基于否证蕴含的极小一阶不可满足子式求解算法

张建民 沈胜宇 李思昆
(国防科学技术大学计算机学院 长沙 410073)

摘 要 解释公式不可满足的原因在软件分析与验证等众多领域都具有非常重要的理论与应用价值,而极小不可满足子公式能够为公式不可满足的原因提供精炼的解释,帮助应用领域的自动化工具迅速定位错误,准确地诊断问题失败的本质缘由.文中针对极小一阶不可满足子式的求解问题,引入了否证蕴含图及其正向与逆向可达结点的概念,并证明了不可满足子式与否证蕴含图之间的关系.基于二者的关系,提出了基于冲突分析与否证蕴含的极小一阶不可满足子式求解算法,并融合了蕴含图剪枝技术,以提高算法效率.通过实验与当前最优的深度优先搜索算法进行了比较,结果表明:文中的算法显著优于深度优先搜索算法,并且随着公式复杂度的增加,性能优势更加明显.

关键词 一阶逻辑公式;可满足模理论问题;极小不可满足子式;消解否证;否证蕴含图
中图法分类号 TP302 **DOI 号**: 10.3724/SP.J.1016.2010.00415

An Algorithm for Extracting Minimal Unsatisfiable Subformulae in First-Order Logic Based on Refutation Implication

ZHANG Jian-Min SHEN Sheng-Yu LI Si-Kun
(*School of Computer, National University of Defense Technology, Changsha* 410073)

Abstract Explaining the causes of infeasibility of formulae has theoretical and practical applications in various fields, such as software verification and analysis. A minimal unsatisfiable subformula can provide a succinct explanation of infeasibility, and help automatic tools to rapidly locate the errors, and determine the underlying reasons for the failure. The authors introduce the definition of refutation implication graph and its forward and backward reachable vertices, and the relationship between the minimal unsatisfiable subformulae and the refutation implication graph. Based on the relationship, the authors propose an algorithm based on conflict analysis and refutation implication, in which a technique called refutation implication graph pruning is implemented. The experimental results on practical benchmarks show that the proposed algorithm outperforms the depth-first search algorithm. While the formulae are becoming more complex, the algorithm is much faster than the depth-first search algorithm.

Keywords first-order formula; satisfiability modulo theories(SMT); minimal unsatisfiable subformulae; resolution refutation; refutation implication graph

1 引 言

近年来,命题逻辑的可满足求解技术得到了飞速的发展,涌现了众多高效的 SAT 求解器,目前 SAT 求解器已经成为软件分析与验证等领域的重要工具.但是,命题逻辑限于其相对较弱的表达能力,无法表达许多领域的问题,例如,软件的分析与

收稿日期:2009-04-17;最终修改稿收到日期:2009-09-14.本课题得到国家自然科学基金(60603088)资助.**张建民**,男,1979 年生,博士研究生,主要研究方向为计算机辅助验证方法.E-mail: jmzhang@nudt.edu.cn.**沈胜宇**,男,1975 年生,博士,副研究员,主要研究方向为计算机辅助设计与验证方法.**李思昆**,男,1941 年生,教授,博士生导师,主要研究领域为计算机辅助设计与验证方法、虚拟现实.

验证[1]、实时系统的验证[2]等;而在另外一些应用中,如汇编代码与 RTL 级 Verilog 代码的验证问题[3],布尔逻辑虽然也能够表达与处理,但由于其抽象层次较低,例如字变量会转换为一组无关的布尔变量,因此大大增加了问题的规模,显著提高了空间与时间的开销.而基于非量化一阶逻辑的可满足问题,即可满足性模理论(Satisfiability Modulo Theories,SMT)问题恰好弥补了 SAT 技术的不足,迅速成为验证领域的研究热点.可满足性模理论问题源于实际应用需求,例如软件的形式化验证、电子设计自动化与人工智能等众多领域的问题,都能够规约为非量化一阶逻辑公式,采用 SMT 求解器来解决.当公式不可满足时,通常要求查找不可满足的原因,诊断与定位问题的错误,这就要求移除公式中与不可满足无关的短句,只保留一部分短句,也就是提取公式的不可满足子式.而极小不可满足子式,即其所有真子式都是可满足的,能够给出关于公式不可满足更加精确的解释,迅速地定位错误,应用也更加广泛[4-6].

近年来,涌现了众多提取布尔不可满足子式的算法[7-15],成为形式验证领域最活跃的分支之一.然而,自从抽象层次更高、表达能力更强的可满足性模理论出现之后,SAT 面临被 SMT 取代的趋势,并且随着 SMT 求解技术的飞速发展,SMT 求解器已经能够解决实际领域中较大规模的问题,为求解 SMT 不可满足子式提供了可靠的理论基础与实践平台,因此 SMT 不可满足子式的求解方法将成为今后研究的重点及主要突破的方向.Cimatti 等[16]首次提出了求解 SMT 不可满足子式的算法 Lemma Lifting,该算法采用分离的布尔不可满足子式求解算法与 SMT 求解器相结合的方法,但该算法并不能保证所得到的 SMT 不可满足子式的极小性.显然,相对于不可满足子式来说,极小不可满足子式的求解难度更大,算法复杂度更高.因此到目前为止,国际上还没有公开发表的针对如何求解 SMT 极小不可满足子式的研究成果.

针对极小 SMT 不可满足子式的求解问题,本文提出基于冲突分析与否证蕴含的极小 SMT 不可满足子式求解算法(Conflict-Analysis and Refutation-Implication Minimal Unsatisfiable Subformulae Extractor,CARI-MUSE).首先给出否证蕴含图及其正向与逆向可达结点的概念,而后证明它们与不可满足子式的关系.算法基于这些结论,通过记录 SMT 求解器在证明公式不可满足性的过程中产生的消解否证,同时构造其否证蕴含图,而后依次选择蕴含图中原始短句,通过检测原始短句的正向不可达结点集的可满足性,来确定该短句是否属于极小不可满足子式;若结点集是可满足的,则将该原始短句加入不可满足子式;否则,从否证蕴含图中删除该短句及其冲突短句,构造更小的否证蕴含图;如此反复迭代,直到遍历否证蕴含图中的原始短句,这时就得到极小 SMT 不可满足子式.为了提高搜索效率,算法中集成了蕴含图剪枝技术,该技术将证明正向不可达结点集的不可满足的步骤,转换为证明正向可达结点集中不存在一条其短句都为假的路径,大大减小了算法的搜索空间.实验结果表明,CARI-MUSE 算法的效率明显高于求解极小不可满足子式的深度优先搜索算法 DFS-MUSE①;并且随着公式复杂度的增加,性能优势更加显著.

2 背景知识

可满足模理论问题基于不包含全称与存在量词的一阶逻辑公式.根据一阶逻辑的定义,一个 n 元项 τ 可以表示为

$$\tau := x \mid f(\tau_1, \cdots, \tau_n) \tag{1}$$

其中 x 表示个体变元或常元,f 表示 n 元函数变元或常元.而非量化一阶逻辑公式 φ 表示为

$$\varphi := P(\tau_1, \cdots, \tau_n) \mid \tau_0 = \tau_1 \mid \neg\varphi_0 \mid \varphi_0 \vee \varphi_1 \mid \varphi_0 \wedge \varphi_1 \tag{2}$$

其中 τ_i 为项,$1 \leq i \leq n$,P 为 n 元谓词变元.

定义 1(可满足性模理论问题). 给出一个非量化的一阶逻辑公式 φ 以及一个赋值模型 M,M 包含一个非空域 $|M|$,其中对于 n 元函数变元 f,$M(f): f \to_n |M|$,对于谓词变元 P,$M(P) \subseteq |M|^n$,对于个体变元 x,$M(x) \in |M|$;公式 φ 中一个项 τ 的赋值为 $M[x] = M(x)$,$M[f(\tau_1, \cdots, \tau_n)] = M(f)(M[\tau_1], \cdots, M[\tau_n])$.那么,$M \models \varphi$ 定义为 $M \models P(\tau_1, \cdots, \tau_n) \Leftrightarrow (M[\tau_1], \cdots, M[\tau_n]) \in M(P)$;$M \models \tau_0 = \tau_1 \Leftrightarrow M[\tau_0] = M[\tau_1]$;$M \models \neg\varphi_0 \Leftrightarrow M \not\models \varphi_0$;$M \models \varphi_0 \vee \varphi_1 \Leftrightarrow M \models \varphi_0$ 或 $M \models \varphi_1$;$M \models \varphi_0 \wedge \varphi_1 \Leftrightarrow M \models \varphi_0$ 且 $M \models \varphi_1$,如果存在这样的赋值模型 M,使得 $M \models \varphi$,那么称公式 φ 是可满足的;否则,称公式 φ 是不可满足的.

可满足性模理论问题实际上是解决非量化一阶

① DFS-MUSE 算法及其实现见 http://www.ssypub.org/~zjm/pubs/unsmt_core_TR.pdf.它是由本文作者设计的极小 SMT 不可满足子式的求解算法.

逻辑公式的可满足性问题，其公式中的函数变元与谓词变元通常基于一些特定的理论域，而这些理论域都源于应用领域中实际问题的抽象. 目前比较常见的理论域包括整数集/实数集上的线性算术(LIA/LRA)、整数集/实数集上的差分逻辑(IDL/RDL)、等式与未解释函数(EUF)、数组(AR)以及位向量(BV)等. SMT 求解技术经过近几年的快速发展，相继出现了 Eager 方法、Lazy 方法以及最新的 DPLL(T)算法[17]，同时 SMT 求解器也逐渐走向成熟与完善，目前已经能够解决实际应用中较大规模的问题，为求解不可满足子式奠定了基础.

定义 2(SMT 不可满足子式). 给出一个不可满足的非量化一阶逻辑公式 φ，$\psi=\wedge_1^n C_i$ 是公式 φ 的一个 SMT 不可满足子式当且仅当 ψ 是不可满足的，并且 $\psi\subseteq\varphi$，即 $\forall i, 1\leq i\leq n$，若 $C_i\in\psi$，则 $C_i\in\varphi$，其中 C_i 表示短句.

定义 3(极小 SMT 不可满足子式). 给出不可满足的非量化一阶逻辑公式 φ 及其一个不可满足子式 ψ，那么 ψ 是极小 SMT 不可满足子式当且仅当 $\forall\phi\subset\psi$，使得 ϕ 是可满足的.

对于 SMT 公式来说，如果其某个不可满足子式的所有真子式都是可满足的，那么它是极小不可满足子式. 显然，相对于不可满足子式来说，极小不可满足子式的求解难度更大，算法复杂度也更高. 但是极小不可满足子式能够给出关于公式不可满足原因更加精确的解释，迅速诊断与定位错误，在实际应用中具有更重要的理论与应用价值.

由于在 SMT 求解器中，都将输入公式转换为合取范式形式从而证明其可满足性，因此可以利用 CNF 公式的消解理论来提取极小不可满足子式. 下面给出消解的定义与消解原理.

定义 4(消解). 设 C_i 与 C_j 为两个短句，若 $l_i\in C_i$ 与 $l_j\in C_j$ 是一对互补的文字，则称 $(C_i\backslash l_i)\vee(C_j\backslash l_j)$ 为 C_i 与 C_j 的消解式，其中 l_i 和 l_j 称为消解基，C_i 和 C_j 称为消解母式.

引理 1[18]. 若短句 C 为 C_i 与 C_j 的消解式，则 $(C_i,C_j)\models C$.

引理 2[18]. 若 $C_i=l_i$ 与 $C_j=l_j$ 为两个单元短句，并且 l_i 和 l_j 是一对互补的文字，则 C_i 与 C_j 的消解式为空短句，即 $(C_i,C_j)\models\perp$.

定义 5(消解序列). 设 S 为短句集，且 C 为短句，若存在短句的有穷序列 $C_0,\cdots,C_n$，满足

(1) $C_n=C$；

(2) 令 $0\leq i\leq n$，则短句 C_i 至少满足下列两个条件之一：

(i) $C_i\in S$；

(ii) $\exists j,k$，使得 $(C_j,C_k)\models C_i$，其中 $0\leq j,k<i$，

那么称短句 C 为 S 的消解结果，表示为 $S|\text{-}|\text{-}C$，并将 $C_0,\cdots,C_n$ 称为由 S 导出 C 的消解序列.

引理 3[18]. 设 S 为短句集，且 C 为短句，若 $S|\text{-}|\text{-}C$，则 $S\models C$.

引理 4(消解原理)[18]. 短句集 S 为不可满足的当且仅当 $S|\text{-}|\text{-}\perp$.

3　算法原理

如果根据定义 3 的方法求解极小 SMT 不可满足子式，假设一个不可满足子式 $|\psi|=n$，那么要进行 2^n-2 次可满足性的判断，才能确定 ψ 的极小性. 所以算法的复杂度非常高，执行效率也比较低. 经过分析，能够得出下面的结论.

引理 5. 给出不可满足的非量化一阶逻辑公式 φ，及其一个不可满足子式 $\psi=\wedge_1^n C_i$，其中 C_i 为短句，那么 ψ 是极小 SMT 不可满足子式，当且仅当从 ψ 中删除任意一个短句 $\forall i, C_i\in\psi, 1\leq i\leq n$，都使得 $\psi\backslash C_i$ 是可满足的.

采用反证法易证引理 5，这里不再赘述. 根据该结论，算法只需将不可满足子式删除其中任意一个短句后，测试剩余短句构成公式的可满足性，只需进行 n 次可满足性判断，即可确定其极小性，从而大大简化了求解极小 SMT 不可满足子式的过程，降低了算法复杂性.

根据消解原理，若短句集是不可满足的，那么经过有限步消解可以得到空短句，因此将消解序列的概念延伸，得到消解否证的定义.

定义 6(消解否证). 给出一个不可满足的 SMT 公式 φ，令集合 $Cla(\varphi)=\{C\mid C$ 为 φ 中的短句$\}$，若 $\{C_0,\cdots,C_n\}$ 是 $Cla(\varphi)$ 导出的消解序列，且 $C_n=\perp$，则称 $R=\{C_0,\cdots,C_{n-1},\perp\}$ 为公式 φ 的一个消解否证.

为了能够高效地求解极小不可满足子式，需要将 SMT 求解器产生的从原始公式到空短句的消解过程记录并转换为一种简洁清晰的数据结构，因此引入了短句蕴含图的概念.

定义 7(短句蕴含图). 给出一个不可满足的 SMT 公式 φ，$G(V,E)$ 为一个有向无环图. 假设 $V=V^r\cup V^c$，其中 V^r 是由 G 中所有始发结点构成的集合，即 V^r 中结点的入度为 0，并且 $\forall v_i\in V^r$，结点 v_i

对应的短句 $C_i\in\varphi$;而 V^c 是由 G 中所有非始发结点构成的集合,由消解结果短句与理论求解器返回的学习短句构成,称为冲突短句,记为 $D=\{C_1^c,C_2^c,\cdots,C_n^c\}$. 那么,

(1) $V^r=Cla(\varphi)$,其中 $Cla(\varphi)=\{C\mid C$ 为 φ 中的短句$\}$;

(2) 对于每一个冲突短句 C_i^c,都存在一个消解序列 $S_i=\{C_{i1},C_{i2},\cdots,C_{ij},C_i^c\}$,使得

(i) $\forall k,1\leq k\leq j$,C_{ik} 满足下列两个条件之一:① $C_{ik}\in Cla(\varphi)$;② $\exists 1\leq l\leq i,C_l^c\in D$,使得 $C_{ik}=C_l^c$;

(ii) $\forall m,1\leq m\leq j$,在 E 中存在唯一的边 e_{im},对应于 $C_{im}\rightarrow C_i^c$;

(3) 出度为 0 的最终结点对应的短句为 $C_n^c=\perp$.

若满足上面的条件,则 $G(V,E)$ 称为公式 φ 的短句蕴含图.

而不可满足子式本质上反映的是短句蕴含图中结点之间的逻辑关系,因此引入短句蕴含图中正向可达结点与逆向可达结点的概念.

定义 8(正向可达结点). 给出 SMT 公式 φ,$G(V,E)$ 为 φ 的一个短句蕴含图,如果从结点 α 经过 $n(n\geq 0)$ 条边能够到达结点 β,即存在一条 α 到 β 的路径,那么称 β 是 α 的正向可达结点. 在 $G(V,E)$ 中,从结点 α 出发所有正向可达结点构成的集合表示为 $FR_v(G,\alpha)$;而从结点 α 出发所有正向不可达的结点构成的集合表示为 $\overline{FR_v}(G,\alpha)$.

定义 9(逆向可达结点). 给出 SMT 公式 φ,$G(V,E)$ 为 φ 的一个短句蕴含图,如果从结点 β 经过 $n(n\geq 0)$ 条边能够到达结点 α,即存在一条 β 到 α 的路径,那么称 β 是 α 的逆向可达结点. 在 $G(V,E)$ 中,到达 α 的所有逆向可达结点构成的集合表示为 $BR_v(G,\alpha)$;而不可到达 α 的结点构成的集合表示为 $\overline{BR_v}(G,\alpha)$.

定理 1. 给出一个不可满足的 SMT 公式 φ,存在一个 φ 的消解序列 $R=\{C_0,\cdots,C_{n-1}\}$,当且仅当存在一个对应于 R 的短句蕴含图 $G(V,E)$.

证明. 首先证明充分性,针对消解序列 R 所包含的短句数 n 采用数学归纳法证明.

当 $n=1$ 时,R 只包含 1 个短句,显然成立;当 $n=3$ 时,假设 $R=\{C_0,C_1,C_2\}$,则消解过程为 $(C_0,C_1)\models C_2$,且 $C_0,C_1\in\varphi$;根据定义 7,$V^r=\{C_0,C_1\}$,$V^c=\{C_2\}$,$E=\{e_1,e_2\}$,其中边 e_1 与 e_2 分别由 C_0 与 C_1 指向 C_2,构成 $G(V^r\cup V^c,E)$,满足短句蕴含图的定义,因此结论成立;

假设 $n\leq m$,结论成立;

当 $n=m+1$ 时,R 产生最终消解结果 C_k 的步骤表示为 $(C_i,C_j)\models C_k$,根据定义 7 构造短句蕴含图,此时存在两种情况:

第 1 种情况. C_i 与 C_j 中只有一个是原始短句,这里假设 $C_j\in\varphi$;而 C_i 的消解序列 $R_i=\{C_0,\cdots,C_i\}$ 包含至多 m 个短句,符合假设条件,那么存在一个对应于 R_i 的短句蕴含图 $G_i(V_i^r\cup V_i^c,E_i)$. 而后令 $V^r=V_i^r\cup\{C_j\}$,$V^c=V_i^c\cup\{C_k\}$,$E=E_i\cup\{e_1,e_2\}$,其中 e_1 与 e_2 分别由 C_i 与 C_j 指向 C_k,构成 $G(V^r\cup V^c,E)$,满足定义 7,因此 $G(V^r\cup V^c,E)$ 为短句蕴含图.

第 2 种情况. $C_i\notin\varphi$,且 $C_j\notin\varphi$,假设 C_i 与 C_j 的消解序列分别为 R_i 和 R_j,其包含的短句数记为 n_i 与 n_j. 由于 $k\leq m$,且 $C_i\notin\varphi,C_j\notin\varphi$,因此 $n_i\leq m-1$,$n_j\leq m-1$,符合假设条件,那么存在 R_i 与 R_j 对应的短句蕴含图 $G_i(V_i^r\cup V_i^c,E_i)$ 和 $G_j(V_j^r\cup V_j^c,E_j)$. 而后,令 $V^r=V_i^r\cup V_j^r$,$V^c=V_i^c\cup V_j^c\cup\{C_k\}$,$E=E_i\cup E_j\cup\{e_1,e_2\}$,其中 e_1 与 e_2 分别由 C_i 与 C_j 指向 C_k,从而构成 $G(V^r\cup V^c,E)$,满足定义 7,因此 $G(V^r\cup V^c,E)$ 为短句蕴含图.

所以,给出一个消解序列 R,则必定存在一个对应于 R 的短句蕴含图 $G(V,E)$.

下面证明结论的必要性,假设存在一个短句蕴含图 $G(V^r\cup V^c,E)$,那么针对结点数 $|V|$ 采用数学归纳法进行证明.

当 $|V|=1$ 时,G 只包含一个空短句,显然成立;当 $|V|=3$ 时,根据定义 7,$V^c=\{C_2\}$,$V^r=\{C_0,C_1\}$,且 $(C_0,C_1)\models C_2$,那么对应于 G 的消解序列 $R=\{C_0,C_1,C_2\}$;

假设 $|V|\leq m$ 时,结论成立;

当 $|V|=m+1$ 时,假设 G 的最终结点为 C_k,且 $(C_i,C_j)\models C_k$,$e_{ik}:C_i\rightarrow C_k$,$e_{jk}:C_j\rightarrow C_k$,那么将 G 中的结点 C_k 以及边 e_{ik} 与 e_{jk} 删除,那么剩余的结点和边分别构成两个子图 $G_i(V_i,E_i)$ 与 $G_j(V_j,E_j)$,符合定义 7,G_i 和 G_j 为短句蕴含图,并且满足 $|V_i|\leq m$,$|V_j|\leq m$. 根据假设条件,可以得到 G_i 与 G_j 的消解序列 R_i 和 R_j;那么令短句集合 $R=\{C\mid C\in R_i$ 或 $C\in R_j$ 或 $C=C_k\}$,由于 $(C_i,C_j)\models C_k$,且 $C_i\in R_i$,$C_j\in R_j$,因此 R 构成 C_k 的消解序列.

所以,给出一个短句蕴含图 $G(V,E)$,则必定存在一个对应于 G 的消解序列 R.

综合上面充分性与必要性的证明,得到结论:存在一个 φ 的消解序列 R,当且仅当存在一个对应于 R 的短句蕴含图 $G(V,E)$. 证毕.

在证明输入公式不可满足性的过程中，通过修改 SMT 求解器，使其记录所有冲突结点的消解序列，以及空短句的消解否证，同时逐步构造短句蕴含图 $G(V,E)$. 通常来讲，并不是 $G(V,E)$ 中所有短句都参与到消解空短句的过程，仅是空短句的逆向可达结点集 $BRv(G,\perp)$ 才参与该过程. 因此能够将 $G(V,E)$ 化简，把所有与空短句消解无关的结点及其边都删除，只保留消解否证中的结点，那么这种非冗余的短句蕴含图称为否证蕴含图. 下面给出否证蕴含图的定义.

定义 10（否证蕴含图）. 给出不可满足公式 φ，及其一个短句蕴含图 $G(V,E)$，如果从 G 中的每个短句 C 出发，都至少存在一条路径到达空短句 $\perp$，那么 G 为否证蕴含图，记为 $G_R(V_\perp,E_\perp)$.

如果将定理 1 的结论扩展到否证蕴含图与消解否证上，是否仍然成立？经过分析，给出下面的结论.

定理 2. 给出一个不可满足的 SMT 公式 φ，存在一个 φ 的消解否证 $R=\{C_0,\cdots,C_{n-1},\perp\}$，当且仅当存在一个对应于 R 的否证蕴含图 $G_R(V_\perp,E_\perp)$.

证明. 将定理 1 中的消解序列的概念扩展到消解否证，而短句蕴含图的概念相应地延伸到否证蕴含图，根据定义 6 与 10，易证.

给出一个不可满足的 SMT 公式 φ 以及 φ 的一个短句蕴含图 $G(V,E)$，那么否证蕴含图 $G_R(V_\perp,E_\perp)=G_R(V^r\cup V^c,E_\perp)$ 的构造方法为：其结点集合 $V_\perp=\{v_i\mid v_i\in BRv(G_R,\perp)\}$，即短句蕴含图中所有空短句 $\perp$ 的逆向可达结点集，其中 $V^r=V_\perp\cap\varphi$，$V^c=V_\perp-V^r$；而边的集合 $E_\perp=\{e_{ij}:v_i\to v_j\mid v_i,v_j\in BRv(G_R,\perp)\}$，即 $E_\perp$ 由所有始点与终点均在 $BRv(G_R,\perp)$ 中的边构成. 那么，根据公式 φ 的否证蕴含图 $G_R(V_\perp,E_\perp)$，如何得到 SMT 不可满足子式？下面的结论给出解决方法.

定理 3. 给出不可满足 SMT 公式 φ 的一个消解否证 R，以及 R 对应的否证蕴含图 $G_R(V_\perp,E_\perp)$，那么 $BRv(G_R,\perp)\cap\varphi$ 中的短句合取构成的公式是 φ 的一个不可满足子式.

证明. 假设 ψ 为 $BRv(G_R,\perp)\cap\varphi$ 中所有短句通过合取构成的公式. 那么 $\forall C\in\psi$，由于 $C\in BRv(G_R,\perp)$，根据定义 9 与 10，表明从短句 C 对应的结点出发，存在一条路径到达空短句 $\perp$，也就是说 C 参与到空短句的消解过程；根据引理 4，$S_\psi\mid\!-\!\mid\!-\perp$，其中 S_ψ 表示由公式 ψ 中短句构成的短句集，则 ψ 是不可满足的. 并且证明过程基于 $\forall C\in\varphi$，所以 $\psi\subseteq\varphi$，且 ψ 是不可满足的，根据定义 2，ψ 为 φ 的一个不可满足子式. 证毕.

下面通过一个例子来说明否证蕴含图及其与不可满足子式的关系. 假设一个 CNF 形式的 SMT 公式 φ 为

$$\begin{aligned}\varphi=&(a\vee\neg(x<0)\vee(y\geq1))\wedge\neg(x<0)\wedge\\&(a\vee(x<0)\vee(y\geq1))\wedge(\neg a\vee(x+y>3))\wedge\\&(a\vee(x<0)\vee\neg(y\geq1))\wedge((x<0)\vee\\&(x+y>3))\wedge(\neg a\vee\neg(x+y>3))\end{aligned}\tag{3}$$

公式 φ 基于整数集上的线性算术（LIA）理论域，其中 x 和 y 为个体变元，a 是命题变元. 图 1 给出了公式 φ 的一个否证蕴含图 $G_R(V^r\cup V^c,E_\perp)$. 根

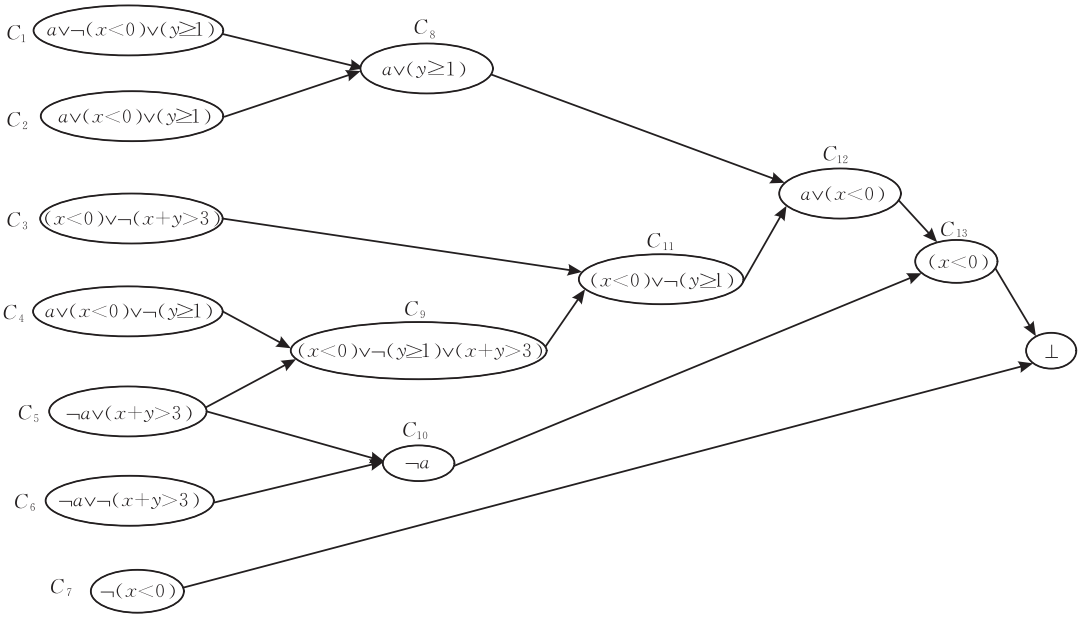


图 1　公式 φ 的否证蕴含图

据定义 10，$V^r=\{C_1,C_2,C_3,C_4,C_5,C_6,C_7\}$，$V^c=\{C_8,C_9,C_{10},C_{11},C_{12},C_{13},\perp\}$，并且 $V^r\cup V^c$ 中的所有结点到空短句都是可达的，即 $V^r\cup V^c=BR_v(G,\perp)$. 在图 1 中，短句 C_4 的正向可达结点集为 $FR_v(G_R,C_4)=\{C_4,C_9,C_{11},C_{12},C_{13},\perp\}$，而短句 C_{12} 的逆向可达结点集为 $BR_v(G_R,C_{12})=\{C_1,C_2,C_3,C_4,C_5,C_8,C_9,C_{11},C_{12}\}$. 根据定理 3，$\varphi$ 的不可满足子式为 $x=C_1\wedge C_2\wedge C_3\wedge C_4\wedge C_5\wedge C_6\wedge C_7$.

4 基于否证蕴含的求解算法

求解极小 SMT 不可满足子式的 CARI-MUSE 算法基于定理 3 的结论，其主要过程为：通过保存 SMT 求解器在证明公式不可满足性时产生的一系列消解步骤，构成消解否证，并转换为一个否证蕴含图；但是定理 3 只能从否证蕴含图中提取不可满足子式，并不保证其极小性，因此算法需要进一步地推演. 首先依次选择蕴含图中的原始短句，删除该短句及其相关的冲突短句，将否证蕴含图中的剩余结点构成一个子公式，调用 SMT 求解器检测其可满足性，此时通过蕴含图剪枝技术加速该过程；若子公式是可满足的，则将当前的原始短句加入到不可满足子式中；否则，表明原始短句不属于极小不可满足子式，因此将该短句及其冲突短句对应的结点与边从蕴含图中移除，同时保证非冗余性，从而构造一个更小的否证蕴含图；而后不断循环，直到遍历否证蕴含图中的所有原始短句，此时就得到极小 SMT 不可满足子式. CARI-MUSE 算法的伪代码如图 2 所示.

CARI-MUSE 算法的输入为 SMT-LIB 格式的公式 φ，目标是得到 φ 的极小不可满足子式. 首先，算法解析输入公式，并利用 SMT 求解器来求解其可满足性，并保存所有的消解步骤，同时构造一个短句蕴含图 $G(V,E)$，但当前的 $G(V,E)$ 对于空短句的消解过程来说可能是冗余的，因此需要将其化简为一个非冗余的短句蕴含图，即否证蕴含图. 这里令 $G_R(V_\perp,E_\perp)=G_R(V^r\cup V^c,E_\perp)$ 表示一个否证蕴含图，其中 V^r 是指 G_R 中所有始发结点的集合，对应于原始公式 φ 中的短句，而 V^c 代表所有非始发结点构成的集合，即中间产生的消解结果短句与理论求解器输出的学习短句.

而后算法利用函数 ChooseClause 选择 V^U 中的一个原始短句 C_α，检测其是否属于极小 SMT 不可满足子式. 在算法构造否证蕴含图 $G(V,E)$ 的过程中，同时求解结点 α 的正向不可达结点集合 $\overline{FR_v}(G_R,\alpha)$. 根据定义 8，正向不可达结点集 $\overline{FR_v}(G_R,\alpha)$ 是指在 $G(V,E)$ 中从 α 出发所有不可达的结点构成的集合. 为了检测结点 α 对应的短句 C_α 是否属于极小不可满足子式，从 $G_R(V^r\cup V^c,E_\perp)$ 中删除 α 的正向可达结点集合 $FR_v(G_R,\alpha)$，即令 $\overline{FR_v}(G_R,\alpha)$ 中结点对应的短句构成的公式 ϕ 作为输入，调用 SMT 求解器求解 ϕ 的可满足性. 由于 $G_R(V_\perp,E_\perp)$ 是否证蕴含图，因此空短句 $\perp$ 总是属于 α 的正向可达结点集合 $FR_v(G_R,\alpha)$，也就是，从 α 出发经过有限条边最终总能到达结点 $v_\perp$，所以空短句 $\perp$ 不会包含在公式 ϕ 中. 那么，算法以 $\overline{FR_v}(G_R,\alpha)$ 转换后的公式 ϕ 作为输入公式，要么证明 ϕ 是可满足的，要么最终产生空短句.

算法. 极小 SMT 不可满足子式的求解算法 CARI-MUSE
输入：SMT 公式 φ
输出：极小不可满足子式 *MinimalUS*
1. 解析公式 φ，转换为内部数据结构；
2. 记录 SMT 求解器的消解过程 S_R；
3. 根据 S_R 为公式 φ 构造一个短句蕴含图 $G(V,E)$；
4. 将 $G(V,E)$ 化简为否证蕴含图 $G_R(V^r\cup V^c,E_\perp)$；
5. $MinimalUS=\varnothing$；
6. $V^U=V^r$；
7. while ($V^U\neq\varnothing$) do
8. $\quad C_\alpha=\text{ChooseClause}(V^U)$；
9. $\quad M=\overline{FR_v}(G_R,\alpha)$；
10. $\quad$ 将 M 转化为子公式 ϕ；
11. $\quad$ if (SMT solver return ϕ is satisfiable) then
12. $\quad\quad MinimalUS=MinimalUS\cup\{C_\alpha\}$；
13. $\quad\quad V^U=V^U-\{C_\alpha\}$；
14. $\quad$ else
15. $\quad\quad$ 记录求解器的消解过程 S_M；
16. $\quad\quad$ 根据 S_M 构造 ϕ 的短句蕴含图 $G_M(V^r_M\cup V^c_M,E_M)$；
17. $\quad\quad V^r_N=V^r-\{C_\alpha\}$；
18. $\quad\quad V^c_N=\overline{FR_v}(G_R,\alpha)\cup V^c_M$；
19. $\quad\quad E_N=\{E_\perp-E_{FR_v(G_R,\alpha)}\}\cup E_M$；
20. $\quad\quad$ 转换为 φ 的短句蕴含图 $G_N(V^r_N\cup V^c_N,E_N)$；
21. $\quad\quad$ 将 G_N 化简为 φ 的否证蕴含图 $G_P(V^r_P\cup V^c_P,E_P)$；
22. $\quad\quad G_R(V^r\cup V^c,E_\perp)\leftarrow G_P(V^r_P\cup V^c_P,E_P)$；
23. $\quad\quad V^U=V^r_P$；
24. return *MinimalUS*.

图 2 CARI-MUSE 算法

如果 SMT 求解器证明 $\overline{FR_v}(G_R,\alpha)$ 对应的公式 ϕ 是可满足的，令 $\varphi\backslash C_\alpha$ 表示从原始公式 φ 中移除短句 C_α 后的子公式，那么表明能够得到 $\varphi\backslash C_\alpha$ 的可满足赋值模型，所以短句 C_α 属于极小 SMT 不可满足子式，将 C_α 加入 *MinimalUS* 集合中，而后跳转到 V^U 中的下一个原始短句. 如果 SMT 求解器返回公式 ϕ 是不可满足的，表示 C_α 不属于当前所求解的极小不可满足子式，此时算法从否证蕴含图 $G_R(V_\perp,E_\perp)$

中移除 C_α 及其冲突短句所对应的结点；但是必须保证删除这些结点与边后，G_R 仍为一个否证蕴含图，即算法要基于 $\overline{FR\upsilon}(G_R,\alpha)$ 中的结点构造一个更小的但完整的否证蕴含图，下面介绍其构造方法.

由于公式 ϕ 是不可满足的，因此求解器返回一个消解过程 S_M，而后算法构造一个公式 ϕ 的短句蕴含图 $G_M(V_M^r \cup V_M^c, E_M)$. 根据定义 7，短句蕴含图的所有始发结点必须是原始公式中的短句，而 G_M 的所有始发结点集合 V_M^r 对应的短句集合为 $\overline{FR\upsilon}(G_R,\alpha)$，因此 G_M 仅为 ϕ 的短句蕴含图，而不构成原始公式 φ 的短句蕴含图，不能将其作为算法后面循环的输入. 但是，$G_M(V_M^r \cup V_M^c, E_M)$ 中冗余的始发结点都是 $G_R(V_\perp, E_\perp)$ 中的冲突短句，并且从结点 α 出发是不可达的，因此可以从集合 $V^r \backslash C_\alpha$ 中提取这些冲突短句的始发结点，并且将原始图 $G_R(V_\perp, E_\perp)$ 中那些从结点 $v_i \in V^r \backslash C_\alpha$ 到 $v_j \in \overline{FR\upsilon}(G_R,\alpha)$ 的边：$v_i \rightarrow v_j$ 添加到 G_M 中，就能够保证其始发结点对应的短句都属于 φ，从而把 G_M 扩展为一个短句蕴含图.

根据前面的步骤，CARI-MUSE 算法能够构造一个新的短句蕴含图 $G_N(V_N^r \cup V_N^c, E_N)$，其始发结点集合 $V_N^r = V^r \backslash C_\alpha$，冲突短句集合 $V_N^c = \overline{FR\upsilon}(G,\alpha) \cup V_M^c$，边的集合是 $E_N = (E_\perp - E_F) \cup E_M$，其中 E_F 表示所有起点或终点属于 $FR\upsilon(G_R,\alpha)$ 的边. 由于 $G_N(V_N^r \cup V_N^c, E_N)$ 可能是冗余的，不构成原始公式 φ 的否证蕴含图，因此要将 G_N 化简为否证蕴含图 $G_P(V_P^r \cup V_P^c, E_P)$. 在化简的过程中，$G_N$ 中的一些原始短句也可能会被删除，因此在每次检测到一个原始短句 C_α 不属于极小不可满足子式时，算法不仅要删除 C_α，而且可能移除其它一些原始短句. 而后算法检 V_P^r 是否为空，若不为空，算法通过函数 ChooseClause 选择下一个原始短句，继续循环；否则，算法终止，此时 *MinimalUS* 即为公式 φ 的极小 SMT 不可满足子式. 另外，改变 ChooseClause 函数的选择策略，以不同的顺序输出短句，能够得到不同的极小 SMT 不可满足子式.

根据定理 3 的结论，算法仅能从 SMT 公式的否证蕴含图中提取不可满足子式，而不能保证其极小性，因此算法必须在否证蕴含图的基础上进一步演绎，才能得到极小 SMT 不可满足子式. 下面给出图 2 所示的 CARI-MUSE 算法的正确性证明.

定理 4.　给出一个不可满足 SMT 公式 φ，那么 CARI-MUSE 算法的最终结果 *MinimalUS* 为 φ 的一个极小 SMT 不可满足子式.

证明.　根据定理 3，算法的 1～4 行得到了 φ 的一个不可满足子式，即否证蕴含图 $G_R(V^r \cup V^c, E_\perp)$ 的结点集 V^r 对应的所有短句的合取，记为 ψ.

而后算法从 V^r 中移除短句 C_α，将 $\overline{FR\upsilon}(G_R,\alpha)$ 转换后的公式 ϕ 作为求解器的输入. 这时存在两种情况：

第 1 种情况(11～13 行)：ϕ 是可满足的，即 $\psi - \{C_\alpha\}$ 是可满足的，而 ψ 是不可满足子式，根据引理 5，C_α 属于极小不可满足子式，保存到 *MinimalUS* 中；

第 2 种情况(15～23 行)：ϕ 是不可满足的，即从 ψ 中移除 C_α 不改变 ψ 的最终消解结果，所以 C_α 不属于极小不可满足子式；而后算法移除 C_α 及其相关的冲突短句，构造一个更小的短句蕴含图 $G_N(V_N^r \cup V_N^c, E_N)$，并将其化简为 φ 的否证蕴含图 $G_P(V_P^r \cup V_P^c, E_P)$；由于保证了下一次循环的输入始终为非冗余的否证蕴含图，G_P 保留了构成极小不可满足子式的短句，那些在化简过程中删除的原始短句对于空短句的消解来说是冗余的，因此它们不属于极小不可满足子式.

算法继续循环，直到 ψ 为空. 由于 $MinimalUS \subseteq \psi$，根据上面的证明，每次加入的短句 C_α 都是必要的，而删除的短句都是冗余，所以，*MinimalUS* 为 φ 的一个极小 SMT 不可满足子式.　　证毕.

以式(3)给出的公式 φ 为例来说明 CARI-MUSE 算法的执行过程. 假设函数 ChooseClause 返回的短句是 C_1，那么算法检测 C_1 是否属于极小不可满足子式. 而 $\overline{FR\upsilon}(G_R,C_1) = \{C_2, C_3, C_4, C_5, C_6, C_7, C_9, C_{10}, C_{11}\}$，那么子公式为 $\phi = C_2 \wedge C_3 \wedge C_4 \wedge C_5 \wedge C_6 \wedge C_7 \wedge C_9 \wedge C_{10} \wedge C_{11}$. 将 ϕ 作为 SMT 求解器的输入，那么得到 ϕ 是不可满足的，并返回 ϕ 的短句蕴含图 $G_M(V_M^r \cup V_M^c, E_M)$，其中 $V_M^r = \{C_2, C_7, C_{10}, C_{11}\}$，$V_M^c = \{a \vee (x<0), a, \perp\}$，令 $C_{14} = a \vee (x<0)$，$C_{15} = a$，$C_{16} = \perp$，$e_{i,j}: C_i \rightarrow C_j$，则 $E_M = \{e_{2,14}, e_{11,14}, e_{14,15}, e_{7,15}, e_{10,16}, e_{15,16}\}$. 但是 V_M^r 中的短句不都属于原始公式 φ，因此算法将其扩展为公式 φ 的短句蕴含图 $G_N(V_N^r \cup V_N^c, E_N)$，如图 3 所示. 由于这个例子比较简单，$G_N(V_N^r \cup V_N^c, E_N)$ 也为 φ 的否证蕴含图，表明算法从 φ 的初始否证蕴含图 $G_R(V^r \cup V^c, E_\perp)$ 中移除了结点 $\{C_1, C_8, C_{12}, C_{13}\}$. 而后算法以 $G_N(V_N^r \cup V_N^c, E_N)$ 作为输入，继续循环.

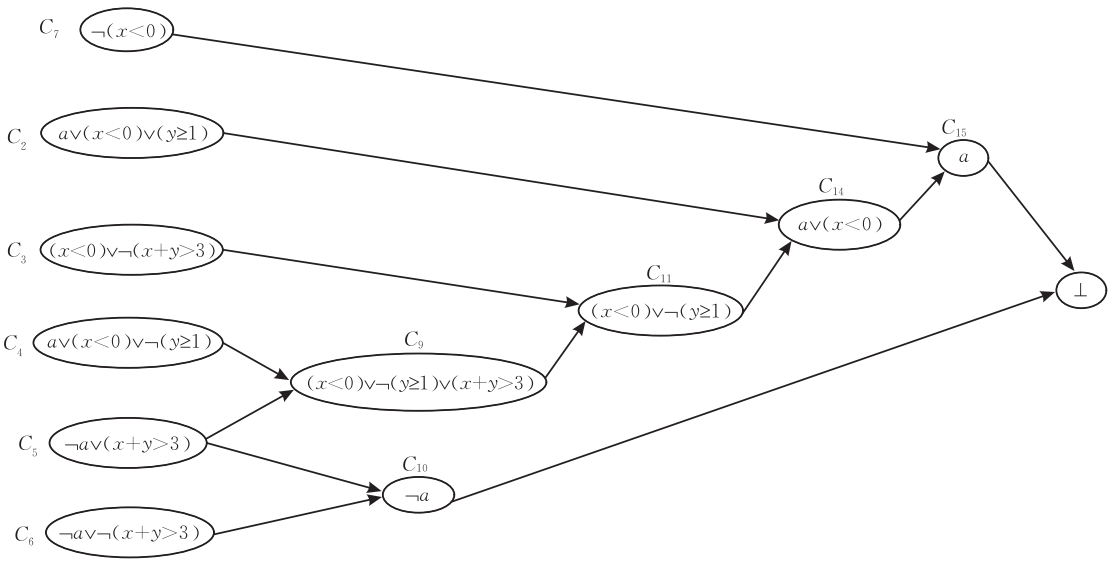


图 3 移除短句 C_1 后的否证蕴含图

5 蕴含图剪枝技术

为了提高 CARI-MUSE 算法的效率，实现了一种基于否证蕴含图的剪枝技术(Refutation Implication Graph Pruning，RIGPruning). 该优化方法的基本思想是：给出一个不可满足的公式 φ，将其划分为两个子公式 $\varphi=\phi\wedge\psi$，那么当一个部分赋值模型 M 使 $\phi=$true 时，则 $\psi=$false. 那么算法求解 $\overline{FR\upsilon}(G_R,\alpha)$ 对应的公式 ϕ 的可满足性赋性时，只要搜索 $FR\upsilon(G_R,\alpha)$ 中所有从结点 α 到空短句的路径，证明不存在一条这样的路径：该路径上的所有短句都为 false，那么就可以表明 ϕ 是不可满足的，这样能够显著地减小变元赋值的搜索空间，加速公式可满足性的判定过程. 下面给出蕴含图剪枝技术原理的证明过程.

定理 5. 给出不可满足的 SMT 公式 φ 及其否证蕴含图 $G_R(V_\perp,E_\perp)$，若一个部分赋值模型 $M\models\overline{FR\upsilon}(G_R,\alpha)$，那么存在一个从 α 到空短句的路径 $P=\{\alpha,\cdots,\upsilon_\perp\}\subseteq FR\upsilon(G_R,\alpha)$，$M$ 使得 $S=\{C_\alpha,\cdots,\perp\}$ 中的所有短句都为 false.

证明. 采用反证法，假设不存在从 α 到空短句的路径 $P=\{\alpha,\cdots,\upsilon_\perp\}$.

根据定义 8 与假设，空短句 $\perp$ 对应的结点 $\upsilon_\perp$ 不包含在 α 的可达结点集合中，即 $\upsilon_\perp\in\overline{FR\upsilon}(G,\alpha)$.

由于 $M\models\overline{FR\upsilon}(G,\alpha)$，即部分赋值模型 M 使得 $\overline{FR\upsilon}(G,\alpha)$ 中的所有短句都为 true，那么 $\overline{FR\upsilon}(G,\alpha)$ 中存在一个结点割集 V_s，其对应的短句通过合取构成的子公式 φ_s 是可满足的. 但是空短句 $\upsilon_\perp\in V_s$，根据引理 4 得到，φ_s 是不可满足的，产生矛盾.

所以，假设错误. 结论成立. 证毕.

根据定理 5 的逆否命题：如果某个赋值模型 M 使得 $FR\upsilon(G,\alpha)$ 中从结点 α 到空短句的某条路径上的每个短句都为 false，那么 $M\models\overline{FR\upsilon}(G,\alpha)$，即 M 为 $\overline{FR\upsilon}(G,\alpha)$ 的可满足赋值模型. 蕴含图剪枝技术利用定理 5 的逆否命题，如果证明 $G_R(V_\perp,E_\perp)$ 中不存在这样的路径 P：P 上所有短句都为 false，那么就说明 $\overline{FR\upsilon}(G,\alpha)$ 是不可满足的. 通常来说，$\overline{FR\upsilon}(G,\alpha)$ 所包含的短句数远远大于 $FR\upsilon(G,\alpha)$ 的短句数，因此在大多数情况下采用这种方法能够大大简化求解 $\overline{FR\upsilon}(G,\alpha)$ 的可满足性的过程，从而提高算法的效率.

以式(3)给出的公式 φ 为例来说明定理 5 的结论，$G_R(V^r\cup V^c,E_\perp)$ 为 φ 的否证蕴含图，则 $\overline{FR\upsilon}(G_R,C_5)=\{C_1,C_2,C_3,C_4,C_6,C_7,C_8\}$，那么使得 $\overline{FR\upsilon}(G_R,C_5)$ 可满足的部分赋值模型 $M=\{a,\neg(x+y>3),\neg(x<0)\}$，那么存在一条路径 $P=\{C_5,C_{10},C_{13},\perp\}\subseteq FR\upsilon(G_R,C_5)$，$M$ 使得 P 中的每个短句都为 false.

函数 RIGPruning 实现了否证蕴含图的剪枝技术，集成于 SMT 求解器中. RIGPruning 的基本运行过程为：首先，将 $FR\upsilon(G,\alpha)$ 中所有结点 α 到结点 $\upsilon_\perp$ 的路径构成一个子图，记为 G_α. 而公式 ϕ 由 $\overline{FR\upsilon}(G_R,\alpha)$ 中所有结点对应的短句的合取构成，而后求解器以 ϕ 与 G_α 作为输入，以深度优先的搜索方式查找 G_α 中未被赋值的变元，并令其为 false. 如果在赋值过程中，产生了真值为 true 的短句或冲突短句，那么就将变元的赋值回溯. 求解器在 G_α 的每条

路径上不断地搜索，直到下面的两种情况之一出现：

① 产生一个部分赋值模型 M，M 使得 G_α 中的某条路径上的短句都为 false，那么求解器继续搜索未赋值的变元，直至得到 $\overline{FR_v}(G,\alpha)$ 的可满足赋值模型；

② 遍历完整个 G_α，表明 G_α 中不存在其上所有短句都为 false 的路径，表明公式 ϕ 是不可满足的；此时 SMT 求解器继续搜索，直到产生 ϕ 的消解否证.

6　实验结果与分析

为了验证算法的有效性，采用业界标准的 SMT Competition 2007 测试集作为基准测试向量；SMT Competition 测试集是一年一度的国际计算机辅助验证 CAV 会议中针对 SMT 求解器进行性能评估与竞赛的测试标准，其所有公式都来源于实际的工业应用. 基于测试集中的公式将基于冲突分析与否证蕴含的算法 CARI-MUSE 与深度优先搜索算法 DFS-MUSE 进行了对比与分析，二者的输入都是 SMT-LIB 格式的公式，算法的运行环境是 2.5GHz 的 Athlon＊2 CPU，内存 2GB，操作系统为 Linux 的机器.

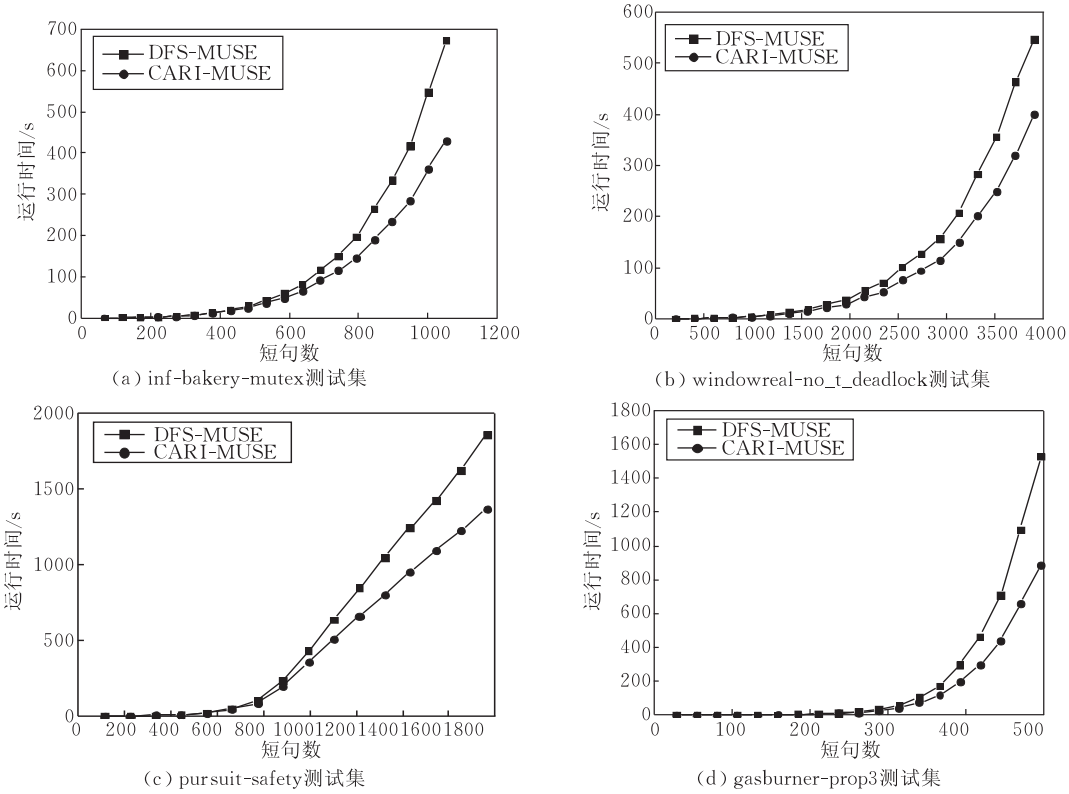
求解极小 SMT 不可满足子式的 CARI-MUSE 算法采用 C＋＋与 STL 实现. 算法中公式的可满足性检测过程基于一个开源的 SMT 求解器 ArgoLib①，它基于 DPLL(T)算法，目前支持的理论域包括线性算术(LRA/LIA)与差分逻辑(RDL/IDL). 两种算法的运行时限都设置为 1800s. 表 1 给出了 CARI-MUSE 算法与 DFS-MUSE 算法基于测试集中 15 个典型公式的实验结果. 表中第 2 列数据是每个公式所包含的变元数；第 3 列表示每个公式所包含的短句数；第 4、5 列给出了求解 SMT 不可满足子式算法(Lemma Lifting＋AMUSE)[16] 的结果短句数与运行时间，该算法不保证不可满足子式的极小性. 第 6、7 列分别是深度优先搜索算法(DFS-MUSE)的运行时间与所提取极小不可满足子式包含的短句数. 第 8、9 列分别是基本冲突分析与否证蕴含算法(不包含蕴含图剪枝的优化过程)与完整 CARI-MUSE 算法的执行时间对比；最后一列是基本与完整 CARI-MUSE 算法得到的极小不可满足子式的长度. 表中所有公式的运行时间都是以 s 为单位.

表 1　CARI 与 DFS 算法在 SMT 测试集上的实验结果

标准测试程序	变量数	公式短句数	求解 unsat core[16]		DFS-MUSE		CARI-MUSE		
			时间/s	结果短句数	时间/s	结果短句数	时间 1/s	时间 2/s	结果短句数
bad_echos_ascend. base	58	259	1.82	115	5.18	11	4.89	4.62	11
sc_init_frame_gap. base	58	265	1.78	119	5.11	13	4.84	4.58	13
good_frame_update. induction	89	439	8.58	208	29.00	161	25.67	23.21	161
good_frame_update. base	89	467	20.56	402	72.81	311	60.03	54.65	311
windowreal-safe2-2	37	404	0.44	279	0.73	188	0.68	0.67	188
windowreal-safe-2	37	411	0.45	286	0.75	195	0.70	0.68	195
lpsat-goal-1	83	1345	0.80	192	1.67	17	1.58	1.50	17
lpsat-goal-2	142	2650	5.52	1693	12.30	1283	11.60	10.72	1283
lpsat-goal-3	201	3955	18.35	2968	43.47	2548	38.22	33.09	2548
windowreal-no_t_deadlock-15	219	2933	45.20	2198	176.96	1351	148.64	130.96	1351
windowreal-no_t_deadlock-16	233	3128	52.36	2342	208.13	1441	173.09	151.90	1441
windowreal-no_t_deadlock-17	247	3323	73.11	2485	293.38	1531	245.61	211.33	1531
windowreal-no_t_deadlock-18	261	3519	85.02	2627	347.24	1622	288.68	246.48	1622
windowreal-no_t_deadlock-19	275	3714	112.57	2764	463.78	1712	381.24	324.64	1712
windowreal-no_t_deadlock-20	289	3909	715.20	2897	547.44	1802	440.80	372.95	1802

图 4 给出了 CARI-MUSE 算法与 DFS-MUSE 算法基于 SMT Competition 2007 测试集中的 4 组典型公式的实验结果，其中横坐标轴表示原始公式所包含的短句数，而纵坐标轴表示算法的运行时间，单位为 s，图中的两条曲线分别表示 CARI-MUSE 算法与 DFS-MUSE 算法随公式的短句数递增时运行时间的变化趋势. 图 4(a)是基于 inf-bakery-mutex 测试集，其公式所包含的短句数范围是 65～1053；图 4(b)对应于 windowreal-no_t_deadlock 测试集，其公式包含的短句数范围是 203～3908；图 4(c)是基于 pursuit-safety 测试集，其公式短句数的范围是 113～1763；而图 4(d)的测试集是 gasburner-prop3，其短句数范围是 28～522.

① Maric F, Janicic P. ArgoLib user manual. http://www.matf.bg.ac.yu/～janicic/argo/ArgoLib/Doc/argo_man.pdf, 2007



(a) inf-bakery-mutex测试集 (b) windowreal-no_t_deadlock测试集

(c) pursuit-safety测试集 (d) gasburner-prop3测试集

图 4 CARI-MUSE 与 DFS-MUSE 基于 4 组 SMT 测试集的对比

图 5 给出了 CARI-MUSE 算法与 DFS-MUSE 算法基于 SMT Competition 2007 测试集中 200 个公式的实验结果,运行时限为 1800s,其中横坐标轴表示冲突分析与否证蕴含算法CARI-MUSE的运行时间,纵坐标轴表示深度优先搜索算法的 DFS-MUSE 的运行时间,二者都采用了对数(lg)坐标,时间单位为 s.

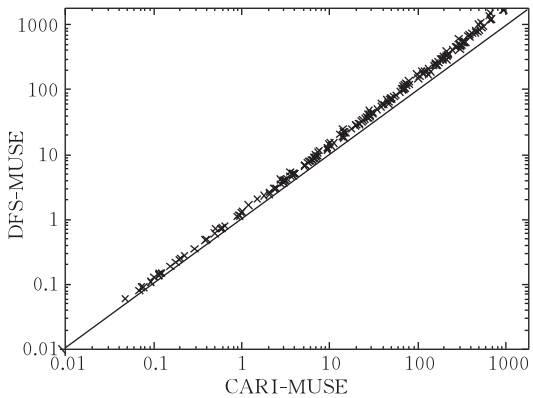


图 5 CARI-MUSE 与 DFS-MUSE 基于 SMT Competition 2007 测试集的对比

从表 1、图 4 与图 5 的实验结果可以看出,CARI-MUSE 算法能够有效地求解 SMT 公式的极小不可满足子式. 从表 1 以及图 4(a)～(d)中能够直观地看出,CARI-MUSE 算法明显优于深度优先搜索算法 DFS-MUSE. 根据表 1 的结果,完整 CARI-MUSE 算法的运行时间小于去除蕴含图剪枝过程的基本算法,因此蕴含图剪枝技术能够加速算法的可满足性判定过程. 另外从表 1 可以看出,CARI-MUSE 算法得到的极小不可满足子式的短句数显著小于 Lemma Lifting+AMUSE 算法得到的不可满足子式,但运行时间要大于后者;而在实际应用中,极小不可满足子式能够给出关于公式不可满足更加精确的解释,迅速地定位错误,应用也更加广泛.

在图 5 中,所有测试集中的公式都位于对角线的上方,但为了数据在图中分布的均匀性,采用了对数坐标系,因此两个算法性能的比值接近于对角线,但是同样能够表明 CARI-MUSE 算法明显优于 DFS-MUSE 算法. 其主要原因是 CARI-MUSE 算法采用了一种更为有效的机制来移除公式中的冗余短句,它每次从否证蕴含图的原始短句集合中删除的短句数为 $n \geq 1$,DFS-MUSE 算法每次循环都只从原始公式中删除一个短句. 另外,根据表 1 的实验结果,对于测试集中的 SMT 公式,极小不可满足子式所包含的短句数远远小于原始公式的短句数,通常占公式总短句数的 1%～50%左右,所以极小不可满足子式能够给出公式不可满足原因更加精确的

解释，更加迅速地诊断与定位错误.

根据图 4(a)～(d)的实验结果分析，可以发现当公式所包含的短句数与变元数较少时，尽管 CARI-MUSE 算法优于 DFS-MUSE 算法，但是两种算法的性能差距并不十分显著；而后随着公式复杂度的增大，即其短句数与变元数逐渐增加时，CARI-MUSE 算法与 DFS-MUSE 算法的运行时间差距越来越大. 表 1 的实验结果也在一定程度上反映了这个规律. 而在图 5 中，随着运行时间的推移，也就是说公式的复杂度越来越大，CARI-MUSE 与 DFS-MUSE 算法的性能比值的总体趋势是越来越偏离对角线，表明 CARI-MUSE 与 DFS-MUSE 算法的性能差距越来越显著. 综上所述，可以给出一个结论：CARI-MUSE 算法明显优于 DFS-MUSE 算法；并且随着公式包含的变元数与短句数不断增加时，CARI-MUSE 算法的性能优势更加显著. 这主要是由于当公式较小时，当把短句蕴含图化简为否证蕴含图时，通常每次只能删除一个原始短句，CARI-MUSE 算法的优势并没有体现出来；但是当公式的复杂度增加时，当短句蕴含图进行约简时，多数情况下会删除多个原始短句，那么就会提高构造极小不可满足子式的效率，而 DFS-MUSE 算法每次循环通常都只从原始公式中删除一个短句，因此这时 CARI-MUSE 算法的性能优势就更加明显.

7　结束语

本文针对极小 SMT 不可满足子式的求解问题，引入了否证蕴含图及其正向与逆向可达结点集的概念，并证明了它们与不可满足子式的关系；提出了基于冲突分析与否证蕴含的极小 SMT 不可满足子式的求解算法. 该算法首先为输入公式构造否证蕴含图，而后选择并移除一个原始短句，通过检测该短句的正向不可达结点集的可满足性，从而确定它是否属于极小不可满足子式. 算法融合了蕴含图剪枝技术，从而显著地减小了算法的搜索空间. 实验结果表明，CARI-MUSE 算法优于深度优先搜索算法 DFS-MUSE，并且随着公式逐渐增大，CARI-MUSE 算法的性能优势更加显著，并通过实验证明了蕴含图剪枝技术的有效性.

致　谢　感谢 ArgoLib 的作者 Filip Maric 提供了求解器的所有源代码，并针对本文的算法提出了许多建议！

参　考　文　献

[1] Ranise S, Deharbe D. Light-weight theorem proving for debugging and verifying units of Code//Proceedings of the International Conference on Software Engineering and Formal Methods. Brisbane, Australia, 2003

[2] Audemard G, Cimatti A, Korni_lowicz A, Sebastiani R. Bounded model checking for timed systems//Proceedings of the 22nd International Conference on Formal Techniques for Networked and Distributed Systems. Houston, Texas, USA, 2002: 243-259

[3] Bozzano M, Bruttomesso R, Cimatti A, Franzen A, Hanna Z, Khasidashvili Z, Palti A, Sebastiani R. Encoding RTL constructs for MathSAT: A preliminary report//Proceedings of the 3rd Workshop on Pragmatics of Decision Procedures in Automated Reasoning, ENTCS 144. Elsevier, 2006: 3-14

[4] Stuckey P J, Sulzmann M, Wazny J. Improving type error diagnosis//Proceedings of the 2004 ACM SIGPLAN Workshop on Haskell. Snowbird, Uppsala, 2004: 80-91

[5] Jain H, Kroening D. Word level predicate abstraction and refinement for verifying RTL verilog//Proceedings of the 42nd Design Automation Conference. Anaheim, San Diego, 2005: 445-450

[6] Simmonds J et al. Exploiting resolution proofs to speed up LTL vacuity detection for BMC//Proceedings of the 7th International Conference on Formal Methods in Computer Aided Design. Austin, 2007: 3-12

[7] Zhang L, Malik S. Extracting small unsatisfiable cores from unsatisfiable Boolean formula//Giunchiglia E, Tacchella A eds. Proceedings of the 6th International Conference on Theory and Applications of Satisfiability Testing. LNCS 2919. Berlin: Springer-Verlag, 2003

[8] Oh Y, Mneimneh M N, Andraus Z S, Sakallah K A, Markov I L. AMUSE: A minimally-unsatisfiable subformula extractor//Proceedings of the 41st Design Automation Conference. San Diego, CA, USA, 2004: 518-523

[9] Li X W, Li G H, Shao M. Formal verification techniques based on Boolean satisfiability problem. Journal of Computer Science and Technology, 2005, 20(1): 38-47

[10] Gershman R, Koifman M, Strichman O. Deriving small unsatisfiable cores with dominator//Ball T, Jones R B eds. Proceedings of the 18th International Conference on Computer Aided Verification. LNCS 4144. Berlin: Springer-Verlag, 2006: 109-122

[11] Dershowitz N, Hanna Z, Nadel A. A scalable algorithm for minimal unsatisfiable core extraction//Biere A, Gomes C P eds. Proceedings of the 9th International Conference on Theory and Applications of Satisfiability Testing. LNCS 4121. Berlin: Springer-Verlag, 2006: 36-41

[12] Zhang J M, Li S K, Shen S Y. Extracting minimum unsatisfiable cores with a greedy genetic algorithm//Sattar A,

Kang B H eds. Proceedings of the 19th Australian Joint Conference on Artificial Intelligence. LNAI 4304. Berlin: Springer-Verlag, 2006: 847-856

[13] Gregoire E, Mazuer B, Piette C. Local-search extraction of MUSes. Constraints, 2007, 12(3): 325-344

[14] Liffiton M H, Sakallah K A. Algorithms for computing minimal unsatisfiable subsets of constraints. Journal of Automated Reeasoning, 2008, 40(1): 1-33

[15] Maaren H, Wieringa S. Finding guaranteed MUSes fast// Buning H K, Zhao X eds. Proceedings of the 11th International Conference on Theory and Applications of Satisfiability Testing. LNCS 4996. Berlin: Springer-Verlag, 2008: 291-304

[16] Cimatti A, Griggio A, Sebastiani R. A simple and flexible way of computing small unsatisfiable cores in SAT modulo theories//Marques-Silva J, Sakallah K eds. Proceedings of the 10th International Conference on Theory and Applications of Satisfiability Testing. LNCS 4501. Berlin: Springer-Verlag, 2007: 334-339

[17] Nieuwenhuis R, Oliveras A, Tinelli C. Solving SAT and SAT modulo theories: From an abstract Davis-Putnam-Logemann-Loveland procedure to DPLL(T). Journal of the ACM, 2006, 53(6): 937-977

[18] Gallier J H. Logic for computer science: Foundations of automatic theorem proving. Harper & Row Publishers Inc., revised in 2003

ZHANG Jian-Min, born in 1979, Ph. D. candidate. His research interests include automatic errors diagnosis and localization, equivalence checking, and SAT-based formal verification.

SHEN Sheng-Yu, born in 1975, Ph. D., associate professor. His research interests include circuit automatic synthesis, formal verification, and electronic design automation.

LI Si-Kun, born in 1941, professor, Ph. D. supervisor. His research interests include electronic CAD, chip design methodologies, and virtual reality.

Background

This work is primarily supported by the National Natural Science Foundation of China under grant No. 60603088 with the title "Methods of automatic program repair based on counterexample explanation", which involves in the research works about automatic faults diagnosis, localization and correction for programs. The authors have made researches on algorithms of errors localization, such as counterexample minimization and explanation, unsatisfiable subformulae extraction and so on.

Although locating and correcting faults is difficult and time-consuming, it becomes one of the hot topics in SAT-based and SMT-based formal verification. Extracting unsatisfiable subformulae from propositional and first-order formulae can provide a succinct explanation of infeasibility, and determine the underlying reasons for the failure. There have been many different contributions to research on finding unsatisfiable Boolean subformulae, mostly based on SAT solvers. However little attention has been concentrated on extraction of unsatisfiable subformulae in Satisfiability Modulo Theories. This work belongs to part of error diagnosis and localization, and focuses on proposing a new algorithm to deriving minimal unsatisfiable subformulae in SMT based on conflict analysis and refutation implication.