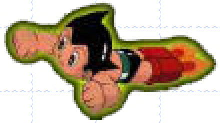


人工智能
Artificial Intelligence

主讲：相明
西安交通大学电信学院计算机系
E_mail：mxiang@mail.xjtu.edu.cn

第3章　确定性推理

3.1 概述

3.1.1 推理方式与分类

- 所谓推理就是按某种策略由已知判断推出另一个判断的思维过程。

- 在人工智能中，推理是由程序实现的，称为推理机。

1. 演绎推理、归纳推理、默认推理

演绎推理：从一般到特殊。例如三段论。

归纳推理：从个体到一般。

默认推理：缺省推理，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。

2. 确定性、不确定性推理
3. 单调推理、非单调推理

推出的结论是否单调增加（演绎推理，缺省推理）

4. 启发式、非启发式推理

所谓启发性知识是指与问题有关且能加快推理进程、求得问题最优解的知识。

5. 基于知识的推理（专家系统）、统计推理、直觉推理（常识性推理）

3.1.2 推理的控制策略

- 推理的控制策略主要包括：推理方向、搜索策略、冲突消解策略、求解策略及限制策略。

1. 正向推理（数据驱动推理）

- 正向推理的基本思想是：从用户提供的初始已知事实出发，在知识库KB中找出当前可适用的知识，构成可适用的知识集KS，然后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。在此之后，再在知识库中选取可适用的知识进行推理。如此重复进行这一过程，直到求得所要求的解。

2 逆向推理

- 逆向推理的基本思想是：首先选定一个假设目标，然后寻找支持该假设的证据，若所需的证据都能找到，则说明原假设是成立的；推理完成。若找不到所需要的证据，则说明原假设不成立，此时需要另作新的假设。

动物识别的例子

- 已知事实：一动物{有毛，吃草，黑条纹}
  - R1：动物有毛 → 哺乳类
  - R2：动物产奶 → 哺乳类
  - R3：哺乳类 ∧ 吃肉 → 食肉类
  - R4：哺乳类 ∧ 吃草 → 有蹄类
  - R5：食肉类 ∧ 黄褐色 ∧ 有斑点→ 猎狗
  - R6：食肉类 ∧ 黄褐色 ∧ 黑条纹→ 虎
  - R7：有蹄类 ∧ 长脖 → 长颈鹿
  - R8：有蹄类 ∧ 黑条纹 → 斑马

3. 混合推理

- 先正向推理后逆向推理
- 先逆向推理后正向推理

4. 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上“碰头”。

5. 求解策略

- 只求一个解，还是求所有解以及最优解。

6. 限制策略

- 限制搜索的深度、宽度、时间、空间等等。

3.1.3 知识匹配

- 所谓模式匹配(知识匹配)是指对两个知识模式(例如两个谓词公式、框架片断、语义网络片断)进行比较，检查这两个知识模式是否完全一致或者近似一致。
- 模式匹配可分为确定性匹配与不确定性匹配。
- 确定性匹配是指两个知识模式完全一致，或者经过变量代换后变得完全一致。

  知识：IF father(x,y) and man(y) THEN son(y,x)

  事实：father(李四，李小四) and man(李小四)
- 不确定性匹配是指两个知识模式不完全一致，但是它们的相似程度又在规定的限度内。

变量代换

定义3-1 代换是一个形如

$$\{t_1/x_1,t_2/x_2,\ldots,t_n/x_n\}$$

的有限集合。

其中$t_1,t_2,\ldots,t_n$是项（常量、变量、函数）；
$x_1,x_2,\ldots,x_n$是（某一公式中）互不相同的变元;
t_i/x_i表示用t_i代换x_i ;

不允许t_i与x_i相同，也不允许变元x_i循环地出现在另一个t_j中。例如：

$\{a/x,f(b)/y,w/z\}$是一个代换

$\{g(y)/x,f(x)/y\}$不是代换

令$\theta = \{t_1/x_1, t_2/x_2, \ldots, t_n/x_n\}$为一个代换，F为表达式，则$F\theta$表示对F用$t_i$代换$x_i$后得到的表达式。$F\theta$称为F的特例。

规则： IF father(x,y) and man(y) THEN son(y,x)

事实： father(李四，李小四) and man(李小四)

$F = father(x,y) \wedge man(y)$

$\theta = \{$李四$/X,$李小四$/Y\}$

$F\theta = father($李四，李小四$) \wedge man($李小四$)$

结论： son(李小四,李四)

代换的复合

定义3-2 设

$$\theta=\{t_1/x_1,t_2/x_2,\ldots,t_n/x_n\}$$

$$\lambda=\{u_1/y_1,u_2/y_2,\ldots,u_m/y_m\}$$

是两个代换，则这两个代换的复合也是一个代换，它是从

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

中删去如下两种元素：

$t_i\lambda/x_i$ 当$t_i\lambda=x_i$

u_i/y_i 当$y_i\in\{x_1,x_2,\ldots,x_n\}$

后剩下的元素所构成的集合，记为$\theta^\circ\lambda$。

- $t_i\lambda$表示对t_i运用λ进行代换。
- $\theta^\circ\lambda$就是对一个公式F先运用θ进行代换，然后再运用λ进行代换：$F(\theta^\circ\lambda)=(F\theta)\lambda$

代换的例子

例如，设有代换

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

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

则

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

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

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

公式集的合一

定义3-3 设有公式集$F=\{F_1,F_2,\ldots,F_n\}$，若存在一个代换λ使得

$F_1\lambda=F_2\lambda=\ldots=F_n\lambda$

则称λ为公式集F的一个合一，且称$F_1,F_2,\ldots,F_n$是可合一的。

- 例如，设有公式集

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

则下式是它的一个合一：

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

- 一个公式集的合一一般不唯一。

最一般的合一

定义3-4 设σ是公式集F的一个合一，如果对任一个合一θ都存在一个代换λ，使得$\theta=\sigma\circ\lambda$
则称σ是一个最一般的合一。

（1）代换过程是一个用项代替变元的过程，因此是一个从一般到特殊的过程。

（2） 最一般合一是唯一的。

求取最一般合一

- 差异集：两个公式中相同位置处不同符号的集合。

例如：F1:P(x,y,z), F2:P(x,f(a),h(b))

则D1={y,f(a)}, D2={z,h(b)}

求取最一般合一的算法：

1. 令$k=0, F_k=F, \sigma_k=\varepsilon$。 ε是空代换。
2. 若F_k只含一个表达式，则算法停止，σ_k就是最一般合一。
3. 找出F_k的差异集D_k。
4. 若D_k中存在元素x_k和t_k，其中x_k是变元，t_k是项，且x_k不在t_k中出现，则置：

   $F_{k+1}=F_k\{t_k/x_k\}$

   $\sigma_{K+1}=\sigma_k{}^\circ\ \{t_k/x_k\}$

   $k=k+1$

   然后转(2)。若不存在这样的x_k和t_k则算法停止。
5. 算法终止，F的最一般合一不存在。

求取最一般合一的例子

例如，设 $F=\{P(a,x,f(g(y))),P(z,f(z),f(u))\}$
求其最一般合一。

令$F_0=F$, $\sigma_0=\varepsilon$。 F_0中有两个表达式，所以σ_0不是最一般合一。

差异集：$D_0=\{a,z\}$。代换： $\{a/z\}$

$F_1= F_0\{a/z\}=\{P(a,x,f(g(y))),P(a,f(a),f(u))\}$ 。

$\sigma_1=\sigma_0 \circ \{a/z\}=\{a/z\}$

差异集： $D_1=\{x,f(a)\}$ 。代换： $\{f(a)/x\}$

$F_2=F_1\{f(a)/x\}=\{P(a,f(a),f(g(y))),P(a,f(a),f(u))\}$ 。

$\sigma_2=\sigma_1 \circ \{f(a)/x\}=\{a/z,f(a)/x\}$

差异集： $D_2=\{g(y),u\}$ 。代换： $\{g(y)/u\}$

$F_3=F_2\{g(y)/u\}=\{P(a,f(a),f(g(y))),\underline{P(a,f(a),f(g(y)))}\}$ 。

$\sigma_3=\sigma_2 \circ \{g(y)/u\}=\{a/z,f(a)/x,g(y)/u\}$ $F_3=F\sigma_3$

3.2 自然演绎推理

- 从一组已知为真的事实出发，直接运用经典逻辑的推理规则推出结论的过程，称为自然演绎推理。其中，基本的推理规则是P规则、T规则、假言推理、拒取式推理等。
- 假言推理的一般形式

$$P, P \rightarrow Q \Rightarrow Q$$

- 拒取式推理的一般形式

$$P \rightarrow Q, \neg Q \Rightarrow \neg P$$

P规则：在推理的任何步骤都可以引入前提。

T规则：推理时，如果前面步骤中有一个或者多个公式永真蕴含公式S，则可把S引入推理过程中。

3.3 归结演绎推理

- 定理证明即证明P→Q(¬P∨Q)的永真性。根据反证法，只要证明其否定(P∧¬Q) 不可满足性即可。
- 海伯伦(Herbrand)定理为自动定理证明奠定了理论基础；鲁滨逊(Robinson)提出的归结原理使机器定理证明成为现实。

3.3.1 海伯论理论

- 在谓词逻辑中，把原子谓词公式及其否定统称为文字。如：$P(x)$， $\neg P(x,f(x))$， $Q(x,g(x))$

定义3-5： 任何文字的析取式称为子句。

例如： $P(x) \lor Q(x)$,

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

定义3-6： 不包含任何文字的子句称为空子句。

子句集

(1) 合取范式：$C_1 \wedge C_2 \wedge C_3 \ldots \wedge C_n$

(2) 子句集：　$S = \{C_1, C_2, C_3 \ldots, C_n\}$

(3)任何谓词公式F都可通过等价关系及推理规则化为相应的子句集S。

把谓词公式化成子句集的步骤(1)

1. 利用等价关系消去“→”和“↔”

   例如公式

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

   可等价变换成 $(\forall x)(\neg(\forall y)P(x,y)\vee\neg(\forall y)(Q(x,y)\rightarrow R(x,y)))$

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

2. 利用等价关系把“¬”移到紧靠谓词的位置上

   上式经等价变换后 $(\forall x)((\exists y)\neg P(x,y)\vee(\exists y)(\neg(\neg Q(x,y)\vee R(x,y))))$

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

3. 重新命名变元，使不同量词约束的变元有不同的名字

   上式经变换后

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

把谓词公式化成子句集的步骤(2)

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

4. 消去存在量词

a.存在量词不出现在全称量词的辖域内，则只要用一个新的个体常量替换受该量词约束的变元。

b.存在量词位于一个或者多个全称量词的辖域内，此时要用Skolem函数$f(x_1,x_2,\ldots,x_n)$替换受该存在量词约束的变元。

上式中存在量词$(\exists y)$及$(\exists z)$都位于$(\forall x)$的辖域内，所以需要用Skolem函数替换，设替换y和z的Skolem函数分别是$f(x)$和$g(x)$，则替换后得到

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

5. 把全称量词全部移到公式的左边

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

6. 利用等价关系把公式化为Skolem标准形

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

Skolem标准形的一般形式是

$$(\forall x_1)(\forall x_2)\cdots(\forall x_n)M$$

其中，M是子句的合取式，称为Skolem标准形的母式。

上式化为Skolem标准形后得到

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

7. 消去全称量词

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

8. 消去合取词，就得到子句集

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

子句集的意义

- 子句集S的不可满足性：对于任意论域中的任意一个解释，S中的子句不能同时取得真值T。

- 定理3-1 设有谓词公式F，其子句集为S，则F不可满足的充要条件是S不可满足。

- 要证明$P \rightarrow Q$ (即$\neg P \vee Q$)永真，只需证明公式$F=(P \wedge \neg Q)$永假，即证明S不可满足。

Herbrand理论

- 为了判断子句集的不可满足性，需要对所有可能论域上的所有解释进行判定。只有当子句集对任何非空个体域上的任何一个解释都是不可满足的，才可断定该子句集是不可满足的。
- 海伯伦构造了一个特殊的论域(海伯伦域)，并证明只要对这个特殊域上的一切解释进行判定，就可知子句集是否不可满足。

3.3.2 鲁滨逊归结原理

- 鲁滨逊归结原理的基本思想：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，就在子句集中选择合适的子句进行归结，一旦通过归结能推出空子句，就说明子句集S是不可满足的。
- 子句集S的不可满足性：对于任意论域中的任意一个解释，S中的子句不能同时取得真值T。一旦S中包含空子句，则S必不可满足。

命题逻辑中的归结原理

定义3-9 若P是原子谓词公式，则称P与¬P为互补文字。在命题逻辑中，P为命题。

定义3-10 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，并将两个子句中余下的部分析取，构成一个新子句C_{12}，则称这一过程为归结。称C_{12}为C_1和C_2的归结式，C_1和C_2为C_{12}的亲本子句。

例：设

$$C_1=\neg P\vee Q,\ C_2=\neg Q\vee R,\ C_3=P$$

C_1与C_2归结得到：$C_{12}=\neg P\vee R$

C_{12}与C_3归结得到：$C_{123}=R$

定理3-4 C_{12}是其亲本子句C_1与C_2的逻辑结论。

证明：设

$C_1=L\vee C`_1$, $C_2=\neg L\vee C`_2$, 则$C_{12}=C`_1\vee C`_2$

$$\because\ C_1=C_1'\vee L\Leftrightarrow\neg C_1'\rightarrow L$$

$$C_2=\neg L\vee C_2'\Leftrightarrow L\rightarrow C_2'$$

$$\therefore\ C_1\wedge C_2=(\neg C_1'\rightarrow L)\wedge(L\rightarrow C_2')$$

由假言三段论

$$(\neg C_1'\rightarrow L)\wedge(L\rightarrow C_2')\Rightarrow\neg C_1'\rightarrow C_2'$$

$$\because\neg C_1'\rightarrow C_2'\Leftrightarrow C_1'\vee C_2'=C_{12}$$

$$\therefore C_1\wedge C_2\Rightarrow C_{12}$$

- 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和C_2后得到新子句集S_1，则由S_1的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=>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。若把C_{12}加入S中得到新子句集S_2，则S与S_2在不可满足的意义上是等价的，即

  S_2的不可满足性<=>S的不可满足性

- 为了要证明子句集S的不可满足性，只要对其中可进行归结的子句进行归结，并把归结式加入子句集S，或者用归结式替换它的亲本子句，然后对新子句集(S_1或者S_2)证明不可满足性就可以了。如果经过归结能得到空子句，则立即可得原子句集S是不可满足的结论。
- 在命题逻辑中，归结原理是完备的。即，若子句集不可满足，则必然存在一个从S到空子句的归结演绎。

谓词逻辑中的归结原理

- 在谓词逻辑中，由于子句中含有变元，所以不能像命题逻辑那样直接消去互补文字，而需要先用最一般合一对变元进行代换，然后才能进行归结。

例如, 设有两个子句

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

由于P(x)与P(a)不同，所以C_1与C_2不能直接进行归结。但是若用最一般合一

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

对两个子句分别进行代换：

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

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

就可对它们进行归结，得到归结式：

$$Q(a)\vee R(y)$$

二元归结式的定义

定义3-11 设C_1与C_2是两个没有相同变元的子句，L_1和$\neg L_2$分别是C_1和C_2中的文字。若σ是L_1和L_2的最一般合一，则称

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

为C_1和C_2的二元归结式，L_1和$\neg L_2$称为归结式上的文字。

例3-6 设

$$C_1=P(a)\vee\neg Q(x)\vee R(x),\quad C2=\neg P(y)\vee Q(b)$$

若选$L_1=P(a)$，$L_2=P(y)$，$\sigma=\{a/y\}$就是L_1与L_2的最一般合一。可得：

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

$$=\neg Q(x)\vee R(x)\vee Q(b)$$

- 若子句C含有可合一的文字，则在进行归结之前应先对这些文字进行合一。记其最一般的合一为σ，称Cσ为子句C的因子。若Cσ是一个单文字，则称它为C的单元因子。

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

$\sigma=\{f(a)/x\}$

$C1\sigma=P(f(a)) \vee Q(f(a))$

$C12 = Q(f(a)) \vee R(b)$

定义3-12 子句C_1和C_2的归结式是下列二元归结式之一：

1. C_1与C_2的二元归结式；
2. C_1与C_2的因子$C_2\sigma_2$的二元归结式；
3. C_1的因子$C_1\sigma_1$与C2的二元归结式；
4. C1的因子$C_1\sigma_1$与C2的因子$C_2\sigma_2$的二元归结式。

- 对于一阶谓词逻辑，归结原理也是完备的。即，若子句集S不可满足，则必然存在一个从S到空子句的归结演绎。

3.3.3 归结反演

- 如欲证明Q为$P_1,P_2,\ldots,P_n$的逻辑结论，只需证明

  $(P_1 \wedge P_2 \wedge \ldots \wedge P_n) \rightarrow Q$永真，即$\neg (P_1 \wedge P_2 \wedge \ldots \wedge P_n) \vee Q$永真，

  或证明 $(P_1 \wedge P_2 \wedge \ldots \wedge P_n) \wedge \neg Q$ 是不可满足的，或证明其子句集是不可满足的。而子句集的不可满足性可用归结原理来证明。应用归结原理证明定理的过程称为归结反演。
- 设F为已知前提的公式集，Q为目标公式(结论)，用归结反演证明Q为真的步骤是：

1. 否定Q，得到¬Q；　　$F=\{P_1,P_2,\ldots,P_n\}$
2. 把¬Q并入到公式集F中，得到{F, ¬Q}；　　$\{P_1,P_2,\ldots,P_n, \neg Q\}$
3. 把公式集{F, ¬Q}化为子句集S；　　$(P_1 \wedge P_2 \wedge \ldots \wedge P_n) \wedge \neg Q$
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。

归结反演的例子

例：已知

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

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

求证：G是F的逻辑结论。

证明：首先把F和¬G化为子句集：

$$F=\{\neg A(x,y)\vee\neg B(y)\vee C(f(x)),\ \neg A(x,y)\vee\neg B(y)\vee D(x,f(x))\}$$

$$\neg G=\{\neg C(z),\ A(a,b),\ B(b)\}$$

然后进行归结：

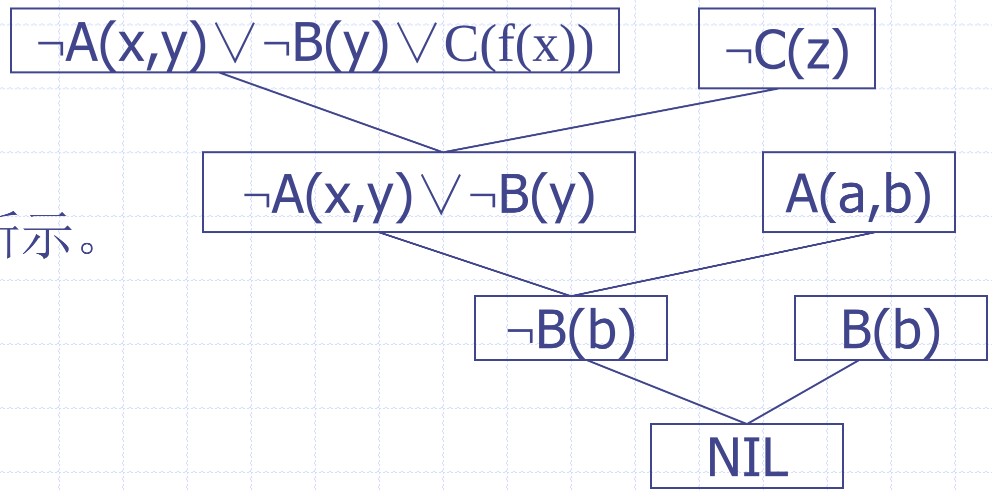
(6)¬A(x,y)∨¬B(y)　　由(1)与(3)归结，{f(x)/z}

(7)¬B(b)　　由(4)与(6)归结，{a/x,b/y}

(8)NIL　　由(5)与(7)归结

所以G是F的逻辑结论。

上述归结过程如右图归结树所示。



- 归结时，并不要求把子句集中所有的子句都用到。
- 在归结过程中，一个子句可以多次被用来进行归结。

3.3.4 归结策略

- 归结的一般过程

设有子句集$S=\{C_1,C_2,C_3,C_4\}$，则对此子句集归结的一般过程是：

1. S内任意子句两两逐一进行归结，得到一组归结式，称为第一级归结式，记为S_1。
2. 把S与S_1内的任意子句两两逐一进行归结，得到一组归结式，称为第二级归结式，记为S_2。
3. S和S_1内的子句与S_2内的任意子句两两逐一进行归结，得到一组归结式，称为第三级归结式，记为S_3。
4. 如此继续，直到出现了空子句或者不能再继续归结为止。

3.3.4 归结策略

- 归结策略可分为两大类：

一类是删除策略；

删除某些无用的子句来缩小归结的范围。

一类是限制策略。

通过对参加归结的子句进行种种限制，尽可能减小归结的盲目性，使其尽快地归结出空子句。

一个归结的例子

例3-8 设有子句集S={P, ¬R,¬P∨Q,¬Q∨R}。归结过程为：

S:	(1)P	
	(2)¬R	
	(3)¬P∨Q	
	(4)¬Q∨R	
S_1:	(5)Q	(1)与(3)归结
	(6)¬Q	(2)与(4)归结
	(7)¬P∨R	(3)与(4)归结
S_2:	(8)R	(1)与(7)归结
	(9)¬P	(2)与(7)归结
	(10)¬P	(3)与(6)归结
	(11)R	(4)与(5)归结
S_3:	(12) NIL	(1)与(9)归结

删除策略

- 纯文字删除法

  如果某文字L在子句集中不存在可与之互补的文字¬L，则称该文字为纯文字。包含纯文字的子句可以删除。

- 重言式删除法

  如果一个子句中同时包含互补文字对，则该字句称为重言式。重言式是永远为真的子句，可以删除。

- 包孕删除法

  设有子句C_1和C_2，如果存在一个代换σ，使得 $C_1\sigma \subseteq C_2$，则称C_1包孕于C_2。C2可删除。

  例如： $C_1 = P(x)$, $C_2 = P(y) \vee Q(z)$，则C_1包孕于C_2

支持集策略

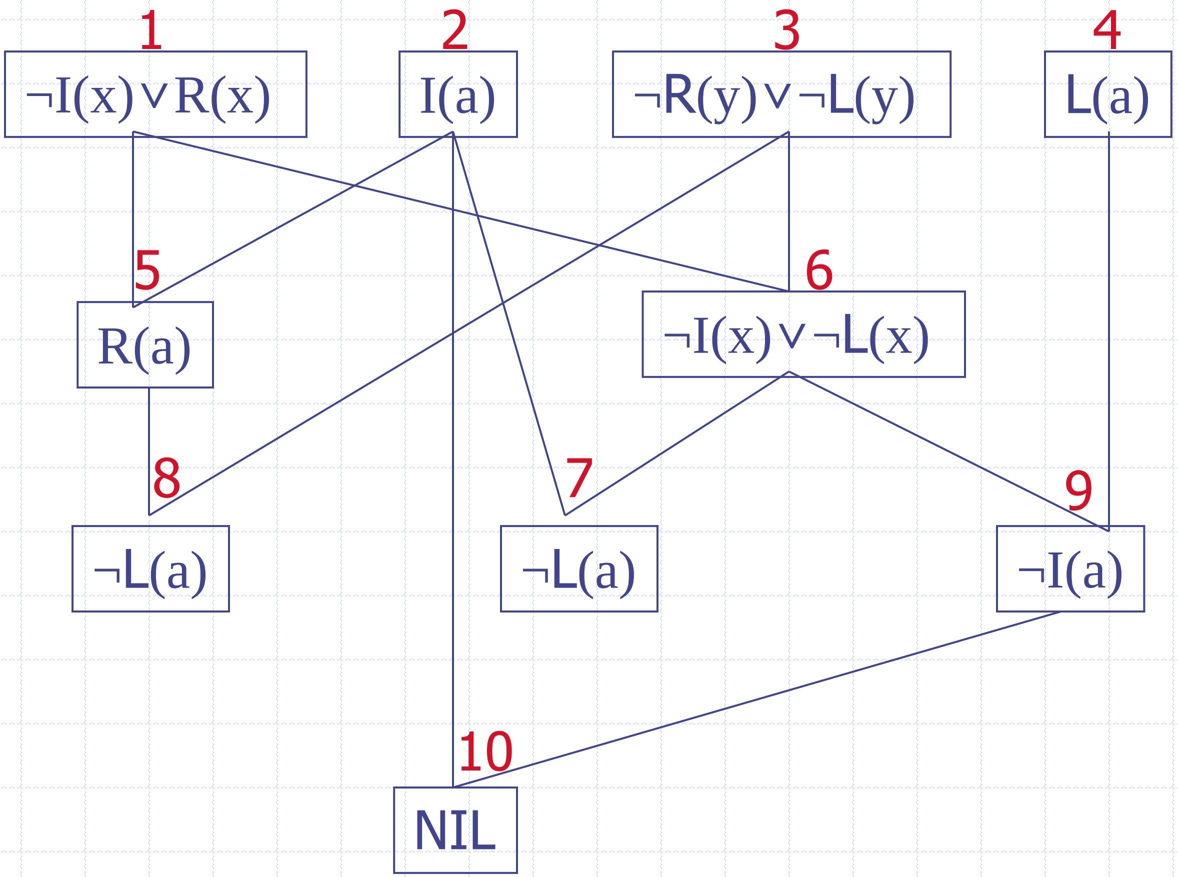
- 对参加归结的子句提出如下限制：每一次归结时，亲本子句中至少有一个是由目标公式的否定所得到的子句，或者是它的后裔。可以证明，支持集策略是完备的。

例3-9 设有子句集S={¬I(x)∨R(x),I(a),¬R(y)∨¬L(y),L(a)}其中¬I(x)∨R(x)是目标公式否定后得到的子句。

用支持集策略进行归结的过程是：

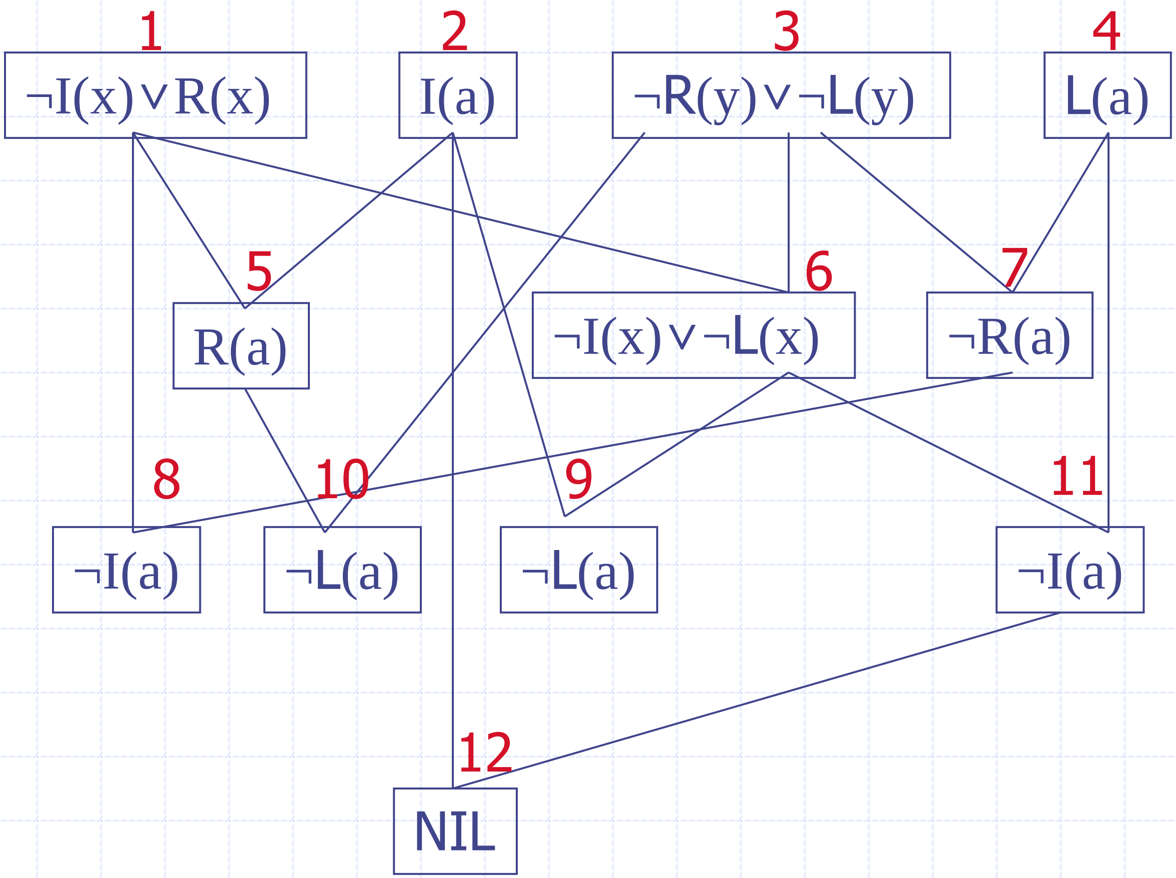
S:	(1)	¬I(x)∨R(x)	
	(2)	I(a)	
	(3)	¬R(y)∨¬L(y)	
	(4)	L(a)	
S_1:	(5)	R(a)	(1)与(2)归结
	(6)	¬I(x)∨¬L(x)	(1)与(3)归结
S_2:	(7)	¬L(a)	(2)与(6)归结
	(8)	¬L(a)	(3)与(5)归结
	(9)	¬I(a)	(4)与(6)归结
S_3:	(10)	NIL	(2)与(9)归结

支持集策略示例



线性输入策略

- 限制：参加归结的两个子句中必须至少有一个是初始子句集中的子句。
- 线性输入策略可限制生成归结式的数量，具有简单、高效的优点。但是它是不完备的。



祖先过滤策略

- 该策略与线性策略比较相似，但放宽了限制。当对两个子句C_1和C_2进行归结时，只要它们满足下述任一个条件就可以归结。

1. C_1和C_2中至少有一个是初始子句集中的子句。
2. C_1和C_2中一个是另外一个的祖先子句。

- 祖先过滤策略是完备的。

单文字子句策略

- 如果一个子句只包含一个文字，则称它为单文字子句。
- 限制：参加归结的两个子句中必须至少有一个是单文字子句。
- 用单文字子句策略归结时，归结式比亲本子句含有较少的文字，这有利于朝着空子句的方向前进，因此它有较高的归结效率。但是，这种归结策略是不完备的。当初始子句集中不包含单文字子句时，归结就无法进行。

3.3.5 应用归结原理求取问题的答案

求解的步骤：

1. 把已知前提用谓词公式表示出来，并且化为相应的子句集。设该子句集的名字为S。
2. 把待求解的问题也用谓词公式表示出来，然后把它否定，并与谓词Answer构成析取式。 Answer是一个为了求解问题而专设的谓词。
3. 把上述析取式化为子句集，并且把该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，则可获得答案。

应用归结原理证明命题

例3-12 设A,B,C三人中有人从不说真话，也有人从不说假话。现在向这三人提出同一个问题：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：“A和B中至少有一个是说谎者”。请证明A不是老实人，即证明$\neg T(A)$。

解：设用T(x)表示x说真话。则已知前提为：

$$T(C) \vee T(A) \vee T(B)$$

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

$$T(A) \rightarrow \neg T(B) \wedge \neg T(C)$$

$$\neg T(A) \rightarrow T(B) \vee T(C)$$

$$T(B) \rightarrow \neg T(A) \wedge \neg T(C)$$

$$\neg T(B) \rightarrow T(A) \vee T(C)$$

$$T(C) \rightarrow \neg T(A) \vee \neg T(B)$$

$$\neg T(C) \rightarrow T(A) \wedge T(B)$$

已知前提：$T(C) \vee T(A) \vee T(B)$

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

$T(A) \rightarrow \neg T(B) \wedge \neg T(C)$

$\neg T(A) \rightarrow T(B) \vee T(C)$

$T(B) \rightarrow \neg T(A) \wedge \neg T(C)$

$\neg T(B) \rightarrow T(A) \vee T(C)$

$T(C) \rightarrow \neg T(A) \vee \neg T(B)$

$\neg T(C) \rightarrow T(A) \wedge T(B)$

子句集S：(1) $\neg T(A) \vee \neg T(B)$

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

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

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

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

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

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

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

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

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

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

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

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

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

下面证明A不是老实人，即证明$\neg T(A)$。对$\neg T(A)$进行否定，并入S中，得到子句集S’，即S’比S多如下子句：

(8)$\neg(\neg T(A))$, 即$T(A)$

应用归结原理对S’进行归结：

(9)$\neg T(A) \vee T(C)$ (1)和(7)归结

(10)$\neg T(A)$ (2)和(9)归结

(11)NIL (8)和(10)归结

在该例中，(3)包孕了(6),(7), (5)包孕了(1),(2), 所以可删除(3),(5)

进一步推广：已知上述条件，求谁是老实人。

把已知前提条件化成子句集，得到S：

(1)¬T(A)∨¬T(B)

(2)¬T(A)∨¬T(C)

(3)T(C)∨T(A)∨T(B)

(4)¬T(B)∨¬T(C)

(5)¬T(C)∨¬T(A)∨¬T(B)

(6) T(A)∨T(C)

(7)T(B)∨T(C)

定义一个新的谓词Ansewer。把¬T(x)∨Ansewer(x)并入S得到S’。即多一个子句：

(8)¬T(x)∨Ansewer(x)

应用归结原理对S’进行归结：

(9)¬T(A)∨T(C)	(1)和(7)归结
(10)T(C)	(6)和(9)归结
(11)Answer(C)	(8)和(10)归结

所以C是老实人，即C从不说假话。

归结演绎推理的特点

优点：

- 简单，便于在计算机上实现。

缺点：

- 必须把逻辑公式化成子句集。
- 不便于阅读与理解。

  $\neg P(x) \vee Q(x)$没有$P(x) \rightarrow Q(x)$直观。

- 可能丢失控制信息。

  下列逻辑公式：

  $(\neg A \wedge \neg B) \rightarrow C$　　$\neg A \rightarrow (B \vee C)$

  $(\neg A \wedge \neg C) \rightarrow B$　　$\neg B \rightarrow (A \vee C)$

  $(\neg C \wedge \neg B) \rightarrow A$　　$\neg C \rightarrow (B \vee A)$

  化成子句后都是: $A \vee B \vee C$

3.4 与/或形演绎推理

- 归结演绎推理要求把有关问题的知识及目标的否定都化成子句形式，然后通过归结进行演绎推理，其推理规则只有一条，即归结规则；
- 与/或形演绎推理不再把有关知识转化成子句集，而把领域知识及已知事实分别用蕴含式及与/或形表示出来，然后通过运用蕴含式进行演绎推理，从而证明某个目标公式。

与/或形演绎推理的特点

优点：

- 不必把公式化为子句集，保留了连接词“→”。这就可直观地表达出因果关系，比较自然。

缺点：

- 对正向演绎推理而言，目标表达式被限制为文字的析取式；
- 对逆向演绎推理，已知事实的表达式被限制为文字的合取式；
- 正、逆双向演绎推理虽然可以克服以上两个问题，但其“接头”的处理却比较困难。

完

谢谢