)$

（2）$\neg P(A) \vee P(B) \vee P(C)$

（3）$\neg P(B) \vee P(C)$

（4）$\neg P(C)$

MATLAB

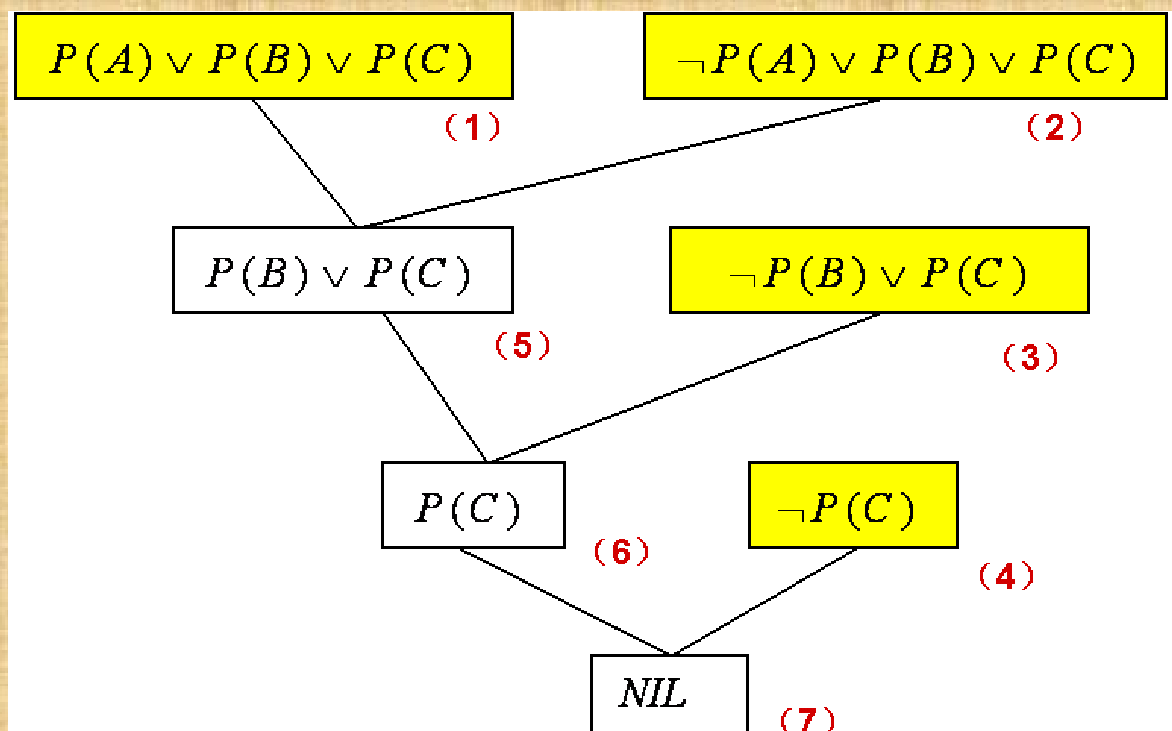
4.5 归结反演

- 应用归结原理进行归结：

（5） $P(B) \vee P(C)$ （1）与（2）归结

（6） $P(C)$ （3）与（5）归结

（7） NIL （4）与（6）归结



MATLAB

4.5 归结反演

- 例4.8 已知：

  规则1：任何人的兄弟不是女性；

  规则2：任何人的姐妹必是女性。

  事 实：*Mary* 是 *Bill* 的姐妹。

  求证： *Mary* 不是 *Tom* 的兄弟。

- 证明：定义谓词

  brother (*x, y*) ： *x* 是 *y* 的兄弟

  sister (*x, y*) ： *x* 是 *y* 的姐妹

  woman (*x*) ： *x* 是女性

MATLAB

4.5 归结反演

- 证明：将规则与事实用谓词公式表示：

（1）$(\forall x)(\forall y)(brother(x,y) \rightarrow \neg woman(x))$

（2）$(\forall x)(\forall y)(sister(x,y) \rightarrow woman(x))$

（3）$sister(Mary, Bill)$

- 把要求证的结论用谓词公式表示出来并否定，得：

（4）$brother(Mary, Tom)$

- 把上述公式化成子句集：

$C_1 = \neg brother(x,y) \vee \neg woman(x)$

$C_2 = \neg sister(x,y) \vee woman(x)$

$C_3 = sister(Mary, Bill)$

$C_4 = brother(Mary, Tom)$

- 将子句集进行归结：

$C_{23} = woman(Mary)$

$C_{123} = \neg brother(Mary, y)$

$C_{1234} = NIL$

MATLAB

4.6 应用归结原理求解问题

- 应用归结原理求解问题的步骤：

（1）已知前提 *F* 用谓词公式表示，并化为子句集 *S*；

（2）把待求解的问题 *Q* 用谓词公式表示，并否定 *Q*，再与 *ANSWER* 构成析取式（$\neg Q \vee ANSWER$）；

（3）把（$\neg Q \vee ANSWER$）化为子句集，并入到子句集 *S* 中，得到子句集 S'；

（4）对 S' 应用归结原理进行归结；

（5）若得到归结式 *ANSWER*，则答案就在 *ANSWER* 中。

MATLAB

4.6 应用归结原理求解问题

- 例4.9 已知：

F_1：王（Wang）先生是小李（Li）的老师。

F_2：小李 与小张（Zhang）是同班同学。

F_3：如果x与y是同班同学，则x的老师也是y的老师。

求：小张的老师是谁？

MATLAB

4.6 应用归结原理求解问题

- **解：定义谓词：**

  $T(x,y)$：x 是 y 的老师。

  $C(x,y)$：x 与 y 是同班同学。

- **把已知前提表示成谓词公式：**

  F_1： $T(Wang, Li)$

  F_2： $C(Li, Zhang)$

  F_3： $(\forall x)(\forall y)(\forall z)(C(x,y) \wedge T(z,x) \rightarrow T(z,y))$

- **把目标表示成谓词公式，并把它否定后与 *ANSWER* 析取：**

$$G: \neg (\exists x) T(x, Zhang) \vee ANSWER(x)$$

MATLAB

4.6 应用归结原理求解问题

- 把上述公式化为子句集：

（1） $T(Wang, Li)$

（2） $C(Li, Zhang)$

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

（4） $\neg T(u, Zhang) \vee ANSWER(u)$

- 应用归结原理进行归结：

（5） $\neg C(Li, y) \vee T(Wang, y)$ （1）与（3）归结

（6） $\neg C(Li, Zhang) \vee ANSWER(Wang)$ （4）与（5）归结

（7） $ANSWER(Wang)$ （2）与（6）归结