



论 文

一个在 Horn 子句中求解极大缩减的算法

罗杰*, 李未

北京航空航天大学计算机学院软件开发环境国家重点实验室, 北京 100191

* 通信作者. E-mail: luojie@nlsde.buaa.edu.cn

收稿日期: 2010-08-04; 接受日期: 2010-11-29

国家重点基础研究发展计划 (批准号: 2005CB321901)、国家高技术研究发展计划 (批准号: 2007AA01Z146) 和软件开发环境国家重点实验室自主课题 (批准号: SKLSDE-2008ZX-01) 资助项目

摘要 在信念修正理论中, 一个核心问题是求解一个公式集合关于事实集合的所有极大协调子集, 即极大缩减. 本文尝试从算法的角度来解决这一问题, 研究在 Horn 子句中求解所有极大缩减的算法. 首先, 本文指出并证明了公式集合和事实集合并集的极小不协调子集与公式集合关于事实集合的极大缩减之间的转化关系. 其次, 给出并证明了 Horn 子句集合极小不协调的一个必要条件. 然后, 基于上述两个结论, 本文提出了一个在 Horn 子句中枚举公式集合和事实集合并集的极小不协调子集的交互式算法和一个通过这些极小不协调子集计算所有极大缩减的算法. 最后, 综合这两个算法, 提出了一个在 Horn 子句中求解所有极大缩减的交互式算法.

关键词 极大缩减 极小不协调集合 极小减集 Horn 子句

1 引言

在软件的进化过程中, 存在着两类对象: 一类是用户预期的结果, 另一类是开发人员编写的有待测试的软件. 软件的测试的第一步就是运行软件, 并观察软件运行的结果与用户预期的结果是否一致. 当软件运行结果与用户的预期不一致时, 软件的开发人员需要依据用户预期的结果, 对软件进行修正, 目标是仅修改软件中导致问题的语句, 而保持软件中与用户预期不矛盾的尽可能多的语句不变.

在科学发现过程中, 一般也存在着两类对象: 一类是受到实验结果和观测数据支持的事实集合 Δ , 另一类是研究者提出的有待验证的科学理论 Γ . 一方面, 人们可以使用科学理论进行逻辑推理, 用所得的逻辑结论解释已有的现象或预测新的现象; 另一方面, 科学家可以用实验和观测支持的事实去检验现有的科学理论, 并根据事实对科学理论作出修改. 也就是说, 当事实与科学理论的逻辑结论相矛盾时, 要以事实集合 Δ 为基础, 对科学理论 Γ 进行修正, 目标是取得现有科学理论 Γ 中与事实集合 Δ 不矛盾的极大子集.

为了描述软件的进化过程和科学发现过程, Li^[1] 提出了 R 演算. R 演算是一个关于逻辑连接词符号和谓词符号的一阶推理系统. 它使用一阶语言作为描述语言. 一阶语言由项和公式两类对象组成. 其中项由下述 Backus 范式结构归纳定义:

$$t ::= c \mid x \mid ft_1 \cdots t_n.$$

项描述的是常量、变量和函数. 而公式则由下述 Backus 范式结构归纳定义:

$$A ::= t_1 = t_2 \mid Pt_1 \cdots t_n \mid \neg A_1 \mid A_1 \wedge A_2 \mid A_1 \vee A_2 \mid A_1 \rightarrow A_2 \mid \forall x A_1 \mid \exists x A_1.$$

如果一个公式中不含有自由变元, 那么我们称其为语句. 公式和语句通常被用来描述命题和软件规约的条文. 在一阶语言中, 如果以某个公式集合为前提, 使用一阶逻辑的推理规则, 不能导出矛盾, 那么我们称这个公式集合是协调的, 否则称其是不协调的. 一个协调的语句集合也被称作形式理论. 一般而言, 只要抽象的层次合理, 科学理论可以用形式理论来描述, 其原因在于科学理论必须是协调的, 否则我们就可以从科学理论出发推导出任意结论, 科学理论也就没有意义了.

在 R 演算中, 科学理论用形式理论 Γ 描述. 而事实集合用 Δ 描述, Δ 是由原子语句或原子语句的否定组成的形式理论. 在本文中, 我们将只讨论有穷形式理论, 即只包含有穷个公式的形式理论. 如无特别说明, 文中的形式理论都是指有穷形式理论. $\Delta \mid \Gamma$ 是事实集合 Δ 和科学理论 Γ 的组合, 描述了科学理论发展过程中的一个状态或某个阶段性成果, 在 R 演算中被称为 R 表达式^[2,3]. 如果某个状态下的科学理论与实验结果支持的事实集合有矛盾, 那么我们称描述这一状态的 R 表达式为不协调的 R 表达式. 而科学理论 Γ 中与事实集合 Δ 协调的极大子集在 R 演算中则被称为 Γ 关于 Δ 的极大缩减 (又被称作 R 缩减)^[2,3]. 极大缩减在软件开发中代表不仅要删除软件 P 中与测试用例有矛盾的语句或模块, 还要使这种删除保持最低限度, 即能不删就不删原则, 使得软件的剩余部分保持极大. 在知识库维护中, 极大缩减可以解释为删除知识库中与实验结果有矛盾的规则, 而且把删除保持在最低限度, 使剩余的部分是知识库的与实验结果不矛盾的极大子集. R 演算的目标是, 用形式化方法推导 Γ 与 Δ 协调的极大子集. 一般而言, 极大缩减可以不是唯一的, 从而科学理论 Γ 关于事实集合 Δ 的极大缩减可以有多个. 研究表明, R 演算不仅与一阶证明系统^[4] 类似, 具有自己的可靠性和完全性, 它还具有可达性, 即每一个极大缩减都可以使用 R 演算形式推导出来^[2,3]. 与一阶证明系统一样, R 演算也是一个形式推理系统. R 为形式化地推导极大缩减提供了理论基础, 但还需要设计有效的算法并开发相应的软件工具, 才能够在计算机的辅助下求解极大缩减.

在一阶语言中, 有一种语句集合具有非常独特的性质: 它本身是不协调的, 但是从中删除任意一个语句之后, 剩下的语句集合就是协调的, 这种语句集合被称为极小不协调集合. 极小不协调集合与极大缩减有着紧密的联系. 举例来说, 如果一个 R 表达式 $\Delta \mid \Gamma$ 中, $\Delta \cup \Gamma$ 恰好构成一个极小不协调集合, 那么只要从 Γ 中删除任何一个语句, 就可以得到 Γ 关于 Δ 的一个极大缩减. 我们将在本文中证明对于一般情况也有类似的结论成立, 即对于任何一个不协调的 R 表达式 $\Delta \mid \Gamma$, 如果 $\Delta \cup \Gamma$ 的子集中极小不协调集合是 $\Gamma_1, \dots, \Gamma_m$, 那么我们可以通过 $\Gamma_1, \dots, \Gamma_m$ 计算出 Γ 关于 Δ 的所有极大缩减. 这一结论表明, 只要我们能够计算出 $\Delta \cup \Gamma$ 的子集中的所有极小不协调集合, 那么我们就可以计算出 Γ 关于 Δ 的所有极大缩减. 也就是说, 基于这一结论, 我们可以设计出一种计算极大缩减的算法. 对于一阶语言中的一般语句集合来说, 计算它的子集中的所有极小不协调集合这一问题复杂度很高. 为了降低复杂度, 本文选择了一阶语言的一个特殊子集 —Horn 子句作为研究的对象. Horn 子句是一阶语言中形如 $\neg A_1 \vee \cdots \vee \neg A_m \vee B$ 或 $\neg A_1 \vee \cdots \vee \neg A_m$ 的语句, 其中 $A_1, \dots, A_m, B$ 是原子语句. 其优点在于: 首先, Horn 子句具有简单的结构和独特的性质, 能够用于提高算法的效率; 其次, Horn 子句被大量应用在 PROLOG 语言中, PROLOG 语言编写的知识库就可以视为一个 Horn 子句集合, 知识库中协调性的维护也可以通过极大缩减的求解算法来实现.

综上所述, 本文使用的方法的核心是: 首先求出 $\Delta \cup \Gamma$ 的所有极小不协调子集, 然后根据这些极小不协调子集求出 Γ 关于 Δ 的所有极大缩减. 这一方法依赖于两个性质: 一是 Horn 子句集合极小不协调的必要条件; 二是极小不协调集合与极大缩减的转化关系. 由此, 求解极大缩减可以分为下述

两个过程: 首先, 对于给定的由 Horn 子句组成的 Δ 和 Γ , 根据第一个性质, 求出 $\Delta \cup \Gamma$ 的所有极小不协调子集 $\{\Gamma_1, \dots, \Gamma_m\}$; 然后, 根据第二个性质, 只要从 Γ 依次删除每个 Γ_i 中某个不在 Δ 中的 Horn 子句, 并保证删除的 Horn 子句最少, 就能得到一个极大缩减. 遍历对 Γ 的所有删除可能, 即可得到 Γ 关于 Δ 的所有极大缩减.

本文由 6 节组成. 在第 2 节中, 我们将引入极小不协调子集的概念并讨论它与极大缩减的关系, 同时证明极小不协调集合与极大缩减的转化关系, 并提出由极小不协调集合计算极大缩减的算法. 在第 3 节中, 将介绍一些在后面的讨论中用到的有关 Horn 子句的概念, 并证明 Horn 子句集合极小不协调的必要条件. 第 4 节给出一个用于枚举不协调 Horn 子句集合中所有极小不协调集合的交互式算法. 第 5 节将综合前面提出的两个算法, 提出一个在 Horn 子句中求解极大缩减的交互式算法. 最后, 第 6 节对本文进行了总结, 并对未来可能的工作作进一步探讨.

2 极小不协调集合

在本节中, 我们将讨论上面提到的极小不协调集合与极大缩减的转化关系, 并给出一个从 $\Delta \cup \Gamma$ 的所有极小不协调子集计算 Γ 关于 Δ 的所有 R 缩减的算法.

首先, 引入一些在本文中需要用到的 R 演算中的概念, 如 R 反驳、 R 缩减和 R 表达式等^[3].

定义 1 (R 反驳) 假设 Γ 是一个形式理论, 而 Δ 是一个由有穷个原子语句或原子语句的否定组成的形式理论. 如果 Δ 与 Γ 不协调, 那么我们称 Δ 是 Γ 的一个 R 反驳.

令 $\Delta = \{\neg B\}$, $\Gamma = \{A, A \rightarrow B\}$, 那么 Δ 就与 Γ 不协调, 它是 Γ 的一个 R 反驳.

定义 2 (R 缩减) 称一个形式理论 Λ 是形式理论 Γ 关于 R 反驳 Δ 的一个 R 缩减如果 Λ 是与 Δ 协调的 Γ 的极大子集.

根据上述定义, 我们知道 $\{A \rightarrow B\}$ 就是 $\{A, A \rightarrow B\}$ 关于 $\{\neg B\}$ 的一个 R 缩减.

定义 3 (R 表达式) 假设 Γ 是一个形式理论而 Δ 是一个由有穷个原子语句或原子语句的否定组成的形式理论. 称 $\Delta | \Gamma$ 是一个 R 表达式. 如果 Δ 是 Γ 的一个 R 反驳, 则称 $\Delta | \Gamma$ 为不协调 R 表达式.

下面我们讨论极小不协调集合与 R 缩减的关系. 首先, 给出极小不协调集合的严格定义.

定义 4 (极小不协调集合) 令 Γ 为一个公式集合. 称 Γ 是一个极小不协调集合, 如果 Γ 不协调并且, 对任意 $A \in \Gamma$, $\Gamma - \{A\}$ 协调. 称 Γ' 是 Γ 的一个极小不协调子集, 如果 $\Gamma' \subseteq \Gamma$.

极小不协调集合具有一个很好的性质, 可以通过删除它的任意一个公式来使得它协调. 例如, $\{A, A \rightarrow B, \neg B\}$ 就是一个极小不协调集合, 从中删除任意一个公式后得到的都是协调集合.

引理 1 令 Γ 为一个有穷集合. 如果 Γ 不协调, 那么 Γ 必存在一个极小不协调子集 Γ' .

引理 1 及以下所有引理的证明见附录 A.

上述引理告诉我们, 如果一个有穷公式集合 Γ 不协调, 那么它必然包含有穷个极小不协调子集. 例如, 对于不协调公式集合 $\{A, A \rightarrow B, \neg B, C \rightarrow D\}$, 我们知道它有且只有一个极小不协调集合 $\{A, A \rightarrow B, \neg B\}$.

假设 Γ 的所有极小不协调子集是 $\Gamma_1, \dots, \Gamma_m$. 下面将证明, 要求得 Γ 的极大协调子集, 只需从 Γ 中依次删除每个 Γ_i 中的一个公式, 其中 $1 \leq i \leq m$.

定义 5 (极小选择) 令 $\Gamma_1, \dots, \Gamma_m$ 为公式集合. 称 Θ 是一个关于 $\{\Gamma_1, \dots, \Gamma_m\}$ 的选择, 如果 $\Theta \subseteq \Gamma_1 \cup \dots \cup \Gamma_m$, 并且, 对任意 $1 \leq i \leq m$, $\Theta \cap \Gamma_i \neq \emptyset$. 称 Θ 是一个关于 $\{\Gamma_1, \dots, \Gamma_m\}$ 的极小选择,

如果不存在另一个关于 $\{\Gamma_1, \dots, \Gamma_m\}$ 的选择 Θ' , 使得 $\Theta' \subset \Theta$.

令 $\Gamma_1 = \{A, \neg A\}$, $\Gamma_2 = \{A, A \rightarrow B, \neg B\}$, $\Gamma_3 = \{B, \neg B\}$, 那么 $\{A, \neg A, B\}$ 是一个关于 $\{\Gamma_1, \Gamma_2, \Gamma_3\}$ 的选择, 而 $\{A, B\}$ 是一个关于 $\{\Gamma_1, \Gamma_2, \Gamma_3\}$ 的极小选择.

定义 6 (极小减集) 令 Γ 是一个不协调公式集合并且 $\{\Gamma_1, \dots, \Gamma_m\}$ 是它的所有极小不协调集合所组成的集合. 称 Θ 是 Γ 的一个减集, 如果 Θ 是关于 $\{\Gamma_1, \dots, \Gamma_m\}$ 的一个选择. 我们称 Θ 是 Γ 的一个极小减集, 如果 Θ 是关于 $\{\Gamma_1, \dots, \Gamma_m\}$ 的一个极小选择.

如果令 $\Gamma = \Gamma_1 \cup \Gamma_2 \cup \Gamma_3 = \{A, \neg A, A \rightarrow B, B, \neg B\}$, 那么 $\Theta_1 = \{A, \neg A, B\}$ 就是 Γ 的一个减集, 而 $\Theta_2 = \{A, B\}$ 是 Γ 的一个极小减集. 从上面的例子中, 可以看出 $\Gamma - \Theta_1 = \{A \rightarrow B, \neg B\}$ 是一个协调集合, 而 $\Gamma - \Theta_2 = \{\neg A, A \rightarrow B, \neg B\}$ 是一个极大协调集合. 更一般的, 可以证明下述引理.

引理 2 令 Γ 为一个不协调集合并且它的所有极小不协调集合是 $\Gamma_1, \dots, \Gamma_m$. $\Gamma - \Theta$ 协调当且仅当 Θ 是 Γ 的一个减集.

根据上述引理, 要保证 $\Gamma - \Theta$ 是 Γ 的极大协调子集, 只需保证 Θ 是一个极小的减集.

引理 3 $\Gamma - \Theta$ 是 Γ 的一个极大协调子集当且仅当 Θ 是 Γ 的一个极小减集.

使用引理 3, 可以证明极小减集和 **R** 缩减具有下述关系.

引理 4 令 $\Delta \mid \Gamma$ 为一个不协调 **R** 表达式. (1) 如果 Θ 是 $\Delta \cup \Gamma$ 的一个极小减集并且 $\Theta \cap \Delta = \emptyset$, 那么 $\Gamma - \Theta$ 就是 Γ 关于 Δ 的一个 **R** 缩减. (2) 如果 Γ' 是 Γ 关于 Δ 的一个 **R** 缩减, 那么存在 $\Delta \cup \Gamma$ 的一个极小减集 Θ 使得 $\Gamma' = \Gamma - \Theta$ 并且 $\Theta \cap \Delta = \emptyset$.

引理 4 中 (1) 的逆命题是不成立的, 其原因是 **R** 演算只能保证所有的极大缩减都能够使用 **R** 演算规则在有穷步内推导出, 但是不能保证使用 **R** 演算推导出来的所有结果都是 **R** 缩减^[2,3], 文献 [3] 中就给出了一个反例. 引理 4 建立起了 **R** 缩减和与 Δ 交集为空的极小减集之间的一一对应关系. 因此, 只需计算出 $\Delta \cup \Gamma$ 的所有与 Δ 交集为空的极小减集, 就可以求出 Γ 关于 Δ 的所有 **R** 缩减. 下面我们将给出一个计算所有极小选择的算法, 并证明它的正确性. 这一算法以一个公式集合组成的集合 $\mathcal{M} = \{\Gamma_1, \dots, \Gamma_m\}$ 为输入, 输出这些公式集合的所有极小选择 Ψ_m .

下面给出一个从 Ψ_{i-1} 计算 Ψ_i 的方法, 其中 $1 \leq i \leq m$.

(1) 首先, 令 $\Psi_0 = \{\emptyset\}$.

(2) 假设 $\Gamma_i = \{A_1, \dots, A_p\}$. 对任意一个 $\Theta \in \Psi_{i-1}$,

(a) 如果 $\Theta \cap \Gamma_i \neq \emptyset$, 那么 Θ 就是 $\Gamma_1, \dots, \Gamma_i$ 的一个极小选择. 接下来将删除 Ψ_i 中包含 Θ 的所有选择. 对 Ψ_i 中的任意集合 Θ'' , 检查 Θ 是否是 Θ'' 的子集. 如果是, 那么将 Θ'' 从 Ψ_i 中删除. 在完成对 Ψ_i 中所有公式集合的检查之后, 将 Θ 加入到 Ψ_i 中.

(b) 如果 $\Theta \cap \Gamma_i = \emptyset$, 那么对任意 $A_j \in \Gamma_i$, 令 $\Theta' = \Theta \cup \{A_j\}$. Θ' 是关于 $\{\Gamma_1, \dots, \Gamma_i\}$ 的一个选择, 我们需要检查 Ψ_i 中是否存在一个集合 Θ'' 使得 Θ'' 是 Θ' 的子集. 如果这样的 Θ'' 不存在, 那么将 $\Theta \cup \{A_j\}$ 加入 Ψ_i .

根据上述方法, 我们可以使用下述算法来计算所有极小选择.

容易看出上述 **R**-subtraction 过程是一个停机过程. 假设 $\{\Gamma_1, \dots, \Gamma_m\}$ 是集合 Λ 的所有极小不协调子集, $|\Gamma_i| = n_i$, $k = \max\{n_1, \dots, n_m\}$, 那么 **R**-subtraction 过程中执行集合操作的次数最多为 $\sum_{i=1}^m \frac{|\Psi_i|(|\Psi_i|-1)}{2}$, 其中 $|\Psi_0| = 1$, $|\Psi_i| \leq \prod_{j=1}^i n_j$. 所以上述过程的时间复杂性为 $O(k^{2m})$. 这表明在极小不协调子集的数量 m 给定时, 其时间复杂性随着 k 呈多项式增长; 而当 k 给定时, 其时间复杂性随 m 呈指数增长. 实际上, 由极小不协调子集计算所有极小减集的问题是计算所有极小命中集合问题 (minimal hitting set problem) 的一种特殊情况. 而极小命中集合对应的判定问题是 Karp 证明的

21 个 NP 完全问题之一. 因此, 由极小不协调子集计算所有极小减集的问题是一个 NP 难问题. 接下来我们将证明 R-subtraction 过程具有下述性质.

Procedure R-subtraction($\mathcal{M}$)

```

 $\Psi_0 := \{\emptyset\};$ 
for  $i = 1, m$  do
   $\Psi_i := \emptyset;$ 
  forall  $\Theta \in \Psi_{i-1}$  do
    if  $\Theta \cap \Gamma_i = \emptyset$  then
      forall  $A \in \Gamma_i$  do
         $\Theta' := \Theta \cup \{A\};$ 
         $b := \text{true};$ 
        forall  $\Theta'' \in \Psi_i$  do
          if  $\Theta' \supset \Theta''$  then
             $b := \text{false};$ 
            break};
        if  $b$  then
           $\Psi_i := \Psi_i \cup \{\Theta'\};$ 
    else
      forall  $\Theta'' \in \Psi_i$  do
        if  $\Theta \subset \Theta''$  then
           $\Psi_i := \Psi_i - \{\Theta''\};$ 
       $\Psi_i := \Psi_i \cup \{\Theta\};$ 
   $i := i + 1;$ 
return  $\Psi_m;$ 

```

引理 5 令 $\mathcal{M} = \{\Gamma_1, \dots, \Gamma_m\}$, 其中 $\Gamma_1, \dots, \Gamma_m$ 是公式集合. 如果过程 R-subtraction 的输入是 $\mathcal{M}$, 那么它的输出 Ψ_m 是 $\mathcal{M}$ 的所有极小选择组成的集合.

根据这一引理可知, 如果 R-subtraction 的输入是 Γ 的所有极小不协调集合, 那么它的输出就是 Γ 的所有极小减集. 下面的例子展示了使用上述算法来计算 Γ 的所有极小减集的过程.

例 1 令 Γ 为 $\{\neg A, A, A \rightarrow B, \neg B, B\}$. $\mathcal{M} = \{\Gamma_1, \Gamma_2, \Gamma_3\}$ 是 Γ 的极小不协调集合组成的集合, 其中 $\Gamma_1 = \{\neg A, A\}$, $\Gamma_2 = \{A, A \rightarrow B, \neg B\}$, $\Gamma_3 = \{\neg B, B\}$.

因为 $\emptyset$ 是 Ψ_0 中唯一的元素并且 $\emptyset \cap \Gamma_1 = \emptyset$, 所以 $\emptyset \cup \{\neg A\}$ 和 $\emptyset \cup \{A\}$ 都将会被加入到 Ψ_1 中. 因此, $\Psi_1 = \{\{\neg A\}, \{A\}\}$.

接下来, 我们将通过 Ψ_1 来计算 Ψ_2 . 因为 $\{\neg A\} \cap \Gamma_2 = \emptyset$, 所以 $\{\neg A, A\}$, $\{\neg A, A \rightarrow B\}$ 和 $\{\neg A, \neg B\}$ 都会被加入到 Ψ_2 中. 由于 $\{A\} \cap \Gamma_2 \neq \emptyset$, 需要删除 Ψ_2 中任何包含 $\{A\}$ 的集合. 已知 $\{\neg A, A\} \supset \{A\}$, 所以 $\{\neg A, A\}$ 将会被删除而 $\{A\}$ 将会被添加到 Ψ_2 中. 因此, $\Psi_2 = \{\{\neg A, A \rightarrow B\}, \{\neg A, \neg B\}, \{A\}\}$.

最后, 我们将计算 Ψ_3 . 因为 $\{\neg A, A \rightarrow B\} \cap \Gamma_3 = \emptyset$, 所以 $\{\neg A, A \rightarrow B, \neg B\}$ 和 $\{\neg A, A \rightarrow B, B\}$ 将会被加入到 Ψ_3 中. 由于 $\{\neg A, \neg B\} \cap \Gamma_3 \neq \emptyset$, 需要删除 Ψ_3 中任何包含 $\{\neg A, \neg B\}$ 的集合. 这也就是说, $\{\neg A, A \rightarrow B, \neg B\}$ 将会被删除, 而 $\{\neg A, \neg B\}$ 将被添加到 Ψ_3 中. 由于 $\{A\} \cap \Gamma_3 = \emptyset$ 并且 Ψ_3 中不存在 $\{A, \neg B\}$ 或 $\{A, B\}$ 的子集, $\{A, \neg B\}$ 和 $\{A, B\}$ 将会被添加到 Ψ_3 之中. 因此, $\Psi_3 = \{\{\neg A, A \rightarrow B, B\}, \{\neg A, \neg B\}, \{A, \neg B\}, \{A, B\}\}$. 这就是 Γ 的所有极小减集组成的集合.

下述引理给出了一种计算 $\Delta \cup \Gamma$ 的所有满足 $\Theta \cap \Delta = \emptyset$ 的极小减集的方法.

引理 6 令 $\Delta \mid \Gamma$ 为一个不协调的 **R** 表达式, 而 $\Gamma_1, \dots, \Gamma_m$ 是 $\Delta \cup \Gamma$ 的所有极小不协调子集. 如果过程 R-subtraction 的输入是 $\{\Gamma_1 - \Delta, \dots, \Gamma_m - \Delta\}$, 那么它的输出是 $\Delta \cup \Gamma$ 的所有满足 $\Theta \cap \Delta = \emptyset$

的极小减集.

根据这一引理和引理 4, 只要我们知道了如何计算 $\Delta \cup \Gamma$ 的所有极小不协调子集, 就能够计算出任意不协调 R 表达式 $\Delta \mid \Gamma$ 中 Γ 关于 Δ 的所有 R 缩减. 下面, 我们将讨论由 Horn 子句组成的不协调集合的极小不协调子集的求解问题.

3 Horn 子句

本节将讨论极小不协调的 Horn 子句集合的性质, 为提出求解所有极小不协调子集的算法做准备. 首先简单回顾一下 Horn 子句相关的概念.

定义 7 (Horn 子句) 一个文字要么是一个正文字, 即原子公式, 要么是一个负文字, 即原子公式的否定. 一个 Horn 子句是一些文字的析取, 其中至多包含一个正文字. Horn 子句被称为确定子句, 如果它包含一个正文字. 确定子句具有下述两种形式: $\{A\}$ 或者 $\{A, \neg B_1, \dots, \neg B_m\}$. 这里的 “,” 表示的是逻辑连接词符号 $\vee$. 我们也称 $\{A\}$ 为原子公式. 一个具有 $\{\neg B_1, \dots, \neg B_m\}$ 形式的 Horn 子句被称为否定子句或目标子句. 特别地, $\square$ 被称为空子句, 表示恒假.

实际上, Horn 子句 $\{A, \neg B_1, \dots, \neg B_m\}$ 对应于公式 $\forall x_1 \dots \forall x_n (A \vee \neg B_1 \vee \dots \vee \neg B_m)$, 其中 $\{x_1, \dots, x_n\}$ 是 $A \vee \neg B_1 \vee \dots \vee \neg B_m$ 中自由变元组成的集合. 举例来说, 一阶语句 $\forall x (A(x) \rightarrow B(x))$ 对应的 Horn 子句是 $\{B(x), \neg A(x)\}$.

在一阶语言中, 替换是一个将项或者公式中的自由变元用一个项来替代的操作, 一般将其记为 $\sigma = [t_1/x_1, \dots, t_n/x_n]$. 特别地, 恒等替换被记为 Id . 对任意公式 A , $\text{Id}(A) = A$.

定义 8 (最广合一子) 对于给定的原子公式集合 $S = \{A_1, \dots, A_n\}$, 我们称一个替换 σ 是 S 的一个合一子, 如果 $\sigma(A_1) = \dots = \sigma(A_n)$. 此时也称 S 是可以合一的. 称替换 σ 是 S 的一个最广合一子, 如果 σ 是 S 的一个合一子并且对于 S 的任意一个合一子 σ' , 都存在一个替换 θ 使得 $\sigma' = \sigma \circ \theta$. 这里 $\circ$ 是左结合的. 公式 $A = \sigma(A_1) = \dots = \sigma(A_n)$ 被称为 $A_1, \dots, A_n$ 的一个最普遍实例.

容易验证 $\sigma_1 = [f(a)/x, a/y, f(f(a))/z]$ 是 $\{P(x, f(x)), P(f(y), z)\}$ 的一个合一子, 而 $\sigma_2 = [f(y)/x, f(f(y))/z]$ 是 $\{P(x, f(x)), P(f(y), z)\}$ 的一个最广合一子, $P(f(y), f(f(y)))$ 是它的一个最普遍实例. 如果令 $\theta = [a/y]$, 那么有 $\sigma_1 = \sigma_2 \circ \theta$ 成立. 这表明任何一个合一子都可以由最广合一子复合上一个替换得到.

定义 9 (分离对) 对于给定的两个 Horn 子句 A 和 A' , 它们的一个分离对是一对满足下述条件的替换 (ρ, ρ') : (1) ρ 是 A 的一个替换并且对于 A 的每一个自由变元 x , $\rho(x)$ 也是一个变元, ρ' 是 A' 的一个替换并且对于 A' 的每一个自由变元 y , $\rho'(y)$ 也是一个变元; (2) ρ 和 ρ' 的值域互不相交.

分离对实际上是一对使得 Horn 子句的自由变元互不相同的替换. Horn 子句 $\{B(x), \neg A(x)\}$ 和 $\{A(x)\}$ 含有相同的自由变元 x , 而分离对 $(\text{Id}, [y/x])$ 将它们分别变为 $\{B(x), \neg A(x)\}$ 和 $\{A(y)\}$, 使得它们的自由变元互不相同. 有了上面这些准备, 下面我们来定义 Horn 子句上的 SLD- 归结方法.

定义 10 (SLD- 反驳) 令 Γ 是一个 Horn 子句组成的集合, Γ_d 是其中的确定子句组成的集合而 $\Gamma_g = \{G_1, \dots, G_q\}$ 是其中的目标子句组成的集合. Γ 的一个 SLD- 派生是一个由满足下述性质的否定子句所组成的序列 $\langle N_0, N_1, \dots, N_p \rangle$:

- (1) $N_0 = G_j$, 其中 G_j 是 Γ_g 中的某个目标子句;
- (2) 对于序列中的任何一个 N_i , $0 \leq i < p$, 如果 $N_i = \{\neg A_1, \dots, \neg A_{k-1}, \neg A_k, \neg A_{k+1}, \dots, \neg A_n\}$, 那么 Γ_d 存在着某个确定子句 $C_{i+1} = \{A, \neg B_1, \dots, \neg B_m\}$ 使得 A_k 和 A 可以合一, 并且对于它们的一个

分离对 (Id, ρ_i) 和 A_k 与 $\rho_i(A)$ 的一个最广合一子 σ_i , 如果 $m > 0$, 那么 $N_{i+1} = \sigma_i(\{\neg A_1, \dots, \neg A_{k-1}, \neg \rho_i(B_1), \dots, \neg \rho_i(B_m), \neg A_{k+1}, \dots, \neg A_n\})$. 如果 $m = 0$, 那么 $N_{i+1} = \sigma_i(\{\neg A_1, \dots, \neg A_{k-1}, \neg A_{k+1}, \dots, \neg A_n\})$. N_{i+1} 被称为 N_i 和 C_i 的归结.

序列 $N_0, C_1, \dots, C_p$ 被称为 SLD- 派生的输入序列. 如果 $N_p = \square$, 那么 SLD- 派生也被称为 SLD- 反驳.

下面我们将给出 Horn 子句中正负谓词符号的定义, 然后再考察由 Horn 子句组成的极小不协调集合的一些特殊性质.

定义 11 (正谓词符号和负谓词符号) 令 C 为一个 Horn 子句, P 是一个谓词符号而 $t_1, \dots, t_k$ 是项. 如果 $Pt_1 \dots t_k$ 是 C 中的一个正文字, 那么称 P 是 C 的一个正谓词符号; 如果 $\neg Pt_1 \dots t_k$ 是 C 中的一个负文字, 那么称 P 是 C 的一个负谓词符号. 用 $S(C)$ 来表示 C 中的正谓词符号组成的集合, 用 $S_-(C)$ 来表示 C 中的负谓词符号组成的集合. 如果 Λ 是一个 Horn 子句集合, 那么 $S(\Lambda)$ 表示 Λ 中所有子句的正谓词符号组成的集合; $S_-(\Lambda)$ 表示 Λ 中所有子句的负谓词符号组成的集合.

由 Horn 子句组成的极小不协调集合具有下述性质.

引理 7 令 Λ 是一个 Horn 子句集合. 如果 Λ 是一个极小不协调集合, 那么 (1) Λ 包含唯一的一个目标子句; (2) Λ 包含至少一个原子公式; (3) $S(\Lambda) = S_-(\Lambda)$.

引理 8 令 Γ 为一个不协调 Horn 子句集合, Horn 子句 $C \in \Gamma$ 并且 P 在 C 中出现. 如果 $P \notin S(\Gamma) \cap S_-(\Gamma)$, 那么 C 不属于 Γ 的任何极小不协调子集.

引理 9 如果 Λ 是一个极小不协调集合, 那么 Λ 可以表示为 $\Phi_1 \cup \dots \cup \Phi_k$ 的形式, 其中 $\Phi_1, \dots, \Phi_k$ 都是包含 Λ 中目标子句并且正负谓词符号集合相同的 Λ 的极小子集.

引理 7 实际上给出了一个 Horn 子句集合是极小不协调集合的必要条件, 而引理 8 则给出了一个 Horn 子句不属于任何极小不协调集合的一个充分条件. 引理 9 则提供了一个求解候选集合的方法.

4 枚举 Horn 子句集合中所有极小不协调子集的算法

由于 Horn 子句集合的协调性是不可判定的, 所以不存在一个能够枚举出 Horn 子句集合的所有极小不协调子集的非交互式算法. 为此, 我们提出了一个交互式的求解算法, 这一算法将会尽可能自动地枚举出极小不协调子集, 并且只在必要时对集合的协调性进行验证, 算法会自动记录它无法判定极小不协调性的集合, 并提交给用户来帮助判断. 这一算法主要基于上一节的引理 7-9, 其基本思路是:

(1) 构造出满足引理 7 中 3 个条件的所有极小子集 $\{\Phi_1, \dots, \Phi_k\}$.

(2) 由上述极小子集生成集合 $L = \{\Phi_{i_1} \cup \dots \cup \Phi_{i_n} \mid 1 \leq i_1, \dots, i_n \leq k, 1 \leq n \leq k\}$. 集合 L 关于集合包含关系构成一个格 (lattice). 从 L 的极小元素 $\Phi_1, \dots, \Phi_k$ 出发, 依包含关系从小到大, 逐个使用搜索 SLD- 反驳的方法来确定 L 中的哪些元素是极小不协调集合, 而哪些元素需要用户来辅助判断.

(3) 将算法和用户判定出的极小不协调集合合并在一起, 我们就得到了所有的极小不协调子集.

根据定义 11, 引理 7 和 8, 满足引理 7 中 (1) 和 (3) 两个条件的极小子集可以按照下述步骤来求解: (a) 首先, 计算出 Γ 中每一个 Horn 子句的正负谓词符号集合. (b) 然后根据引理 8 中的条件排除不可能属于任何极小不协调子集的 Horn 子句. (c) 最后, 从剩下的 Horn 子句集合中找出包含一个目标子句且正负谓词符号集合相同的极小子集. 其中, 步骤 (a), (b) 可以通过下述 Prepare 过程来完成. 这一过程以 Γ 为输入, 输出一个二元组 (ST_g, ST_d) , 其中 ST_g 和 ST_d 分别是 Γ 中目标子句和确定

子句对应的正谓词符号集合、负谓词符号集合和下标构成的三元组所组成的集合. 其具体作法如下:

Procedure Prepare(Γ)

```

 $ST := \emptyset, S := \emptyset, T := \emptyset;$ 
forall  $C_i \in \Gamma$  do
    switch  $C_i$  do
        case  $\{\neg B_1, \dots, \neg B_m\}$  ( $P_j$  is the predicate symbol in  $B_j$ )
             $ST := ST \cup \{(\emptyset, \{P_1, \dots, P_m\}, i)\};$ 
             $T := T \cup \{P_1, \dots, P_m\};$ 
        case  $\{A, \neg B_1, \dots, \neg B_m\}$  ( $P_j$  is the predicate symbol in  $B_j$  and  $Q$  is the predicate symbol in  $A$ )
             $ST := ST \cup \{(\{Q\}, \{P_1, \dots, P_m\}, i)\};$ 
             $S := S \cup \{Q\};$ 
             $T := T \cup \{P_1, \dots, P_m\};$ 
        case  $\{A\}$  ( $Q$  is the predicate symbol in  $A$ )
             $ST := ST \cup \{(\{Q\}, \emptyset, i)\};$ 
             $S := S \cup \{Q\};$ 
     $ST_g := \emptyset, ST_d := \emptyset;$ 
    forall  $(S', T', i) \in ST$  do
        if  $S' \cup T' \subseteq S \cap T$  then
            if  $S' = \emptyset$  then
                 $ST_g := ST_g \cup \{(S', T', i)\};$ 
            else
                 $ST_d := ST_d \cup \{(S', T', i)\};$ 
    return  $(ST_g, ST_d);$ 
    
```

下面我们将讨论如何在 Prepare 过程的计算结果的基础上实现步骤 (c), 即计算出满足引理 7 中性质 (1) 和 (3) 的极小子集. 性质 (1) 表明这些极小子集中必须包含唯一一个目标子句, 所以我们以一个目标子句 G 为出发点, 求出所有包含 G 并且正负谓词符号集合相同的极小子集. 我们将这过程记为 Candidates, 其具体作法如下:

(1) Φ 是一个用于存放 Horn 子句的集合, 它的初始值为 ST_g 中的一个目标子句. S 是 Φ 中正谓词符号的集合, T 是 Φ 中负谓词符号的集合. ST_d 是过程 Prepare 返回的结果之一, 是由 Γ 中那些可能属于某个极小不协调集合的确定子句的正负谓词符号集合和下标三元组 (S', T', i) 所组成的集合. 集合 C 中存储的是候选集合, 即所有包含目标子句并且满足引理 7 的性质 (3) 的极小子集.

Procedure Candidates(S, T, Φ, Γ, ST_d)

```

 $C := \emptyset, \Sigma := \{(S, T, \Phi)\}, U := T - S;$ 
if  $U = \emptyset$  then
     $C := C \cup \{\Phi\};$ 
else
    forall  $P \in U$  do
         $\Sigma' := \emptyset;$ 
         $\Omega := \{T_i \mid (\{P\}, T_i, i) \in ST_d\};$ 
        forall  $(S', T', \Phi') \in \Sigma$  do
             $S'' := S' \cup \{P\};$ 
            forall  $T_i \in \Omega$  do
                 $\Sigma' := \Sigma' \cup \{(S'', T' \cup T_i, \Phi' \cup \{\Gamma[i]\})\};$ 
         $\Sigma := \Sigma';$ 
    forall  $(S', T', \Phi') \in \Sigma$  do
         $C' := \text{Candidates}(S', T', \Phi', \Gamma, ST_d);$ 
         $C := C \cup C';$ 
    return  $C;$ 
    
```

(2) 由于这一过程的初始输入是一个目标子句, 所以在递归计算过程中一直有 $T \supseteq S$ 成立. 令 $U := T - S$, 那么 U 就是那些没有对应正谓词符号的 T 中的负谓词符号组成的集合.

(a) 如果 $U = \emptyset$, 那么 Φ 中正负谓词符号相同, 它将被加入到 $\mathcal{C}$ 中.

(b) 否则, 对于 U 中的任意谓词符号 P , 我们将找到一个包含正谓词符号 P 的确定子句 C_P , 将它加入 Φ , 得到 Φ' . 将递归的计算出所有包含 Φ' 的候选集合并将它们加入 $\mathcal{C}$.

一旦将包含目标子句 G 并且正负谓词符号集合相同的所有极小子集求出, 那么根据引理 9, 我们就可以在它们的并集构成的格 L 上, 依据集合的包含关系从小到大来检查 L 中的哪些集合是满足引理 7 中 (2) 的, 其中哪些还是不协调的. 如果某个集合 Φ 满足引理 7 中 (2) 而且是不协调的, 并且 L 中比它小的集合都是协调的, 那么集合 Φ 就是一个极小不协调集合. 我们将这一过程记为 MIS^* , 其具体作法如下:

(1) 在这一过程中将用到一个已有的算法来检查 Horn 子句集合的协调性, 这就是查找给定 Horn 子句集合中的 SLD- 反驳的算法. 这一类算法在逻辑程序设计语言 PROLOG 中已经得到广泛的研究和应用, 本文不再对此做详细的讨论. 我们将采用一个满足下述性质的停机深度优先搜索算法 ConsistencyCheck:

- (i) 如果给定的 Horn 子句集合存在一个 SLD- 反驳, 那么 ConsistencyCheck 将返回 No;
- (ii) 如果在遍历了所有可能后没有找到一个 SLD- 反驳, 那么 ConsistencyCheck 将返回 Yes;
- (iii) 如果遇到了一个无穷的路径, 算法被强行终止, 那么 ConsistencyCheck 将返回 Unknown.

(2) 使用 Ω 来记录那些满足引理 7 中 (1) 和 (3) 两个条件的候选集合, 使用 E_G^m 来记录极小不协调集合, 使用 E_G^n 来记录那些需要用户来判定的集合. Ω 将被初始化为 $\mathcal{C}_G$. E_G^m 和 E_G^n 则被初始化为空集.

(3) 对于 Ω 中的任意集合 Λ , 首先检查引理 7 中的条件 (2) 是否满足.

Procedure MIS*($\mathcal{C}$)

```

 $E^m := \emptyset, E^n := \emptyset, \Omega := \mathcal{C};$ 
while  $\Omega \neq \emptyset$  do
  forall  $\Phi \in \Omega$  do
    if  $\Phi$  contains at least one atomic formula then
      if there exists an  $\Phi' \in E^m$  such that  $\Phi \supset \Phi'$  then
         $\Omega := \Omega - \{\Phi\};$ 
      else if there exists an  $\Phi' \in E^n$  such that  $\Phi \supset \Phi'$  then
         $E^n := E^n \cup \{\Phi\};$ 
      else
         $state := \text{ConsistencyCheck}(\Phi);$ 
        if  $state = \text{No}$  then
           $E^m := E^m \cup \{\Phi\};$ 
           $\Omega := \Omega - \{\Phi\};$ 
        else if  $state = \text{Unknown}$  then
           $E^n := E^n \cup \{\Phi\};$ 
      else
         $\Omega := \Omega - \{\Phi\};$ 
     $\Omega := \{\Phi_1 \cup \Phi_2 \mid \Phi_1 \in \Omega, \Phi_2 \in \mathcal{C} \text{ and } \Phi_1 \not\supseteq \Phi_2\};$ 
return  $(E^m, E^n);$ 

```

(a) 如果条件满足, 那么将先检查是否存在 E_G^m 中的一个集合是 Λ 的子集. 如果这样的集合存在,

那么 Λ 就不是极小不协调集合, 需要将它从 Ω 中删除; 否则, 将再检查它是否有一个不可判定的子集. 如果有, 那么将 Λ 加入 E_G^n 中; 否则, 将使用 ConsistencyCheck 来判定 Λ 是否协调. 如果算法判断 Λ 不协调, 那么 Λ 是一个极小不协调集合, 将它加入 E_G^m , 如果算法不能判断 Λ 是否协调, 那么将 Λ 从 Ω 中删除并将其加入到 E_G^n 中.

(b) 否则, 将 Λ 从 Ω 中删除.

(4) 令 $\Omega = \{\Phi_1 \cup \Phi_2 \mid \Phi_1 \in \Omega, \Phi_2 \in \mathcal{C} \text{ 并且 } \Phi_1 \not\supseteq \Phi_2\}$, 进入下一次循环, 直到 Ω 为空.

由于每一个极小不协调子集必然包含唯一的一个目标子句, 所以我们只需要对 Γ 中的每一个目标子句 G , 求出所有包含 G 的极小不协调子集组成的集合 M_G , 那么 $\bigcup_{G \in \Gamma_g} M_G$ 就是 Γ 的所有极小不协调子集. 因此, 可以使用上述 3 个过程, 给出下述求解 Horn 子句集合所有极小不协调子集的算法 MIS.

Procedure MIS(Γ)

```

 $E^m := \emptyset, E^n := \emptyset;$ 
 $(ST_g, ST_d) := \Gamma;$ 
forall  $(S', T', i) \in ST_g$  do
     $\mathcal{C}_i := \text{Candidates}(S', T', \{\Gamma[i]\}, \Gamma, ST_d);$ 
     $(E_i^m, E_i^n) := \text{MIS}^*(\mathcal{C}_i);$ 
     $E^m := E^m \cup E_i^m;$ 
     $E^n := E^n \cup E_i^n;$ 
if  $E^n \neq \emptyset$  then
    print  $E^n;$ 
     $F^m := \text{read in user's input};$ 
     $E^m := E^m \cup F^m;$ 
return  $E^m;$ 
    
```

引理 10 对于输入的任意 Horn 子句集合 Γ , 过程 MIS 输出 Γ 的所有极小不协调子集.

5 Horn 子句中极大缩减的求解算法

基于之前的讨论, 使用上一节提出的枚举极小不协调子集的算法 MIS 和通过极小不协调子集求解极大缩减的算法 R-subtraction, 我们可以设计出下述求解 $\mathbf{R}$ 表达式 $\Delta \mid \Gamma$ 中 Γ 关于 Δ 的所有 $\mathbf{R}$ 缩减的算法 R-contractions, 作法如下:

Procedure R-contractions(Δ, Γ)

```

 $\mathcal{M} := \emptyset, \mathcal{R} := \emptyset;$ 
 $E^m := \text{MIS}(\Delta \cup \Gamma);$ 
forall  $\Phi \in E^m$  do
     $\mathcal{M} := \mathcal{M} \cup \{\Phi - \Delta\};$ 
 $\Psi := \text{R-subtraction}(\mathcal{M});$ 
forall  $\Theta \in \Psi$  do
     $\mathcal{R} := \mathcal{R} \cup \{\Gamma - \Theta\};$ 
return  $\mathcal{R};$ 
    
```

根据引理 4,6,10, 我们知道上述算法的输出是 Γ 关于 Δ 的所有 $\mathbf{R}$ 缩减. 下面将通过一个例子来展示这个算法的执行过程.

例 2 令 Δ 为 $\{\{\neg C\}\}$, 而 Γ 为 $\{\{A\}, \{B, \neg A\}, \{C, \neg B\}, \{D, \neg A\}, \{C, \neg D\}, \{F, \neg E\}\}$. 如果要使用上面定义的过程 R-contractions 来计算 Γ 关于 Δ 的所有 $\mathbf{R}$ 缩减, 那么首先要执行的是过程 MIS. 而过程 MIS 中最先被执行的过程是 Prepare, 它的输入是 $\Delta \cup \Gamma$. 在执行完 Prepare 的第一个 forall 循环之后, 我们得到

$$ST = \{(\{A\}, \emptyset, 1), (\{B\}, \{A\}, 2), (\{C\}, \{B\}, 3), (\{D\}, \{A\}, 4), (\{C\}, \{D\}, 5), (\emptyset, \{C\}, 6), (\{F\}, \{E\}, 7)\},$$

$$S = \{A, B, C, D, F\}, \quad T = \{A, B, C, D, E\}.$$

Prepare 中的第二个 forall 循环将会剔除掉那些不可能属于任何极小不协调集合的 Horn 子句在 ST 中对应的三元组, 并根据剩下的那些三元组对应的子句是目标子句还是确定子句分为 ST_g 和 ST_d 两个集合. 因此, Prepare 的输出就是 (ST_g, ST_d) , 其中

$$ST_g = \{(\emptyset, \{C\}, 6)\}, ST_d = \{(\{A\}, \emptyset, 1), (\{B\}, \{A\}, 2), (\{C\}, \{B\}, 3), (\{D\}, \{A\}, 4), (\{C\}, \{D\}, 5)\}.$$

由于 ST_g 中只包含一个元素 $(\emptyset, \{C\}, 6)$, MIS 中的 forall 语句只循环一次. 也就是说, 过程 Candidates 将会以 $(\emptyset, \{C\}, \{\{\neg C\}\}, \Delta \cup \Gamma, ST_d)$ 为输入被调用, 然后 MIS* 将以 Candidates 的输出为输入被调用. Candidates $(\emptyset, \{C\}, \{\{\neg C\}\}, \Delta \cup \Gamma, ST_d)$ 的执行将会输出 $\mathcal{C} = \{\Gamma_1, \Gamma_2\}$, 其中 $\Gamma_1 = \{\{\neg C\}, \{C, \neg B\}, \{B, \neg A\}, \{A\}\}$, $\Gamma_2 = \{\{\neg C\}, \{C, \neg D\}, \{D, \neg A\}, \{A\}\}$.

由于上述 Horn 子句中不包含变元, 我们知道协调性检查算法 ConsistencyCheck 可以判定 Γ_1 和 Γ_2 都不协调. 因此, MIS* 的输出就是 (E^m, E^n) , 其中 $E^m = \{\Gamma_1, \Gamma_2\}$, $E^n = \emptyset$. 因为 $E^n = \emptyset$, MIS 将直接输出 $\Delta \cup \Gamma$ 的所有极小不协调集合组成的集合 E^m .

R-contractions 中的第一个 forall 循环将包含在 Δ 中的子句从 E^m 中的极小不协调集合中删除, 得到 $\mathcal{M} = \{\{\{C, \neg B\}, \{B, \neg A\}, \{A\}\}, \{\{C, \neg D\}, \{D, \neg A\}, \{A\}\}\}$. 然后 R-subtraction 以 $\mathcal{M}$ 为输入, 通过一个类似于例 1 中的计算过程, 得到

$$\Psi = \{\{\{A\}\}, \{\{C, \neg B\}, \{D, \neg A\}\}, \{\{C, \neg B\}, \{C, \neg D\}\}, \{\{B, \neg A\}, \{D, \neg A\}\}, \{\{B, \neg A\}, \{C, \neg D\}\}\}.$$

因此, 经过 R-contractions 中的第二个 forall 循环, 我们得到了 Γ 关于 Δ 的所有 $\mathbf{R}$ 缩减.

$$\begin{aligned} \mathcal{R} = & \{\{\{B, \neg A\}, \{C, \neg B\}, \{D, \neg A\}, \{C, \neg D\}, \{F, \neg E\}\}, \{\{A\}, \{B, \neg A\}, \{C, \neg D\}, \{F, \neg E\}\}, \\ & \{\{A\}, \{B, \neg A\}, \{D, \neg A\}, \{F, \neg E\}\}, \{\{A\}, \{C, \neg B\}, \{C, \neg D\}, \{F, \neg E\}\}, \\ & \{\{A\}, \{C, \neg B\}, \{D, \neg A\}, \{F, \neg E\}\}\}. \end{aligned}$$

6 结论

本文提出一个在 Horn 子句中计算所有极大缩减的交互式算法. 这一算法的思路是, 对于给定的不协调 $\mathbf{R}$ 表达式 $\Delta \mid \Gamma$, 首先求出 $\Delta \cup \Gamma$ 的所有极小不协调子集, 然后通过这些极小不协调子集求出 Γ 关于 Δ 的所有极大缩减. 相比于其他计算极大缩减的方法, 这一方法的优点是可以一次求出所有的极大缩减. 这一算法使用了寻找 SLD- 反驳的方法来做 Horn 子句集合的协调性检查, 而寻找 SLD- 反驳是 PROLOG 语言的一个基础算法 [4], 有很多经过优化的算法可供我们使用. 采用这一方法也使得我们的算法更易于集成到 PROLOG 中, 为 PROLOG 语言编写的知识库提供协调性维护的功能. 同

时, 也必须指出这一算法有它的局限性. 由于 Horn 子句集合的协调性也是不可判定的, 在 Horn 子句集合中含有变元的情况下, 可能需要用户的协助才能枚举出它的所有极小不协调子集.

在以后的工作中, 我们将考虑针对一阶语言提出求解所有极大缩减的算法, 探讨在一阶语言的哪些子集上算法可以自动的完成对所有极大缩减的计算, 以及在哪些子集上计算所有极大缩减问题的难度.

对于 Horn 子句信念修正中的缩减函数已经有了研究. 需要指出的是, 本文所研究的极大缩减求解算法与其他基于 AGM 理论的缩减函数不尽相同. 其主要区别在于本文所针对的极大缩减与信念修正的 AGM 理论中所描述的缩减不同. AGM 理论中缩减的概念最早是由 Gärdenfors 等于 1985 年提出的 [5], Gärdenfors 等将这一类缩减称为部分匹配缩减. 随后 AGM 理论的后继研究者又提出了一些新型的缩减, 例如, 完全匹配缩减、基于认知牢固度的缩减、E- 缩减、I- 缩减等 [6–8]. 但是这些缩减有一个共同的特性, 那就是必须保证缩减结果的唯一性, 为此它们都依赖于逻辑推理以外的信息. 然而有些情况下, 我们并不希望缩减依赖于逻辑推理以外的信息, 也不需要保证缩减结果的唯一性. 例如在软件的进化过程和科学发现过程中, 不希望使用逻辑推理以外的信息来保证缩减结果的唯一性, 而是希望能够得到所有可能的缩减结果. 极大缩减正是为描述这样的过程而提出的. 极大缩减和 AGM 理论中缩减的另一个不同点在于, 在 AGM 理论中, 缩减一般是指信念系统关于某个语句的缩减. 但是在极大缩减中, 缩减则是指公式集合 Γ 关于事实反驳 Δ 的缩减, 这里面 Δ 不是一个语句, 而是一个由原子语句或原子语句的否定组成的协调语句集合.

致谢 感谢李贺认真的阅读本文初稿并提出了宝贵的建议.

参考文献

- 1 Li W. A logical framework for evolution of specifications. In: Programming Languages and Systems—ESOP'94. LNCS 788. Berlin: Springer-Verlag, 1994. 394–408
- 2 Li W. R-calculus: an inference system for belief revision. Comput J, 2007, 50: 378–390
- 3 Li W. Mathematical Logic—Foundations For Information Science. Basel: Birkhaeuser Publish, 2009
- 4 Gallier J H. Logic for Computer Science: Foundations of Automatic Theorem Proving. New York: Harper & Row, 1986
- 5 Alchourrón C E, Gärdenfors R, Makinson D. On the logic of theory change: partial meet contraction and revision functions. J Symbolic Logic, 1985, 50: 510–530
- 6 Gärdenfors P. Knowledge in Flux. Modeling the Dynamics of Epistemic States. Cambridge: The MIT Press, 1988
- 7 Gärdenfors P, Makinson D. Revisions of Knowledge Systems Using Epistemic Entrenchment. In: 2nd Conference on Theoretical Aspects of Reasoning about Knowledge, 1988. 83–95
- 8 Delgrande J P. Horn Clause Belief Change: Contraction Functions. In: Proc KR, 2008. 156–165
- 9 Booth R, Meyer T, Varzinczak I J. Next Steps in Propositional Horn Contraction. In: Proc IJCAI, 2009. 702–707

附录 A 定理的证明

引理 1 证明 由于 Γ 是有穷的, 我们可以逐个的考察 Γ 中的公式 A , 如果 $\Gamma - \{A\}$ 也不协调, 那么将 A 从 Γ 中删除; 如果 $\Gamma - \{A\}$ 协调, 那么保持 Γ 不变. 这一过程将在有穷步内终止, 最后得到的集合 Γ' 将满足性质: Γ' 不协调, 但是对于 Γ' 中的任何一个公式 A , $\Gamma' - \{A\}$ 协调. 也就是说, Γ' 是一个极小不协调集合.

引理 2 证明 必要性: 如果 Θ 是 Γ 的一个减集, 那么 $\Gamma - \Theta$ 协调. 我们将使用反证法来证明. 假设 $\Gamma - \Theta$ 不协调, 那么 $\Gamma - \Theta$ 存在一个极小不协调子集 Γ' . 由于 $\Gamma \supseteq \Gamma - \Theta$, Γ' 也是 Γ 的极小不协调子集. 根据前提可知, 存在

一个 $1 \leq k \leq m$ 使得 $\Gamma' = \Gamma_k$. 由于 $\Theta \cap \Gamma_k \neq \emptyset$, 那么必然存在一个 $A \in \Theta \cap \Gamma_k$. 这也就是说, $A \in \Theta$ 并且 $A \in \Gamma_k$. 所以 $A \notin \Gamma - \Theta$. 又因为 $\Gamma' = \Gamma_k$ 是 $\Gamma - \Theta$ 的一个子集, $A \notin \Gamma_k$. 这就与 $A \in \Gamma_k$ 相矛盾. 因此, $\Gamma - \Theta$ 协调.

充分性: 如果 $\Gamma - \Theta$ 协调, 那么 Θ 是 Γ 的一个减集. 我们将使用反证法来证明. 假设 Θ 不是 Γ 的一个减集, 即存在一个 $1 \leq k \leq m$ 使得 $\Theta \cap \Gamma_k \neq \emptyset$. 由于 Γ_k 是 Γ 的子集, 已知 $\Gamma - \Theta \supseteq \Gamma_k$. 因为 Γ_k 不协调, $\Gamma - \Theta$ 也不协调. 这就导致了矛盾. 因此, Θ 是 Γ 的一个减集.

引理 3 证明 必要性: 如果 Θ 是 Γ 的一个极小减集, 那么 $\Gamma - \Theta$ 是 Γ 的一个极大协调子集. 根据引理 2, 已知 $\Gamma - \Theta$ 是协调的. 只需证明它也是极大的. 我们将使用反证法来证明. 假设 $\Gamma - \Theta$ 不是 Γ 的极大协调子集, 那么存在 Γ 的一个协调子集 Γ' 使得 $\Gamma' \supset \Gamma - \Theta$. 根据引理 2, $\Theta' = \Gamma - \Gamma'$ 是 Γ 的一个减集. 由于 $\Theta' \subset \Theta$, 这与 Θ 是 Γ 的一个极小减集相矛盾. 因此, $\Gamma - \Theta$ 是 Γ 的一个极大协调子集.

充分性: 如果 $\Gamma - \Theta$ 是 Γ 的一个极大协调子集, 那么 Θ 是 Γ 的一个极小减集. 根据引理 2, 我们知道 Θ 是 Γ 的一个减集. 只需证明它还是极小的. 我们将使用反证法来证明. 假设 Θ 不是 Γ 的极小减集, 那么存在一个 $\Theta' \subset \Theta$ 使得 Θ' 是 Γ 的一个减集. 根据引理 2, $\Gamma - \Theta'$ 是协调的. 因为 $\Gamma - \Theta' \supset \Gamma - \Theta$, 所以 $\Gamma - \Theta$ 不是 Γ 的极大协调子集. 这与前提相矛盾. 因此, Θ 是 Γ 的一个极小减集.

引理 4 证明 (1) 根据引理 3, $\Delta \cup \Gamma - \Theta$ 是 $\Delta \cup \Gamma$ 的一个极大协调子集. 因为 $\Theta \cap \Delta = \emptyset$, 所以 $\Delta \cup \Gamma - \Theta = \Delta \cup (\Gamma - \Theta)$. 所以 $\Gamma - \Theta$ 与 Δ 协调. 假设 $\Gamma - \Theta$ 不是 Γ 关于 Δ 的 R 缩减, 那么存在 Γ 的一个协调子集 Γ' 使得 $\Gamma' \supset \Gamma - \Theta$. 由于 $\Delta \cup \Gamma' \supset \Delta \cup (\Gamma - \Theta)$, 这就与 $\Delta \cup \Gamma - \Theta$ 是 $\Delta \cup \Gamma$ 的一个极大协调子集相矛盾. 因此, $\Gamma - \Theta$ 是 Γ 关于 Δ 的一个 R 缩减.

(2) 令 $\Theta = \Gamma - \Gamma'$. 显然 $\Gamma' = \Gamma - \Theta$. 由于 Γ' 是 Γ 关于 Δ 的一个 R 缩减, $\Delta \cup \Gamma'$ 是 $\Delta \cup \Gamma$ 的一个极大协调子集并且 $\Theta \cap \Delta = \emptyset$. 根据引理 3, $\Delta \cup \Gamma' = \Delta \cup \Gamma - \Theta$ 是 $\Delta \cup \Gamma$ 一个极大协调子集当且仅当 Θ 是 $\Delta \cup \Gamma$ 的一个极小减集. 所以 Θ 就是使得引理成立的一个极小减集.

引理 5 证明 我们将使用数学归纳法来证明这个引理.

(1) 首先证明 Ψ_1 是 $\{\Gamma_1\}$ 的所有极小选择组成的集合.

由于 $\Psi_0 = \{\emptyset\}$, 上述过程的第一个 forall 语句将只循环一次并且 $\Theta = \emptyset$. 所以 $\Theta \cap \Gamma_1 = \emptyset$ 并且接下来的 if 语句的第一部分将被执行. 假设 Γ_1 是 $\{A_1, \dots, A_p\}$. 因为 Ψ_1 的初始值是 $\emptyset$, 过程中的第 3 个 forall 语句将不会被执行并且 b 为 true. 因此, 第 2 个 forall 循环语句会将 $\{A_1\}, \dots, \{A_p\}$ 加入到 Ψ_1 中. 在 for 语句的第一次循环结束后, Ψ_1 的值是 $\{\{A_1\}, \dots, \{A_p\}\}$. 容易验证, Ψ_1 就是 $\{\Gamma_1\}$ 的所有极小选择组成的集合.

(2) 假设 Ψ_{i-1} 是 $\{\Gamma_1, \dots, \Gamma_{i-1}\}$ 的所有极小选择组成的集合. 我们将证明 Ψ_i 是 $\{\Gamma_1, \dots, \Gamma_i\}$ 的所有极小选择组成的集合.

令 $\Theta \in \Psi_{i-1}$. 如果 $\Theta \cap \Gamma_i = \emptyset$, 那么对于 Γ_i 中的任意公式 A , $\Theta \cup \{A\}$ 是 $\{\Gamma_1, \dots, \Gamma_i\}$ 的一个选择并且只有在 Ψ_i 中不存在 $\Theta \cup \{A\}$ 的子集的情况下, 它才会被加入 Ψ_i . 否则, Θ 就是 $\{\Gamma_1, \dots, \Gamma_i\}$ 的一个极小选择, 并且在删除了 Ψ_i 中包含 Θ 的所有集合后, Θ 会被加入到 Ψ_i 中. 因此, Ψ_i 将会一直保持下述性质: 对于 Ψ_i 中的任意两个集合 Θ_1 和 Θ_2 , $\Theta_1 \not\supseteq \Theta_2$. 所以只有下述集合中的极小集合会在第 i 次循环完成后任然留在 Ψ_i 之中.

$$\{\Theta \mid \Theta \in \Psi_{i-1} \text{ and } \Theta \cap \Gamma_i \neq \emptyset\} \cup \{\Theta \cup \{A\} \mid A \in \Gamma_i, \Theta \in \Psi_{i-1} \text{ and } \Theta \cap \Gamma_i = \emptyset\}.$$

由于 $\{\Gamma_1, \dots, \Gamma_i\}$ 的所有极小选择都包含在上述集合中, Ψ_i 就是 $\{\Gamma_1, \dots, \Gamma_i\}$ 的所有极小选择组成的集合.

根据归纳法, Ψ_m 是 $\mathcal{M}$ 的所有极小选择组成的集合.

引理 6 证明 根据引理 5, 如果 R -subtraction 的输入是 $\{\Gamma_1 - \Delta, \dots, \Gamma_m - \Delta\}$, 那么它的输出 Ψ_m 是 $\{\Gamma_1 - \Delta, \dots, \Gamma_m - \Delta\}$ 的所有极小选择组成的集合. 假设 $\Theta \in \Psi_m$. 根据定义 5, $\Theta \subseteq (\Gamma_1 - \Delta) \cup \dots \cup (\Gamma_m - \Delta)$ 并且 $\Theta \cap (\Gamma_i - \Delta) \neq \emptyset$. 所以 $\Theta \cap \Gamma_i \neq \emptyset$ 并且 $\Theta \cap \Delta = \emptyset$. 因为 Θ 还是极小的, 所以 Θ 是 $\Delta \cup \Gamma$ 的满足 $\Theta \cap \Delta = \emptyset$ 的一个极小减集. 所以 Θ 是 $\{\Gamma_1, \dots, \Gamma_m\}$ 的一个减集. 由于 Ψ_m 包含所有与 Δ 没有交集的极小选择, Ψ_m 就是 $\Delta \cup \Gamma$ 的所有满足 $\Theta \cap \Delta = \emptyset$ 的极小减集.

引理 7 证明 根据一阶推理系统的可靠性和完全性^[3], 如果 Λ 不协调, 那么它不可满足. 再根据 Horn 子句上 SLD- 归结方法的完全性^[4], 如果 Λ 不可满足, 那么存在一个 Λ 上的 SLD- 反驳. 如果 Λ 是极小不协调的, 那么 Λ 中的每一个子句都将在 SLD- 反驳中被使用. 否则, 将会导出 Λ 不是极小不协调的矛盾. 假设 SLD- 反驳的输入序列是 $N_0, C_1, \dots, C_p$. 已知 $\Lambda = \{N_0, C_1, \dots, C_p\}$.

(1) 根据 SLD- 反驳的定义和 $\Lambda = \{N_0, C_1, \dots, C_p\}$, 我们知道 N_0 是 Λ 中唯一一个目标子句.

(2) 根据 SLD- 反驳的定义, $N_p = \square$. 由于 N_p 是 N_{p-1} 和 C_p 的归结, 这表明 N_{p-1} 仅包含一个原子公式的否定并且 C_p 只包含一个原子公式. 因为 $C_p \in \Lambda$, 所以 Λ 包含至少一个原子公式.

(3) 令 P 为 $S(\Lambda)$ 中的任意一个谓词符号. 不妨设 $P \in S(C_i)$ 并且 A 是其中使用了 P 的文字. 因为 N_i 是 N_{i-1} 和 C_i 的归结, 所以 N_{i-1} 中存在一个文字 $\neg A_k$ 使得 A_k 和 A 可以合一. 所以 A_k 是一个包含谓词 P 的原子公式. 因此, $P \in S_{-}(N_{i-1}) \subseteq S_{-}(\{N_0, C_1, \dots, C_{i-1}\}) \subseteq S_{-}(\Lambda)$.

令 P 为 $S_{-}(\Lambda)$ 中的任意一个谓词符号. 由于 $N_p = \square$, P 必然在 $S_{-}(N_0), \dots, S_{-}(N_{p-1})$ 中某个负谓词集合里出现, 并且一定会被消去. 不妨设 $P \in S_{-}(N_i)$, $\neg A_k$ 是第 i 次归结时 N_i 中选定的文字并且 $P \in S_{-}(\neg A_k)$. 令 C_{i+1} 为 $A \vee \neg B_1 \dots \vee \neg B_m$. 由于 A_k 和 A 可以合一, A 是包含谓词 P 的一个正文字. 因此, $P \in S(C_{i+1}) \subseteq S(\Lambda)$.

引理 8 证明 因为 $P \notin S(\Gamma) \cap S_{-}(\Gamma)$, 所以 $P \in S(\Gamma), P \notin S_{-}(\Gamma)$ 或者 $P \notin S(\Gamma), P \in S_{-}(\Gamma)$. 使用反证法证明. 在 $P \in S(\Gamma), P \notin S_{-}(\Gamma)$ 的情况下, 由于 P 在 C 中出现, $P \in S(C)$. 假设 Γ 的一个极小不协调子集 Λ 包含 C . 根据引理 7, $S(\Lambda) = S_{-}(\Lambda)$. 由于 $C \in \Lambda$, 我们知道 $P \in S(\Lambda)$. 因为 $P \notin S_{-}(\Gamma)$, 所以 $P \notin S_{-}(\Lambda)$. 这也就是说 $S(\Lambda) \neq S_{-}(\Lambda)$, 这就导致了矛盾.

在 $P \notin S(\Gamma), P \in S_{-}(\Gamma)$ 的情况下, 证明是类似的.

引理 9 证明 只需要证明: 对任意 $C \in \Lambda$, 存在一个包含 Λ 中目标子句并且正负谓词符号集合相同的 Λ 的极小子集 Φ_i , 使得 $C \in \Phi_i$.

根据引理 7 的证明过程, 我们知道存在 Λ 的一个 SLD- 反驳. 假设这个 SLD- 反驳的输入序列是: $N_0, C_1, \dots, C_p$. 并令 $T_0 = S_{-}(N_0)$, $S_i = S(C_i)$, $T_i = S_{-}(C_i)$, 其中 $1 \leq i \leq p$. 显然 $T_p = \emptyset$. 下面先证明: 这一输入序列中的任何一个确定子句 C_i 都有一条到 N_0 的路径, 即存在 Λ 中的确定子句序列 $C_{i_1}, \dots, C_{i_s} = C_i$, 使得 $P_{j-1} \cap S_{i_j} = \emptyset$ 并且 $Q_{j-1} \cap S_{i_j} \neq \emptyset$ 对于所有 $1 \leq j \leq s$ 成立. 其中 P_j 和 Q_j 的定义如下: $P_0 = \emptyset, Q_0 = T_0$, $P_j = P_{j-1} \cup S_{i_j}$, $Q_j = (Q_{j-1} \cup T_{i_j}) - P_j$, $1 \leq j \leq s$.

(1) 如果 $i = 1$, 那么 N_0, C_1 就是一条从 C_1 到 N_0 的路径.

(2) 如果上述结论对 $i < m$ 成立, 那么当 $i = m$ 时, 由于 $S_{-}(N_{m-1}) \cap S_m \neq \emptyset$, 所以 C_m 中的正谓词符号 B 是 $N_0, C_1, \dots, C_{m-1}$ 中某个 Horn 子句的负谓词符号. 如果 B 是 N_0 的负谓词符号, 那么 N_0, C_m 就是一条从 C_m 到 N_0 的路径. 如果 B 是某个 C_i 的负谓词符号, 不妨设 B 第一次出现的子句是 C_q , 即对 $j < q$, B 不是 C_j 的负谓词符号, 其中 $1 \leq q \leq m-1$. 根据归纳假设可知, 存在一条从 C_q 到 N_0 的路径 $N_0, \dots, C_q$, 那么根据路径的定义, $N_0, \dots, C_q, C_m$ 就是一条从 C_m 到 N_0 的路径.

由归纳法可知, 对于 $1 \leq i \leq p$, 存在一条从 C_i 到 N_0 的路径. 由于 $\Lambda = \{N_0, C_1, \dots, C_p\}$, 所以这也就是说: 对于 Λ 中的任意确定子句 C_i , 存在一条从 C_i 到 N_0 的路径 $N_0, C_{i_1}, \dots, C_{i_s} = C_i$. 如果 $Q_s = \emptyset$, 那么 $\Phi_i = \{N_0, C_{i_1}, \dots, C_{i_s}\}$ 就是一个包含 C_i 和 N_0 并且正负谓词符号集合相同的 Λ 的极小子集. 如果 $Q_s \neq \emptyset$, 那么令 $S = \bigcup_{i=1}^p S_i$. 由于 $T_i \subseteq S$, $0 \leq i \leq p$, 所以对任意 j 有 $Q_j \subseteq S - P_j$ 成立. 由于 S 中只包含有穷个谓词符号, 所以存在一个 $n \leq |S|$ 使得 $N_0, C_{i_1}, \dots, C_{i_s}, \dots, C_{i_n}$ 是一条经过 C_i 的路径并且 $Q_n = \emptyset$. 因此, $\Phi_i = \{N_0, C_{i_1}, \dots, C_{i_s}, \dots, C_{i_n}\}$ 就是一个包含 C_i 和 N_0 并且正负谓词符号集合相同的 Λ 的极小子集.

由于 Λ 中的每一个 Horn 子句 C_i 都包含在某一个 Φ_i 中, 所以必然存在包含 Λ 中目标子句 N_0 并且正负谓词符号集合相同的 Λ 的极小子集 $\Phi_1, \dots, \Phi_k$, 使得 $\Lambda = \Phi_1 \cup \dots \cup \Phi_k$.

引理 10 证明 根据引理 8, 我们知道过程 Prepare 输出的是所有可能属于某个极小不协调集合的 Horn 子句的正负谓词符号集合和下标组成的三元组的集合, 并且根据其是确定子句还是目标子句分类为 ST_d 和 ST_g . 根据引理 7 的 (1), 每一个极小不协调集合都包含唯一一个目标子句. 所以先对每一个目标子句求出包含它的所有极小不协调集合, 再把这些极小不协调集合合并, 就得到了 Γ 的所有极小不协调子集. MIS 的第一个 forall 循环实际上就是这样做的. 首先 Candidates 计算出所有包含第 i 个目标子句并且满足引理 7 的 (3) 的 Γ 极小子集. 接下来过程 MIS* 找出这些极小子集的并集中哪些满足引理 7 的 (2) 并且不协调的, 哪些是不满足引理 7 的 (2) 的, 哪些是协调的, 哪些是无法判定的. 根据引理 9, 存放在 E_i^m 中的是包含第 i 个目标子句的极小不协调子集, 存放在 E_i^n 中的是包含第 i 个目标子句, 但算法无法判定是否不协调的候选集合. 将所有这些 E_i^m 和 E_i^n 分别合并就得到了由算法判定出的极小不协调子集组成的集合 E^m 和算法无法判定的候选集组成的集合 E^n . 由于用户从 E^n 中挑选出的极小不协调集合存放在 F^m 中, E^m 和 F^m 的并集就是 Γ 的所有极小不协调子集.

An algorithm to compute maximal contractions for Horn clauses

LUO Jie* & LI Wei

State Key Laboratory of Software Development Environment, School of Computer Science and Technology, Beihang University, Beijing 100191, China

*E-mail: luojie@nlsde.buaa.edu.cn

Abstract In the theory of belief revision, the computation of all maximal subsets (maximal contractions) of a formula set with respect to a set of facts is one of the key problems. In this paper, we try to solve this problem by studying the algorithm to compute all maximal contractions for Horn clauses. First, we point out and prove the conversion relationship between minimal inconsistent subsets of union of the formula set and the set of facts and maximal contractions of the formula set with respect to the set of facts. Second, we prove a necessary condition of a set of Horn clauses to be minimal inconsistent. Then, based on these two conclusions, we propose an interactive algorithm to enumerate all minimal inconsistent subsets of a given set of Horn clauses and a second algorithm to compute maximal contractions from these minimal inconsistent subsets. Finally, we proposed an interactive algorithm to compute maximal contractions for Horn clauses by composing the above two algorithms.

Keywords maximal contraction, minimal inconsistent set, minimal subtraction, Horn clause