归结原理
(The Resolution Principle)

张文生　研究员

中国科学院自动化研究所

一．起因；

二．命题逻辑的归结原理；

三．置换与合一；

四．一阶谓词的归结原理；

五．归结原理的完备性；

六．效率的提高

Davis-Putnam的工作

- 预备知识：
- 空子句永假；（注：空子句是指集合中有一个空子句）；
- 空集合永真；（空集合是指没有元素的集合）；
- 重言式子句：包含互补对的子句。

- 四条规则:

注意：下面规则的应用不改变子句集的不相容性。

- 规则一：重言式规则
  - **S**中的重言式子句，不会为**S**的不可满足提供任何信息，应该删除。
  - 例如：**S={P∨~P，Q，R∨P}**
    S的逻辑含义是（**P∨~P**）∧**Q∧(R∨P)=**
    Q∧(R∨P)，从而删去重言式**P∨~P**，不影响**S**的真值。
  - **S'={Q，R∨P}**

- 规则二：单文字规则(**one-literal rule**)
  - 如果在**S**中存在只有一个文字的基础子句**L**, 消去在**S**中带有这个文字**L**的所有子句得到**S'**, 如果**S'**为空, 则**S**是相容的;

    否则, 从**S'**中删去**~L**, 得到**S''**，则 **S''**不可满足当且仅当**S**不可满足.
  - 单文字: 在**S**中存在只有一个文字的基础子句**L.**
  - 例子: **S={L，L∨P，~L∨Q，S∨~R}**
    - **S' = {~L∨Q，S∨~R}**　　　**S''= {Q，S∨~R}**
  - **S**不可满足, 则在所有解释下**S**都为假;
    - **L=F;**
    - **L=T;**
      - **~L=F.**

- 规则三：纯文字规则
  - L是S的纯文字. 从S中删除含L的子句得S’，如果S’为空集，那么S是可满足的;
    否则, S’不可满足当且仅当S不可满足.
  - 纯文字：如果文字L出现于S中，而~L不出现于S中，L称为S的纯文字.
  - 例子: S={A∨B, A∨~B, ~B, B}
    - S’= {~B, B};
  - S不可满足, 在A为真的情况下不可满足;
    - A=1 → A∨B=1, A∨~B=1;
    - S’不可满足, 当然S不可满足;

规则四

- 分裂规则

  - $S=(L\vee A_1)\wedge\ldots\wedge(L\vee A_m)\wedge(\sim L\vee B_1)\wedge\ldots\wedge(\sim L\vee B_n)\wedge R$

    其中：A_i, B_i, R中不含L和~L（自由）。
    令$S'=\{A_1\wedge\ldots\wedge A_m\wedge R\}$，
    $S''=\{B_1\wedge\ldots\wedge B_n\wedge R\}$
    则S不可满足 当且仅当 S'和S''同时是不可满足，
    即：$S'\vee S''$同时是不可满足。

例1. $S=\{P \vee Q \vee \sim R,\ P \vee \sim Q,\ \sim P,\ R,\ U\}$

- 对**U**使用纯文字: **{P∨Q∨~R, P∨~Q, ~P, R}**
- 对**~P**使用单文字: **{Q∨~R, ~Q, R}**
- 对**~Q**使用单文字: **{~R, R}**
- 对**R** 使用单文字: **{□}**
- **S**不可满足**;**

- 注意：如果**~ L**是单文字基础子句，当**~L**从这个子句集合中被删去后，则这个子句为空子句。

例2. S={P∨Q, ~Q, ~P∨Q∨~R}

- 对**~Q**使用单文字**: {P, ~P∨~R}**
- 对**P**使用单文字**: {~R}**
- 对**~R**使用纯文字**: {□}**
- **S**可满足**;**

例3. $S=\{P\vee\sim Q,\ \sim P\vee Q,\ Q\vee\sim R,\ \sim Q\vee\sim R\}$

- 用规则4：$S_1=\{\sim Q,\ Q\vee\sim R,\ \sim Q\vee\sim R\}$
  $S_2=\{Q,\ Q\vee\sim R,\ \sim Q\vee\sim R\}$

- 在S_1中，对**~Q**使用单文字：**{~R}**
- 在S_2中，对**Q**使用单文字：**{~R}**
- 则得到：$S_1 \vee S_2=\{\sim R\}$
- 对**~R** 使用纯文字：**{}**
- **S**可满足**;**

例4. $S=\{P \vee Q, P \vee \sim Q, R \vee Q, R \vee \sim Q\}$

- 用规则3： **$\{Q \vee \sim R,\ \sim Q \vee \sim R\}$**
- 用规则3： **$\{\}$**

- **S**可满足**;**

注意：

- 这些规则对于命题逻辑是十分有效的，但是，对于一阶谓词逻辑则需要寻找基础实例集。
- Davis的工作在理论上是十分重要的，它对于后来的归结原理的证明方法起了重要的作用。

回顾/起因

- 自动定理证明
- 定理: $(F_1 \wedge F_2 \wedge \ldots \wedge F_n) \rightarrow G$
- 证明公式永真
- 证明公式的非永假
- 化为前束合取范式
- 化为Skolem范式
- 化为子句集
- 子句集不可满足

- Herbrand定理(Version 1)
  - 子句集S是不可满足的，当且仅当对应于S的任一棵完备语义树, 都存在一棵有限的封闭语义树。
- Herbrand定理(Version 2)
  - 子句集S是不可满足的，当且仅当存在一个有限不可满足的S的基础实例集合S'。
- Gilmore的方法(1960)
- Davis-Putnam: 提高效率

- 困难:
  - 生成基础实例集合是指数复杂性的.
- 例子

例子

- $S=\{P(x,g(x),y,h(x,y),z,k(x,y,z)), \sim P(u,v,e(v),w,f(v,w),x)\}$
  - $H_0=\{a\}$
  - $H_1=\{a,g(a),h(a,a),k(a,a,a),e(a),f(a,a)\}$
- 基础实例集：
  - $S_0=\{ P(a,g(a),a,h(a,a),a,k(a,a,a)), \sim P(a,a,e(a),a,f(a,a),a)\}$
  - S_1有$6*6*6 + 6*6*6*6 = 1512$个元素;
  - H_5有10^{64}数量级的元素，S_5有10^{256}数量级的元素.

- 如果可以不从基础实例出发，而直接从S出发，则可以避免大的计算量。
- 归结原理:
  - 检查子句集S中是否含有空子句，或者能*推导*出空子句。
- 推导不是任意定义的，必须保证推导出的结果是原来子句集合（逻辑公式）的逻辑结论。
- 例如，证明c_1->~c_2永真.

- 归结原理是J.A.Robinson在1965年提出的，被认为是定理机器证明的重大突破。

一．起因；

二．命题逻辑的归结原理；

三．置换与合一；

四．一阶谓词的归结原理；

五．归结原理的完备性；

六．效率的提高：

二.命题逻辑的归结原理

- 归结原理是Davis-Putnam单文字规则的扩展;
  - 如果在S中存在只有一个文字的基础子句L, 消去在S中带有这个文字L的所有子句得到S’, 如果S’为空, 则S是相容的; 否则, 从S’中删去~L, 得到S’’. S’’不可满足当且仅当S不可满足.
- 例:
  - C_1: P
  - C_2: ~P$\vee$Q
  - C_3: Q
- 互补对：原子与原子的非构成互补对；

- 扩展
  - 例子：
  - C_1: $P \vee Q$
  - C_2: $\sim P \vee R$
  - C_3: $Q \vee R$

- C_1: 打伞 ∨ 不下雨
- C_2: 不打伞 ∨ 不被淋湿
- C_3: 不下雨 ∨ 不被淋湿

- 归结原理：
  - 对任何两个子句C_1和C_2，如果一个在C_1中的文字L_1和一个在C_2中的文字L_2构成互补对，则分别从C_1和C_2中删除L_1和L_2，并将C_1和C_2的剩余部分构成析取式C，则C称为C_1和C_2的归结式。
- 例：
  - C_1: $P \vee Q \vee \sim T$
  - C_2: $\sim P \vee R$
  - C: $Q \vee \sim T \vee R$ （ C_1与C_2的归结式）

- 例：
  - C_1: $\sim P \vee Q$
  - C_2: $\sim P \vee R$
  - C_1与C_2没有互补对，所以没有归结式！
- 互补对每次只能取一对：
  - C_1: $P \vee Q$
  - C_2: $\sim P \vee \sim Q$
  - C_3: $Q \vee \sim Q$ / $P \vee \sim P$
  - C_3不能为空！

定理

- 子句C_1和C_2的归结式C是C_1和C_2的逻辑结论。
- 证明：
  - 令$C_1=L \vee C_1'$, $C_2=\sim L \vee C_2'$.
  - $C = C_1' \vee C_2'$.
  - 设$C_1 \wedge C_2$为真，证C为真。

- 事实上：
  - $C_1 = L \vee C_1' = \sim C_1' \rightarrow L$
  - $C_2 = \sim L \vee C_2' = L \rightarrow C_2'$
  - $C_1 \wedge C_2$为真, $\sim C_1' \rightarrow C_2' = C_1' \vee C_2'$为真.

- C_1: 打伞 $\vee$ 不下雨 = 下雨$\rightarrow$打伞
- C_2: 不打伞 $\vee$ 不被淋湿 = 打伞$\rightarrow$不被淋湿
- C_3: 不下雨 $\vee$ 不被淋湿 = 下雨$\rightarrow$不被淋湿

归结的例子

设公理集：

P,

$(P \wedge Q) \rightarrow R$,

$(S \vee T) \rightarrow Q$,

T

求证：R

子句集：

(1) P

(2) $\sim P \vee \sim Q \vee R$

(3) $\sim S \vee Q$

(4) $\sim T \vee Q$

(5) T

(6) $\sim R$（目标求反）

化子句集：

$(P \wedge Q) \rightarrow R$

$\Rightarrow \sim(P \wedge Q) \vee R$

$\Rightarrow \sim P \vee \sim Q \vee R$

$(S \vee T) \rightarrow Q$

$\Rightarrow \sim (S \vee T) \vee Q$

$\Rightarrow (\sim S \wedge \sim T) \vee Q$

$\Rightarrow (\sim S \vee Q) \wedge (\sim T \vee Q)$

$\Rightarrow \{\sim S \vee Q, \sim T \vee Q\}$

子句集：

(1) P
(2) ~P∨~Q∨R
(3) ~S∨Q
(4) ~T∨Q
(5) T
(6) ~R（目标求反）

归结：

(7) ~P∨~Q	(2, 6)
(8) ~Q	(1, 7)
(9) ~T	(4, 8)
(10) nil	(5, 9)

- 例2
  - S={P∨Q，~P∨Q，P∨~Q，~P∨~Q}

(1) P∨Q

(2) ~P∨Q

(3) P∨~Q

(4) ~P∨~Q

(5) Q (1,2)

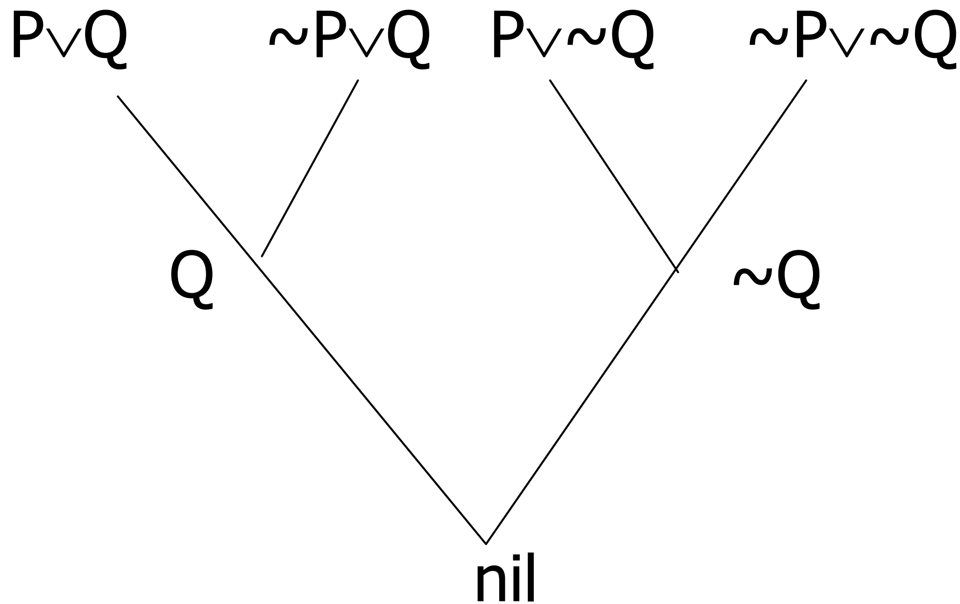
(6) ~Q (3,4)

(7) nil (5,6)

- 定义: 推演
  - 给定一个子句集合S，从S到子句C的一个推演是一个有限的子句序列C_1 ,…, C_k，使得每个C_i 或是S中的一个子句，或是C_1到C_{i-1}中的某些子句的一个归结式，而C_k=C。如果C=nil，则这个推演（推导）称为S的一个证明，或反演。

推演树(deduction tree)

- S={P∨Q，~P∨Q，P∨~Q，~P∨~Q}



归结定理完备性

- 如果S不相容，则一定存在一个S的反演。

三. 置换与合一

- 例：
  - C_1： $P(x) \vee Q(x)$
  - C_2： $\sim P(f(x)) \vee R(x)$
- 没有互补对;
- 例：
  - C_1： $P(y) \vee Q(y)$ $\{y/x\}$
  - C_1： $P(f(x)) \vee Q(f(x))$ $\{f(x)/y\}$
  - C： $R(x) \vee Q(f(x))$

置换(substitution)

- 定义: 置换是一个形如$\{t_1/v_1,\ldots, t_n/v_n\}$的有限集，其中每个$v_i$是变量，$t_i$是不同于$v_i$的项（常量、变量或函数）($v_i \neq t_i$)。当$i \neq j$时，$v_i \neq v_j$。
  - 无元素组成的置换称为空置换，记为ε;
- 例子：
  - $\{a/x, w/y, f(s)/z\}$, $\{g(x)/x\}$是置换;
  - $\{x/x\}$, $\{y/f(x)\}$不是置换;

实例(instance)

- 置换的结果称为实例;
- 定义: 令$\theta=\{t_1/v_1,\ldots, t_n/v_n\}$是一个置换。E是一个表达式。则$E\theta$是一个同时用项$t_i$代替E中变量$v_i$所得到的表达式($1\le i\le n$)。$E\theta$称为E的实例。
- 表达式:
  - 不一定是公式;
  - 项, 项集, 原子, 原子集, 文字, 子句, 子句集。
- 例子:
  - $E=P(x, y, z)$, $\theta=\{a/x, f(b)/y, c/z\}$
    - $E\theta=P(a, f(b), c)$
  - $E=P(x, y, z)$, $\theta=\{y/x, z/y\}$
    - $E\theta=P(y, z, z)$. $E\theta\neq P(z, z, z)$. (同时)

- 例子：
  - $E=P(x, y, z)$, $\theta=\{a/x, f(b)/y, c/z\}$
    - $E\theta=P(a, f(b), c)$
  - $E=P(x, g(y), h(x,z))$, $\theta=\{a/x, f(b)/y, g(w)/z\}$
    - $E\theta=P(a, g(f(b)), h(a,g(w)))$
  - $E=P(x, y, z)$, $\theta=\{y/x, z/y\}$
    - $E\theta=P(y, z, z)$. $E\theta\neq P(z, z, z)$. (同时)

置换的复合(composition)

- 例子:
  - $E=P(x, y, z)$
  - $\theta=\{a/x, f(z)/y, w/z\}$
  - $E\theta=P(a, f(z), w)$
  - $\lambda=\{t/z, g(b)/w\}$
  - $E\theta\lambda=P(a, f(t), g(b))$
  - $\theta\lambda=\{a/x, f(t)/y, g(b)/z\}$

复合置换

- **定义**：令$\theta=\{t_1/x_1,\ldots,t_n/x_n\}$, $\lambda=\{u_1/y_1,\ldots,u_m/y_m\}$是两个置换.则$\theta$与$\lambda$的复合是一个置换, 记为$\theta\circ\lambda$. (先$\theta$后$\lambda$)
  - 构成$\{t_1\lambda/x_1,\ldots,t_n\lambda/x_n, u_1/y_1,\ldots,u_m/y_m\}$;
  - 如果$y_j\in\{x_1,\ldots,x_n\}$, 则删除u_j/y_j;
  - 如果$t_k\lambda=x_k$, 则删除$t_k\lambda/x_k$;

- 例子:
  - $\theta=\{t_1/x_1, t_2/x_2\}=\{f(y)/x, z/y\}$
  - $\lambda=\{u_1/y_1, u_2/y_2, u_3/y_3\}=\{a/x, b/y, y/z\}$
  - $\theta\circ\lambda=\{t_1\lambda/x_1, t_2\lambda/x_2, u_1/y_1, u_2/y_2, u_3/y_3\}$
    $=\{f(b)/x, y/y, a/x, b/y, y/z\}$
  - 删除$\{a/x, b/y\}$;
  - 删除$\{y/y\}$;

- 置换的复合满足结合律;
  - $(\theta \circ \lambda) \circ \mu = \theta \circ (\lambda \circ \mu)$
- 但一般不满足交换律;
  - $\theta = \{a/x\}$, $\lambda = \{b/x\}$
  - $\theta \circ \lambda = \{a/x\}$
  - $\lambda \circ \theta = \{b/x\}$
  - $\theta \circ \lambda \neq \lambda \circ \theta$

合一(unification)

- **$E_1\theta=E_2\theta$?**
- 定义:如果$E_1\theta=\ldots=E_n\theta$，则称置换θ为 $\{E_1, \ldots, E_n\}$的合一子(unifier).
- 定义：如果对$\{E_1, \ldots, E_n\}$存在这样的合一子，则称集合$\{E_1, \ldots, E_n\}$是可合一的.
- 例**1:**
  - **E={P(a,y), P(x, f(b))}, θ={a/x, f(b)/y}.**
  - **E={P(a,b), P(x, f(b))}**

- 合一子不一定唯一
  - $E=\{P(a,y), P(x, f(b))\}$
  - $\theta_1=\{a/x, f(b)/y\}$ (唯一)
  - $E=\{P(x,y), P(x,f(b))\}$
  - $\theta_1=\{a/x, f(b)/y\}$ (不唯一)
  - $\theta_2=\{b/x, f(b)/y\}$

最一般合一子

- mgu(most general unifier)

- 定义：如果对**E**的每个合一子θ**,** 都存在一个置换λ**,** 使得θ**=**γ°λ，则称合一子γ是集合$\{E_1,\ldots,E_n\}$的最一般合一子**.**

- 例子:
  - $E=\{P(x,y), P(x,f(b))\}$
    - $\theta_1=\{a/x, f(b)/y\}$
    - $\theta_2=\{b/x, f(b)/y\}$
    - $\gamma=\{f(b)/y\}$
    - $\theta_1= \gamma \circ \{a/x\}$
    - $\theta_2= \gamma \circ \{b/x\}$

- 是否存在寻找**E**的**mgu**的一般算法?
- 如何寻找**E**的**mgu?**

- 合一算法的考虑**:**
  - 消除两个谓词之间项的差别**. {P(x,…), P(a,…)}**

- 非空表达式集**W**的差别集**:**
  - 从左向右**,** 在**W**中的所有表达式**,** 遇到第一个不相同符号**,** 提取从这个符号开始的子表达式**,** 由此构成一个集合**,** 称为**W**的差别集**,** 记为**D.**

- 例子**:**
  - **W={P(x,f(y,z),z，w), P(x,a), P(x, g(z),z,b)}**
  - **D={f(y,z), a, g(z)}**

合一算法

W的合一算法**:**

- **1. K=0, W_k=W, $\gamma_k=\varepsilon$.**
- **2.** 如果W_k是单一的**,** 停机**,** γ_k是**W**的**mgu.**
  否则，求出W_k的差别集D_k**.**
- **3.** 如果在D_k中存在元素v_k与t_k**,** 使v_k是一个未出现在t_k中的变量**,** 转**4;**
  否则，停机**, W**是不可合一的**.**
- **4.** 令$\gamma_{k+1}=\gamma_k \circ \{t_k/v_k\}$，　$W_{k+1}=W_k \circ \gamma_{k+1}$.
- **5. K=K+1.** 转**2.**

- 换名:
  - {P(f(x), x), P(x, a)};
  - D={f(x), x}.
  - 换名: {P(f(y), y), P(x, a)};
  - mgu: {f(a)/x, a/y}

例1:

- 求W={P(a,x,f(g(y))), P(z,f(z),f(u))}的mgu.
  - D_0={a,z}. $\gamma_1 = \varepsilon \circ$ {a/z}= {a/z}.
  - $W_1 = W_0 \cdot \gamma_1$ ={P(a,x,f(g(y))), P(a,f(a),f(u))}
  - D_1={x,f(a)}. $\gamma_2 = \gamma_1 \circ$ {f(a)/x}= {a/z, f(a)/x}.
  - $W_2 = W_1 \cdot \gamma_2$ ={P(a,f(a),f(g(y))), P(a,f(a),f(u))}
  - D_2={g(y),u}. $\gamma_3 = \gamma_2 \circ$ {g(y)/u}={a/z,f(a)/x, g(y)/u}
  - $W_3 = W_2 \cdot \gamma_3$ ={P(a,f(a),f(g(y)))}
  - γ_3是mgu.

例2:

- 求$W=\{Q(f(a), g(x)), Q(y, y)\}$的mgu.
  - $D_0=\{f(a),y\}$. $\gamma_1= \varepsilon\circ\{f(a)/y\}= \{f(a)/y\}$.
  - $W_1= W_0\cdot\gamma_1 =\{Q(f(a), g(x)), Q(f(a), f(a))\}$
  - $D_1=\{g(x), f(a)\}$.
  - 不可合一, 没有mgu.

例3:

- 求$W=\{P(f(y), y), P(x, a)\}$的mgu.
  - $D_0=\{f(y),x\}$.
  - $\gamma_1= \varepsilon \circ\{f(y)/x\}= \{f(y)/x\}$.
  - $W_1= W_0 \cdot \gamma_1 =\{P(f(y), y), P(f(y), a)\}$

  - $D_1=\{y, a\}$.
  - $\gamma_2=\gamma_1\circ\{a/y\}= \{f(y)/x\} \circ\{a/y\} = \{f(a)/x, a/y\}$.
  - $W_2= W_1 \cdot \gamma_2 =\{P(f(a),a)\}$
  - γ_2是mgu.

定理

- 如果**W**是一个有限非空的可合一的表达式集合**,** 则合一算法一定终止在第二步**,** 并且最后一个γ_k是**W**的**mgu.**

一．起因；

二．命题逻辑的归结原理；

三．置换与合一；

四．一阶谓词的归结原理；

五．归结原理的完备性；

六．效率的提高：

四. 一阶谓词的归结原理

- 定义: (因子)
  - 如果一个子句C中, 两个或更多的文字有相同的谓词符号, 且它们有mgu λ, 则$C\lambda$称为C的因子.
  - 如果$C\lambda$是单元子句, 则称$C\lambda$为C的单元因子.

- 例子:
  - $C=P(x) \vee P(f(y)) \vee \sim Q(x)$
  - $P(x)$和$P(f(y))$, $\lambda=\{f(y)/x\}$
  - $C\lambda= P(f(y)) \vee \sim Q(f(y))$

一阶谓词的归结

- 定义: (一阶谓词的归结)
  - 令C_1与C_2是两个子句, 它们没有共同的变量. 设L_1与L_2分别是C_1与C_2的两个文字, 如果L_1与$\sim L_2$有mgu λ, 则子句

$$(C_1\lambda - L_1\lambda) \cup (C_2\lambda - L_2\lambda)$$

称为C_1与C_2的二元归结式;

L_1与L_2称为被归结文字。

例子：

- C_1: $P(x) \vee Q(x)$
- C_2: $\sim P(a) \vee R(x)$

- 重命名C_2: $\sim P(a) \vee R(y)$
- $L_1=P(x)$, $L_2=\sim P(a)$
- L_1与$\sim L_2$有mgu $\lambda=\{a/x\}$

- $(C_1\lambda - L_1\lambda) \cup (C_2\lambda - L_2\lambda)$
  $=(\{P(a),Q(a)\} - \{P(a)\}) \cup (\{\sim P(a), R(y)\} - \{\sim P(a)\}$
  $=\{Q(a)\} \cup \{R(y)\}$
  $=\{Q(a), R(y)\}$
- $Q(a) \vee R(y)$ 是C_1与C_2的二元归结式.

归结式

- 定义：子句C_1与C_2的<u>归结式</u>包括：
  - C_1与C_2的二元归结式;
  - C_1的因子与C_2的二元归结式;
  - C_1与C_2的因子的二元归结式;
  - C_1的因子与C_2的因子的二元归结式;

- 例子:
  - $C_1=P(x) \vee P(f(y)) \vee R(g(y))$
  - $C_2=\sim P(f(g(a))) \vee Q(b)$
  - C_1的因子C_1'是$P(f(y)) \vee R(g(y))$
  - C_1'与C_2的二元归结式是 $R(g(g(a))) \vee Q(b)$

例1

- F_1: $(\forall x)(C(x) \rightarrow (W(x) \wedge R(x)))$
- F_2: $(\exists x)(C(x) \wedge O(x))$
- G: $(\exists x)(O(x) \wedge R(x))$
- 证明G是F_1与F_2的逻辑结论.

- 化成标准式:

(1) $\sim C(x) \vee W(x)$
(2) $\sim C(x) \vee R(x)$
(3) $C(a)$
(4) $O(a)$
(5) $\sim O(x) \vee \sim R(x)$

- 归结:

(6) $R(a)$	$\{a/x\}$	(2)(3)
(7) $\sim O(a)$	$\{a/x\}$	(5)(6)
(8) nil		(4)(7)

例2

1．能阅读的都是有文化的. **R（x），L（x）**
2．海豚是没有文化的.　　**D（y），~L（y）**
3．某些海豚是有智能的.　**D（z），I（z）**
证明：某些有智能的并不能阅读.　**I（w），~R(w)**

将文字叙述变成逻辑公式表达：

$(\forall x)\,(R(x) \rightarrow L(x))$

$(\forall y)\,(D(y) \rightarrow \sim L(y))$

$(\exists z)\,(D(z) \wedge I(z))$

$(\exists w)\,(I(w) \wedge \sim R(w))$

$(\forall x)(R(x) \rightarrow L(x))$
$(\forall y)(D(y) \rightarrow \sim L(y))$
$(\exists z)(D(z) \wedge I(z))$
$(\exists w)(I(w) \wedge \sim R(w))$

- 化为子句集:

(1) $\{\sim R(x) \vee L(x)\}$

(2) $\{\sim D(y) \vee \sim L(y)\}$

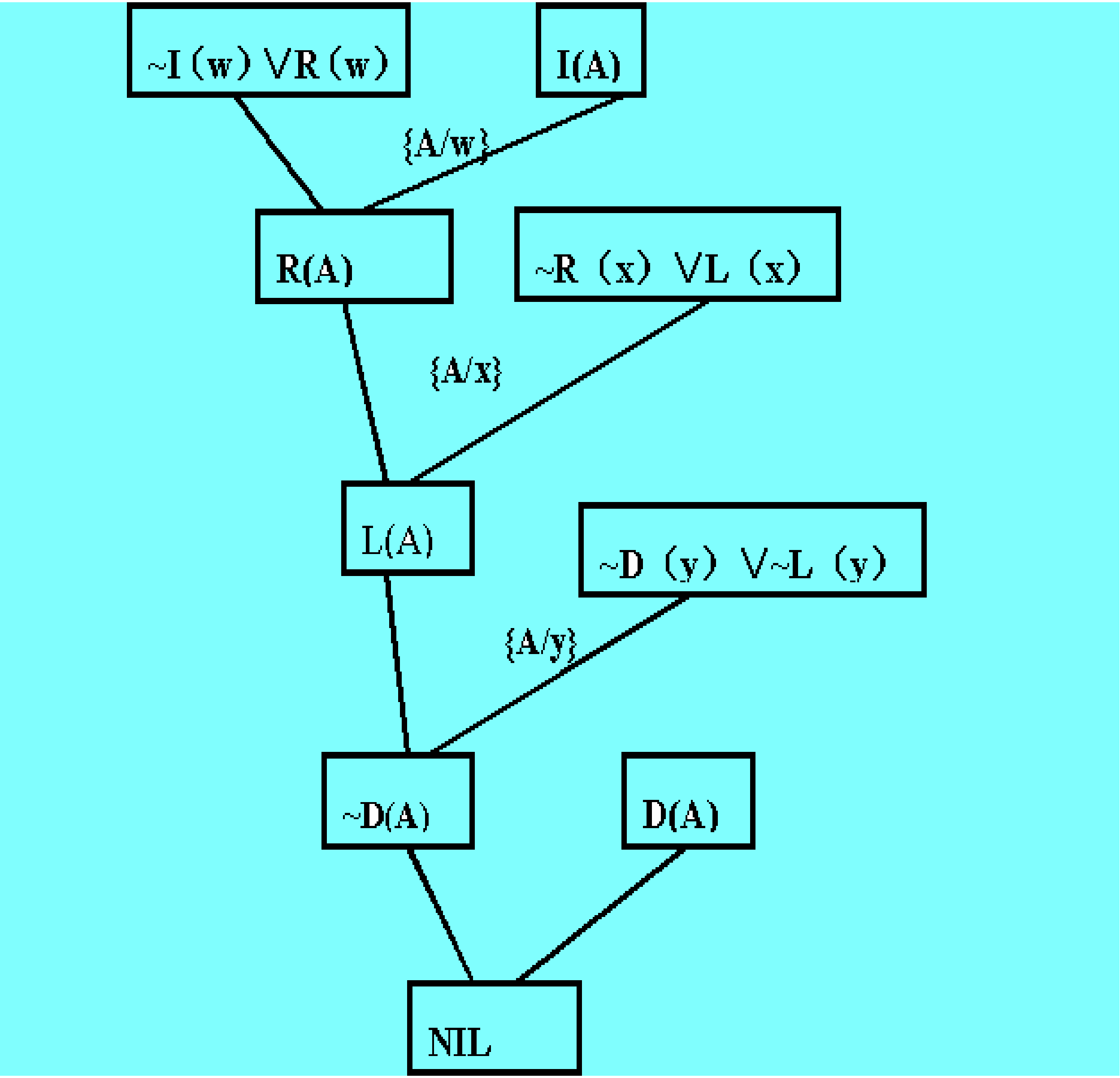
(3) $\{D(A), I(A)\}$

- 目标的否定

$(\forall w)[\sim I(w) \vee R(w)]$

(4) $\{\sim I(w) \vee R(w)\}$

子句集:
~R（x）∨L（x）
~D（y）∨~L（y）
D(A）
I（A）
~I（w）∨R（w）



一．起因；

二．命题逻辑的归结原理；

三．置换与合一；

四．一阶谓词的归结原理；

五．归结原理的完备性；

六．效率的提高：

五. 归结原理完备性

- 定理:

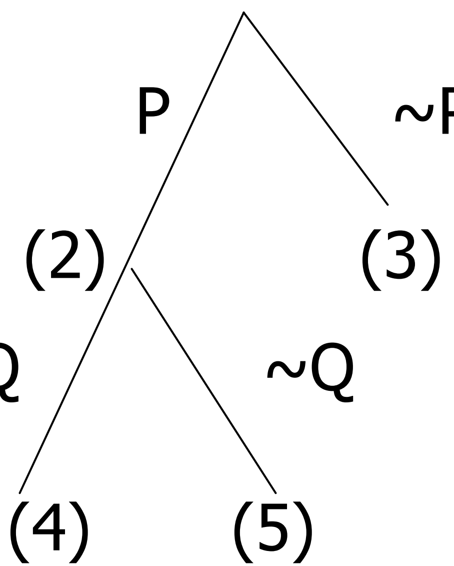
  一个子句集合**S**是不相容的，当且仅当存在一个从**S**到**nil**的反演。

- 难点: 存在性.
  - 构造性
  - $\mathbf{C_1}$ **,…,** $\mathbf{C_k}$
  - 语义树

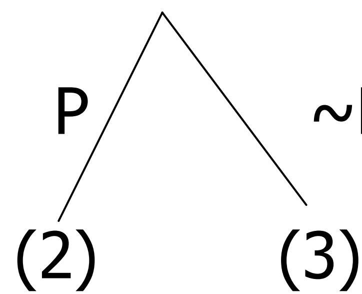
语义树与归结原理

- 子句集合**S**不相容/封闭语义树/推理节点
- 例子**:**

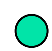
{P, ~P∨Q, ~P∨~Q}



{P, ~P∨Q, ~P∨~Q, ~P}



{P, ~P∨Q, ~P∨~Q, ~P, nil}



语义树的倒塌

提升引理

- 引理：
  - 如果C_1'和C_2'是子句C_1和C_2的实例, C'是C_1'和C_2'的归结式, 则存在C_1和C_2的归结式C, 使C'是C的实例.

例子

- $C_1 : P(x) \vee Q(f(x))$
- $C_2 : \sim Q(y) \vee R(y)$

- $C_1' : P(a) \vee Q(f(a)) \qquad \{a/x\}$
- $C_2' : \sim Q(f(a)) \vee R(f(a)) \qquad \{f(a)/y\}$

- $C' : P(a) \vee R(f(a))$
- $C : P(x) \vee R(f(x))$

归结原理完备性

- 定理**:** 一个子句集合**S**是不相容的， 当且仅当存在一个从**S**到**nil**的反演。
  - 设子句集合**S**不相容。
  - 设存在一个从**S**到**nil**的反演。

一．起因；

二．命题逻辑的归结原理；

三．置换与合一；

四．一阶谓词的归结原理；

五．归结原理的完备性；

六．效率的提高；

效率的问题

- 归结原理比**Herbrand**定理有了明显的进步;
- 盲目的归结会产生组合爆炸问题;
- 不必要的归结式 → 不必要的归结式;
- 例子:

例子

- $S=\{P\vee Q, \sim P\vee Q, P\vee\sim Q, \sim P\vee\sim Q\}$

- 盲目归结过程:
- $S_0=S$
- $S_i=\{C_1, C_2$的归结式$|C_1\in S_0\cup S_1\cup\ldots\cup S_{i-1}, C_2\in S_{i-1}\}$
- 具体过程:
  - S_0: (1)$P\vee Q$ (2)$\sim P\vee Q$ (3)$P\vee\sim Q$ (4)$\sim P\vee\sim Q$
  - S_1: (5)Q (1)(2) (6)P (1)(3)
  - (7)$Q\vee\sim Q$ (1)(4) (8)$P\vee\sim P$ (1)(4) ……. (12)$\sim Q$
  - S_2: (13) $P\vee Q$ (1)(7) (14)$P\vee Q$ (1)(8)
  - …….
  - (39) nil (5)(12)

效率的提高

- 1965: Wos, G.A.Robinson, Curson, 支持集归结;
- 1967: Slagle, 语义归结;
- 1970: Loveland, Luckham, 线性归结;
- 1971: Boyer, 锁归结;
- 1978: 刘叙华, 锁语义归结;
- 1979: 王湘浩,刘叙华, 广义归结;

分类

- 限制参加归结的子句;
- 限制子句中被归结的文字;
- 限制归结的方式;