DOI:10.3969/j.issn.1673-4785.201212058
网络出版地址:http://www.cnki.net/kcms/detail/23.1538.TP.20131012.1813.002.html

不可满足子式研究

殷明浩,李欣
(东北师范大学 计算机科学与信息技术学院,吉林 长春 130117)

摘 要:为了广泛有效地将不可满足子式应用于知识验证、产品规划、硬件和软件的设计与验证等领域,对不可满足子式进行了相关研究.对当前不可满足子式的主要相关算法进行了概述评论、分类归纳,并从计算复杂性角度介绍了其子类、参数复杂性以及 QBF 中的极小不可满足子式.总结了近 10 年来不可满足子式的理论与算法,讨论了不可满足子式的未来研究发展方向.研究有利于进一步发现不可满足的根本原因,从而进行有针对性地改进,并对相关人员的研究提供帮助.

关键词:可满足性问题;不可满足子式;可满足模理论;局部搜索
中图分类号:TP311 **文献标志码**:A **文章编号**:1673-4785(2013)06-0497-08

中文引用格式:殷明浩,李欣. 不可满足子式研究[J]. 智能系统学报, 2013, 8(6): 497-504.
英文引用格式: YIN Minghao, LI Xin. Research on an unsatisfiable subformula[J]. CAAI Transactions on Intelligent Systems, 2013, 8(6): 497-504.

Research on an unsatisfiable subformula

YIN Minghao, LI Xin
(School of Computer Science and Information Technology, Northeast Normal University, Changchun 130117, China)

Abstract: In recent years, an increasing number of researchers have started to focus their attention on unsatisfiable subformulas, especially in regards to the extremely small and the minimal unsatisfiable subformulas. The unsatisfiable subformula (US) has a wide range of practical applications, including knowledge validation, product programs, and design and verification of hardware and software. An unsatisfiable subformula may be very helpful in diagnosing the causes of unfeasibility in large systems. In the past 10 years, research on an unsatisfiable subformula has been developed quickly. In this paper, the authors discuss the algorithm in relation to an unsatisfiable subformula, introduce the subcategory of an unsatisfiable subformula from the viewpoint of calculation complexity, parameter complexity and the research on an unsatisfiable subformula in QBF. Finally, the authors discuss the future direction of research on an unsatisfiable subformula.

Keywords: satisfiability problem; unsatisfiable subformula; satisfiability module theory; local search

可满足性问题(satisfiability problem, SAT)是当前人们研究得最多的约束可满足性问题,也是当今计算机科学和人工智能研究的核心问题.在实际应用领域中的很多问题,例如等价性检查、大规模集成电路的自动布线、自动测试向量生成、硬件与软件的属性验证等都可转化为约束可满足问题,即能够归约为一阶逻辑公式或命题逻辑.然而,仅仅判断命题逻辑的可满足性是无法满足实际要求的.当公式不可满足时,为了查找不可满足的原因,往往要删除无关子句进而提取不可满足子式.目前,研究人员对不可满足子式的算法研究和相应计算的复杂性表现出了极大兴趣.寻找最小不可满足子式的决策问题是 $\sum_2$-complete 问题[1],寻找一个极小不可满足子式是 D^p-complete 问题.目前有很多算法是关于极小[2-11]和最小不可满足子式[12-14]的提取,这些提取不可满足子式的算法主要应用于形式验证领域.极小不可满足子式可以更好地解释不可满足的根源所

收稿日期:2012-12-27. 网络出版日期:2013-10-12.
基金项目:国家自然科学基金资助项目(61070084,60803012).
通信作者:殷明浩. E-mail: ymh@nenu.edu.cn.

在,最小不可满足子式可以精确地定位错误原因,但是其求解难度极大.随着 SMT(satisfiability modulo theories)技术的不断发展,它开始逐渐取代 SAT.相对于 SAT,SMT 具有更好的抽象能力,更接近于高层设计,极具发展潜力.它使用字级建模,不必像 SAT 那样将问题转化为位级处理,从而减少了空间和时间的开销.SMT 可以看作是 SAT 问题的一个扩展,它极大地补充了 SAT 的不足之处.求解其极小不可满足子式算法也极具实际意义.SMT 技术是近几年发展起来的,目前提取 SMT 中极小不可满足子式的算法比较少. Cimatti[15-16] 提出了一个新颖的方式 lemma-Lift 来提取 SMT 极小不可满足子式;张建民[17]提出了基于 DPLL(T)的深度优先搜索的极小 SMT 不可满足子式求解算法 DFS-MUSE 和宽度优先搜索的极小 SMT 不可满足子式的求解算法 BFS-MUSE.2010 年,他又提出了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 CARI-MUSE[18],该算法的效果明显好于 DFS-MUSE 算法,性能随着时间复杂度的增加更加明显.

研究不可满足子式的复杂性对不可满足子式求解算法也极具重大意义.H.K.Büning 和赵希顺[19]对不可满足的子类进行了研究,介绍了各子类的特性,并在文献中对子类的复杂性进行了证明[20]. H. Fleischne[21]、S. Szeide[22]将固定参数的概念与极小不可满足子式中固定差值概念相关联,证明了固定参数的极小不可满足子式的计算复杂性.带量词的布尔公式 QBF 作 SAT 公式的自然扩展具有紧凑的空间结构以及更强大、更直观的表达能力,对 QBF 中极小不可满足子式的研究有利于探索新的 QBF 算法.赵希顺[23]证明了 QBF 中,对 QCNF 公式固定差值为 1 的极小不可满足子式可以在多项式时间内有解.

1 相关定义

在计算机科学领域,不可满足子式的研究最初源于 SAT 问题,首先介绍本文涉及的相关概念.

定义 1 命题可满足性问题.

给定一个 CNF 表达式 F 和一个变量集合$\{V=x_1,x_2,x_3\}$.如果变量集合 V 中的一组赋值使得表达式 F 的值为真,那么称表达式 F 是可满足的;否则,表达式 F 为不可满足.

定义 2 不可满足子式/不可满足核(US/UC).

对于一个公式 φ,ϕ 是 φ 的一个不可满足子式,当且仅当 φ 是不可满足的,且 $\phi \subseteq \varphi$.

显然,对于同一个问题实例,可能存在不同的不可满足子式,每个不可满足子式的子句数也可能不同.其中一些不可满足子式可能是其他不可满足子式的子集.在最坏情况下,不可满足子式就是原始公式本身.

定义 3 极小不可满足子式.

对于一个不可满足的公式 φ 的一个不可满足子式 ϕ,那么 φ 是极小不可满足子式,当且仅当 $\forall\ \phi \in \varphi$,使得 ϕ 是可满足的.

定义 4 最小不可满足子式.

给出一个不可满足的公式 φ 以及 φ 的所有不可满足子式构成的集合 $\{\phi_1,\phi_2,\cdots,\phi_n\}$,如果 $\phi_k \in \{\phi_1,\phi_2,\cdots,\phi_n\}$ 是最小不可满子式,当且仅当 $\forall \phi_i \in \{\phi_1,\phi_2,\cdots,\phi_n\}$,$1 \leqslant i \leqslant n$,使得 $|\phi_k| \leqslant |\phi_i|$.

定义 5 量化布尔公式-QBF 问题.

一个 QCNF 公式 $\varphi = Q_1x_1Q_2x_2\cdots Q_nx_n\phi$,其中 $Q_i \in \{\exists,\forall\}$,$\phi$ 是一个 CNF 公式,用矩阵形式表达 φ. $x_1, x_2,\cdots, x_n$ 分别被 $Q_1,Q_2,\cdots,Q_n$ 量化.有时,可以简写为 $\varphi = Q\phi$.

定义 6 QBF 极小不可满足子式(MF).

当且仅当 φ 是不可满足的,并且从 φ 中移出任何一个子句将导致公式变为可满足,一个 QBF 公式 φ 被称为极小不满足,不可满足子式的子式可以表示为 MF(minimal false formulas).

定义 7 可满足性模理论(SMT).

SMT 是 SAT 的一种扩展,其命题不仅包括布尔公式,还包括其他特定理论中的一阶逻辑约束,如整数与实数算术类型、位向量、递归及数据类型等.

定义 8 SMT 不可满足子式.

给出一个不可满足的无量词一阶逻辑公式 φ,当且仅当 ϕ 是不可满足的,并且 $\phi \subseteq \varphi$,ϕ 是公式 φ 的一个 SMT 不可满足子式.

定义 9 极小 SMT 不可满足子式.

给出不可满足的无量词一阶逻辑公式 φ,及其一个不可满足子式 ϕ,那么 ϕ 是极小 SMT 不可满足子式,当且仅当 $\forall\ \phi \in \varphi$,使得 ϕ 是可满足的.

定义 10 约束可满足问题(CSP).

由一个变量集合和一个约束集合组成.问题的一个状态是由对一些或全部变量的一个赋值定义的完全赋值;每个变量都参与的赋值.问题的解是满足所有约束的完全赋值,或更进一步,使目标函数最大化.

定义 11 极小 CSP 不可满足子式.

一个极小 CSP 不可满足子式,其约束的有关子集是不可满足的,并且其真子集是可满足的.

2　不可满足子式求解算法

按照问题的解决方式,求解布尔不可满足子式的方法可划分为基于 DPLL 搜索[2,4-6,12-14]与局部搜索[7-11].通常来说 ,基于 DPLL 搜索方法能够找到问题的精确解,但其面临的主要挑战是计算时间随问题规模的增大呈指数级增长.而局部启发式搜索尽管有时无法给出精确解,但其运算速度快,求解效率高.

根据不可满足子式的大小可以分为 2 类:1)极小不可满足子式的提取算法[2-11];2)最小不可满足子式的提取算法[12-14].相对而言,最小不可满足子式的求解难度更大,算法的复杂度也更高.

2.1　极小不可满足子式求解

当前主流的不可满足子式提取算法关于极小不可满足子式的提取.形式验证与人工智能规划问题均可转为可满足性问题,一组可满足的赋值代表了一种可行方案;反之,说明原问题存在错误,寻找不可满足子式可以精确地对问题进行定位.

2.1.1　基于冲突树的搜索算法

Han 和 Lee[24]提出了一个获得所有极小不可满足子式的算法.该方法的思想是有序遍历集合 S 的所有子集.它由 2 部分组成:1)用 CS-tree(冲突树)的不同结点表示集合 S 的不同子集,并且没有哪 2 个结点表示相同的子集.CS-tree 定义了子集间的遍历顺序,因此保证了同一子集不被重复搜索.为了实现这个目标,一个结点由集合 D 和集合 P 2 部分组成,表示为 $D \cup P$.集合 D 的元素一定要出现在所有后续子集中,而集合 P 的元素不需要.如果一个结点有 $|P|$ 个孩子,它由逐个消减集合 P 约束,并加入到集合 D 中获得.函数 all subsets (D,P,ϕ) 遍历 CS-tree 的所有结点,返回 $D \cup X(X \subset P)$.当迭代访问一个根结点时,每次迭代以深度优先方式遍历 CS-tree,并且每次内部调用 all subsets( )函数访问这个孩子结点的后代.2)利用 CS-tree 探测所有极小不可满足子式.算法以深度优先方式遍历树,测试每个结点的标志 $D \cup P$,判断其可满足性.如果结点可满足,那么就不必访问它的孩子节点.否则,继续寻找孩子节点来寻找不可满足子式.如果遍历所有孩子结点没有发现子集 A 使得 $A \subset D$.那么集合 D 就是极小不可满足集合.文献[2]在以上基本算法中加入了预处理,利用约束独立性、利用始终满足的子句、蕴涵、廉价求解器、增量式求解器、无视约束等策略.其中,使用增量式求解器极大地改进了算法,它减少了对算法规则的依赖性,能够快速地发现可满足子集.

2.1.2　OMUS 与 ASMUS 算法

文献[14]中,É. Grégoire 提出了基于启发式局部搜索算法 OMUS.OMUS 算法是通过计算关键子句分值来获得一个极小不可满足子式的启发式算法.算法主要分为 2 部分:1)在局部算法中根据子句布尔值为假的频度计算子句的分值,移除确定不可能出现在不可满足子式中的子句.具体来说,对于一个不可满足子式,如果在局部搜索时终未能找到它的一个赋值模型,那么将移除子式中分值最低的子句.获得的子公式将被记录在堆栈中.2)检查最后一个未能赋值的子式是否可满足,如果子公式是不可满足的,那么它就是不可满足子式的近似解.否则,重复检查栈顶子句超集的不可满足性,直到一个集合被证明是不可满足的.该过程将获得一个近似的极小不可满足子式.然后,调用一个精细调节(fine tunes)过程,OMUS 算法最终获得一个不可满足子式.

É. Grégoire 在文献[8]中提出了 ASMUS 算法.ASMUS 以 OMUS 为基础,可以近似求解所有不可满足子式的集合.MAX-SAT 可以提供极小的不可满足子式数量.MAX-SAT 解中的剩余不可满足子句一定尽可能多地属于那些不可满足子式的交集.那么对于每个子句,记录最小的子句数.在探测到一个不可满足子式后,把不可满足子式中分值最高的子句删除掉.很显然,ASMUS 是不完全的.

2.1.3　基于消解反驳的局部搜索算法

局部搜索是一种求解优化问题的算法,算法从解空间的一个点出发,每一步从当前候选解移动到一个邻居候选解,去寻找较好的候选解,候选解的质量是由一个评估函数来界定的.局部搜索算法在处理规模较大的问题时,其收敛速度快,求解效率高.上文提到的 OMUS 与 ASMUS 实质上是启发式局部搜索与计数的组合.消解是判定 SAT 问题的基本方法之一.一个公式经过一系列的消解步骤最终产生空子句,则公式是不可满足的.文献[9-11]中采用局部搜索方法作为解空间的搜索策略.算法在搜索过程中建立证明公式不可满足性的消解序列,并从其中提取不可满足子式.算法描述如下:算法启发式或随机地选择 2 个子句进行消解,最终得到一个消解序列或达到最大迭代次数.搜索过程融合了一些基于启发式的 SAT 推理技术,包括二元子句消解、等价约简和单元子句传播.算法首先寻找公式中是否存在单元子句,如果存在,则在公式上进行单元子句传播,并进一步判断能否产生空子句;如果存在二元子句,则执行二元子句消解与等价性约简,启发式地选择消解子句.若 2 类子句都不存在,则随机选取 2

个子句进行消解.最后采用一个递归函数从空子句开始回溯,逐步遍历整个消解序列,提取布尔不可满足子式.

2.1.4 基于最大可满足性的求解算法

目前,有很多研究人员利用极大可满足性和极小不可满足性之间的二元关系提取极小不可满足子式.Bailey 和 Stuckey 在文献[25]中已经阐述了这种关系.基于最大可满足性的 CAMUS[4],Digger[13]能够提取极小不可满足子式;贪心遗传算法和蚁群算法[14]能够提取最小不可满足子式.文献[26]中提出了一种在 CSP 中提取极小不可满足子式的混合算法.该算法也是基于最大可满足性求解极小不可满足子式.求解 CSP 极小不可满足子式的算法一般分为直接算法和间接算法.间接算法利用极小不可满足子式和极小修正集(MCS)之间的关系.算法主要是寻找一个完全的极小修正集,然后从中提取极小不可满足子式.另一方面,直接算法通过产生子集和测试其可满足性直接地在 CSPs 约束的子集空间搜索不可满足子式.混合算法融合了直接和间接方法.混合算法像间接算法那样不再计算全部的极小修正集,而是寻找极小修正集中较小的子集.这就是所谓的"主干"集合,即极小修正子集发现的不可消减的碰集.每个主干集表示不可满足子式的一个簇.通过直接方法搜索主干的超级空间,这些主干集随后逐个被扩展为完全的不可满足子式.实验表明,基于最大可满足性的混合算法比单纯直接算法或间接算法效率高很多.混合算法通过在每步中搜索一个较小的空间提高效率.

2.1.5 基于关联赋值的算法

文献[27]中算法提出了关联赋值的概念,将赋值与子句联系起来.关联赋值的概念来自于布劳威尔不动点近似算法在可满足性问题方面的应用.它将子句视为实体,也就是一个有序的赋值集合,并且从可能的赋值中选择一个赋值与其关联.这些赋值形成一个完整的序列,在序列中优先选择所有可满足赋值,再选择所有不可满足赋值.当达到帕累托最优后,有可能找到这些子句的一个子集来表示不可满足性.在帕累托最优中,所有子句已经选择了一个惟一的不满足自己的赋值.帕累托最优证明了子句的选择权与不可满足公式之间的冲突.当子句证明了公式的不可满足性后,利用文献[27]中的理论去实现算法,寻找一个不可满足子式.然而这种算法的效率并不高,文献[12]中提出了一个简单有效的极小不可满足子式抽取算法 MiniUnsat.MiniUnsat 中包含了一个高效率 SAT 求解器 MiniSat 2.0.为了进一步提高效率,在 MiniSat 的分支选择中加入了启发式策略.当每个变量决策时,MiniSat 的分支默认选择赋值为负值的一支.在算法的第 1 轮中,并不发生任何变化.第 1 轮之后,每个尚未删除的子句都在前 1 轮中有了一个关联赋值.除了新的关键子句移动到这个赋值的前面,随后的子句仍将保持同样的顺序.程序因此确定了变量的分支方向,这将引导求解器按着以前关联赋值的方向寻找新的赋值.

2.2 最小不可满足子式求解

不可满足子式反应了问题的存在.最小不可满足子式比极小不可满足子式更能精确地定位错误.研究最小不可满足算法,也是研究人员的最终目的.目前关于最小不可满足子式的提取研究还相对较少.

2.2.1 基于辅助变量的算法

算法 AMUSE[3]和文献[12]中算法通过引入一组辅助变量求解不可满足子式.AMUSE 是对一个基于 DPLL 构架的 SAT 求解器的扩展.它不再寻找一个可满足的真值赋值而是寻找其中蕴含的不可满足子式.为此引入了一组辅助变量,通过自底向上的方式搜索不可满足树的空间:从一个空子句开始,逐步添加子句,直到获得一个不可满足子式.为了改善搜索效率,其中使用了冲突子句学习优化策略.但是 AMUSE 算法无法保证获得极小不可满足子式,尤其随着问题规模的扩大,效率会明显下降.文献[6]中提出了一个获得最小不可满足子式的模型:$\omega_i' = \{\neg\ s_i\} \cup \omega_i$,其中新子句 ω_i' 是由子句 ω_i 与选择变量 s_i 构成.选择变量 s_i 决定是否选择子句 ω_i. 对每个 S 中变量赋值,产生的子式可能是满足的,也可能是不满足的.对于不可满足的子式,S 中有多少变量赋值为 1,就意味着有多少子句包含在不可满足子式里.那么最小不可满足子式就是 S 集合中赋值为 1 的、数量最少的子公式.通常可以使用高效的求解器来实现这个模型.公式中变量被组织成 2 个互斥的集合:变量集合 S 和变量集合 X.首先决策变量 S,然后决策变量 X.因此每次变量 S 赋值,都定义了一个潜在的子式.如果对于一个赋值,所有的子句都满足,那么回溯到最近未决策的变量 S.否则,每次回溯要先改变 X 赋值,再改变 S 赋值,直到没有变量 X 的赋值可以改变,则 S 中赋值为 1 最少的,就是最小不可满足子式.

2.2.2 分支限界算法

文献中[13]中 CAMUS-min 和 Digger 都是基于分支限界的算法.CAMUS-min 算法是 CAMUS 的改进,它比 CAMUS 多了分支限界过程.CAMUS 利用极大可满足性和极小不可满足性之间的二元关系提取

极小不可满足子式,一般分为 2 步:1)获得公式中有所极大可满足子式的补集;2)通过在递归树中寻找补集的极小碰集来提取极小不可满足子式.CAMUS-min 与 CAMUS 不同之处在第 2 步,不直接寻找极小碰集,而在递归中增加了分支限界功能来减小树的搜空间,产生最小的碰集.CAMUS-min 中,一个贪婪的启发式策略 MIS-quick 为算法提供下界,如果递归树所包含的不可满足子式没有当前不可满足子式小,CAMUS-min 删除递归树中的这些分支,因此运行时间大幅减少,并且最后的不可满足子式一定是最小不可满足子式.Digger 使用一个递归的分支限界树来获得最小不可满足子式.树中的每个结点都是原始公式的一个子集,并且上下界由子集中的最小不可满足子式限定.因此算法在根结点处启动递归调用.Digger 不同于标准的分支限界树,不光遍历叶子结点,也遍历包括根结点的所有结点.对于每个结点,算法利用不相交的极小修正集,启发式选择子句的一个样本集,这不同于 CAMUS-min 中寻找全部极小修正集.实质上,Digger 利用不相交的极小修正集关系将一个问题分解为更容易处理的子公式,同时在搜索最小不可满足子式过程中提供了一个非常有效的下界.

2.2.3　贪心遗传算法

贪心遗传算法[14]可以视为贪心算法与遗传算法的混合算法.贪心算法是指在对问题求解时,总是做出在当前看来是最好的选择.也就是说,不从整体最优上加以考虑,它所做出的仅是在某种意义上的局部最优解.遗传算法(genetic algorithm)是一类借鉴生物界的进化规律演化而来的随机化搜索方法.它由美国的 J.Holland 教授于 1975 年首先提出,其主要特点是直接对结构对象进行操作,不存在求导和函数连续性的限定;具有内在的隐并行性和更好的全局寻优能力;采用概率化的寻优方法,能自动获取和指导优化的搜索空间,自适应地调整搜索方向,不需要确定规则.遗传算法是全局进化方法,可以避免进入局部最优,但是在初期缺乏充分有效启发信息的情况下局部收敛速度较慢,并且对初始种群十分依赖.所以将贪心算法与遗传算法有效地结合起来,保留了 2 种算法的优点,极大地避免了 2 种算法的不足.贪心遗传算法的基本策略是:首先采用贪心算法快速计算不可满足子式的近似最优解,为遗传算法构造一个较优的初始种群,缩减其搜索空间,而后利用遗传算法的全局性反复精化近似解,从而得到更小的不可满足子式.实验结果表明,在运算效率以及单位时间内删除的短句数方面,显著高于分支限界算法,而贪心遗传算法无论在运行时间还是不可满足子式的大小上都优于蚁群算法.

2.3　SMT 中不可满足子式求解算法

随着 SMT 技术的不断发展,它开始逐渐取代 SAT.SMT 具有更好的抽象能力,更接近于高层设计,极其具有发展潜力.它使用字级建模,不必像 SAT 中将问题转化为位级处理,从而减少了空间和时间的开销.SMT 可以看做是 SAT 问题的一个扩展,它极大地补充了 SAT 的不足之处.求解其极小不可满足子式算法也是非常具有实际意义的.SMT 技术是近几年发展起来的,目前提取 SMT 中极小不可满足子式的算法比较少.

2.3.1　Lemma-lifting 方式

Cimatti 等[15-16]提出了求解 SMT 不可满足子式的算法—Lemma-lifting 方式.Lemma-lifting 方式将 SMT 求解器与外部命题核提取器结合起来,通过一种直接的方式求解 SMT.SMT 求解器保存并返回在证明公式不可满足性的过程中获得的特定理论引理.外部命题核提取器随后调用原始 SMT 问题和特定理论引理的布尔抽象.算法是基于以下 2 个重要的观察:1)在搜索过程中,SMT 求解器发现的特定理论引理是在 T 理论中考虑的有效子句,因此它们不影响 T 中一个公式的可满足性;2)原始 SMT 公式与特定理论引理的合取是命题不可满足的.因此,当外部命题核提取器找到原始 SMT 公式与特定理论引理的合取的一个核时,应当删除其中的特定理论引理,获得一个精简的原始公式的子集.虽然在原理上很简单,但是该方式的概念很有趣,本质上讲,SMT 求解器动态地将理论信息适当地转化为布尔标准.而且,该方式还有以下优点:1)易于实现与升级.由于 SMT 求解器与布尔不可满足子式提取算法是相互独立的,因此能够将最新的求解器或算法进行组合,从而提高 SMT 不可满足子式的求解效率;2)寻找较小的不可满足子式非常有效;3)内核提取不需要复杂的 SMT 推理.但是,该算法有如下缺点:1)需要额外的存储空间用来保存 SMT 求解器产生的引理;2)无法保证最终求得的不可满足子式是极小不可满子式.

2.3.2　基于 DPLL 构架求解算法

DPLL 是一种基于回溯的算法,在成熟运用于 SAT 求解器后,基于 DPLL(T) 的 SMT 求解技术也得到飞速发展.DPLL(T) 由布尔引擎 DPLL(X) 与特定的理论求解器 T-slover 组成.文献[16-17]就是基于 DPLL(T) 构架求解不可满足子式.文献[17]提出了深度优先方式的极小不可满足子式算法(DFS-

MUSE)和深度优先方式的极小不可满足子式算法(BFS-MUSE).这 2 个算法从不可满足子式中删除任意一个子句,然后测试其余子句构成公式的可满足性,从而判断是否为极小不可满足子式;而不必测试不可满足子式所有的真子式,将复杂度从指数降为线性.算法中也融合了剪枝技术、冲突子句学习技术用于剔除不必要的冗余不可满足性测试,从而缩小了搜索空间,提高了搜索效率.算法中也融合了动态排序技术,降低了刚测试过子句的优先级,避免重复测试.DFS-MUSE 的一个重要特点是:当选择搜索树中不同的分支,会得到不同的极小 SMT 不可满足子式.BFS-MUSE 与 DFS-MUSE 的区别在于搜索策略不同.BFS-MUSE 是以宽度优先的方式对搜索树进行遍历,首先考虑同样大小的子式的可满足性,然后再考虑较小的子式.DFS-MUSE 是以深度优先的方式对搜索树进行遍历,先对当前子树内子式的可满足性测试,找出最小的子式,然后再测试其他子树的可满足性.

2.3.3 基于否正蕴涵图的求解算法

文献[18]中提出了基于否正蕴涵与冲突分析提取 SMT 中极小不可满足子式的算法 CARI-MUSE.否证蕴含图是指对一个不可满足的 SMT 子式及其子句蕴含图,从该蕴含图的每个子句出发,至少存在一条路径可以达到空子句.否证蕴含图与不可满足子式存在以下密切关系:对于不可满足公式 F 的一个消解反驳 R 和 R 对应的否证蕴含图 G,那么 G 中逆向可达结点集合与公式 F 的交集就是 F 的一个不可满足子式.CARI-MUSE 算法以上述结论为依据,利用 SMT 求解器记录在证明公式不可满足性的过程中产生的消解反驳序列,并将其转化为否证蕴含图,提取出不可满足子式.为了进一步获得极小不可满足子式,依次删除蕴含图中原始子句及其相关的冲突短句,调用 SMT 求解器来判断否证蕴涵图的剩余子公式(结点)的可满足性,以确定该子句是否为不可满足的原因;如果剩余子公式是可满足的,说明公式已经是极小不可满足子式;否则,从否证蕴含图中删除该子句及其冲突短句,形成更小的否证蕴含图;算法迭代的遍历否证蕴含图中所有原始子句,最后得到极小 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,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-MUSE;并且随着公式复杂度的增加,性能优势更加明显.

3 不可满足子式的复杂性

研究不可满足子式的复杂性,对不可满足子式算法的提出和改进有重大意义.

3.1 不可满足子式的子类

研究人员对不可满足子式的子类感兴趣有以下原因:1)有利于发展新的可满足性算法,例如,对极小不可满足子式结构的深入了解,可能会改进 DP 算法;2)可能有助于求解难度大的公式.例如利用差值特性产生新的多项式时间可解的公式类[28];3)不可满足子式的决策问题是 D^p -complete,但是不可满足子式的一些子类的复杂性可能相对容易.BÜNING H K 和赵希顺[19-20]将不可满足子式进行分类研究,并讨论了这些子类的复杂性.MARG-MU 类是指,如果一个极小不可满足子式是临界类,那么移除任意一个文字事件,将导致公式成为非极小不可满足子式.也就是说 MARG-MU 类中没有多余文字.MAX-MU 类由最大的极小不可满足子式构成,即公式中再加入一个文字将不再是极小不可满足子式.Unique-SAT 指公式在移除任意一个子句后,只有一个可满足的真值分配.带有这种性质的极小不可满足子式,就是 Unique-MU 类.Almost-Unique-MU 类,是 Unique-MU 较小的改进,即至多有一个子句 $f \in F$, 使得 $F - \{f\}$ 有多于一个可满足的真值分配.Dis-MU 类指一个极小不可满足子式 F 可以分裂为 2 个独立的公式 G 与 H,且公式 G 与 H 中不再包含互补文字.目前一些证据[29]表明 Unique-SAT 不是 D^p -complete 的,那么 Unique-MU 也不太可能是 D^p -complete.Almost-Unique-MU 要比 Unique-MU 复杂,因为文中证明了它是 D^p -complete.但是未能找到 Dis-MU 归约到 Unique-SAT 方法.文献[20]中证明了以下层次结构:

$$\text{Unique-SAT} \approx_p \text{Unique-MU} \leqslant_p \text{Dis-MU}$$

$$\text{MU} \approx_p \text{MARG-MU} \leqslant_p \text{MAX} - \text{MU} \approx_p \text{Almost} - \text{Unique} - \text{MU}$$

式中:$\leqslant_p$ 表示多项式时间可约.$A \approx_p B$ 是 $A \leqslant_p B$, $B \leqslant_p A$ 的缩写.

3.2 不可满足子式参数复杂性

Downey 和 Fellows 提出了一种解决 NP 难问题的新的研究方法,即很多难解和无法判定的问题通过引入参数可以得到时间复杂度为 $O(f(k) \cdot n^{\alpha})$ 的固定参数可解(FTP)方法.极小不可满足子式也可以参数化,通常用 MU (k) $(k \geqslant 1)$ 表示固定差值为 k 的极小不可满足子式.参数 k 是子句数减去变量数的差值.文献[21]中提出了一个多项式时间算法来

判断一个公式是否属于 MU(k).因为其使用了“遍历全部 k 个子集”的策略,所以算法非常依赖参数 k. H. Fleischne 等利用双边图匹配的某些相应赋值限制寻找一个可满足的真值赋值.算法的时间复杂度为 $n^{O(k)}$,精确上界 $O(\ln k + 1/2)$. (l 为公式长度,n 为公式的变量数).S. Szeide[22] 提出了一个算法,并得到一个新的结果.算法产生于 SAT 问题的固定参数可解问题:对于一个 CNF 公式 F,如果其子集中子句个数超过变量个数最多 k 个,那么可以在 $O(2^k n^3)$ 时间内判断公式 F 的可满足性.参数 k 就是 F 的最大差值,可以用图匹配算法有效计算.最后,基于以上结论,利用图论中的最大差值和 q-expanding 双边图[30] 计算出改进的结果为 $O(2^k n^4)$.

3.3　QBF 中的极小不可满足子式

近年来,SAT 技术已经成功应用在限界模型检验(bounded model checking,BMC)并推广到限界的模型检验.但是直接将基于 SAT 的 BMC 推广到非限界的模型检验的方法不能克服软件规模增大导致的状态空间爆炸问题,即模型检验过程中软件规模的增大可能导致命题逻辑公式的长度呈指数倍的增长.带量词的布尔公式 QBF 作为 SAT 公式的自然扩展具有紧凑的空间结构和更强大、更直观的表达能力.

赵希顺等[23] 对 QBF 中的极小不可满足问题(minimal falsity,MF)进行了研究.对不可满足的子式研究,有可能产生新的 QBF 求解算法.QBF 的最小不可满足公式 QCNF 指的是该公式不可满足但其任一真子公式均可满足.一个 QCNF 公式的差值不同于命题公式的差值,其差值是子句数和存在量词之间的差值.对 QCNF 公式固定差值为 1 的极小不可满足子式可表示为 MF(1).文献[23]中得出重要结论:对 QCNF 公式固定差值为 1 的极小不可满足子式可以在多项式时间有解.这个证明依赖于公式 MU(1)的结构. 对于 MF(k)($k \geqslant 2$)是否存在多项式时间可解,仍然是开放的.

4　总结与展望

不可满足子式的研究给研究人员带了多方面的挑战,但是也给研究人员带来多方面机会.不可满足子式的研究出现了以下研究方面的热点.

1)使用符号化技术,如 BDD 技术能够紧凑地表示公式,是缓解内存瓶颈的关键技术.文献[31]借助于 BDD 判断公式是不可满足的并且其全部子公式是满足的,从而证明公式是极小不可满足子式.

2)混合求解算法:每种求解策略有其自身优点和缺点.混合求解有利于发挥各自算法的优势,避免其缺点,尽可能提高算法的效率.例如文献[26]中提出了一种在 CSP 中提取极小不可满足子式的混合算法,这是直接算法和间接算法的混合.文献[14]求解最小不可满足子式可以视为贪心算法与遗传算法的混合算法.

3)新领域中扩展:在形式验证等实际应用中,QBF、CSP、SMT 都比 SAT 表达力强.尤其 SMT 技术得到飞速发展,并且在验证领域获得很好的效果.研究 SMT 的极小不可满足子式将是未来主要方向.

4)参数复杂性:将固定差值的极小不可满足子式转化为固定参数问题,并且求解其计算复杂性.但是,对于 QBF 中,MF(k)(k>1)是否可以在多项式时间内判定,依然是开放的.

参考文献:

[1] GUPTA A. Learning abstractions for model checking[D]. Pittsburgh: Carnegie Mellon University, 2006: 1-180.

[2] De La BANDA M G, STUCKEY P J, WAZNY J. Finding all minimal unsatisfiable subsets[C]//Proceedings of the 5th ACM SIGPLAN International Conference on Principles and Practice of Declaritive Programming. New York, USA, 2003: 32-43.

[3] OH Y, MNEIMNEH M N, ANDRAUS Z S. AMUSE: a minimally unsatisfiable subformula extractor[C]//Proceedings of the 41st Annual Design Automation Conference. New York, USA, 2004: 518-523.

[4] LIFFITON M, SAKALLAH K. On finding all minimally unsatisfiable subformulas[C]//Theory and Applications of Satisfiability Testing. St. Andrews, Scotland, 2005: 173-186.

[5] MARQUES-SILVA J, LYNCE I. On improving MUS extraction algorithms[M]//Lecture Notes in Computer Science, Heidelberg: Springer, 2011, 6695: 159-173.

[6] Van MAAREN H, WIERINGA S. Finding guaranteed MUSes fast[M]. Heidelberg: Springer, 2008: 291-304.

[7] GRÉGOIRE É, MAZURE B, PIETTE C. Extracting MUSes[C]//Proceedings of the 17th European Conference on Artificial Intelligence 2006. Riva del Garda, Italy, 2006: 387-391.

[8] GRÉGOIRE É, MAZURE B, PIETTE C. Local-search extraction of MUSes[J]. Constrain, 2007, 12(3): 325-344.

[9] ZHANG J, SHEN S, LI S. Finding unsatisfiable subformulas with stochastic method[M]. Heidelberg: Springer, 2007, 4881: 385-394.

[10] ZHANG J, SHEN S, LI S. A heuristic local search algorithm for unsatisfiable cores extraction[M]. Heidelberg: Springer, 2007, 4707: 649-659.

[11] ZHANG J, SHEN S, LI S. Tracking unsatisfiable subformulas from reduced refutation proof[J]. Journal of Software, 2009, 4(1): 42-49.

[12] LYNCE I, MARQUES-SILVA J P. On computing minimum unsatisfiable cores[C]//International Symposium on Theo-

ry and Applications of Satisfiability Testing. Vancouver, Canada, 2004: 305-310.

[13] LIFFITON M, MNEIMNEH M, LYNCE I, et al. A branch and bound algorithm for extracting smallest minimal unsatisfiable subformulas[J]. Constraints, 2009, 14(4): 415-442.

[14] ZHANG Jianmin, SHEN Shengyu, LI Sikun. Algorithms for deriving minimum unsatisfiable Boolean subformulae[J]. Acta Electronica Sinica, 2009, 37(5): 993-999.

[15] CIMATTI A, GRIGGIO A, SEBASTIANI R. A simple and flexible way of computing small unsatisfiable cores in SAT modulo theories[M]. Heidelberg: Springer, 2007, 4501: 334-339.

[16] CIMATTI A, GRIGGIO A, SEBASTIANI R. Computing small unsatisfiable cores in satisfiability modulo theories [J]. Journal of Artificial Intelligence Research, 2011, 40: 701-728.

[17] ZHANG Jianmin, SHEN Shengyu, ZHANG Jun, et al. Extracting minimal unsatisfiable subformulas in satisfiability modulo theories [J]. Computer Science and Information Systems, 2011, 8: 693-710.

[18] ZHANG Jianmin, SHEN Shengyu, LI Sikun. An algorithm for extracting minimal unsatisfiable subformulae in first-order logic based on refutation implication[J]. Chinese Journal of Conputers, 2010, 33(3): 415-426.

[19] BÜ NING H K, ZHAO X. On the structure of some classes of minimal unsatisfiable formulas [J]. Discrete Applied Mathematics, 2003, 130(2): 185-207.

[20] BÜ NING H K, ZHAO X. The complexity of some subclasses of minimal unsatisable formulas[J]. Journal on Satisfiability Boolean Modeling and Computation, 2007, 3: 1-17.

[21] FLEISCHNER H, KULLMANN O, SZEIDER S. Polynomial-time recognition of minimal unsatisable formulas with fxed clause-variable difference [J]. Theoretical Computer Science, 2002, 289(1): 503-516.

[22] SZEIDER S. Minimal unsatisable formulas with bounded clause-variable difference are fixed-parameter tractable [M]. Heidelberg: Springer, 2003: 548-558.

[23] BÜ NING H, ZHAO X. Minimal false quantified Boolean formulas[M]. Heidelberg: Springer, 2006: 339-352.

[24] HAN B, LEE S J. Deriving minimal conflict sets by CS-trees with mark set in diagnosis from first principles[J]. IEEE Transactions on Systems, Man, and Cybernetics, 1999, 29(2): 281-286.

[25] BAILEY J, STUCKEY P J. Discovery of minimal unsatisfiable subsets of constraints using hitting set dualization [M]. Heidelberg: Springer, 2005: 174-186.

[26] SHAH I. A hybrid algorithm for finding minimal unsatisfiable subsets in over-constrained CSPs [J]. International Journal of Intelligent Systems, 2011, 26(11): 1023-1048.

[27] Van MAAREN H. Pivoting algorithms based on Boolean vector labeling[J]. Acta Mathematica Vietnamica, 1997, 22(1): 183-198.

[28] DAVYDOV G, DAVYDOVA I, BÜNING H K. An efficient algorithm for the minimal unsatisfiability problem for a subclass of CNF[J]. Annals of Mathematics and Artificial Intelligence, 1998, 23: 229-245.

[29] TORAN J. Complexity classes defined by counting quantifiers[J]. Journal of the ACM, 1991, 38(3): 753-774.

[30] LOVASZ L, PLUMMER M D. Matching theory[M]. Amsterdam: North-Holland Publishing, 1986: 29.

[31] HUANG J P. MUP: a minimally unsatisfiability prover [C]//Proceedings of the 10th Asia and South Pacific Design Automation Conference. Shanghai, China, 2005: 432-437.

作者简介:

殷明浩,男,1979 年生,主要研究方向为自动推理和智能规划. 主持过多项国家自然科学基金等项目,发表学术论文多篇,其中被 SCI 检索 20 余篇.

李欣,男,1982 年生,硕士研究生,主要研究领域为自动推理和智能规划. 发表学术论文多篇.