第 7 卷第 4 期
1996 年 4 月

软件学报
JOURNAL OF SOFTWARE

Vol. 7, No. 4
Apr. 1996

表达式的覆盖、分解与划分*

周生炳　　戴汝为

(中国科学院自动化研究所　北京　100080)

摘要　本文把简单表达式(项和原子)视为语言 L 的 Herbrand 域或 Herbrand 基中的集合. 作者提出覆盖表达式的概念,得到 2 个表达式之间覆盖关系的判别准则. 对多个表达式,作者提出表达式的分解概念及相应的分解算法,在此基础上,本文给出 1 个表达式覆盖多个表达式的等价条件. 根据集合的划分公式,得到划分表达式的方法. 最后定义 1 个变换把合取式转换为简单表达式,从而方便地把简单表达式的结果推广到合取式. 本文是作者提出的一种标记逻辑程序的过程语义的理论基础.

关键词　生成集合,标记逻辑程序,表达式的覆盖、分解与划分,差异矩阵.

本文讨论表达式的分解与覆盖问题,它是标记逻辑程序[1,2]ALP(annotated logic program)过程语义的基础. 直观上,对查询 $A_1:\mu_1\&\ldots\&A_k:\mu_k$,我们要确定它的正确与否,即是否存在 $A_1,\ldots,A_k$ 的某些实例 $A'_1,\ldots,A'_K$,使 $A'_1:\mu_1\&\ldots\&A'_k:\mu_k$ 成立. 用支持模型的语言来刻划,即 $M(A'_i)\in D\mu_i\cup D\top$? 为此,我们要考虑 G 中所有以 $A_i:\rho_i$ 为头的子句,这里 $\rho_i\in D\mu_i$,看看该子句的体 $B_{i1}:\varphi_{i1}\&\ldots\&B_{ir_i}:\varphi_{ir_i}$ 是否成立. 于是,问题转化为对 $B_{i1}:\varphi_{i1}\&\ldots\&B_{ir_i}:\varphi_{ir_i}$ 的考察. 如此继续下去,最后归结到对 G 中单位子句 $A:\mu\leftarrow$ 的考察.

在传统逻辑程序或 Subrahmanian 的标记逻辑中[3,4],如果存在 A 的一个(SLD－或 $SLDNF$－或 $SLDnh$－)反驳,即 A 得到某些证据的支持,则 A 为真. 但是,由 ALP 的特点,即使存在这么一条"真值"传递链,也不能据此肯定原命题. 因为,对每一个以 $A_i:\rho_i$ 为头的子句,G 中可能有"反对"它的子句,即有以 $A_i:\beta_i(\beta_i>\rho_i,\beta_i\in D\mu_i)$ 为头的子句. 如果以 $A_i:\beta_i$ 为头的子句的体成立,那么,至少该子句所"反对"的那些子句(即标记小于 β_i)不能作为命题成立的依据. 也就是说,在 ALP 中,确定一个命题的成立与否,必须全面考察支持它和反对它的证据. 形象地说,就是支持和反对的双方进行博弈或辩论,胜者即为最后的结论. 基于这种想法,我们引入一种特殊的博弈树,叫做 SLD－博弈树,在它上面施行某种删除(剪枝)策略,可最终确定命题成立与否.

---

* 本文研究得到国家攀登计划、国家 863 高科技项目基金资助. 作者周生炳,1962 年生,讲师,现在清华大学计算机系智能技术与系统国家重点实验室作博士后,主要研究领域为非单调推理,逻辑程序. 戴汝为,1932 年生,研究员,中国科学院院士,主要研究领域为人工智能,模式识别,智能控制,思维科学.

本文通讯联系人:周生炳,北京 100084,清华大学计算机系

本文 1995－01－17 收到修改稿

以查询 $A:\mu(\mu\in\{t,f\})$* 为根节点的 SLD－博弈树中所有成功路径(从根节点到空节点的路径)确定一个表达式$(A:\mu)\theta$(θ 是该路径的计算回答). 树中路径可大致分为“支持”路径和“反对”路径,这些路径相应的表达式确定的元素集合(即生成集合)并不相同. 形象地说,支持和反对双方有各自的“势力范围”. 划分好势力范围后,查询的确定就可应用简单树的删除策略来解决.[2,5~7]例如,设某个支持路径为$(A:\mu)\theta$,所有反对路径为$(A:\mu)\theta_1,\ldots,(A:\mu)\theta_k$,如果$[A\theta]-\bigcup_{i=1}^{k}[A\theta_i]\neq\varnothing$,那么对该集合中的任意元素 H,$M(H)\in D\mu\cup D\urcorner$,这里 M 是程序的支持模型. 因此,关键问题在于如何划分各自的势力范围. 也就是说,在多个反对链中,识别出真正的反对链、部分反对链以及伪反对链. 这对于树的剪枝是至关重要的. 本文的目的就是提供回答上述问题的理论基础,它对 ALP 的查询赋值问题起着关键作用.

1 简单表达式的覆盖

称文字和项为简单表达式. 假定读者熟悉诸如合一、合一的合成、mgu(最一般合一)、合一算法、表达式的换名式(Renaming)等概念见文献[4,8]. 本文的讨论在固定的 Herbrand 域或 Herbrand 基上进行,该 Herbrand 域由 $n(n>0)$个常量和 $m(m\geqslant 0)$个函数生成,Herbrand 基由这些常量、函数以及若干个(至少 1 个)文字生成.

定义 1.1. 称表达式 E 的所有基础实例的集合为 E 的生成集合,记为$[E]$.

由定义可知,表达式的生成集合就是它在 Herbrand 域或 Herbrand 基中的外延. 显然,$H\in[E]$当且仅当存在基础替换 λ,使 $H=E\lambda$.

定义 1.2. 称表达式 E_2 覆盖 E_1,记为 $E_1\leqslant E_2$,如果$[E_1]\subseteq[E_2]$.

引理 1.3. **设 E_1 与 E_2 可合一,E_1,E_2 不含公共变量,它们的 mgu 为 σ,则$[E_1\sigma]=[E_2\sigma]=[E_1]\cap[E_2]$.

定理 1.4. 设 E_1,E_2 不含公共变量,则 E_1 和 E_2 可合一当且仅当$[E_1]\cap[E_2]\neq\varnothing$.

定义 1.5. 项 t 的深度 $deep(t)$定义如下:

(1) $deep(t)=0$,t 是常量或变量;

(2) $deep(t)=\max\limits_{i}\{deep(t_i)\}+1$,$t=f(t_1,\ldots,t_n)$.

定义 1.6. 文字 $A(t_1,\ldots,t_k)$的深度定义为:$deep(A(t_1,\ldots,t_k))=\max\limits_{i}\{deep(t_i)\}$.

定理 1.7. $[E\theta]=[E]$当且仅当 θ 为空替换或换名替换.

推论 1.8. $E_1\leqslant E_2$ 当且仅当存在替换 θ,使 $E_2\theta=E_1\xi$,其中 ξ 是换名替换或空替换.

2 表达式的分解

对一个表达式 E 与多个表达式 $E_i(1\leqslant i\leqslant k)$之间的覆盖关系

$$[E]\subseteq\bigcup_{i}[E_i]$$

怎样判断呢? 这个问题比 2 个表达式之间的关系复杂得多. 为此,我们把表达式按 Herbrand

---

* 查询的一般形式是 $A_1:\mu_1\&\ldots\&A_n:\mu_n$,其中 $\mu_i\in\{t,f\}$.

** 限于篇幅,本文大多数结果的证明均略去,详细证明见文献[2].

域的生成方式分解,再对分解项进行比较.

设语言 L 含 n 个常量 $C=\{c_1,...,c_n\}$,m 个函数 $F=\{f_1^{n_1},...,f_m^{n_m}\}$,其中,$n_i$ 表示函数 f_i 的元数,于是 L 的 Herbrand 域由这若干个常量和函数生成.按照定义 1.1,下述关系是显而易见的:

$$Con.\quad [c]=\{c\}\qquad c\in C$$

$$Var.\quad [x]=C\cup[f_1]\cup...\cup[f_m]\qquad x\text{ 是变量符}$$

$$Fun.\quad [f(x_1,...,x_k)]=\{\bigcup_{j=1}^{n}[f(c_j,x_2,...,x_k)]\}\cup\{\bigcup_{j=1}^{m}[f(f_j,x_2,...,x_k)]\}$$

其中,内层函数 f_j 的变量不同于 $x_2,...,x_k$.

我们看到,分解式 Var 和 Fun 可以无限进行下去.但是,对于固定的深度,分解是有限的.例如,对 $f_i(x_1,...,x_k)$,限定分解式的每一项的深度不超过 2,则共有 $(n+m)^k$ 个分解项.可见随着深度的增加,分解项的项数呈指数增长.

定义 2.1. 称 $Spec(E)=\{E_1,...,E_p\}$ 是表达式 E 的谱分解,如果满足:

(i) $[E]=\bigcup_{i=1}^{p}[E_i]$

(ii) $[E_i]\cap[E_j]=\varnothing\ (i\neq j)$

记为 $E=E_1\dotplus...\dotplus E_p$ 或 $E=\sum_{F\in Spec(E)}F$.

上述对变量 x 和函数 $f(x_1,...,x_n)$ 的分解式可表示为

$$Var.\quad x=c_1\dotplus...\dotplus c_n\dotplus f_1(x_{11},...,x_{1n_1})\dotplus...\dotplus f_m(x_{m1},...,x_{mn_m})$$

$$\text{或 } Var.\quad x=\sum_{t\in Spec(x)}t$$

$$\begin{aligned}Fun.\quad f(x_1,...,x_k)=&f(c_1,x_2,...,x_k)\dotplus...\dotplus f(c_n,x_2,...,x_k)\\&\dotplus f(f_1(x_{11},...,x_{1n_1}),x_2,...,x_k)\dotplus...\\&\dotplus f(f_m(x_{m1},...,x_{mn_m}),x_2,...,x_k)\end{aligned}$$

引理 2.2. 设 $E=E(t_1,...,t_k)(x_1,...,x_\tau)$,$x_1,...,x_\tau$ 是 E 的全部变量.把某个变量 x_i 按上述关系 $Var.$ 分解,则

$$\begin{aligned}E=&E(t_1,...,t_k)(x_1,...,x_{i-1},c_1,x_{i+1},...,x_\tau)\dotplus...\\&\dotplus E(t_1,...,t_k)(x_1,...,x_{i-1},c_n,x_{i+1},...,x_\tau)\\&\dotplus E(t_1,...,t_k)(x_1,...,x_{i-1},f_1(y_{11},...,y_{1\sigma_1}),x_{i+1},...,x_\tau)\dotplus...\\&\dotplus E(t_1,...,t_k)(x_1,...,x_{i-1},f_m(y_{m1},...,y_{m\sigma_m}),x_{i+1},...,x_\tau)\end{aligned}$$

即 $E=\sum_{u\in Spec(x_i)}E(t_1,...,t_k)(x_1,...,x_{i-1},u,x_{i+1},...,x_\tau)$ 是 E 的谱分解,其中 σ_i 是 f_i 的元数,$y_{ij}\neq x_s(i=1,...,m,j=1,...,\sigma_i,s=1,...,\tau)$,即分解引进的变量不同于已有的变量.

称变量 x 的分解式($Var.$)是深度为 1 的标准分解.我们看到,($Fun.$)实际上是引理2.2的一个特例.对给定深度 d,可反复应用($Var.$)直至每项深度$=d$ 或不能分解为止(因为某些变量换成常量).这样的分解称为变量 x 的深度 d 谱,记为 $x=\sum_{t\in Spec^d(x)}t$.

定义 2.3. 表达式 E_1,E_2 的差异集 $dis(E_1,E_2)$ 是如下二元组的集合:首先找出 E_1 和 E_2 不同的第 1 个符号,然后从 E_1,E_2 中抽出占有这个符号位置的子表达式,这 2 个子表达式记为 $\{t_1^1,t_1^2\}$,它是 $dis(E_1,E_2)$ 的第 1 个元素.把这 2 个子表达式抽出来后,以空位代替,再对这

2 个含空位的表达式施行上述的步骤,得到新的二元组$\{t_2^1,t_2^2\}$,它是$dis(E_1,E_2)$的第 2 个元素,以空位代入t_2^1,t_2^2的位置.如此进行下去,得到的全部二元组的集合就是差异集$dis(E_1,E_2)$.

从定义可以看出,这里的差异集与文献[4,8]定义的差异集不同,后者是前者的第 1 个二元组.

定义 2.4. 设E_1,E_2可合一,对$dis(E_1,E_2)$中第 1 个这样的二元组$\{t_1,t_2\}$,其中,t_1,t_2中有且仅有一个是变量,另一个不是基础项,不妨设t_1是变量x,则把E_1中变量x按t_2的深度分解,即 $E_1=\sum_{u\in Spec^{deep(t_2)}(x)} E_1(t_1,...,t_k)(u)=\sum_{u\in Spec^{deep(t_2)}(x)} E_1(t_1,...,t_k)\{x/u\}$

上述分解称为E_1相对于E_2的一次谱分解.如果t_2是变量,则上述分解对E_2进行,称为E_2相对于E_1的一次谱分解.

现在我们就可以定义分解算法并据此判断$[E]\subseteq\bigcup_i[E_i]$.

E相对于$E_1,...,E_k$的分解算法定义如下:

```
Procedure Decompose(E;E1,...,En)
  置 Left={E},Right={E1,...,En}
  Repeat
    对任意 E∈Left,寻找 F∈Right,使 E 相对于 F 可分解或 F 相对于 E 可分解,分别置
      Left=(Left-{E})∪Spec(E)
    或 Right=(Right-{F})∪Spec(F)
  Until Left 中任意表达式与 Right 中任意表达式相互均不可分解.
```

其中$Spec(E)$表示表达式E按定义 2.4 分解后所有分解项的集合.

下面,我们研究可合一表达式合一以后的深度变化情况.

设x在项$t=t(t_1,...,t_k)$中出现多次,那么x的每次出现都有一个深度,多次出现对应多个深度,称为x的深度分布.定义如下:

定义 2.5. 设$t=x$,则变量x在项t中出现的深度为 0.如果x在项$t=f(t_1,...,t_k)$中只出现一次,即x在项t_i中出现,x在t中出现的深度为x在t_i中出现的深度+1.

定义 2.6. 设x在项$t=f(t_1,...,t_k)$中出现s次.把x在t中的s次出现分别替换为$x_1,...,x_s$,使$\{x_1,...,x_s\}\cap Var(t)^*=\varnothing$,$x_i\neq x_j(i\neq j)$,替换后的项记为$t'$,则$x_1,...,x_s$在$t'$中只出现一次.设它们在$t'$中出现的深度分别为$d_1,...,d_s$,则称$\langle d_1,...,d_s\rangle$为$x$在项$t$中的深度分布,记为$d_t^x$.对文字$A(t_1,...,t_k)$,设$d_{t_i}^x=\langle d_{i1},...,d_{is_i}\rangle$,则$x$在$A$中的深度分布$d_A^x=\langle d_{11},...,d_{1s_1},...,d_{k1},...,d_{ks_k}\rangle$.

例 2.7:令$t=f(x,g(y,h(x,h(x,x))))$,x在t中出现 4 次,$d_t^x=\langle 1,3,4,4\rangle$.

定义 2.8. 设$d_E^x=\langle d_1,...,d_r\rangle$,令$d(x,E)=max\{d_1,...,d_r\}$,叫作$x$在$E$中的最大深度,$d(E)=\sum_{x\in Var(E)} d(x,E)$称为$E$的深度上界.

引理 2.9. 设E_1,E_2可合一,它们的mgu为θ,则$deep(E_1\theta)=deep(E_2\theta)\leqslant d(E_1)+d(E_2)+max\{deep(E_1),deep(E_2)\}$.

---

* $Var(E)$表示表达式E中出现的全部变量.

引理 2.10. 设 E_1,E_2 可合一,那么过程 Decompose(E_1,E_2)结束后,*Left* 和 *Right* 中分别有且仅有一个元素 E_1,E_2 可合一. 换句话说,设 E_1,E_2 的 *mgu* 为 θ,那么 *Left* 和 *Right* 中均存在唯一的元素 u,v,使$[u\sigma_1]=[v\sigma_2]=[E_1\theta]=[E_2\theta]$,其中 σ_1,σ_2 分别为 u,E_1 和 v,E_2 的 *mgu*.

利用上述结果,容易推出:

定理 2.11. 如果 E 与 $E_i(i=1,2,...,k)$可合一,那么过程 Decompose$(E;E_1,...,E_k)$ 必在有限步骤内结束.

注意,为便于判别覆盖关系,可以假设集合 *Left* 中的元素与 *Right* 中的元素无公共变量,这一点总是可以做到的.

定理 2.12. 设 E 与 $E_1,...,E_k$ 可合一,E 与 $E_i(i=1,...,k)$无公共变量,则$[E]\subseteq\bigcup_i[E_i]$当且仅当过程 Decompose$(E;E_1,...,E_n)$结束后,$\forall\ D\in Left$,存在 $F\in Right$,使 $D\leqslant F$.

证明:显然,下面的关系成立:$[E]=\bigcup_{D\in Left}[D]$,$\bigcup_i[E_i]=\bigcup_{F\in Right}[F]$
由此可知充分性成立. 以下证明必要性,仅对项进行讨论,表达式为文字的情形是类似的.

设 $[t]\subseteq\bigcup_i[t_i]$,则对过程 Decompose 结束后的集合 *Left*, *Right*, $\forall\ v\in Left$, $\exists\ u\in Right$,使 v,u 可合一. 如果不然,即有 $v\in Left$,使对 *Right* 中任意 u,v 与 u 不可合一. 根据定理1.4,$[v]\cap[u]=\varnothing$. 因此$[v]\cap(\bigcup_{u\in Right}[u])=\varnothing$,即$[t]\not\subseteq\bigcup_i[t_i]$,与假设矛盾.

设 $v\in Left$,令 Unificable$(v)=\{u\,|\,u\in Right, u$ 与 v 可合一$\}=\{u_1,...,u_k\}$.

当 $deep(v)=0$ 时,若 $v=c$,则 u_i 为常量 c 或为变量,显然有$[c]\subseteq[u_i]$;若 $v=x$ 为变量,则 u_i 为常量或变量,并且至少有一个 u_i 为变量,否则将有$[v]\not\subseteq\bigcup[u_i]$,从而与假设矛盾. 设 $u_i=y$,显然$[v]=[x]=[y]=[u_i]$,即 $v\leqslant u_i$.

当 $deep(v)\geqslant 1$ 时,v,u_i 有如下形式:

$$v=f(v_1,...,v_\tau)$$
$$u_i=f(u_{i1},...,u_{i\tau})$$

因为 v 与 u_i 相互不可分解,则 v 与 u_i 的差异集中任意元素$\{x,t\}$,t 是变量或基础项. 把 v 与 u_i 的差异集中所有这样的元素$\{x,t\}$,其中 x 是 v 中的变量,列为如下形式的“矩阵”(称为差异矩阵):

$$\left|\begin{array}{l}
\{x_1,t_{11}^{(1)}\}...\{x_1,t_{1r_1}^{(1)}\}\{x_2,t_{21}^{(1)}\}...\{x_2,t_{2r_2}^{(1)}\}...\{x_l,t_{l1}^{(1)}\}...\{x_l,t_{lr_l}^{(1)}\}\\
\{x_1,t_{11}^{(2)}\}...\{x_1,t_{1r_1}^{(2)}\}\{x_2,t_{21}^{(2)}\}...\{x_2,t_{2r_2}^{(2)}\}...\{x_l,t_{l1}^{(2)}\}...\{x_l,t_{lr_l}^{(2)}\}\\
\qquad\qquad\qquad\qquad\cdots\cdots\\
\{x_1,t_{11}^{(k)}\}...\{x_1,t_{1r_1}^{(k)}\}\{x_2,t_{21}^{(k)}\}...\{x_2,t_{2r_2}^{(k)}\}...\{x_l,t_{l1}^{(k)}\}...\{x_l,t_{lr_l}^{(k)}\}
\end{array}\right|$$

u_i 中与 x_j 对应的项的集合$\{t_{j1}^{(i)},...,t_{jr_j}^{(i)}\}$记为 $term(x_j,u_i)$,如果有 $1\leqslant i\leqslant k$,使 $term(x_\xi,u_i)\cap term(x_\zeta,u_i)=\varnothing(\xi\neq\zeta)$,并且 $term(x_j,u_i)(j=1,2,...,l)$中的项均为变量,那么,令

$$\theta=\{t_{j\xi}^{(i)}/x_j\,|\,j=1,2,...,l,\xi=1,2,...,r_j\}$$

则有 $u_i\theta=v$,由推论 1.8,$v\leqslant u_i$. 下面证明这个事实.

首先,必有 u_i,$1\leqslant i\leqslant k$,使 $term(x_j,u_i)(j=1,2,...,l)$中的项都是变量. 否则,每一行都

有 x_j 对应基础项，设矩阵中全部基础项为 $\{t_1,\ldots,t_\pi\}$，则任取基础替换 λ

$$\lambda=\{x_1/w_1,\ldots,x_l/w_l\}$$

使 $w_j\overline{\in}\{t_1,\ldots,t_\pi\}$，$j=1,2,\ldots,l$，则 $v\lambda\overline{\in}[u_i]$，$1\leqslant i\leqslant\pi$，这是不可能的. 不失一般性，可设矩阵中不含基础项.

把矩阵中含 $x_j(1\leqslant j\leqslant l)$ 的若干列记为 Π_j，则可移动 Π_j，使前面的 $\pi(\pi\leqslant l)$ 块中每个 x_j，均有 u_ξ，使 $term(x_j,u_\xi)\cap term(x_i,u_\xi)\neq\varnothing(j\neq i,i,j\leqslant\pi)$. 这是可以做到的，因为不满足上述条件的 Π_i 可以移到后面. 不妨设 Π_i 均满足上述条件.

取这样的基础替换 λ　　　$\lambda=\{x_1/w_1,\ldots,x_l/w_l\}$

使 $w_i\neq w_j(i\neq j)$，则 $v\lambda\overline{\in}[u_i]$，$i=1,2,\ldots,k$. 事实上，对任意 u_i，设 $term(x_\xi,u_i)\cap term(x_\zeta,u_i)\neq\varnothing$，那么，$u_i$ 中有一个变量 y 对应 v 中 2 个不同变量 x_ξ,x_ζ，即 $dis(v,u_i)$ 中有元素 $\{x_\xi,y\}$，$\{x_\zeta,y\}$. 因此 $[u_i]$ 中任意元素在对应 x_ξ 和 x_ζ 的位置上均为相同项，而 $v\lambda$ 在这 2 个位置上分别为 w_ξ 和 w_ζ，可见 $v\lambda\overline{\in}[u_i]$. 用下面的图式表示，可以清楚地看出这一点：

$$v\lambda=f(v_1(\ldots,w_\xi,\ldots,w_\zeta,\ldots),\ldots,v_\tau(\ldots,w_\xi,\ldots,w_\zeta,\ldots))$$

$$u_i=f(v_1(\ldots,y,\ldots,y,\ldots),\ldots,v_\tau(\ldots,y,\ldots,y,\ldots))$$

这是不可能的. 这说明有 u_i，使 $term(x_\xi,u_i)\cap term(x_\zeta,u_i)=\varnothing(1\leqslant\xi,\zeta\leqslant l)$，于是有 $v\leqslant u_i$. □

3　表达式的划分

对表达式 E 和表达式族 $\{E_i\}_{i=1}^k$，E 与 E_i 可合一，我们的目的是找出 $[E]$ 与 $\bigcup_{i\leqslant k}[E_i]$ 的全部关系. 首先，E 有一个初步的划分：

$$[E]=([E]-\bigcup_{i=1}^{n}[E_i])\cup([E]\cap(\bigcup_{i=1}^{n}[E_i]))$$

如果 $[E]-\bigcup_{i\leqslant k}[E_i]\neq\varnothing$，那么，过程 Decompose$(E;E_1,\ldots,E_k)$ 结束以后，至少有一个 $v\in Left$，对任意 $u\in Right$，有 $v\not\leqslant u$. 沿用定理 2.12 的记号，有

$$[v]\not\subseteq\bigcup_{u\in Unificable(v)}[u]$$

定理 2.12 的证明中实际上已经给出了求一个元素 $w\in[v]-\bigcup_{u\in Unificable(v)}[u]$ 的方法，即任取基础替换 λ：　　　$\lambda=\{x_1/w_1,\ldots,x_l/w_l\}$

使 $w_i\neq w_j(i\neq j)$，并且 $w_i\overline{\in}\{t_1,\ldots,t_\tau\}$，其中 t_i 是矩阵中全部基础项. 这个元素当然也是 $[E]-\bigcup_{i\leqslant k}[E_i]$ 中的元素.

由于分解过程的引入，我们可把对 E 的划分转化为对 $Left$ 中每个元素的划分. 对 $v\in Left$ 同样有一个初步的划分：

$$[v]=([v]-\bigcup_{u\in Unificable(v)}[u])\cup([v]\cap\bigcup_{u\in Unificable(v)}[u])$$

如上所述，若 $[v]-\bigcup_{u\in Unificable(v)}[u]\neq\varnothing$，我们能确定该集合中的一个元素，而对 $[v]\cap\bigcup_{u\in Unificable(v)}[u]$，我们将结合集合划分公式和差异矩阵，把它划分成一系列不相交集合，并在每个集合中取一个生成元.

定义 3.1. 设 A 是一个集合，如果存在一族集合 B_i，$i=1,2,\ldots,k$，使 $A=\bigcup_{i=1}^{k}B_i$ 并且 $B_i\cap B_j=\varnothing(i\neq j)$，则称集族 $\{B_i\}_{i=1}^k$ 是 A 的一个划分.

任意集族$\{B_i\}_{i=1}^n$未必是$B=\bigcup_{i=1}^n B_i$的划分，但是，下面的结果保证可对B_i施行集合运算，使运算结果成为B的划分.

定理 3.2.

$$\bigcup_{i=1}^n B_i=\bigcup_{k=1}^n \bigcup_{i_l<i_{l+1}} (\bigcap_{\sigma=1}^k B_{i_\sigma}-(\bigcup_{\substack{\tau=1\\ j_\tau\neq i_\sigma}}^{n-k} B_{j_\tau}))=$$

$$\bigcup_{k=1}^n \bigcup_{\substack{i_1<i_2<\ldots<i_k\\ j_1<j_2<\ldots<j_{n-k}\\ \{i_1,\ldots,i_k\}\cap\{j_1,\ldots,j_{n-k}\}=\varnothing\\ \{i_1,\ldots,i_k\}\cup\{j_1,\ldots,j_{n-k}\}=\{1,2,\ldots,n\}}} (B_{i_1}\cap B_{i_2}\cap\ldots\cap B_{i_k}-(B_{j_1}\cup\ldots\cup B_{j_{n-k}}))$$

上述等式中，右边各项显然互不相交，因此右边的集族构成集合$\bigcup_{i=1}^k B_i$的划分. 注意定理3.2中k个集合的并集被分割为2^k-1个互不相交集合的并，显然，如果B_i中有不相交集合，则其划分后的集合数将大大降低.

这样，$\bigcup_{i=1}^k [v]\cap[u_i]$就可根据此式进行划分. 划分以后，不相交集合很多，但是这些集合中相当多是空集. 我们将给出判断空集的的准则，并对非空集合求出生成元.

v与$u_i(i=1,2,\ldots,k)$的差异矩阵为

$$\begin{vmatrix} \{t_1,t_{11}^{(1)}\}\ldots\{t_1,t_{1r_1}^{(1)}\}\{t_2,t_{21}^{(1)}\}\ldots\{t_2,t_{2r_2}^{(1)}\}\ldots\{t_l,t_{l1}^{(1)}\}\ldots\{t_l,t_{lr_l}^{(1)}\} \\ \{t_1,t_{11}^{(2)}\}\ldots\{t_1,t_{1r_1}^{(2)}\}\{t_2,t_{21}^{(2)}\}\ldots\{t_2,t_{2r_2}^{(2)}\}\ldots\{t_l,t_{l1}^{(2)}\}\ldots\{t_l,t_{lr_l}^{(2)}\} \\ \cdots\cdots \\ \{t_1,t_{11}^{(k)}\}\ldots\{t_1,t_{1r_1}^{(k)}\}\{t_2,t_{21}^{(k)}\}\ldots\{t_2,t_{2r_2}^{(k)}\}\ldots\{t_l,t_{l1}^{(k)}\}\ldots\{t_l,t_{lr_l}^{(k)}\} \end{vmatrix}$$

其中每个元素$\{t_\xi,t_{\xi j}^{(i)}\}(1\leqslant j\leqslant r_\xi)$是$v$与$u_i$的差异集中的元素，$t_\xi$是$v$中的子表达式，$t_{\xi j}^{(i)}$是$u_i$中的子表达式，它们或为变量，或是基础项. 矩阵中有些位置为空，因为v与u_i对应位置上的子表达式是相同的基础项.

首先看看v与u_i合一后u_i中那些在差异矩阵中的子表达式的变化. 显然，所有$t_{\xi j}^{(i)}(1\leqslant j\leqslant r_\xi),\xi=1,2,\ldots,l$都被替换成同一个项. 如果$term(t_\xi,u_i)\cap term(t_\zeta,u_i)\neq\varnothing$，则合一以后，所有$t_{\xi j}^{(i)}(j=1,2,\ldots,r_\xi),t_{\zeta j}^{(i)}(j=1,2,\ldots,r_\zeta)$都被替换为相同的项(基础项或变量). 为便于比较，我们可以使u_i合一后只出现v中的变量.

由于v与u_i相互均不可分解，而且v与u_i可合一，那么$u_i\theta_i$之间可合一，它们之间的区别在于变量的出现次数和顺序不同，这里θ_i指v与u_i的mgu. 我们把v中用以替换u_i中子表达式$t_{\xi j}^{(i)}$的子表达式按照$t_{\xi j}^{(i)}$在差异矩阵中的顺序排列，得到如下矩阵：

$$\begin{vmatrix} t_{11}\ldots t_{11} & t_{12}\ldots t_{12} & \ldots & t_{1l}\ldots t_{1l} \\ t_{21}\ldots t_{21} & t_{22}\ldots t_{22} & \ldots & t_{2l}\ldots t_{2l} \\ & \cdots\cdots & & \\ t_{k1}\ldots t_{k1} & t_{k2}\ldots t_{k2} & \ldots & t_{kl}\ldots t_{kl} \end{vmatrix}$$

其中$t_{ij}\in\{t_1,\ldots,t_l\}\cup\{w_1,\ldots,w_s\}$，$w_i$是$u_j$在差异矩阵中的所有基础项，这个矩阵称为替换矩阵.

在差异矩阵中，若$term(t_\xi,u_i)\cap term(t_\zeta,u_i)\neq\varnothing$，那么，替换矩阵中$t_{i\xi}=t_{i\zeta}$. 由于$[v]-\bigcup_{u\in Unificable(v)}[u]\neq\varnothing$，每一行都满足这样的性质. 划分集合$[v]\cap\bigcup_{u\in Unificable(v)}[u]$，对应划分公

式的一项为 $[v]\cap[u_{j_1}]\cap\ldots\cap[u_{j_\eta}]-([u_{i_1}]\cap[v]\cup\ldots\cup[u_{i_{k-\eta}}]\cap[v])$

交集 $[v]\cap[u_{j_1}]\cap\ldots\cap[u_{j_\eta}]=[v\theta]=[u_{j_i}\theta](1\leqslant i\leqslant\eta)$,这里 θ 是 $v,u_{j_1},\ldots,u_{j_\eta}$ 的 mgu,它可由上述替换矩阵确定.

取替换矩阵的第 $j_1,\ldots,j_\eta$ 行构成新的矩阵,则 $v,u_{j_1},\ldots,u_{j_\eta}$ 合一后,这个新矩阵的若干块(至少 2 块)中的元素变得相同,不妨设前 $\xi(\xi\leqslant l)$ 块的元素相同. 在余下的 $k-\eta$ 行(记为 $i_1,\ldots,i_{k-\eta}$ 行)中,如果某一行(不妨设为 i_1 行)中使 $t_{i_1\zeta}=t_{i_1\omega}(\omega\neq\zeta)$ 的元素都在前 ξ 块,即 $\zeta\leqslant\xi,\omega\leqslant\xi$,那么,$[v]\cap[u_{j_1}]\cap\ldots\cap[u_{j_\eta}]\subseteq[v]\cap[u_{i_1}]$,因此 $[v]\cap[u_{j_1}]\cap\ldots\cap[u_{j_\eta}]-([v]\cap[u_{i_1}]\cup\ldots\cup[v]\cap[u_{i_{k-\eta}}])$ 是空集. 否则,$i_1,\ldots,i_{k-\eta}$ 中的每一行都有 $\zeta>\xi$,使 $t_{i_j\omega}=t_{i_j\zeta}(\omega\neq\zeta)$,$1\leqslant j\leqslant k-\eta$. 取基础替换 λ 如下:

$$\lambda=\{t/w,t_{\xi+1}/w_1,\ldots,t_l/w_{(l-\xi)}\}$$

其中 t 是前 ξ 块中的变量,后 $l-\xi$ 块中的变量互不相同,$w_i\neq w_j(i\neq j)$,$w_i\neq w$. 于是,$u_{j_i}\theta\lambda\in[v]\cap([u_{j_1}]\cap\ldots\cap[u_{j_\eta}]-[u_{i_1}]\cup\ldots\cup[u_{i_{k-\eta}}])$. 事实上,由 λ 的定义,显然有 $u_{j_i}\theta\lambda\in[v]\cap[u_{j_1}]\cap\ldots\cap[u_{j_\eta}]$. 注意到 $[u_{i_j}]$ 中的任意元素在对应 $t_{i_j\zeta}$ 和 $t_{i_j\omega}$ 的位置上均为相同项,而 $u_{j_i}\theta\lambda$ 在这 2 个位置上均为不同项(因为 $\zeta>\xi$),因此,$u_{j_i}\theta\lambda\overline{\in}[v]\cap[u_{i_j}]$,$j=1,2,\ldots,k-\eta$,即 $u_{j_i}\theta\lambda\overline{\in}[v]\cap[u_{i_1}]\cup\ldots\cup[v]\cap[u_{i_{k-\eta}}]$. 如果把上述划分中的每个集合看成一个等价类,则对非空集合,这样得到的任意元素就是该等价类的生成元.

至此我们已解决本节开始提出的问题,它在 ALP 的查询赋值问题中将起重要作用.

4 合取式的覆盖与分解

本节讨论合取式 $E_1=A_1\&\ldots\&A_k$,$E_2=B_1\&\ldots\&B_k$ 之间,$E=B_1\&\ldots\&B_k$ 与 $E_i=B_{i1}\&\ldots\&B_{ik}$,$i=1,2,\ldots,m$ 之间的覆盖关系及判别算法.

定义 4.1. 设 $E=A_1\&\ldots\&A_k$ 是文字 $A_1,\ldots,A_k$ 的合取,令

$$[E]=\{E\theta=A_1\theta\&\ldots\&A_k\theta\mid\theta\text{ 是基础替换}\}$$

称 $[E]$ 为合取式 E 的生成集合.

定义 4.2. 合取式 E_2 覆盖 E_1,记为 $E_1\leqslant E_2$,如果 $[E_1]\subseteq[E_2]$.

下面的结果是明显的:

引理 4.3. 设 $E_1=A_1\&\ldots\&A_k$,$E_2=B_1\&\ldots\&B_m$. 如果 $E_1\leqslant E_2$,则 $k=m$,并且 $A_i\leqslant B_i$,$1\leqslant i\leqslant k$.

定义 4.4. 设 $E=A_1(t_{11},\ldots,t_{1r_1})\&\ldots\&A_k(t_{k1},\ldots,t_{kr_k})(x_1,\ldots,x_r)$,$x_1,\ldots,x_r$ 是 E 中出现的所有变量. 引进一个函数 χ_E,称为 E 的特征函数,它的定义域为 $U_L x\ldots xU_L$,其中 U_L 是语言 L 的 Herbrand 域,值域是一个特殊的元素集合,记为 Π_E,每个元素形如

$$\chi_E(a_{11},\ldots,a_{1r_1},\ldots,a_{k1},\ldots,a_{kr_k})$$

其中 $a_{ij}\in[t_{ij}]$,$i=1,2,\ldots,k$,$j=1,2,\ldots,r_k$,并且

$$(a_{11},\ldots,a_{1r_1},\ldots,a_{k1},\ldots,a_{kr_k})=(t_{11},\ldots,t_{1r_1},\ldots,t_{k1},\ldots,t_{kr_k})\theta$$

$\theta=\{x_1/v_1,\ldots,x_r/v_r\}$ 是基础替换.

χ_E 是语言 L 以外的函数,合取式 E 不同,则对应的 χ_E 不同,并且,χ_E 的元数视 E 不同而不同. 因此,χ_E 不是严格意义上的函数,它是我们为叙述方便而引进的一个工具.

定义 4.5. 合取式 E_1 称为 E_2 的变体(*Variant*),如果有换名替换 θ,使 $E_1\theta=E_2$.

引理 4.6. E_1 是 E_2 的变体,当且仅当存在换名替换 θ,σ,使 $E_1\theta=E_2,E_2\sigma=E_1$.

证明:类似于文献[4]中相应简单表达式的结果的证明.

引理 4.7. E_1 是 E_2 的变体,当且仅当 $[E_1]=[E_2]$当且仅当 $[\chi_{E_1}]=[\chi_{E_2}]$.

定义 4.8. 设 $\chi_E=\chi_E(t_1,...,t_r)(x_1,...,x_r)$,并且 $x_i(1\leqslant i\leqslant r)$按深度 l 分解,那么

$$\chi_E=\sum_{u\in Spec^l(x_i)}\chi_E(t_1,...,t_r)(x_1,...x_{i-1},u,x_{i+1},...,x_r)$$

$$=\sum_{u\in Spec^l(x_i)}\chi_E(t_1,...,t_r)\{x_i/u\}$$

是 χ_E 相应的分解,即 $\chi_E(t_1,...,t_r)$中含 x 的项同时分解.

由此可见,引进 χ_E 可把合取式的分解转化为普通项(因而是简单表达式)的分解,并且各合取项间的联系亦在 χ_E 中保留下来.因此,合取式之间的覆盖关系等问题的判断就可转化为简单项之间覆盖关系的判断,从而直接引用前面的结果即可.这样,以下的结论就不难想象了.

定理 4.9. 合取式 E_1,E_2 可合一,当且仅当$[E_1]\cap[E_2]\neq\varnothing$,当且仅当$[\chi_{E_1}]\cap[\chi_{E_2}]\neq\varnothing$.

定理 4.10. 设 E_1,E_2 是合取式,那么 $E_1\leqslant E_2$ 当且仅当 $\chi_{E_1}\leqslant\chi_{E_2}$.

定理 4.11. 设 $E,E_1,...,E_k$ 是合取式,E_i 与 E 可合一,那么$[E]\subseteq\bigcup_i[E_i]$当且仅当$[\chi_E]\subseteq\bigcup_i[\chi_{E_i}]$.

参考文献

1 周生炳,戴汝为.基于标记逻辑的非单调推理(Ⅰ)(Ⅱ).计算机学报,1995,(9).

2 周生炳.标记逻辑程序理论研究——说明语义与过程语义[博士论文].中国科学院自动化研究所,1994.

3 Blair H A, Subrahmanian V S. Paraconsistent logic programming. Theoretical Computer Science 68(1989),1989. 135~154.

4 Lloyd J W. Foundations of logic programming. Berlin: Springer-Verlag, 1987.

5 Zhou S, Dai R. The SLD-game tree for the covered acyclic ALP. In: Shi Zhongzhi(ed.), Proceedings of the 3rd Pacific Rim International Conference on Artificial Intelligence, Beijing: International Academic Publishers, 1994. 175~180.

6 周生炳,戴汝为.有限 SLD-博弈树的分裂与化简.何新贵主编,人工智能新进展,第三届全国人工智能联合学术会议(重庆),北京:清华大学出版社,1994.151~157.

7 周生炳,戴汝为.SLD-博弈树及其删除策略.中国科学(A辑),1995,(10).

8 刘叙华,姜云飞.定理机器证明.北京:科学出版社,1987.

COVERING, DECOMPOSITION, AND PARTITION OF EXPRESSIONS

Zhou Shengbing Dai Ruwei

(*Institute of Automation The Chinese Academy of Sciences Beijing* 100080)

Abstract In this paper, a simple expression E, i. e., a term or an atom, is viewed as a set, the "so—called" generated set of E, of the Herbrand universe U_L or Herbrand base B_L of the language L. The authors give a criteria to determine whether an expression covers another one, and present the notion of decomposition expressions and the decomposition algorithm on which a necessary and sufficient condition for the covering relation between one and several other expressions is based. Then, they show how to partition a union of the sets generated by the simple expressions according to the partition formula in the set theory. Finally, they generalize generis these above results into the conjunctions through defining a transformation which converts a conjunction into a simple expression. This paper provides the theoretic foundation of the procedural semantics of authors' annotated logic program.

Key words Generated set, annotated logic program, covering, decomposition, and partition of expression, disagreement matrix.