ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.21, No.11, November 2010, pp.2711−2724
doi: 10.3724/SP.J.1001.2010.03664

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

基于约束的多面体抽象域的弱接合*

陈立前, 王 戟+, 刘万伟

(国防科学技术大学 计算机学院 并行与分布处理国家重点实验室,湖南 长沙 410073)

Weak Join for the Constraint-Based Polyehdra Abstract Domain

CHEN Li-Qian, WANG Ji+, LIU Wan-Wei

(National Laboratory for Parallel and Distributed Processing, School of Computer, National University of Defense Technology, Changsha 410073, China)

+ Corresponding author: E-mail: wj@nudt.edu.cn

Chen LQ, Wang J, Liu WW. Weak join for the constraint-based polyehdra abstract domain. *Journal of Software*, 2010,21(11):2711−2724. http://www.jos.org.cn/1000-9825/3664.htm

Abstract: The main tractability problem of the constraint-based polyhedra abstract domain can be derived from the costly (strong) join operation, that is, the convex hull computation. This paper presents a series of cheap weak join operations as a sound substitution for the convex hull operation of the constraint-based polyhedra domain. To achieve a trade-off between efficiency and precision, a heuristic strategy is proposed which dynamically combines both strong join and weak join during program analysis. Experimental results show that the weak join operation can significantly improve the efficiency, scalability and robustness of the constraint-based polyhedra domain.

Key words: static analysis; abstract interpretation; polyhedra abstract domain; convex hull; strong join; weak join

摘 要: 基于约束的多面体抽象域的处理能力主要受限于其高代价的(强)接合操作,即两多面体的凸闭包计算.针对基于约束的多面体抽象域提出了一系列低代价的弱接合操作,以作为凸闭包计算的可靠替代候选.为了能够在分析效率和精度之间取得合理权衡,还提出了一种启发式策略,以把强、弱接合动态地、有机地结合起来进行程序分析.实验结果表明,弱接合能够极大地提升基于约束的多面体抽象域的效率、可扩展性和鲁棒性.

关键词: 静态分析;抽象解释;多面体抽象域;凸闭包;强接合;弱接合

中图法分类号: TP311 **文献标识码**: A

抽象解释[1,2]是一种对数学结构进行近似(或抽象)的通用理论.该理论在静态分析中的一个重要应用是数值程序分析[3],旨在自动产生程序中某程序点处的数值不变式,即每次程序执行均满足的数值变量间的关系(如 $x-y\leqslant 1$).基于抽象解释的数值程序分析在编译优化、程序分析与验证等方面都有着广泛的应用[4],可用来分析程序中是否有除零错、数组越界、整数溢出等运行时错误,也可以用来分析程序中的断言以及契约中的前置条件、后置条件和不变式.

抽象域是抽象解释理论中的一个核心概念,由两部分构成:一个计算机可表示的对象(称为域元素)集合以

* Supported by the National Natural Science Foundation of China under Grant Nos.60725206, 60921062, 60803042, 90818024 (国家自然科学基金); the Hu'nan Provincial Natural Science Foundation of China under Grant No.07JJ1011 (湖南省自然科学基金)

Received 2008-12-30; Accepted 2009-05-21

及一个用来操纵这些对象的操作(称为域操作)集合.在程序分析中,程序的状态集合通过抽象域中的域元素来近似,程序语义动作(赋值、测试、控制流接合、循环等)通过抽象域中的域操作来建模.数值抽象域的设计与实现是基于抽象解释的数值程序分析的关键.近年来,该领域出现了许多在表达能力、运行效率等方面各有特色的数值抽象域,包括区间域[5]、多面体域[6]、八边形域[7]、每不等式两变量域[8]、模板约束矩阵域[9]、子多面体域[10]等.

其中,多面体抽象域由 Cousot 和 Halbwachs 于 1978 年提出[6],是表达能力最强、应用最广泛的数值抽象域之一.多面体抽象域可用来推导程序中变量之间的线性关系,即线性不变式(形如$\sum a_i x_i \leqslant c$).经过 30 多年的发展,多面体抽象域在软硬件系统分析与验证领域已取得了广泛应用[11].目前,多面体抽象域的一些开源实现库已经开发出来,包括 PPL 库[12]、Polylib 库[13]、NewPolka 库[14](现已集成到 APRON 数值抽象域库[15]中)等.

虽然这些实现库在实现方面做了很多优化,但由于多面体抽象域本身固有的高复杂度,多面体抽象域在许多应用中仍受到可扩展性和易处理性方面的限制[16].目前,多面体域的实现[12−14]通常都是基于一种对偶表示[6],即约束表示(constraint representation)和帧表示(frame representation).在约束表示中,一个多面体通过一组有穷线性约束的交来描述.而在帧表示中,一个多面体通过其生成子(顶点和射线)的有穷集合来描述.一些域操作(如,交和测试)在约束表示中实现更为高效,而另一些域操作(如,投影和接合)则在帧表示中实现更为高效.因此,在基于多面体的分析过程中,经常需要把一种表示转化成另一种表示.这种对偶转化一般采用 Chernikova 算法来实现,但是 Chernikova 算法计算代价较高,并可能产生相对输入指数级的输出[17].在基于对偶表示的多面体抽象域中,这种经常性出现的、代价昂贵的对偶转化严重削弱了多面体抽象域的执行效率和可扩展性.

为了避免多面体抽象域的高复杂度,一些研究建议抛弃通用多面体抽象域,而采用表达能力稍弱的弱关系型抽象域(weakly relational abstract domain)[3].本质上,这些弱关系型抽象域都是受限形式的多面体,复杂度较低并存在特定的高效算法来实现各种域操作.这种弱关系型抽象域包括八边形抽象域(形如$\pm x \pm y \leqslant c$)[6]、每不等式两变量抽象域(形如 $ax+by \leqslant c$)[8]、模板约束矩阵域(又称模板多面体抽象域,形如$\sum a_i x_i \leqslant c$,其中 a_i 的取值事先固定)[9]等.实践表明,这些弱关系型抽象域能够很好地改进多面体抽象域的复杂度,执行效率较高,可扩展性较好.但是,这些弱关系型抽象域均由于表达能力的限制,不能在分析中推导出可能需要的形式任意的不变式,从而可能不能证明某些用户感兴趣的程序性质.

作为多面体对偶表示的一种替代候选,Simon 和 King[18]采用基于约束的凸闭包算法,把接合操作(即凸闭包计算)转化为投影操作,以消除帧表示,从而避免对偶转化所带来的复杂度瓶颈.这种基于约束表示的接合操作的出现使得只基于约束表示的多面体抽象域的实现成为可能.基于同样的思想,Chen,Mine 和 Cousot[19]给出了只基于约束表示的多面体抽象域的完整实现,并提出了一个可靠的浮点多面体抽象域,进一步验证了只基于约束的多面体抽象域的可行性.在基于约束的多面体抽象域中,凸闭包计算是通过转化为 Fourier-Motzkin 变量消除来实现的,而 Fourier-Motzkin 变量消除序列一般会产生大量甚至指数级的冗余约束,从而可能引发冗余约束的组合爆炸问题[20].正如文献[18]中所述,基于约束的凸闭包计算方法适合于低维稀疏约束系统.但对于高维的或稠密的约束系统,由于计算过程中会产生大量冗余约束,该方法的执行效率将受到很大影响.线性约束系统中的冗余约束通常需要采用线性规划技术逐一消除,而在凸闭包计算过程中,这种可能出现的大规模的线性规划问题将会极大地降低线性规划求解器的执行效率,甚至导致线性规划求解器出现“数值不稳定性(numerical instability)”问题而不能找到最优解.因此,基于约束的接合操作成为基于约束表示的多面体抽象域的主要计算瓶颈,极大地制约了其执行效率和可扩展性,难以在实际程序分析中广泛采用.

Sankaranarayanan 等人[9,21]称其模板约束矩阵域中的接合操作为弱接合(weak join),以区别传统通用多面体抽象域中的(强)接合,即凸闭包.但是,模板约束矩阵域的主要缺陷在于,该域只能发现预先设计好的固定形式的变量间线性关系(称为模板),而不能发现其他新的约束关系.与之不同的是,本文将面向通用多面体抽象域.

本文为基于约束的通用多面体抽象域设计了弱接合操作,在不损失太多精度的前提下提高该抽象域的执行效率、可扩展性和易处理性.其基本思想是,结合多面体凸闭包的几何性质,利用包络和界等易于计算的启发式信息,把稠密的、复杂的多面体约束表示稀疏化、简单化,通过低代价的弱接合操作求得凸闭包的上近似,并

保证可靠性.为了能够在程序分析的计算效率与精度之间取得合理权衡,本文还提出了一种启发式策略,把强、弱接合动态地、有机地结合起来进行程序分析.实验表明,弱接合操作能够极大地提高基于约束的多面体抽象域的分析效率和可扩展性,尤其对于稠密的、大型的约束系统,弱接合操作的应用能够极大地提高基于约束的多面体分析的易处理性和鲁棒性.

本文第1节首先回顾基于约束的多面体抽象域的接合操作的实现方法.第2节为基于约束的多面体抽象域设计弱接合操作.第3节提出一个启发式策略,以在实际程序分析中把强、弱接合操作有机结合起来.第4节讨论本文弱接合操作的基于浮点的可靠实现方法.第5节给出并比较基于弱接合的与基于强接合的多面体分析的实验结果.第6节对本文进行总结并讨论未来的工作.

1 基于约束的多面体抽象域的接合

在基于约束表示的多面体抽象域中,在有理数$\mathbb{Q}$上,一个(凸)多面体 P 可通过一个线性不等式系统 $P=\{Ax\leqslant b\}$来描述,其中,$A\in\mathbb{Q}^{m\times n}$是一个有理数矩阵,$b\in\mathbb{Q}^m$是一个有理数向量,$m$ 是不等式系统中约束的数目,n 是不等式系统中变量的个数.其语义在几何上对应位于该多面体内的点的集合,即$\gamma(P)=\{x\in\mathbb{Q}^n|Ax\leqslant b\}$,其中,每个点 x 代表一种可能的环境(或称状态),即对所有变量 x 的一种可能赋值.注意,等式型约束$\sum a_ix_i=b$ 可通过一对≤型约束来表示,即$\sum a_ix_i\leqslant b\wedge\sum(-a_i)x_i\leqslant -b$,而形如$\sum a_ix_i<b$ 的严格不等式则可以可靠地抽象为$\sum a_ix_i\leqslant b$.一个不等式约束φ被一个多面体 P 蕴含,记为 $P\vDash\varphi$,当且仅当 $\gamma(P)$中的所有点都满足不等式φ.本文定义多面体域上的序关系如下:$P_1\sqsubseteq P_2$当且仅当$\gamma(P_1)\subseteq\gamma(P_2)$,即$\forall\varphi_2\in P_2.\ P_1\vDash\varphi_2$,并称 P_2 是 P_1 的一个上近似.两个多面体 P_1 和 P_2 的交 $P_1\sqcap P_2$ 定义为这两个多面体的约束的集合并所对应的多面体,从而$\gamma(P_1\sqcap P_2)=\gamma(P_1)\cap\gamma(P_2)$.

基于约束表示的多面体抽象域的完整实现可以归结到两个基本原操作的实现:投影和线性规划[19].其中,接合操作通过转化归结到投影操作,但是,投影操作消除变量的过程中可能引入冗余约束,而冗余约束的消除则需要通过线性规划技术来实现.

1.1 投 影

投影是基于抽象解释的静态分析中的一个重要操作.在多面体抽象域中,它可以用来从给定多面体中消除关于某特定变量 x_i 的信息,而不影响到其他变量之间的关系.投影操作可以通过经典的 Fourier-Motzkin 变量消除算法来实现,以从定义多面体 P 的约束系统中消除所有 x_i 的出现:

$$Fourier(P,x_i)=\left\{(-a_i^-)c^+ + a_i^+c^- \left| \begin{array}{l} c^+=\left(\sum_k a_k^+x_k\leqslant b^+\right)\in P,\ a_i^+>0 \\ c^-=\left(\sum_k a_k^-x_k\leqslant b^-\right)\in P,\ a_i^-<0 \end{array}\right.\right\}\cup\left\{\left(\sum_k a_kx_k\leqslant b\right)\in P\mid a_i=0\right\}.$$

1.2 线性规划

线性规划(linear programming,简称 LP)是在一组给定有穷线性约束(称为可行空间)下寻找某线性函数(称为目标函数)的最优值的方法.线性规划的理论和方法都已发展得很成熟,在科学与工程领域应用非常广泛.目前,该领域研究人员已经开发了许多高效的、可扩展到成千上万规模变量数和约束数的线性规划算法和工具.

求最小目标函数值的线性规划问题具有如下形式:min e subject to P,其中,P 是一个多面体的约束表示,称为可行空间;e 是一个线性表达式,形如$\sum_i a_ix_i$,称为目标函数.与之对应的是,求最大目标函数值的线性规划问题:max e subject to P.

一个不等式约束$\varphi:\sum_i a_ix_i\leqslant b$ 是否被一个多面体 P 蕴含,可以通过线性规划来判定.求解 LP 问题:μ=max $\sum_i a_ix_i$ subject to P,如果$\mu\leqslant b$,则φ是被多面体 P 蕴含,即 $P\vDash\varphi$,否则 $P\nvDash\varphi$.

1.3 冗余约束消除

一个多面体的约束表示不是唯一的,两个不同的约束表示可能在几何上对应同一个多面体.比如,约束集合$\{x=0,y=0\}$与$\{x=0,y=x\}$在几何上表示 x–y 平面上的同一个点(0,0).在实现时,为了保证执行效率,一般希望约束越少越好,因而需要消除冗余约束.若一个不等式$\varphi\in P$ 能够被 P 中其他约束蕴含,即 $P\backslash\{\varphi\}\vDash\varphi$,则称不等式$\varphi$在 P 中

是冗余的.对于 P 中任意一个不等式约束 $\varphi:\sum_i a_i x_i \leqslant b$,可通过求解 LP 问题:$\mu=\max \sum_i a_i x_i$ subject to $P\backslash\{\varphi\}$,来检查 φ 是否是冗余的.如果 $\mu \leqslant b$,则 φ 是冗余的,能够从 P 中删除.该过程可重复进行,直到 P 中没有更多的不等式可删除为止.

1.4 接 合

在基于抽象解释的程序分析中,为了抽象控制流接合(join),需要计算程序变量的环境的并(union).然而,多面体的并不是封闭的,即两多面体的并不一定是一个多面体.包含两多面体 P 和 P' 的并的最小多面体是这两个多面体的凸闭包(convex hull,简称 CH),记作 $P \sqcup_{CH} P'$.两多面体的凸闭包可能包含一些不在原来两个多面体内的点.

例如,对于图 1 中所示的程序,*brandom* 表示随机布尔值.在执行完 if 语句后,在①处,程序变量 x,y 的可能取值为 $(x=0 \wedge y=0) \vee (x=1 \wedge y=1)$,几何上对应 x–y 平面上的点(0,0)和点(1,1),即多面体 $P=\{x=0,y=0\}$ 和多面体 $P'=\{x=1,y=1\}$ 的并.这两个点的并不是一个凸的多面体,包含这两个点的并的最小多面体是连接这两个点的线段,即这两个多面体的凸闭包 $P \sqcup_{CH} P'=\{y=x,0 \leqslant x \leqslant 1\}$.

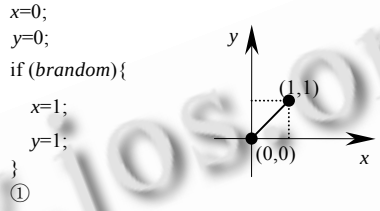

Fig.1 A join operation in the polyhedra abstract domain

图 1 多面体抽象域上的接合操作

在基于对偶表示的多面体抽象域[6]中,两多面体的凸闭包通过帧表示上的生成子(即顶点和射线)的并来计算,然后通过应用 Chernikova 算法把并结果的帧表示转化成约束表示,实现冗余生成子的消除.在基于约束的多面体域中,一般采用文献[18]中给出的方法来计算该闭包.其主要思想是:两个多面体 P 和 P' 的凸闭包可以通过考虑分别来自这两个多面体的任意两点 $z \in P$ 和 $z' \in P'$ 之间的凸组合(convex combination)来构造.两点 $z \in P$ 和 $z' \in P'$ 之间的凸组合定义为 $x=\sigma_1 z+\sigma_2 z'$,其中,$\sigma_1,\sigma_2 \in [0,1]$ 且 $\sigma_1+\sigma_2=1$,几何上 x 代表了 z 与 z' 之间连线线段上的任意一点.

给定 $\gamma(P)=\{x \in \mathbb{Q}^n | Ax \leqslant b\}$ 和 $\gamma(P')=\{x \in \mathbb{Q}^n | A'x \leqslant b'\}$,$P$ 和 P' 的凸闭包为

$$\gamma(P_H)=\left\{x \in \mathbb{Q}^n \middle| x=\sigma_1 z+\sigma_2 z' \wedge \sigma_1+\sigma_2=1 \wedge \sigma_1 \geqslant 0 \wedge \sigma_2 \geqslant 0 \wedge Az \leqslant b \wedge A'z' \leqslant b'\right\},$$

其中,$\sigma_1,\sigma_2 \in \mathbb{Q}$ 和 $x,z,z' \in \mathbb{Q}^n$.为了消除非线性等式 $x=\sigma_1 z+\sigma_2 z'$,我们引入 $y=\sigma_1 z$ 和 $y'=\sigma_2 z'$,并把上述系统松弛成

$$\gamma(P_{CH})=\left\{x \in \mathbb{Q}^n \middle| x=y+y' \wedge \sigma_1+\sigma_2=1 \wedge \sigma_1 \geqslant 0 \wedge \sigma_2 \geqslant 0 \wedge Ay \leqslant \sigma_1 b \wedge A'y' \leqslant \sigma_2 b'\right\},$$

从 $\gamma(P_{CH})$ 中投影掉 σ_1,σ_2,y,y' 即可得到 P 和 P' 的凸闭包(的拓扑闭包).

几何上,不难看出,凸闭包计算具有单调性.

性质 1(凸闭包的单调性). 给定多面体 P_1,P_1',P_2 和 P_2',若 $P_1 \sqsubseteq P'_1$ 且 $P_2 \sqsubseteq P_2'$,则 $P_1 \sqcup_{CH} P_2 \sqsubseteq P_1' \sqcup_{CH} P_2'$.

例 1:给定多面体 $P_1=\{-4x-y \leqslant 34^{(1)},-2x+3y \leqslant 24^{(2)},x+3y \leqslant 6^{(3)},x+y \leqslant 0^{(4)},2x-3y \leqslant -5^{(5)},x-4y \leqslant 0^{(6)}\}$,$P_2=\{-x-2y \leqslant 2^{(7)},-2x-y \leqslant -5^{(8)},-4x+y \leqslant -7^{(9)},-3x+4y \leqslant 11^{(10)},-x+6y \leqslant 41^{(11)},2x+3y \leqslant 53^{(12)},2x+y \leqslant 39^{(13)},2x-y \leqslant 33^{(14)},3x-4y \leqslant 52^{(15)},x-4y \leqslant 28^{(16)}\}$,$P_1$ 与 P_2 的凸闭包 $P_1 \sqcup_{CH} P_2=\{-4x-y \leqslant 34^{(1)},-2x+3y \leqslant 24^{(2)},-x+6y \leqslant 41^{(11)},2x+3y \leqslant 53^{(12)},2x+y \leqslant 39^{(13)},2x-y \leqslant 33^{(14)},3x-4y \leqslant 52^{(15)},x-4y \leqslant 28^{(16)},-4x+13y \leqslant 76^{(17)},-3x-16y \leqslant 56^{(18)}\}$,如图 2 所示.

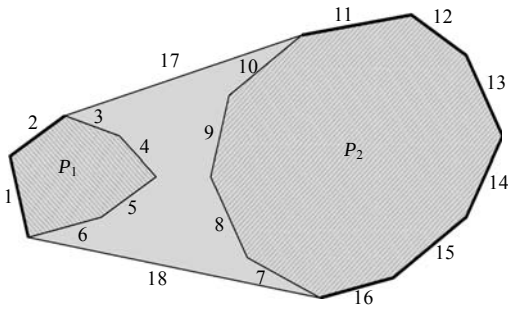

Fig.2 Polyhedral convex hull

图 2 多面体凸闭包

2 基于约束的多面体抽象域的弱接合

多面体的接合操作具有很高的复杂度,运行代价很昂贵,但却是多面体抽象域中经常用到的一个操作[16].在基于约束的多面体抽象域中,按前一节所述方法,接合操作可转化为投影操作,通过经典的 Fourier-Motzkin 变量消除算法来实现.使用 Fourier-Motzkin 变量消除算法的一个主要缺陷在于该算法可能引入指数级的冗余约束[18],从而需要耗费大量线性规划求解来消除这些冗余约束.这是基于约束的多面体凸闭包计算方法代价昂贵的根源.

为了降低多面体接合操作的代价以提高多面体分析的效率,本节将在抽象解释框架下设计低计算代价的多面体弱接合,记作$\sqcup_W$.为了区别传统接合操作,本文把文献[6]中定义的接合操作称为强接合(strong join)操作,记为$\sqcup_S$.在基于约束的多面体抽象域中,$\sqcup_S$通过计算两多面体的凸闭包得到,即$\sqcup_S=\sqcup_{CH}$.基于抽象解释理论,弱接合$\sqcup_W$是可靠的当且仅当 $P_1\sqcup_S P_2\sqsubseteq P_1\sqcup_W P_2$,即多面体弱接合操作所得多面体应是强接合计算所得多面体的上近似(over-approximation).几何上,所有位于强接合计算所得多面体内的点应在弱接合计算所得多面体内.

从构成上看,两个多面体的凸闭包的约束表示可按来源分为两部分:旧关系型约束和新关系型约束.旧关系型约束是指这些约束在凸闭包的输入参数多面体的约束表示中已出现过.新关系型约束是指这些约束在凸闭包的输入参数多面体的约束表示中并没有出现,而是在凸闭包计算过程中新产生的.在如图 2 所示的凸闭包中,用粗线标识的$\{-4x-y\leqslant 34^{(1)},-2x+3y\leqslant 24^{(2)},-x+6y\leqslant 41^{(11)},2x+3y\leqslant 53^{(12)},2x+y\leqslant 39^{(13)},2x-y\leqslant 33^{(14)},3x-4y\leqslant 52^{(15)},x-4y\leqslant 28^{(16)}\}$是旧关系型约束,而$\{-4x+13y\leqslant 76^{(17)},-3x-16y\leqslant 56^{(18)}\}$则是新关系型约束.

定义 1(包络). 给定两个多面体 P_1 和 P_2,P_1 和 P_2 的(凸)包络(envelope)定义为:$env(P_1,P_2)=S_1\cup S_2$,其中,$S_1=\{\varphi_1\in P_1|P_2\models\varphi_1\}$,$S_2=\{\varphi_2\in P_2|P_1\models\varphi_2\}$.

事实上,两个多面体的包络构成了这两个多面体凸闭包中的旧关系型约束.设 $i\in\{1,2\}$,对于任意约束$\varphi\in P_i$,如果$\varphi\in env(P_1,P_2)$,则称约束φ是 P_i 中的包络约束;否则,称φ是 P_i 中的非包络约束.值得注意的是,两个多面体包络的计算只需简单的蕴含检查就可以在凸闭包计算之前低代价地确定下来.

性质 2. 两多面体 P_1 和 P_2 的包络是其凸闭包的上近似,即 $P_1\sqcup_{CH}P_2\sqsubseteq env(P_1,P_2)$.

从定义 1 不难看出,包络可以通过(m_1+m_2)个线性规划求解的代价计算得到,其中 m_1,m_2 分别为 P_1,P_2 约束表示中不等式的数目.

在例 1 中,$env(P_1,P_2)=\{-4x-y\leqslant 34^{(1)},-2x+3y\leqslant 24^{(2)},-x+6y\leqslant 41^{(11)},2x+3y\leqslant 53^{(12)},2x+y\leqslant 39^{(13)},2x-y\leqslant 33^{(14)},3x-4y\leqslant 52^{(15)},x-4y\leqslant 28^{(16)}\}$.

定义 2(界盒). 如果一个多面体 $P=\{Ax\leqslant b\}$的约束表示形如$\pm x_i\leqslant c$(称为界约束),其中,c 是一个有理数常量或$+\infty$,则该多面体 P 是一个盒(box).一个多面体 P 的界盒(bounding box)是包含该多面体的最小的盒,记为 $BB(P)$.

给定两个盒 $B_1=\{a\leqslant x\leqslant b\}$和 $B_2=\{c\leqslant x\leqslant d\}$,则这两个盒在区间抽象域(interval domain)上的接合定义为

$B_1 \sqcup_I B_2=\{\min(a,c)\leqslant x\leqslant \max(b,d)\}$.对于 n 维的多面体,其界盒可以通过最多 $2n$ 次线性规划计算得到.显然,一个多面体的界盒是该多面体的上近似,即 $P\sqsubseteq BB(P)$.两个盒 B_1 和 B_2 在区间抽象域上的接合,是这两个盒在多面体抽象域上凸闭包的上近似,即 $B_1\sqcup_{CH} B_2\sqsubseteq B_1\sqcup_I B_2$.两个多面体 P_1 和 P_2 对应界盒的凸闭包是这两个多面体凸闭包的上近似,即 $P_1\sqcup_{CH} P_2\sqsubseteq BB(P_1)\sqcup_{CH} BB(P_2)$.

性质 3. 两个多面体 P_1 和 P_2 的界盒在区间抽象域上的接合所得到的盒多面体,等价于这两个多面体凸闭包的界盒,即 $BB(P_1\sqcup_{CH} P_2)=BB(P_1)\sqcup_I BB(P_2)$.

根据性质 3,两个多面体凸闭包的界盒的计算也可以在凸闭包计算之前以较低的代价确定下来,即通过计算 $BB(P_1)\sqcup_I BB(P_2)$得到.在基于约束的多面体抽象域的实现(如 FPPol[19])中,通常会为每个变量维护并及时更新其上、下界信息.

在例 1 中,$BB(P_1)=\{-9\leqslant x\leqslant -1,-2\leqslant y\leqslant 4\}$,$BB(P_2)=\{2\leqslant x\leqslant 18,-5\leqslant y\leqslant 9\}$,$BB(P_1\sqcup_{CH}P_2)=BB(P_1)\sqcup_I BB(P_2)=\{-9\leqslant x\leqslant 18,-5\leqslant y\leqslant 9\}$,$BB(P_1)\sqcup_{CH} BB(P_2)=\{-9\leqslant x\leqslant 18,-5\leqslant y\leqslant 9,-5x+11y\leqslant 89,-3x-11y\leqslant 49\}$.

下面,将基于上述低计算代价可得到的旧关系型约束(即包络)和界约束,为基于约束的多面体抽象域设计弱接合操作.其主要目标是在不损失太多精度的情况下,采用轻量级方法产生凸闭包的上近似,以逼近新关系型约束.

2.1 基于模板的弱接合

类似于模板约束矩阵抽象域[9]的思想,如果用户能够提供一些模板(template)约束,则可以把这些模板约束作为弱接合操作结果多面体中约束的候选,并通过线性规划求得凸闭包的上近似.对于多面体抽象域,其模板约束形如$\sum_i a_i x_i\leqslant c$,其中变量 x_i 的系数 a_i 是一个固定的常数,常量项 c 是一个可变的参数,如 $2x+3y\leqslant c$.给定两个多面体 P_1 和 P_2,对于模板$\sum_i a_i x_i\leqslant c$,通过求解如下线性规划问题:$v_1=\max \sum_i a_i x_i$ subject to P_1 及 $v_2=\max \sum_i a_i x_i$ subject to P_2,我们可以把约束$\sum_i a_i x_i\leqslant \max(v_1,v_2)$加入到弱接合结果中.

如果用户不能提供模板约束,本文的策略是选择两个多面体 P_1,P_2 中约束较少的那个多面体中的约束作为模板约束.这里,不妨设 P_1 中的约束较少,不等式$\sum_i a_i x_i\leqslant c_1$ 是 P_1 中的任意一个约束.求解线性规划问题:$v=\max \sum_i a_i x_i$ subject to P_2,如果 $v\neq+\infty$,则把约束$\sum_i a_i x_i\leqslant \max(c_1,v)$加入到弱接合结果中.考虑到界约束的低计算代价及其在程序值范围分析中的重要性,本文把$\pm x_i\leqslant c$ 也作为模板约束,并通过 $BB(P_1)\sqcup_I BB(P_2)$计算得到.

基于模板的弱接合的优点在于其计算代价低,并能保证结果约束系统的约束不会太多.尤其是当参与接合操作的两个多面体一个很简单、约束较少,而另一个很复杂、约束很多时,基于模板的弱接合能够得到比较简单的结果,有利于后续分析.

对于例 1 中的多面体,基于模板的弱接合的结果为$\{-2x+3y\leqslant 24,-4x-y\leqslant 34,x+3y\leqslant 40,x+y\leqslant 23,2x-3y\leqslant 36,x-4y\leqslant 28,x\leqslant 18,y\leqslant 9,-y\leqslant 5\}$,如图 3 所示.

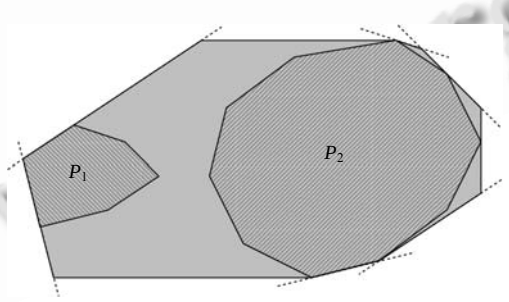

Fig.3 Template based weak join

图 3 基于模板的弱接合

2.2 基于包络和界信息的弱接合

给定两个多面体 P_1 和 P_2,定义 $EEBB(P_1,P_2)=env(P_1,P_2)\sqcap(BB(P_1)\sqcup_I BB(P_2))$,则 $EEBB(P_1,P_2)$是由 P_1 和 P_2 的包

络及其凸闭包之界盒所能确定的最小多面体.由 $P_1\sqcup_{CH}P_2\sqsubseteq env(P_1,P_2)$且 $P_1\sqcup_{CH}P_2\sqsubseteq BB(P_1)\sqcup_I BB(P_2)$,可以得到如下性质:

性质 4. 给定两个多面体 P_1 和 P_2,$EEBB(P_1,P_2)$是 P_1 和 P_2 凸闭包的上近似,即 $P_1\sqcup_{CH}P_2\sqsubseteq EEBB(P_1,P_2)$.

$EEBB(P_1,P_2)$精确地保留了两个多面体的包络和界信息,但是不能发现新关系型约束.为了能够产生和逼近凸闭包中的新关系型约束,本文基于包络和界信息定义如下弱接合操作:

- 设$i\in\{1,2\}$,定义$EB(P_i)=BB(P_i)\cup(env(P_1,P_2)\cap P_i)$,即为由$P_i$中的包络约束及$P_i$的界盒所确定的最小多面体.定义弱接合操作:$P_1\sqcup_{EB}P_2=(EB(P_1)\sqcup_{CH}EB(P_2))\sqcap EEBB(P_1,P_2)$.
- 对偶地,定义$NEB(P_i)=BB(P_i)\cup\{\varphi\in P_i|\varphi\notin env(P_1,P_2)\}$,即为由$P_i$中的非包络约束及$P_i$的界盒所确定的最小多面体.定义弱接合操作:$P_1\sqcup_{NEB}P_2=(NEB(P_1)\sqcup_{CH}NEB(P_2))\sqcap EEBB(P_1,P_2)$.
- 根据参与凸闭包计算的每个输入多面体中的包络约束数和非包络约束数,启发式地确定其上近似.如果该多面体的包络约束少于非包络约束,则使用包络约束及界约束来作为该多面体的上近似;否则,使用非包络约束及界约束来作为该多面体的上近似.记使用该策略为输入多面体P_i确定的上近似为$XB(P_i)$,定义弱接合操作:$P_1\sqcup_{XB}P_2=(XB(P_1)\sqcup_{CH}XB(P_2))\sqcap EEBB(P_1,P_2)$.

显然,$EB(P_i)$,$NEB(P_i)$,$XB(P_i)$都是P_i的上近似,并且都比P_i要稀疏.

性质 5. 给定两个多面体 P_1 和 P_2,$P_1\sqcup_{EB}P_2$,$P_1\sqcup_{NEB}P_2$,$P_1\sqcup_{XB}P_2$ 都是 P_1 和 P_2 凸闭包的上近似,即 $P_1\sqcup_{CH}P_2\sqsubseteq P_1\sqcup_{EB}P_2$,$P_1\sqcup_{CH}P_2\sqsubseteq P_1\sqcup_{NEB}P_2$,$P_1\sqcup_{CH}P_2\sqsubseteq P_1\sqcup_{XB}P_2$.

通常,由于 $EB(P_i)$,$NEB(P_i)$,$XB(P_i)$的约束数都比 P_i 要少,也比 P_i 要稀疏,因而 $P_1\sqcup_{EB}P_2$,$P_1\sqcup_{NEB}P_2$,$P_1\sqcup_{XB}P_2$ 的计算效率和鲁棒性都会优于 $P_1\sqcup_{CH}P_2$.为了进一步提高 $P_1\sqcup_{EB}P_2$,$P_1\sqcup_{NEB}P_2$ 与 $P_1\sqcup_{XB}P_2$ 的精度,在实现时,我们把$EB(P_1)\sqcup_{CH}EB(P_2)$,$NEB(P_1)\sqcup_{CH}NEB(P_2)$,$XB(P_1)\sqcup_{CH}XB(P_2)$执行产生的新关系型约束(未在 P_1,P_2 中出现的非界约束)作为模板约束,采用基于模板的弱接合的思想对弱接合结果进行缩紧.

$P_1\sqcup_{EB}P_2$,$P_1\sqcup_{NEB}P_2$ 与 $P_1\sqcup_{XB}P_2$ 的优点在于它们能够产生新关系型约束,并且结果的精度相对较高.

对于例 1 中的两个多面体,$EEBB(P_1,P_2)=\{-4x-y\leqslant 34,-2x+3y\leqslant 24,-x+6y\leqslant 41,2x+3y\leqslant 53,2x+y\leqslant 39,2x-y\leqslant 33,3x-4y\leqslant 52,x-4y\leqslant 28,-y\leqslant 5\}$.采用模板缩紧技术后,$P_1\sqcup_{EB}P_2$ 的结果为$\{-4x-y\leqslant 34,-2x+3y\leqslant 24,-x+6y\leqslant 41,2x+3y\leqslant 53,2x+y\leqslant 39,2x-y\leqslant 33,3x-4y\leqslant 52,x-4y\leqslant 28,-19x+48y\leqslant 306,-3x-10y\leqslant 44,-y\leqslant 5\}$,$P_1\sqcup_{NEB}P_2$ 的结果为$\{-4x-y\leqslant 34,-2x+3y\leqslant 24,-x+6y\leqslant 41,2x+3y\leqslant 53,2x+y\leqslant 39,2x-y\leqslant 33,3x-4y\leqslant 52,x-4y\leqslant 28,-15x+52y\leqslant 311,-3x-17y\leqslant 61\}$,$P_1\sqcup_{XB}P_2$ 的结果为$\{-4x-y\leqslant 34,-2x+3y\leqslant 24,-x+6y\leqslant 41,2x+3y\leqslant 53,2x+y\leqslant 39,2x-y\leqslant 33,3x-4y\leqslant 52,x-4y\leqslant 28,-15x+43y\leqslant 262,-3x-16y\leqslant 56\}$,如图 4 所示.

2.3 基于两变量约束的弱接合

类似于八边形抽象域[7]和每不等式两变量抽象域[8]的思想,本文采用两变量约束的动机基于如下观察:在实际程序分析中,最多只涉及两变量的不变式通常占有意义的、用户感兴趣的不变式中的大多数,在许多情况下,这种不变式足以分析和验证用户所关注的程序性质.本文基于两变量约束的弱接合的主要目标是产生合适的两变量约束作为上近似来逼近凸闭包.

定理 1. 给定两个盒 B_1 和 B_2,B_1 和 B_2 的凸闭包多面体的约束表示中每个不等式约束最多只涉及两个变量,即两个盒的凸闭包多面体可以精确地在每不等式两变量抽象域中计算得到.

根据定理 1,两个盒 B 和 B'的凸闭包可以在每不等式两变量抽象域中计算得到.考虑任意两个变量 x 和 y,在 x–y 平面上,盒 B,B'各自定义了一个矩形(可能不限界).在 x–y 平面上,两个矩形的凸闭包将由一些形如$\pm x\leqslant c$的单变量约束(即界约束,对应x,y变量的最小下界和最大上界),以及一些形如$ax+by\leqslant c$的两变量约束构成.几何上,这些两变量约束可以通过计算连接两矩形顶点的直线所确定的半平面得到.无论这两个矩形的布局如何,其凸闭包最多产生 4 个两变量约束,分别连接两个矩形各自的左上与左上、左下与左下、右上与右上、右下与右下顶点,如图 5 所示.

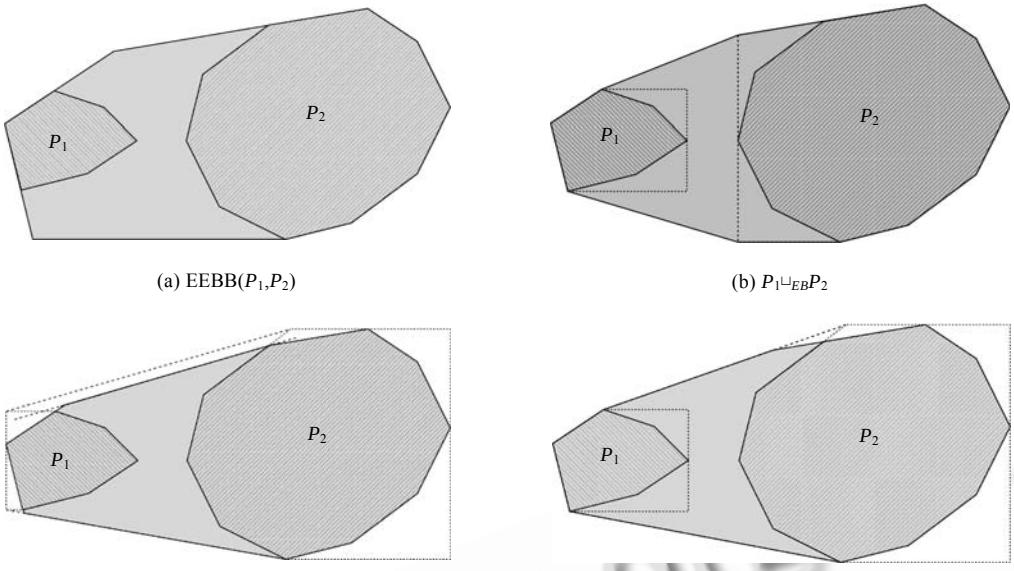

Fig.4 Weak join based on envelope and bounds information

图 4 基于包络和界信息的弱接合

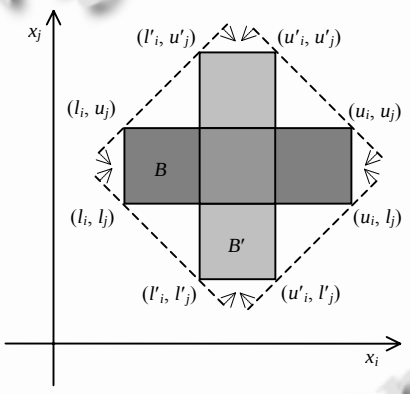

Fig.5 Polyhedral convex hull of boxes

图 5 盒的多面体凸闭包

给定两个 n 维盒 $B=\{l\leqslant x\leqslant u\}$ 和 $B'=\{l'\leqslant x\leqslant u'\}$,其中 l,u,l',u' 都是一个 n 维有理数常数(可能取值$\pm\infty$)向量,x 是一个 n 维向量,下面给出计算 B 和 B' 的多面体凸闭包的具体算法.

算法 1. 盒的凸闭包计算算法.

输入:参与凸闭包计算的两个盒 B 和 B' 的约束表示.

输出:盒的凸闭包多面体 P_B.

1. 在 x 向量中任意选择两个变量 x_i 和 x_j,盒 B 和 B' 在 x_i–x_j 平面上投影后得到两个矩形 $B_{ij}=\{l_i\leqslant x_i\leqslant u_i, l_j\leqslant x_j\leqslant u_j\}$ 和 $B'_{ij}=\{l'_i\leqslant x_i\leqslant u'_i, l'_j\leqslant x_j\leqslant u'_j\}$,其中 $l_i,u_i,l_j,u_j,l'_i,u'_i,l'_j,u'_j\in\mathbb{Q}\cup\{\pm\infty\}$.记 B 和 B' 的凸闭包多面体为 P_B.
   1.1(界约束). 设 $k\in\{i,j\}$,若 $\max\{u_k,u'_k\}\neq+\infty$,则把 $x_k\leqslant\max\{u_k,u'_k\}$ 放入 P_B 中;若 $\min\{l_k,l'_k\}\neq-\infty$,则把 $-x_k\leqslant -\min\{l_k,l'_k\}$ 放入 P_B 中.

1.2(两变量约束). 分如下 4 种情况考虑:

1.2.1(左上左上). 若$-\infty<l_i<l'_i\wedge u_j<u'_j<+\infty$或$-\infty<l'_i<l_i\wedge u'_j<u_j<+\infty$,则把$(u_j-u'_j)x_i+(l'_i-l_i)x_j\leqslant l'_iu_j-l_iu'_j$放入$P_B$中.

1.2.2(左下左下). 若$-\infty<l_i<l'_i\wedge-\infty<l'_j<l_j$或$-\infty<l'_i<l_i\wedge-\infty<l_j<l'_j$,则把$(l'_j-l_j)x_i+(l_i-l'_i)x_j\leqslant l_il'_j-l_jl'_i$放入$P_B$中.

1.2.3(右上右上). 若$u'_i<u_i<+\infty\wedge u_j<u'_j<+\infty$或$u_i<u'_i<+\infty\wedge u'_j<u_j<+\infty$,则把$(u'_j-u_j)x_i+(u_i-u'_i)x_j\leqslant u_iu'_j-u_ju'_i$放入$P_B$中.

1.2.4(右下右下). 若$-\infty<l'_j<l_j\wedge u'_i<u_i<+\infty$或$-\infty<l_j<l'_j\wedge u_i<u'_i<+\infty$,则把$(l_j-l'_j)x_i+(u'_i-u_i)x_j\leqslant l_ju'_i-u_il'_j$放入$P_B$中.

2. 在 x 向量中选择其他两个变量,重复第 1 步,直到所有变量对都被考虑为止.

算法 1 最终将最多产生 $2n$ 个界约束(单变量约束).由于对于每个 x_i–x_j 平面而言,算法 1 最多产生 4 个两变量约束,算法 1 总共最多产生 $2n(n-1)$个两变量约束.因此,算法 1 最终将最多产生 $2n^2$ 个不等式约束.

定义基于两变量约束的弱接合操作:$P_1\sqcup_{TVPI}P_2=(BB(P_1)\sqcup_{CH}BB(P_2))\sqcap EEBB(P_1,P_2)$.根据定理 1,两个多面体的界盒的凸闭包,即 $BB(P_1)\sqcup_{CH}BB(P_2)$,可以采用算法 1 在每不等式两变量抽象域上计算得到,并且其计算代价较低.

性质 6. 给定两个多面体 P_1 和 P_2, $P_1\sqcup_{TVPI}P_2$ 是 $P_1\sqcup_{CH}P_2$ 的上近似,即 $P_1\sqcup_{CH}P_2\sqsubseteq P_1\sqcup_{TVPI}P_2$.

类似地,把 $BB(P_1)\sqcup_{CH}BB(P_2)$得到的两变量约束作为模板约束,采用基于模板的弱接合的思想对凸闭包结果进行缩紧可以提高精度.

基于两变量约束的弱接合操作的优点在于它能够避免凸闭包计算过程中的大量的冗余约束消除计算以及凸闭包计算潜在的指数级输出,保证接合结果中的约束数在 $O(n^2)$数量级内,并能产生程序分析中用户感兴趣的两变量约束.

对于例 1 中的两个多面体,采用模板缩紧技术后,$P_1\sqcup_{TVPI}P_2$ 的结果为$\{-4x-y\leqslant 34,-2x+3y\leqslant 24,-x+6y\leqslant 41,2x+3y\leqslant 53,2x+y\leqslant 39,2x-y\leqslant 33,3x-4y\leqslant 52,x-4y\leqslant 28,-5x+11y\leqslant 74,-3x-11y\leqslant 46,-y\leqslant 5\}$,如图 6 所示.

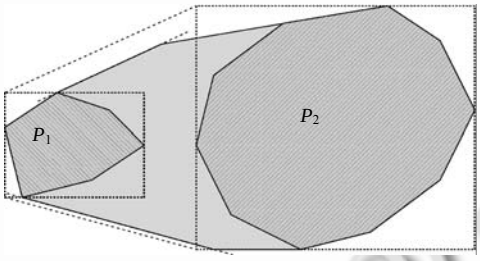

Fig.6 Weak join based on two-variable constraints

图 6 基于两变量约束的弱接合

3 启发式结合策略

前一节提出的弱接合操作$\sqcup_W$均满足 $P_1\sqcup_SP_2\sqsubseteq P_1\sqcup_WP_2$,即弱接合所得多面体均是强接合所得多面体的一个上近似,因此这些弱接合操作都是可靠的.相对强接合,弱接合执行效率较高但存在一定程度的精度损失.同时,本文提出的弱接合操作也各有优缺点:基于模板的弱接合执行效率很高,但是不能产生新关系型约束;基于包络和界信息的弱接合可以产生任意形式的新关系型约束,但是依然存在凸闭包计算过程中产生指数级冗余约束的隐患;基于两变量约束的弱接合计算比较简单,但是只能产生最多涉及两个变量的新关系型约束.在实际程序分析中,为了在执行效率和精度之间取得折中,需要把强接合与这些弱接合动态地结合起来进行程序分析.为此,本文提出一种启发式的结合策略,以在程序分析过程中根据输入多面体的参数特性动态地选择合适的接合

操作.

给定输入多面体,可以对其复杂度和稀疏程度进行定量评价,评价指标包括:多面体的维数、多面体的约束数、系数矩阵中非 0 系数的个数、系数矩阵中非 0 系数的复杂度(小整数、大整数、还是浮点数)等.据此,给定两个输入多面体,本文按如下策略选择接合操作:(1) 若两者都很简单(如约束少、系数矩阵稀疏),则结合策略选择强接合操作,以尽量不损失精度;(2) 若两者都很复杂(如约束多、系数矩阵稠密),则结合策略选择基于两变量约束的弱接合操作,以提高效率;(3) 若其中一个很简单而另一个非常复杂,则结合策略选择基于模板的弱接合操作,其模板约束由比较简单的输入多面体的约束组成;(4) 若两者非包络约束之和远大于(小于)包络约束之和,则使用基于包络(非包络)约束与界约束的弱接合;(5) 对于一般情况,结合策略选择基于包络/非包络约束和界约束的弱接合:如果一个多面体中的包络(非包络)约束较少,则使用包络(非包络)约束和界约束来作为该多面体的上近似.

此外,结合策略还将根据程序分析的不同阶段来选择合适的接合操作.在基于抽象解释的程序分析中,一般采用“迭代+加宽(widening)”的方法来计算程序不动点.在第 1 次迭代完成之前,尽可能地选择强接合操作来产生约束,以尽量不丢失最终可能稳定的约束,即不变式.在随后的迭代中,经常会出现参与接合操作的两个多面体中存在许多相同约束的情况,此时使用弱接合可极大地提高分析效率.在加宽算子应用之后,包络约束和界约束往往在最终稳定的不变式中占很大比率,此时,使用弱接合操作可以提高分析效率,而又不会损失太多的精度.

4 基于浮点的可靠实现方法

通常,多面体抽象域的实现都是基于任意精度有理数的,以保证实现的可靠性,但是这种实现会影响多面体抽象域的可扩展性[16].任意精度的有理数不能被当前硬件直接支持,而是通过软件方法来实现,因而运行速度慢并且占用空间多.与此同时,实际程序分析中,这种实现容易导致大数问题,耗费大量 GCD 计算,影响到分析的效率,如果使用线性规划,还会影响到 LP 求解器的数值稳定性.而浮点数能够被当前硬件直接支持,运行速度快,而且基于相同位数,浮点数比机器整数所能表示的数值范围大得多.比如,64 位机器整数的表示范围约为[–9.2e+18,+9.2e+18],而 64 位浮点数的表示范围约为[–1.7e+308,+1.7e+308].因此,在很大程度上,浮点数能够缓解大数带来的问题.同时,由于 IEEE 浮点数的非均匀分布特性,浮点数支持一种渐进式的精度损失.但是,由于浮点算术的舍入误差,浮点实现的可靠性难以得到保证.

文献[19]提出了一种基于约束的多面体抽象域的可靠的浮点实现方法.其中,两个多面体的凸闭包操作通过一个可靠的浮点 Fourier-Motzkin 变量消除算法和一种轻量的基于浮点的严格线性规划技术来实现.同样地,本文中的弱接合也可以基于这些方法采用浮点数来实现.文献[19]中浮点 Fourier-Motzkin 变量消除算法和基于浮点的严格线性规划技术的可靠性也保证了本文基于浮点实现的弱接合的可靠性,即弱接合的浮点实现总是输出一个比多精度有理数实现更保守的结果,即一个上近似的多面体.

对于算法 1 中的两变量约束产生方法,其浮点实现作如下考虑以保证可靠性.例如,对于算法 1 的 1.2.4(右下右下)情形,若$-\infty<l'_j<l_j \wedge u'_i<u_i<+\infty$成立,则把$((l_j \ominus_{-\infty} l'_j) \oslash_{-\infty} (u_i \ominus_{+\infty} u'_i))x_i - x_j \leqslant ((l_j \ominus_{-\infty} l'_j) \oslash_{-\infty} (u_i \ominus_{+\infty} u'_i)) u'_i \ominus_{+\infty} l'_j$作为结果加入到 P_B 的约束表示中.其中,$\{\oplus_r, \ominus_r, \otimes_r, \oslash_r\}$表示四则浮点算术,舍入模式 $r \in \{+\infty, -\infty\}$,其中,$+\infty$,$-\infty$分别表示向上、向下舍入.对于算法 1 的其他情形,按照类似方法可适配到浮点运算上并保证可靠性.

5 实验结果与分析

本文对基于约束的浮点多面体抽象域 FPPol[19]进行了扩展,采用双精度浮点数并使用线性规划工具 GLPK[22]实现了本文提出的弱接合操作.为了比较相同输入下,强接合和弱接合操作的执行效率,本文针对一些来自实际程序分析中的不同规模、不同复杂度的多面体输入进行了测试.为了评估应用弱接合在实际程序分析中的效果,本文把扩展后的 FPPol 适配到数值抽象域 APRON 库[15]中,并使用静态分析工具 Interproc[23]进行了一系列实验,在结果不变式和性能等方面比较了基于强接合的程序分析与基于弱接合的程序分析.

本文采用的实验平台为:Fedora 9 Linux 操作系统,2GB 物理内存,Intel P4 2.8GHz 单核 CPU 处理器.本文的测试程序源于 StInG[24]和 LpInv[9],这些测试程序原本是以线性迁移系统形式给出的,本文使用 Interproc 支持的 spl 语言重写了这些测试程序.这些测试程序都是控制流接合密集型程序,符合本文的实验需求.Interproc 采用传统的基于抽象解释的不动点迭代方法,并支持延迟的加宽策略.表 2 中的“加宽算子延迟参数”给出了加宽算子应用之前的迭代次数.在实验过程中,如果约束系统太复杂,则 GPLK 可能会面临“数值不稳定”问题,不能找到最优解,从而不能消除约束系统的冗余约束,导致基于约束的多面体分析失效.对于这种情形,表 1、表 2 中使用“×”来标注.

- 单纯接合操作实验

首先,本文使用 FPPol 抽象域,采用传统基于强接合的多面体分析方法对这些测试程序进行分析.我们把每个测试程序分析过程中所产生的较为复杂的、耗时较多的、甚至导致分析失效的一个强接合操作的输入多面体记录下来,作为比较单个强、弱接合操作执行效率的测试用例.单纯单个接合操作测试的实验结果如表 1 所示.表 1 给出了参与该次接合操作的两个输入多面体的变量数以及各自的约束数.对于这些输入,表 1 给出了这些多面体的强接合和各种弱接合操作的运行时间以及计算过程中出现的最大约束数.

从表 1 可以看出,大部分情况下弱接合操作的执行效率都要优于强接合操作的执行效率.尤其是,基于模板和基于两变量约束的弱接合总是大大优于强接合的执行效率,并且鲁棒性好,不会出现失效情况.但是,在某些情况下,由于接合操作输入多面体的拓扑布局的特点,基于约束和界信息的弱接合操作的执行效率可能比强接合操作的执行效率要低.通过比较接合操作结果多面体,我们发现,不同弱接合的结果约束各有特色,精度各有差异,难以断言哪个弱接合操作一定比另一个弱接合操作更精确.一般而言,基于包络约束和界信息的弱接合比基于模板和基于两变量约束的弱接合要精确.综上所述,在实际程序分析中,应根据输入多面体参数的特点来动态地选择合适的弱接合操作.

Table 1 Comparision of strong join and weak join for constraint-based polyhedra domain

表 1 基于约束的多面体强接合和弱接合结果比较

Programs where join comes from	Number of variables	Number of constraints	Computation time in seconds (the largest number of constraints appearing during the join)					
			Strong join	Weak join based on				
				Templates	Envelopes & bounds	Nonenvelopes & bounds	Envelopes/ nonenvelopes & bounds	Two-Variable constraints
Our Example 1	2	6,10	1.323(793)	0.002(10)	0.049(132)	0.020(102)	0.014(80)	0.002(14)
see-saw	2	9,8	1.654(762)	0.002(12)	0.037(103)	0.034(106)	0.003(13)	0.001(12)
berkeley	4	8,8	0.034(52)	0.002(15)	0.029(40)	0.014(34)	0.031(63)	0.001(17)
dragon	5	54,45	×(≥6927)	0.034(54)	×(≥6129)	0.024(105)	0.035(107)	0.005(103)
heap	5	28,17	80.370(4145)	0.003(14)	0.061(34)	169.988(5766)	0.051(34)	0.004(21)
robot	6	8,6	0.014(89)	0.001(12)	0.013(42)	0.007(62)	0.007(62)	0.003(19)
lifo	7	22,14	3.793(260)	0.004(27)	0.187(71)	0.603(121)	0.081(45)	0.006(47)
cars	7	39,17	240.088(9996)	0.032(31)	8.497(731)	×(≥2288)	1.232(156)	0.027(31)
barber	8	14,50	78.803(2464)	0.017(30)	1.818(220)	5.758(504)	7.150(509)	0.009(60)
barberm4-2	8	39,28	×(≥4495)	0.011(44)	417.086(4600)	0.202(91)	0.220(105)	0.004 (78)
swim-pool-1	9	15,14	0.037(46)	0.002(28)	0.027(38)	0.018(37)	0.019(40)	0.003(38)
train-rm03	9	22,22	78.653(4824)	0.008(25)	138.792(7323)	0.004(47)	0.013(47)	0.002(46)
cars-2p	10	47,26	40.428(1659)	0.029(32)	0.031(46)	25.208(2739)	0.109(99)	0.442(146)
csm	13	24,25	0.100(53)	0.027(49)	0.078(67)	0.233(91)	0.248(104)	0.005(67)
scheduler-2p	14	21,21	20.589(2395)	0.005(26)	7.393(754)	0.008(45)	0.026(47)	0.002(44)
multipool	18	32,29	0.580(69)	0.020(61)	0.884(91)	7.834(900)	7.957(900)	0.015(92)
consprod	18	52,36	×(≥1008)	0.020(72)	×(≥2097)	×(≥1332)	×(≥1332)	0.013(109)
incdec	32	47,47	2.914(109)	0.047(47)	4.341(93)	×(≥1948)	×(≥1948)	0.061(154)
mesh2x2	32	54,46	5.295(121)	0.040(106)	6.911(158)	1.994(166)	2.541(174)	0.026(159)
bigjava	44	77,75	3.758(174)	0.031(159)	3.989(231)	8.320(232)	7.796(241)	0.120(286)

- 程序分析实验

由于目前尚没有公开可用的基于约束表示的有理数多面体域的实现库,这里选择基于对偶表示的有理数多面体域 NewPolka[14]来对本文的测试程序进行分析以获得精确分析结果,作为本文实验的标准参考.理论上,

基于对偶表示的有理数多面体域的分析结果应与使用强接合的基于约束的有理数多面体域的结果相同.表 2 给出了基于对偶表示的有理数多面体域 NewPolka 的分析结果.其中,对于某些高维测试程序,NewPolka 由于基于有理数实现所带来的时空复杂度问题,不能在规定的时间(1 小时)内完成分析,本文使用“>1h”标记.对于基于约束表示的浮点多面体域 FPPol,表 2 分别给出了基于纯强接合和基于强、弱接合结合策略的分析结果.对于基于强、弱接合结合策略的情形,表 2 还给出了整个程序分析过程中使用强接合和弱接合的次数.

性能.从表 2 可以看出,通常情况下,基于强、弱接合结合策略的多面体分析比基于强接合的多面体分析的效率要高,并且鲁棒性更好.在基于强接合的多面体分析过程中,由于中间产生的某些约束系统的高复杂性,导致 GPLK 出现“数值不稳定”问题,不能消除冗余约束,极大地削弱了分析的效率甚至导致分析无法继续下去(用“×”来标注).在这种基于传播的分析中,之前复杂(约数多、稠密)的结果将会影响之后的分析.在基于强、弱接合结合策略的多面体分析中,当中间过程的某些约束系统比较复杂时,结合策略将选用弱接合操作,在提高单次接合操作效率的同时,还能保证输出结果的简单性,不会影响到后续分析.如第 3 节所述,每次接合操作执行之前,我们都会对输入多面体的复杂度和稀疏程度进行定量评估,以选择合适的接合操作.比如,只要两个输入多面体的稠密度(系数矩阵 A 中非 0 系数个数/矩阵大小)之和超过 0.65 且系数矩阵中非 0 系数个数均大于 45,我们就会选用弱接合操作.而对于某些高维程序(如 bigjava),只要两输入多面体稠密度之和大于 0.1 且系数矩阵中非 0 系数个数均超过 120,我们就会选用弱接合操作.

Table 2 Comparision of strong join-based and weak join-based polyhedra analysis

表 2 基于强接合的多面体分析和基于弱接合的多面体分析比较

Programs	Number of variables	Widening delay parameter	Computation time in seconds (number of iterations)			Precision comparison
			Newpolka: Rational polyhedra based on double description method	FPPol: Floating-Point polyhedra based on only constraints		
				Strong join	Combining strong join and weak join ⟨Number of strong join, Number of weak join⟩	
see-saw	2	9	0.056(11)	6.802(11)	1.089(11) ⟨124,7⟩	=
berkeley	4	9	0.086(12)	0.886(12)	0.862(12) ⟨215,7⟩	=
dragon	5	9	49.123(12)	×	3.779(11) ⟨242,524⟩	≠
heap	5	9	0.372(11)	×	0.529(11) ⟨218,25⟩	>
robot	6	9	0.201(11)	1.097(11)	1.154(11) ⟨98,0⟩	=
lifo	7	9	1.389(11)	×	28.946(12) ⟨260,489⟩	≠
cars	7	5	>1h	×	3.901(6) ⟨2,42⟩	<
barber	8	2	53.899(6)	×	1.528(6) ⟨202,75⟩	>
barberm4-2	8	1	53.612(6)	×	3.127(5) ⟨330,5⟩	≠
swim-pool-1	9	9	1.553(11)	17.817(11)	15.471(11) ⟨667,70⟩	≠
train-rm03	9	9	1.758(11)	×	0.889(4) ⟨154,14⟩	>
cars-2p	10	5	22.753(8)	×	760.495(9) ⟨270,23⟩	≠
csm	13	9	0.964(11)	6.940(11)	6.742(11) ⟨624,0⟩	≠
scheduler-2p	14	9	1.418(12)	×	67.442(12) ⟨596,6⟩	=
multipool	18	4	>1h	16.801(6)	16.499(6) ⟨431,0⟩	<
consprod	18	6	174.451(8)	×	86.339(8) ⟨323,367⟩	≠
incdec	32	3	>1h	×	208.307(6) ⟨233,670⟩	<
mesh2x2	32	5	>1h	×	88.029(7) ⟨377,640⟩	<
bigjava	44	3	>1h	×	173.577(6) ⟨904,106⟩	<

精度.由于对于某些测试程序,基于强接合的多面体分析失效,本文把使用基于对偶表示的有理数多面体域 NewPolka 的分析结果作为标准参考,并与使用浮点多面体 FPPol 基于强、弱接合结合策略所得结果不变式进行了比较.比较结果见表 2“精度比较”栏:“=”表示两种方法分析结果一样,“>”表示基于 NewPolka 所得不变式更强,“<”表示基于 FPPol 所得不变式更强,“≠”表示两种方法分析结果不可比.通过比较和分析所得不变式我们发现对于某些测试程序,基于强、弱接合的结合策略所产生的不变式不弱于甚至强于基于 NewPolka 产生的不变式.因为在多面体分析中,分析的主要精度损失源于加宽算子.理论上,由于加宽算子的非单调性,基于强、弱接合的结合策略所产生的不变式不一定弱于基于强接合的分析.

在多面体分析中,一种常用的策略是延迟加宽算子的策略,以尽量减少加宽算子带来的精度损失.但是,这种延迟策略可能导致分析过程中,中间产生的约束表示越来越复杂,接合操作执行代价越来越高,并且由于线性

规划工具的使用,分析的鲁棒性可能越来越差.此时,弱接合的使用将是一个很好的选择:虽然每次弱接合操作应用可能带来一些精度损失,但至少可以保证分析的成功完成,而且最终分析结果的精度不一定差.

6 结 论

本文为基于约束的多面体抽象域设计并实现了一系列可靠的、低计算代价的弱接合操作,作为计算代价昂贵的强接合(凸闭包)的替代候选,以提高多面体分析的执行效率和可扩展性.在此基础上,还提出了一种启发式的结合策略,能够在多面体分析过程中动态地选择合适的接合操作,把多种弱接合操作和强接合操作有机地结合起来,在分析的效率和精度之间取得权衡.实验结果表明,在基于多面体的程序分析中,这些弱接合操作的应用不仅能够有效地提高分析的效率,而且能够提高基于约束的多面体抽象域的易处理性和鲁棒性.

本文将来的工作将针对基于对偶表示的多面体抽象域中高复杂度的对偶转化问题开展研究,旨在保证可靠性的前提下通过上近似方法降低对偶转化方法的复杂度,并消除对偶转化的潜在的指数级输出问题.

致谢 在本文工作的研究过程中,得到了 Antoine Mine 和 Axel Simon 的大力帮助和讨论,在此表示感谢.

References:

[1] Cousot P, Cousot R. Abstract interpretation: A unified lattice model for static analysis of programs by construction or approximation of fixpoints. In: Proc. of the 4th POPL. New York: ACM Press, 1977. 238−252.

[2] Cousot P. The verification grand challenge and abstract interpretation. In: Meyer B, Woodcock J, eds. Verified Software: Tools, Theories, Experiments (VSTTE 2005). LNCS 4171, Berlin: Springer-Verlag, 2008. 189−201.

[3] Miné A. Weakly relational numerical abstract domains [Ph.D. Thesis]. Paris: Ecole Normale Superieure, 2004.

[4] Li MJ, Li ZJ, Chen HW. Program verification techniques based on the abstract interpretation theory. Journal of Software, 2008, 19(1):17−26 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/19/17.htm [doi: 10.3724/SP.J.1001.2008. 0017]

[5] Cousot P, Cousot R. Static determination of dynamic properties of programs. In: Robinet B, ed. Proc. of the 2nd Int'l Symp. on Programming. Paris: Dunod, 1976. 106−130.

[6] Cousot P, Halbwachs N. Automatic discovery of linear restraints among variables of a program. In: Proc. of the 5th POPL. New York: ACM Press, 1978. 84−97.

[7] Miné A. The octagon abstract domain. Higher-Order and Symbolic Computation, 2006,19(1):31−100.

[8] Simon A, King A, Howe JM. Two variables per linear inequality as an abstract domain. In: Leuschel M, ed. Proc. of the LOPSTR 2002. LNCS 2664, Berlin: Springer-Verlag, 2003. 71−89.

[9] Sankaranarayanan S, Sipma H, Manna Z. Scalable analysis of linear systems using mathematical programming. In: Cousot R, ed. Proc. of the VMCAI 2005. LNCS 3385, Berlin: Springer-Verlag, 2005. 25−41.

[10] Laviron V, Logozzo F. SubPolyhedra: A (more) scalable approach to infer linear inequalities. In: Jones N, Muller-Olm M, eds. Proc. of the VMCAI 2009. LNCS 5403, Berlin: Springer-Verlag, 2009. 229−244.

[11] Bagnara R, Hill PM, Zaffanella E. Applications of polyhedral computations to the analysis and verification of hardware and software systems. Theoretical Computer Science, 2009,410(46):4672−4691.

[12] Bagnara R, Hill PM, Zaffanella E. The parma polyhedra library: Toward a complete set of numerical abstractions for the analysis and verification of hardware and software systems. Science of Computer Programming, 2008,72(1-2):3−21.

[13] Loechner V. PolyLib. 1997. http://icps.u-strasbg.fr/~loechner/polylib/

[14] Jeannet B. NewPolka. 2006. http://pop-art.inrialpes.fr/people/bjeannet/newpolka/

[15] Jeannet B, Miné A. Apron: A library of numerical abstract domains for static analysis. In: Bouajjani A, Maler O, eds. Proc. of the CAV 2009. LNCS 5643, Berlin: Springer-Verlag, 2009. 661−667.

[16] Que DN. Robust and generic abstract domain for static program analysis: The polyhedral case. Technical Report, E290, Ecole des Mines de Paris, 2006.

[17] Le Verge H. A note on Chernikova's algorithm. Technical Report, 635, IRISA, 1992.

[18] Simon A, King A. Exploiting sparsity in polyhedral analysis. In: Hankin C, Siveroni I, eds. Proc. of the 12th Int'l Static Analysis Symp. (SAS 2005). LNCS 3672, Berlin: Springer-Verlag, 2005. 336−351.

[19] Chen LQ, Miné A, Cousot P. A sound floating-point polyhedra abstract domain. In: Ramalingam G, ed. Proc. of the 6th Asian Symp. on Programming Languages and Systems (APLAS 2008). LNCS 5356, Berlin: Springer-Verlag, 2008. 3−18.

[20] Imbert JL. About redundant inequalities generated by Fourier's algorithm. In: Jorrand P, Sgurev V, eds. Proc. of the 4th Int'l Conf. on Artificial Intelligence: Methodology, Systems, Applications (AIMSA'90). Amsterdam: North-Holland, 1990. 117−127.

[21] Sankaranarayanan S, Colón MA, Sipma H, Manna Z. Efficient strongly relational polyhedral analysis. In: Emerson EA, S.Namjoshi K, eds. Proc. of the VMCAI 2006. LNCS 3855, Berlin: Springer-Verlag, 2006. 111−125.

[22] Makhorin A. The GNU linear programming kit. 2000. http://www.gnu.org/software/glpk/

[23] Lalire G, Argoud M, Jeannet B. Interproc. 2007. http://pop-art.inrialpes.fr/people/bjeannet/bjeannet-forge/interproc/

[24] Sankaranarayanan S, Sipma HB, Manna Z. Constraint-Based linear-relations analysis. In: Giacobazzi R, ed. Proc. of the 11th Int'l Static Analysis Symp. (SAS 2004). LNCS 3148, Berlin: Springer-Verlag, 2004. 53−68.

附中文参考文献:

[4] 李梦君,李舟军,陈火旺.基于抽象解释理论的程序验证技术.软件学报,2008,19(1):17−26. http://www.jos.org.cn/1000-9825/19/17.htm [doi: 10.3724/SP.J.1001.2008.0017]

陈立前(1982－),男,湖南茶陵人,博士,主要研究领域为程序分析与验证,抽象解释.

刘万伟(1980－),男,博士,CCF 会员,主要研究领域为模型检验,定理证明.

王戟(1969－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为高可信软件技术,软件方法学,软件工程.