基于区间线性模版约束的程序分析

姜加红 尹帮虎 陈立前
(国防科技大学计算机学院 长沙 410073)

摘 要 抽象解释是一种对程序语义进行可靠近似的通用理论,该理论在保证可靠性的前提下,可为程序变量的值范围分析提供一个通用的框架.抽象域是抽象解释框架的核心,在该框架下面向数值性质分析的数值抽象域得到了广泛的关注.其中,模板多面体抽象域的表达能力覆盖了程序分析中常用的弱关系型抽象域,如区间抽象域、八边形抽象域等.该文对经典的基于线性模版约束的模版多面体抽象域进行扩展,以支持区间线性模版约束,从而得到一个新的数值抽象域——区间线性模版约束抽象域,可以用来推导变量间形如"$\sum_k [a_k,b_k]x_k \leq c$"的区间线性不等式关系(其中区间系数$[a_k,b_k]$为分析前预先确定的常量).该抽象域采用"弱解"作为区间线性约束的解语义,可以表达某类非凸(甚至非联通)性质,因而比经典的模版多面体抽象域表达能力更强.区间线性模板约束抽象域的域元素可以看作是一系列模版多面体的析取,几何上,其域元素与每个象限的交均是一个凸的模板多面体(可以为空).该文给出了区间线性模版约束抽象域的域表示和域操作,基于该抽象域的静态分析算法主要基于区间线性规划来实现.进一步,该文讨论了基于区间线性模版约束抽象域中区间线性模版的生成方法.最后,该文在开源数值抽象库 APRON 中实现了区间线性模版约束抽象域,并开展了程序分析实验.初步的实验结果表明区间线性模版约束抽象域可以有效地分析程序中的析取行为.

关键词 抽象解释;区间线性模版约束;区间线性规划;程序静态分析;非凸性质
中图法分类号 TP301 **DOI 号** 10.11897/SP.J.1016.2018.00545

Program Analysis Based on Interval Linear Template Constraints

JIANG Jia-Hong YIN Bang-Hu CHEN Li-Qian
(*School of Computer Science, National University of Defense Technology, Changsha* 410073)

Abstract The problem of automatically inferring numerical invariants in a program has received wide attention in the analysis and verification of programs. Abstract interpretation is a general theory to soundly approximate program semantics. It provides a general framework to analyze value ranges of program variables, guaranteeing the soundness of the analysis. Abstract domain is key to the framework of abstract interpretation, which achieves a trade-off between efficiency and precision, and especially various numerical abstract domains have been proposed under this framework. In particular, the expressiveness of the template constraint matrix domain (TCM) subsumes most weakly relational abstract domains that are commonly used in practical program analysis, for example, interval abstract domain ($a \leq x \leq b$), octagon abstract domain ($\pm x \pm y \leq c$), etc. During the analysis and verification of real-life systems, due to uncertainty, the application data in the model or program may not be known exactly, which is then often provided or modelled in terms of intervals. Moreover, in practice, floating-point arithmetic and non-linear expressions

收稿日期:2016-11-02;在线出版日期:2017-09-08.本课题得到国家"九七三"重点基础研究发展规划项目基金(2014CB340703)、国家自然科学基金(61532007)、上海市高可信计算重点实验室开放基金(07dz22304201504)资助.**姜加红**,男,1989 年生,博士研究生,主要研究方向为抽象解释、SMT. E-mail: jhjiang@nudt.edu.cn. **尹帮虎**,男,1989 年生,博士研究生,主要研究方向为抽象解释、程序分析. **陈立前**(通信作者),男,1982 年生,博士,助理研究员,中国计算机学会(CCF)会员,主要研究方向为程序分析与验证、形式化方法. E-mail: lqchen@nudt.edu.cn.

are often abstracted into linear expressions with interval coefficients. In other words, interval coefficients appear naturally during program analysis in practice. Hence, abstract domains that can infer interval linear relationships among variables are desired. This paper extends classical template constraint matrix domain which is based on linear template constraints, to support interval linear template constraints, and proposes a new numerical domain—interval template constraint matrix domain (itvTCM), which could infer interval linear inequality relations among variables in the program in the form of "$\sum_k [a_k, b_k] x_k \leq c$" (where the interval coefficient $[a_k, b_k]$ is determined before analysis). itvTCM makes use of "weak solution" as the semantics of the solution of interval linear constraints, which could represent certain non-convex (even non-connected) properties, and thus it is more expressive than TCM. Each itvTCM element could be considered as a disjunction of multiple TCMs but without using any explicit disjunctive operations. From the geometric point of view, each itvTCM element maps each orthant to a convex polyhedron (maybe an empty polyhedron). This paper provides domain representation and domain operations (such as join, meet, widening, etc.) of itvTCM, and most domain operations of itvTCM are implemented based on interval linear programming. Theoretically, the complexity of some domain operations of itvTCM is at worst exponential of that of the corresponding domain operations in classic TCM. However, in practice, we could alleviate this problem through restricting the number of interval coefficients. In this paper, we also discuss how to generate templates for itvTCM. Finally, we have implemented itvTCM in the numerical abstract domain library APRON, and conducted experiments. The preliminary experimental results show that itvTCM is useful to capture disjunctive behaviors of a program.

Keywords abstract interpretation; interval linear template constraint; interval linear programming; program static analysis; non-convex property

1 引 言

基于抽象解释的程序分析的精度很大程度上取决于所选取的抽象域[1]. 多面体抽象域是目前表达能力最强、应用最为广泛的数值抽象域之一[2]. 但是由于其高复杂度,在许多实际应用中多面体抽象域在可扩展性方面存在限制. 为了降低多面体抽象域的复杂度,同时又能推导出实际的线性不变式,Sankaranarayanan 等人提出了模版多面体抽象域(Template Constraint Matrix domain,简称 TCM)[3-4],采用形如"$\boldsymbol{A} \cdot \boldsymbol{x} \leq \boldsymbol{b}$"的一组线性不等式,其中系数矩阵 $\boldsymbol{A}$ 在分析前预先确定,$\boldsymbol{x}$ 为环境中变量构成的向量,右值的约束常数向量 $\boldsymbol{b}$ 由分析时自动推导得到[4]. 但是,模版多面体抽象域的域操作通过线性规划来实现,其具有多项式时间复杂度,并且模版多面体抽象域的表达能力涵盖了目前实际静态分析中常用的弱关系型线性抽象域(如八边形抽象[5]、八面体抽象域[6]等). 所以由于模版多面体表达能力的代表性及其多项式的时间复杂度,模版多面体抽象域自提出以来一直受到了学术界的广泛关注.

当前大多数数值抽象域(如区间抽象域、八边形抽象域、模版多面体抽象域等)均是基于一系列线性表达式的合取,对应的几何图形区域都是凸的,因此只能表达凸性质. 而实际分析中,程序的行为在具体语义或收集语义下一般都是非凸的. 比如,程序中经常使用"if-then-else"语句来进行分情况讨论. 另外,用户关心程序的非凸数值性质,如检查程序中没有"除零错"需要验证形如"$x \neq 0$"这一非凸性质. 如基于凸模版多面体抽象域对图 1 中的程序进行分析时,由于第 1～5 行的程序对应的行为是析取行为,

```
1.  if (brandom()){
2.      y=2*x+1;
3.  }else{
4.      y=3*x+2;
5.  }
6.  if(y>3){
7.      z=1/x;
8.  }
```

图 1 程序示例 program1

基于凸抽象域分析会得到 top(即为全集)，导致第 7 行的分析不精确，从而会导致“除零错”误报.

针对线性模版约束抽象域的凸性局限性，本文利用区间线性代数中区间线性约束的非凸性，来对线性模版约束抽象域进行扩展以支持区间线性约束不变式的自动推导. 以图 1 中示例程序为例，假定变量构成的向量为$[x,y]^{\mathrm{T}}$，系数矩阵设定为$[[2,3],[-1,-1]]$，则在第 6 行行头(即 if-then-else 语句结束处)可以得到程序的不变式为$[2,3]x+[-1,-1]y\leq -1\wedge[2,3]x+[-1,-1]y\geq -2$，因此在分析第 7 行的赋值语句之前，我们可以得到$[2,3]x>1$，得到$x>0$，从而消除第 7 行的“除零错”误报.

本文将基于区间线性代数提出一个新的抽象域，区间线性模版约束抽象域(Interval Linear Template Constraint Matrix Domain，简称 itvTCM). 区间线性模版约束抽象域可以看作经典模版多面体抽象域的区间线性扩展版本. 区间线性模版约束抽象域可以表达变量之间形如“$\sum_k [a_k,b_k]x_k\leq c$”的区间线性不等式关系. 与模版多面体中类似，区间线性模版约束中变量的区间系数是固定的. 基于区间线性不等式系统的弱解语义[7]，区间线性模版约束抽象域能够天然地表达某类非凸性质. 本质上，区间线性模版约束抽象域的域元素可以看作多个模版多面体的析取. 因此，与经典模版多面体抽象域相比，区间线性模版约束抽象域可以天然地刻画非凸信息. 进一步，本文将讨论区间线性模版的生成策略，以及如何基于区间线性模版约束抽象域对程序进行静态分析.

最后，本文在抽象解释框架下设计并实现了一个分析工具原型，并进行实验，结果表明区间线性模版约束抽象域可以有效地分析程序的非凸性质.

本文第 2 节介绍本文需要的预备知识；第 3 节给出抽象域的域表示和域操作；第 4 节阐述区间线性模版的生成策略；第 5 节描述本文的工具原型的实现并给出初步的实验结果；第 6 节介绍本文的相关工作；第 7 节对本文做总结并对下一步工作给予展望.

2　预备知识

本节简单介绍抽象解释的基本理论、经典区间抽象域、模版多面体抽象域、区间线性代数以及区间线性规划的相关内容.

2.1　抽象解释理论

2.1.1　Galois 连接

在基于区间抽象域的静态分析是用一个区间来刻画变量值范围的最大和最小值. 抽象解释理论是通过一个 Galois 连接来表示具体域和抽象域之间的映射关系.

设$\langle D,\sqsubseteq\rangle$和$\langle D^{\sharp},\sqsubseteq^{\sharp}\rangle$是两个给定的偏序集，其中$\langle D,\sqsubseteq\rangle$为定义在具体域上的偏序集合，$\langle D^{\sharp},\sqsubseteq^{\sharp}\rangle$为定义在抽象域上的偏序集合，函数$\alpha:D\to D^{\sharp}$及$\gamma:D\to D^{\sharp}$构成的函数对$(\alpha,\gamma)$称为 D 与 $D^{\sharp}$ 之间的 Galois 连接，iff. $\forall x\in D,x^{\sharp}\in D^{\sharp},\alpha(x)\sqsubseteq^{\sharp}x^{\sharp}\Leftrightarrow x\sqsubseteq\gamma(x^{\sharp})$，其中函数 α 为抽象化算子，称 γ 为具体化算子. 抽象域与具体域之间的关系如图 2 所示.

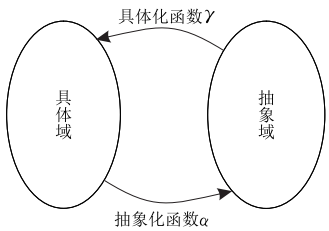

图 2　Galois 连接示意图

由定义中性质$\alpha(x)\sqsubseteq^{\sharp}x^{\sharp}$(亦即$x\sqsubseteq\gamma(x^{\sharp})$)可知，$x^{\sharp}$是 x 的可靠抽象，且$\alpha(x)$是 x 在 $D^{\sharp}$ 中最精确的可靠近似. 在基于抽象解释框架的程序分析中，将程序的集合语义作为具体偏序集合，抽象语义作为抽象偏序集合，通过在抽象域上的迭代计算来对待分析程序的实际行为进行上近似.

对于给定的 Galois 连接$\langle D,\sqsubseteq\rangle$和$\langle D^{\sharp},\sqsubseteq^{\sharp}\rangle$，抽象域 $D^{\sharp}$ 上的函数 $f^{\sharp}$，具体域 D 上的函数 f，当$\forall x^{\sharp},(\alpha\circ f\circ\gamma)(x^{\sharp})\sqsubseteq^{\sharp}f^{\sharp}(x^{\sharp})$，我们称 $f^{\sharp}$ 是 f 的可靠抽象. 若$f^{\sharp}=\alpha\circ f\circ\gamma$，则称 $f^{\sharp}$ 为 f 的最佳抽象；若$\gamma\circ f^{\sharp}=f\circ\gamma$，则称 $f^{\sharp}$ 为 f 的精确抽象. 在给定 Galois 连接的前提下，f 的最佳抽象总是存在的，但其精确抽象不一定存在.

2.1.2　不动点理论

设$\langle D,\sqsubseteq,\sqcup,\sqcap,\bot,\top\rangle$，$\langle D^{\sharp},\sqsubseteq^{\sharp},\sqcup^{\sharp},\sqcap^{\sharp},\bot^{\sharp},\top^{\sharp}\rangle$为两个完备格，函数对$(\alpha,\gamma)$为两者之间的 Galois 连接，$f$ 和 $f^{\sharp}$ 分别为两个完备格上的单调函数，若 $f^{\sharp}$ 为 f 的可靠抽象，则$\alpha(LFP(f))\sqsubseteq^{\sharp}LFP(f^{\sharp})$，亦即$LFP(f)\sqsubseteq\gamma(LFP(f^{\sharp}))$.

假设完备格$\langle D^{\sharp},\sqsubseteq^{\sharp},\sqcup^{\sharp},\sqcap^{\sharp},\bot^{\sharp},\top^{\sharp}\rangle$上没有无穷递增链，若 $f^{\sharp}$ 为 $D^{\sharp}$ 上的单调函数，$x_0^{\sharp}$ 是 $f^{\sharp}$ 的前向不动点，则迭代过程$x_{i+1}^{\sharp}=f^{\sharp}(x_i^{\sharp})$将在

有穷步终止.这种迭代序列被称为 Kleene 迭代序列.Kleene 迭代是抽象解释框架下求解程序不动点的一个常用方法.抽象解释框架确保了在抽象语义下求得的程序的不动点是实际语义不动点的上近似,即迭代过程的可靠性.

2.1.3 加宽算子

当完备格上没有无穷递增链时,Kleene 迭代确保了迭代过程的终止性.但完备格上的递增链长度很大甚至无穷,当前的迭代过程可能时间消耗过大甚至不能终止.接下来,本文介绍如何确保含无穷递增链的完备格上 Kleene 迭代过程的终止性,以及如何加快 Kleene 迭代过程的收敛速度.

设$\langle D^{\sharp},\sqsubseteq^{\sharp}\rangle$是一个偏序集,二元操作符$\triangledown$是加宽算子(Widening operator)的充要条件为:

(1) $\forall a_1,a_2\in D^{\sharp}$, $a_1\sqcup a_2\sqsubseteq a_1\triangledown a_2$;

(2)对于所有递增序列$\{a_n\}$,序列$\{a_n^{\triangledown}\}$最终收敛,其中$\{a_n^{\triangledown}\}$定义为:如果$n=0$, $a_n^{\triangledown}=a_n$,否则$a_n^{\triangledown}=(a_{n-1}^{\triangledown})\triangledown a_n$.

抽象解释的框架中,通过加宽算子对抽象语义对上界逼近,从而确保抽象域上的迭代过程在有穷步内收敛.加宽操作是对迭代趋势的一个可靠猜测,能够确保有穷步内迭代的收敛,但可能会带来精度损失.

2.2 经典区间抽象域

基于区间抽象域的静态分析在所有程序点为每一个变量维护一个区间,用以刻画变量在所有执行中的最大值和最小值信息.基于抽象解释理论,实数域(具体域)与区间域(抽象域)之间的映射关系可以通过如下的 Galois 连接来刻画,即

$$(\wp(\mathbb{R}),\leq)\underset{\alpha_i}{\overset{\gamma_i}{\rightleftarrows}}(ItvSet,\subseteq_i).$$

其中实数域上的偏序关系直接使用小于等于关系;$ItvSet$ 是实数 $\mathbb{R}$ 上的区间集合$\{[a,b]\mid a\in\mathbb{R}\cup\{-\infty\},b\in\mathbb{R}\cup\{-\infty\},a\leq b\}\cup\{\bot_i\}$,其中$\bot_i$表示空区间. $ItvSet$ 上的偏序关系$\subseteq_i$定义为:对于 $I_1=[a,b]$, $I_2=[a',b']$, $I_1\subseteq_i I_2$当且仅当$a\geq a'\wedge b\leq b'$或者$I_1=\bot_i$.

具体化函数γ_i负责将一个区间($I=[a,b]$)映射成一个实数集合,其定义为:$\gamma_i(I)\triangleq I=\bot_i?\varnothing:\{x\in R\mid a\leq x\leq b\}$;抽象函数$\alpha_i$负责将一个实数集合($S$)映射成一个区间,其定义为$\alpha_i(S)\triangleq S=\varnothing?\bot_i:[\min(S),\max(S)]$.如图 3 所示,变量 x 和 y 在实际执行中对应的取值为图 3 中标注的点,则其在区间抽象域上的域表示分别为$[x_0,x_1]$和$[y_0,y_1]$.

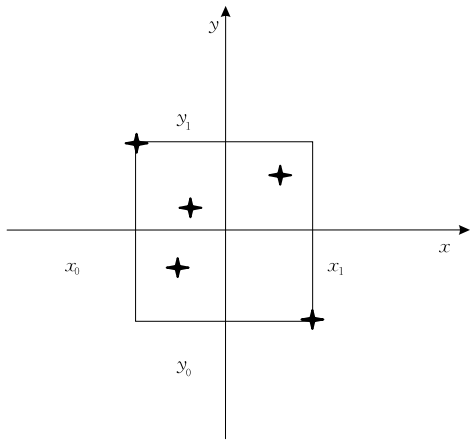

图 3 区间抽象域示例图

一个抽象域的基本域操作主要有交操作、接合操作、算术操作、加宽操作等.下面介绍区间抽象域的一些基本域操作($I_1=[a,b]$, $I_2=[a',b']$):

(1)交操作(Meet):

$$I_1\bigcap\nolimits_i I_2\triangleq\begin{cases}[\max(a,a'),\min(b,b')], & \text{若}\max(a,a')\leq\min(b,b')\\ \bot_i, & \text{否则}\end{cases}.$$

(2)接合操作(Join):

$$I_1\bigcup\nolimits_i I_2\triangleq\begin{cases}I_1, & \text{若 } I_2=\bot_i\\ I_2, & \text{若 } I_1=\bot_i\\ [\min(a,a'),\max(b,b')], & \text{否则}\end{cases}.$$

(3)区间算术(Arithmetic):

$I_1+_i I_2\triangleq[a+a',b+b']$;

$I_1-_i I_2\triangleq[a-b',b'-a']$;

$I_1\times_i I_2\triangleq[\min\{a\times a',a\times b',b\times a',b\times b'\},\ \max\{a\times a',a\times b',b\times a',b\times b'\}]$.

(4)加宽操作(Widening):

$$I_1\triangledown_i I_2\triangleq\begin{cases}I_1, & \text{若 } I_2=\bot_i\\ I_2, & \text{若 } I_1=\bot_i\\ [a\leq a'?a:-\infty,b\geq b'?b:+\infty], & \text{否则}\end{cases}.$$

(5)迁移函数(Translation Function):

对于赋值语句$var:=expr$,在抽象环境$X^{\sharp}$下,其赋值迁移函数定义为$[\![var:=expr]\!]^{\sharp}X^{\sharp}\triangleq X^{\sharp}[var\mapsto[\![expr]\!]^{\sharp}X^{\sharp}]$,其中,$[\![expr]\!]^{\sharp}X^{\sharp}$为表达式$expr$当前环境$X^{\sharp}$下,通过区间算术运算得到的抽象语义值.

对于测试迁移语句,设$X^{\sharp}(x)=[a,b]$,则有

$$[\![x\leq c]\!]^{\sharp}X^{\sharp}\triangleq\begin{cases}\bot_i, & \text{若}a>c\\ X^{\sharp}[x\mapsto[a,\min\{b,c\}]], & \text{否则}\end{cases}$$

和

$[\![x\geq c]\!]^\sharp X^\sharp \triangleq \begin{cases}\perp_i, & \text{若 } b<c\\ X^\sharp[x\mapsto[a,\min\{b,c\}]], & \text{否则}\end{cases}$，

其中 $x\leq c$ 和 $x\geq c$ 是基本约束，任意形式的约束可以抽象成一个或多个这样的基本约束.

2.3　模版多面体抽象域

模版多面体抽象域由 Sankaranarayanan 等人首先提出，用以刻画变量在给定模版下的线性约束关系[4]. 模版抽象域是域表示形如“$\boldsymbol{A}\cdot\boldsymbol{x}\leq\boldsymbol{b}$”的凸抽象域，其中 $\boldsymbol{A}$ 是一个 $m\times n$ 的系数矩阵(分析前确定)，$\boldsymbol{x}$ 是一个 $n\times 1$ 列向量(由当前分析中的变量环境确定)，$\boldsymbol{b}$ 是抽象域表示的右值约束变量，在分析中计算得到. 如图 4 所示是一个模版多面体示例，对应的系数矩阵为 $\begin{bmatrix}1&0\\-1&0\\1&-1\\-1&1\end{bmatrix}$.

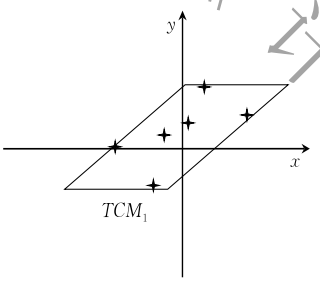

图 4　线性模版多面体示例图

系数矩阵的每一行对应一条线性模版约束，因此模版多面体抽象域可以看作是多条线性模版约束的合取. 一条线性模版约束是指形如“$\boldsymbol{t}\cdot\boldsymbol{x}\leq b$”的约束(其中 $\boldsymbol{t}$ 是一个常数行向量，即一个模版，b 为约束值). 如图 5 所示，线性约束 l_0，l_1，l_2(彼此平行)为给定线性模版下的一组约束，图中打点处为程序变量实际取值，则线性约束 l_0，l_1 为程序变量之间约束关系的上近似，l_1 为该模版下的最佳上近似.

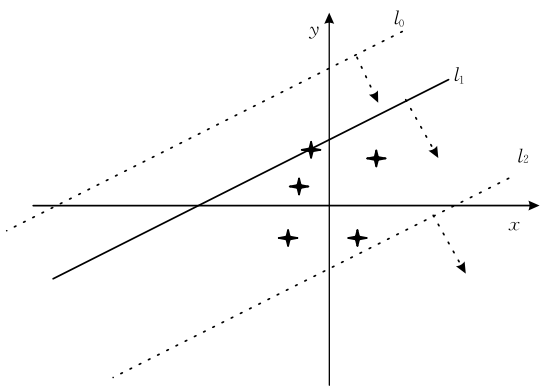

图 5　单模版线性约束示例图

经典的数值抽象域，如区间抽象域、八边形抽象域等均可以看作是模版多面体抽象域在设置特定模版下的一种特殊表示. 若设定模版多面体的系数矩阵为 $\begin{bmatrix}1&0\\-1&0\\0&1\\0&-1\end{bmatrix}$，则可求得变量 x 在区间抽象域中的域表示为“$x\in[x_0,x_1]$，$y\in[y_0,y_1]$”. 类似地，八边形抽象域对应模版多面体的系数矩阵为

$$\begin{bmatrix}1&1\\1&-1\\-1&1\\-1&-1\\1&0\\-1&0\\0&1\\0&-1\end{bmatrix}.$$

2.4　区间线性不等式系统

设 $\boldsymbol{IA}\in\mathbb{IR}^{m\times n}$ 为一个 $m\times n$ 的区间矩阵，矩阵中每一个元素均是一个区间 $itv_{i,j}(1\leq i\leq m,1\leq j\leq n)$；$\boldsymbol{A}\in\mathbb{R}^{m\times n}$ 为一个 $m\times n$ 的实数矩阵，矩阵中每一个元素均是一个实数 $a_{i,j}(1\leq i\leq m,1\leq j\leq n)$. 若 $\forall i,j.1\leq i\leq m,1\leq j\leq n$，有 $a_{i,j}\in itv_{i,j}$，我们称 $\boldsymbol{A}\in\boldsymbol{IA}$.

令 $\boldsymbol{b}\in\mathbb{R}^m$ 为一个 m 维的常数向量，则我们称 $\boldsymbol{IA}\cdot\boldsymbol{x}\leq\boldsymbol{b}$ 为一个区间线性不等式系统. 区间线性不等式系统 $\boldsymbol{IA}\cdot\boldsymbol{x}\leq\boldsymbol{b}$ 表示所有线性不等式系统 $\boldsymbol{A}\cdot\boldsymbol{x}\leq\boldsymbol{b}$，其中 $\boldsymbol{A}\in\boldsymbol{IA}$.

在区间线性不等式系统中，若 $\exists\boldsymbol{A}\in\boldsymbol{IA}$ 使得 $\boldsymbol{A}\cdot\boldsymbol{x}\leq\boldsymbol{b}$，则我们称向量 $\boldsymbol{x}$ 为区间线性不等式系统 $\boldsymbol{IA}\cdot\boldsymbol{x}\leq\boldsymbol{b}$ 的一个弱解[7]. 并且，我们称集合 $\{x\in\mathbb{R}^n:\exists\boldsymbol{A}\in\boldsymbol{IA}.\ \boldsymbol{A}\cdot\boldsymbol{x}\leq\boldsymbol{b}\}$ 为区间线性不等式系统 $\boldsymbol{IA}\cdot\boldsymbol{x}\leq\boldsymbol{b}$ 的弱解集合.

总体来说，弱解集合可以是非凸的，甚至是非连通的. 一个象限是 n 维欧几里德空间中 2^n 子集中的一个，通过限制每个笛卡尔坐标轴为非负或者非正. 在一个给定的象限内，变量保持了确定的符号，因此区间系数可以转化为一个常数系数(区间的上界或者下界). 弱解集合与每个象限的交可以通过一个凸多面体来表示，一个 n 维区间不等式系统最差情况下需要 2^n 个不同的凸多面体来表达.

2.5　区间线性规划

设 $\boldsymbol{IA}\in\mathbb{IR}^{m\times n}$ 为一个 $m\times n$ 的区间矩阵，$\boldsymbol{b}\in\mathbb{R}^m$ 为一个 m 维的约束向量，$\boldsymbol{IC}\in\mathbb{IR}^n$ 为一个 n 维的区间向量. 则我们称如下一族线性规划问题 $f(\boldsymbol{IA},\boldsymbol{b},\boldsymbol{IC})=\max\{\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x},\ \text{s. t. }\boldsymbol{A}\cdot\boldsymbol{x}\leq\boldsymbol{b}\}$ 为一个区间线性规

划问题,其中 $\boldsymbol{A}\in\boldsymbol{IA},\boldsymbol{c}\in\boldsymbol{IC}$[8]. 一种求解区间线性规划问题的简单方法是将该问题分解为 2^n 个基本线性规划问题,即对每个变量的正负值进行讨论,每个象限对应一个线性规划问题,最后再将这 2^n 个线性规划问题的结果进行综合,得到整个区间线性规划问题的结果.

例 1. 考虑如下区间线性规划问题,设变量 $\{x,y\}$ 满足区间线性约束 $[[-1,1],[2,3]]\begin{bmatrix}x\\y\end{bmatrix}\leq 1$,求解 $\max\{3x+5y\}$.

根据求解区间线性规划的思路,我们对变量的正负值进行分类讨论,将区间多面体投影到每一个象限中去. 当 $x>0,y>0$ 时,$-x<x,2y<3y$,故将区间线性约束 $[[-1,1],[2,3]]\begin{bmatrix}x\\y\end{bmatrix}\leq 1$ 转化为 $[-1,2]\begin{bmatrix}x\\y\end{bmatrix}\leq 1$;当 $x>0,y\leq 0$ 时,$-x<x,2y>3y$,故将区间线性约束 $[[-1,1],[2,3]]\begin{bmatrix}x\\y\end{bmatrix}\leq 1$ 转化为 $[-1,3]\begin{bmatrix}x\\y\end{bmatrix}\leq 1$;当 $x\leq 0,y>0$ 时,$-x>x,2y<3y$,故将区间线性约束 $[[-1,1],[2,3]]\begin{bmatrix}x\\y\end{bmatrix}\leq 1$ 转化为 $[1,2]\begin{bmatrix}x\\y\end{bmatrix}\leq 1$;当 $x\leq 0,y\leq 0$ 时,$-x>x,2y>3y$,故将区间线性约束 $[[-1,1],[2,3]]\begin{bmatrix}x\\y\end{bmatrix}\leq 1$ 转化为 $[1,3]\begin{bmatrix}x\\y\end{bmatrix}\leq 1$. 这样我们将一个区间线性规划问题的求解转化为 4 个基本的线性规划问题的求解.

当然,这种思路存在很多的优化策略. 实际中,文献[8]根据语法将区间线性规划问题分成了几个类型,然后根据不同的类型设计不同的求解算法.

3 区间线性模版约束抽象域

本节将介绍区间线性模版约束抽象域的域表示和域操作. 在此之前,首先回顾一下模版多面体抽象域的设计. 模版多面体抽象域的域表示可以由一个线性不等式组来表示,记作 $\boldsymbol{A}\cdot\boldsymbol{x}\leq\boldsymbol{b}$,其中,$\boldsymbol{x}$ 是一个由程序变量构成的一个 n 维向量(n 为环境中变量的维数),$\boldsymbol{A}$ 为一个 $m\times n$ 的系数矩阵(矩阵的元素为常量,在分析前预先给定),约束向量 $\boldsymbol{b}$ 是一个 m 维向量(向量 $\boldsymbol{b}$ 的每一个元素决定每一行的约束上界,将由分析自动推导出来). $\mathbb{R}^n$ 内所有满足不等式组 $\boldsymbol{A}\cdot\boldsymbol{x}\leq\boldsymbol{b}$ 的点构成的集合称为对应的抽象状态的具体化语义. 模版确定了系数矩阵元素的取值. 当模版确定后,不同的约束向量 $\boldsymbol{b}$ 的值决定了模版多面体抽象域的不同抽象状态.

例 2. 如图 6 中所示的一个线性模版约束示例,系数矩阵为一个 3×3 的矩阵,$\boldsymbol{x}$ 为一个 3 维向量,约束向量为一个 3 维向量.

$$\begin{bmatrix}1&1&1\\1&2&3\\-1&2&3\end{bmatrix}\begin{bmatrix}x_1\\x_2\\x_3\end{bmatrix}\leq\begin{bmatrix}c_1\\c_2\\c_3\end{bmatrix}$$

图 6 线性模版约束示例

3.1 区间线性模版约束抽象域的域表示

itvTCM 的域表示基于一个区间线性不等式组. 与 TCM 的主要不同之处在于,itvTCM 对应系数矩阵的元素不是常数而是一个区间. itvTCM 的域表示可以通过一个区间线性不等式系统 $\boldsymbol{IA}\cdot\boldsymbol{x}\leq\boldsymbol{b}$ 来表示,其中 $\boldsymbol{IA}\in\mathbb{IR}^{m\times n}$ 为一个 $m\times n$ 的系数矩阵(矩阵的元素为区间),$\boldsymbol{b}\in\mathbb{R}^m$ 是一个常数向量,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. 其具体语义在代数上对应区间线性不等式系统 $\boldsymbol{IA}\cdot\boldsymbol{x}\leq\boldsymbol{b}$ 的弱解集合. 与 TCM 中类似,itvTCM 对应的区间系数矩阵是在分析前预先确定的.

例 3. 如图 7 中所示的一个区间线性模版约束示例,系数矩阵是一个 3×3 的矩阵,矩阵的每个元素为一个区间,$\boldsymbol{x}$ 为一个 3 维向量,常数向量为一个 3 维向量.

$$\begin{bmatrix}[-1,1]&[1,1]&[1,1]\\[1,1]&[2,2]&[3,3]\\[-1,1]&[2,2]&[3,3]\end{bmatrix}\begin{bmatrix}x_1\\x_2\\x_3\end{bmatrix}\leq\begin{bmatrix}c_1\\c_2\\c_3\end{bmatrix}$$

图 7 区间线性模版约束示例

本文将给定系数矩阵的区间线性不等式系统 $\boldsymbol{IA}\cdot\boldsymbol{x}\leq\boldsymbol{b}$ 对应的弱解集合在几何上的图形区域称为区间模版多面体. 从域表示上来看,区间模版多面体可以看作是经典模版多面体的区间拓展版本,因为经典凸模版多面体仅支持标准的线性不等式(不含区间系数). 不难看出,区间模版多面体具有如下性质:

(1) 非凸性. 一个区间模版多面体可以是非凸的,但它跟 $\mathbb{R}^n$ 上每个象限的交是凸的,甚至一个区间模版多面体可以是非连通的;

(2) 关于交闭合. 两个区间模版多面体的交仍然是一个区间模版多面体;

(3) 关于并封闭. 两个区间模版多面体的并仍然是一个区间模版多面体.

总体而言，一个区间模版多面体在几何上拥有较为复杂的形态. 令程序变量集合为$\{x,y\}$，给定的区间系数矩阵为$[[-1,1],[0,0]]$，则区间模版多面体$[[-1,1],[0,0]]\begin{bmatrix}x\\y\end{bmatrix}\leq 1$对应的集合图形如图 8 中虚线箭头指示区域所示，可以看出此区间模版多面体是非凸的，但其与每个象限的交均是一个凸的多面体.

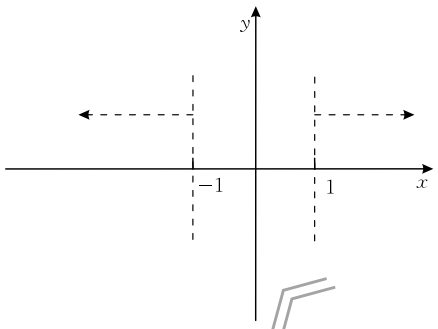

图 8　区间线性模版多面体示例图

本文将区间模版多面体看作是经典模版多面体的简单基数幂[9]：把每个象限映射到一个凸的模版多面体. 这主要基于如下事实：一个区间模版多面体与一个象限的交必然是一个凸的多面体(包含空多面体). 最坏情况下，一个 n 维区间模版多面体可以看作 2^n 个模版多面体的集合并.

3.2　区间线性模版约束抽象域的域操作

令 $itvTCM_1\triangleq \boldsymbol{IA}'\cdot\boldsymbol{x}\leq \boldsymbol{c}'$，$itvTCM_2\triangleq \boldsymbol{IA}''\cdot\boldsymbol{x}\leq \boldsymbol{c}''$，为两个 itvTCM 域元素. 由于其模版一致，故区间系数矩阵相同. 我们设区间系数矩阵 $\boldsymbol{IA}$ 的大小为 $m\times n$. 向量 $\boldsymbol{c}'$，$\boldsymbol{c}''$分别为对应的 m 维约束向量，设 $\boldsymbol{c}'\triangleq[c_1',\cdots,c_m']^{\mathrm{T}}$，$\boldsymbol{c}''\triangleq[c_1'',\cdots,c_m'']^{\mathrm{T}}$. 则区间线性不等式 $\sum_k[\underline{A}_{ik},\overline{A}_{ik}]x_k\leq c\ (1\leq i\leq m,1\leq k\leq n)$ 为 itvTCM 对应的第 i 行约束.

3.2.1　约简操作

在分析过程中，区间线性不等式组中可能产生冗余约束或部分不等式约束的上界不够精确. 为了得到抽象域域表示的规范型，使得域表示唯一，我们设计了约简操作(Reduce). 约简操作的大体思路是：通过区间线性规划对冗余约束进行精化，使得区间线性不等式的右值为最小值. 约简操作的过程中需要对区间不等式组的每一行进行精化，如调用 m 次形如“$\max\left(\sum_k[\underline{A}_{ik},\overline{A}_{ik}]x_k\right)$，s. t. $\boldsymbol{IA}\cdot\boldsymbol{x}\leq\boldsymbol{c}$”的区间线性规划，为第 i 行求得新的上界.

例 4.　以图 6 中的区间线性约束模版为例，若约束向量$[c_1,c_2,c_3]^{\mathrm{T}}=[5,8,10]^{\mathrm{T}}$，则约简操作会为每一行约束调用一次区间线性规划，并求得每一行的新的上界. 约简后的区间线性不等式约束为

$$\begin{bmatrix}[-1,1] & [1,1] & [1,1]\\ [1,1] & [2,2] & [3,3]\\ [-1,1] & [2,2] & [3,3]\end{bmatrix}\begin{bmatrix}x_1\\x_2\\x_3\end{bmatrix}\leq\begin{bmatrix}5\\8\\8\end{bmatrix}.$$

3.2.2　包含测试($\sqsubseteq_{itv}$)

我们通过约束向量上的序关系来定义 itvTCM 上的包含关系，即 $itvTCM_1\sqsubseteq_{itv}itvTCM_2$ iff. $c'\leqslant c''$.

例 5.　设区间模版多面体 $itvTCM_1$ 和 $itvTCM_2$ 的系数矩阵如图 6 中所示，若 $\boldsymbol{c}'=[5,8,10]^{\mathrm{T}}$，$\boldsymbol{c}''=[15,8,20]^{\mathrm{T}}$，则根据包含测试的定义可知 $\boldsymbol{c}'\leqslant\boldsymbol{c}''$，故 $itvTCM_1\sqsubseteq_{itv}itvTCM_2$. 若 $\boldsymbol{c}'=[5,8,10]^{\mathrm{T}}$，$\boldsymbol{c}''=[4,8,20]^{\mathrm{T}}$，则 $itvTCM_1$ 与 $itvTCM_2$ 之间相互不存在包含关系.

3.2.3　交操作($\sqcap_{itv}$)

令 $\boldsymbol{c}'''\triangleq\langle\min(c_1',c_1''),\cdots,\min(c_m',c_m'')\rangle$，我们称向量$\boldsymbol{c}'''$为向量 $\boldsymbol{c}'$ 和 $\boldsymbol{c}''$的下界. 则 $itvTCM_1\sqcap_{itv}itvTCM_2\triangleq Reduce(\boldsymbol{IA}\cdot\boldsymbol{x}\leq\boldsymbol{c}''')$.

例 6.　设区间模版多面体 $itvTCM_1$ 和 $itvTCM_2$ 的系数矩阵如图 6 中所示，若 $\boldsymbol{c}'=[5,13,17]^{\mathrm{T}}$，$\boldsymbol{c}''=[4,12,20]^{\mathrm{T}}$，则在交操作中求得 $\boldsymbol{c}'''=[4,12,17]^{\mathrm{T}}$，约简后的区间模版多面体为

$$\begin{bmatrix}[-1,1] & [1,1] & [1,1]\\ [1,1] & [2,2] & [3,3]\\ [-1,1] & [2,2] & [3,3]\end{bmatrix}\begin{bmatrix}x_1\\x_2\\x_3\end{bmatrix}\leq\begin{bmatrix}4\\12\\12\end{bmatrix}.$$

3.2.4　结合操作($\sqcup_{itv}$)

令 $\boldsymbol{c}'''\triangleq\langle\max(c_1',c_1''),\cdots,\max(c_m',c_m'')\rangle$，我们称向量$\boldsymbol{c}'''$为向量 $\boldsymbol{c}'$，$\boldsymbol{c}''$的上界. 则 $itvTCM_1\sqcup_{itv}itvTCM_2\triangleq Reduce(\boldsymbol{IA}\cdot\vec{\boldsymbol{x}}\leq\boldsymbol{c}''')$.

例 7.　设区间模版多面体 $itvTCM_1$ 和 $itvTCM_2$ 的系数矩阵如图 6 中所示，若 $\boldsymbol{c}'=[5,13,17]^{\mathrm{T}}$，$\boldsymbol{c}''=[4,12,20]^{\mathrm{T}}$，则在结合操作中求得$\boldsymbol{c}'''=[5,13,20]^{\mathrm{T}}$，约简后的区间模版多面体为

$$\begin{bmatrix}[-1,1] & [1,1] & [1,1]\\ [1,1] & [2,2] & [3,3]\\ [-1,1] & [2,2] & [3,3]\end{bmatrix}\begin{bmatrix}x_1\\x_2\\x_3\end{bmatrix}\leq\begin{bmatrix}5\\13\\13\end{bmatrix}.$$

3.2.5　加宽操作(∇_{itv})

设 $itvTCM_1\sqsubseteq_{itv}itvTCM_2$，则 $\boldsymbol{c}'\leqslant\boldsymbol{c}''$. 首先我们定义两个向量的加宽操作：$\boldsymbol{c}'\nabla\boldsymbol{c}''\triangleq\langle wid(c_1',c_1''),\cdots,wid(c_m',c_m'')\rangle$，其中 $wid(c_i',c_i'')\triangleq\begin{cases}\infty, & \text{若 } c_i''>c_i'\\ c_i'', & \text{否则}\end{cases}$ $(1\leq i\leq m$，若 $wid(c_i',c_i'')$的值为“∞”意味着第 i 维约束为空约束). 则 $itvTCM_1\nabla_{itv}itvTCM_2\triangleq Reduce$

($\boldsymbol{IA}\cdot\boldsymbol{x}\leq\boldsymbol{c}'\triangledown\boldsymbol{c}''$).

例 8. 设区间模版多面体 $itvTCM_1$ 和 $itvTCM_2$ 的系数矩阵如图 6 中所示,若 $\boldsymbol{c}'=[5,10,7]^{\mathrm{T}}$,$\boldsymbol{c}''=[5,10,8]^{\mathrm{T}}$,则在加宽操作中求得 $\boldsymbol{c}'''=[5,10,+\infty]^{\mathrm{T}}$,约简后的区间模版多面体为

$$\begin{bmatrix}[-1,1] & [1,1] & [1,1]\\ [1,1] & [2,2] & [3,3]\\ [-1,1] & [2,2] & [3,3]\end{bmatrix}\begin{bmatrix}x_1\\x_2\\x_3\end{bmatrix}\leq\begin{bmatrix}5\\10\\10\end{bmatrix}.$$

3.2.6 测试迁移操作

条件测试$[\![\sum_k[a_k,b_k]x_k\leq c]\!]^{\sharp}(itvTCM)$的结果是在区间模版多面体 $itvTCM$ 中添加区间线性约束$\sum_k[a_k,b_k]x_k\leq c$,其中$[\![\cdot]\!]^{\sharp}(itvTCM)$表示一个程序语义动作在区间模版多面体 $itvTCM$ 上的应用效果.程序可能出现更复杂的条件测试语句,如包含析取、非线性或浮点表达式的语句,这些复杂语句都可以抽象成形如"$\sum_k[a_k,b_k]x_k\leq c$"形式的区间线性约束.

在具体实现中,我们将$[\![\sum_k[a_k,b_k]x_k\leq c]\!]^{\sharp}(itvTCM)$转化为一系列区间线性规划问题,即在约束空间中添加测试语句对应的约束,并与当前抽象域状态对应的区间线性不等式组一起作为区间线性规划的约束,依次以各个模版分别作为规划的目标函数,求解新的右值常数向量.如我们通过区间线性规划 $\max\left(\sum_k[\underline{A}_{ik},\overline{A}_{ik}]x_k\right)$, s. t. $\left(\boldsymbol{IA}\cdot\boldsymbol{x}\leq\boldsymbol{c}\wedge\sum_k[a_k,b_k]x_k\leq c\right)$,求得第 i 行的新上界.

例 9. 设当前的区间模版多面体为

$$itvTCM_1=\begin{bmatrix}[-1,1] & [1,1] & [1,1]\\ [1,1] & [2,2] & [3,3]\\ [-1,1] & [2,2] & [3,3]\end{bmatrix}\begin{bmatrix}x_1\\x_2\\x_3\end{bmatrix}\leq\begin{bmatrix}5\\10\\8\end{bmatrix},$$

条件测试语句为 $x_1+x_2+x_3<-2$,则 $itvTCM_1$ 在该条件测试语句作用后的结果为

$$\begin{bmatrix}[-1,1] & [1,1] & [1,1]\\ [1,1] & [2,2] & [3,3]\\ [-1,1] & [2,2] & [3,3]\end{bmatrix}\begin{bmatrix}x_1\\x_2\\x_3\end{bmatrix}\leq\begin{bmatrix}-2\\10\\8\end{bmatrix}.$$

3.2.7 赋值迁移操作

赋值迁移操作$[\![x_j:=exp]\!]^{\sharp}(itvTCM)$的结果为$([\![x_j'-exp=0]\!]^{\sharp}(itvTCM))[x_j'/x_j]$,其中,引入新鲜变量 x_j'是用于保存表达式 exp 的值,这对不可逆赋值语句如 $y:=[-1,1]y+[-2,2]$来说很有必要.在具体实现中,与测试迁移操作中类似,我们也是将$[\![x_j:=exp]\!]^{\sharp}(itvTCM)$转化为区间线性规划问题,只是规划的目标函数需要用变量 x_j'替换掉 x_j.如我们通过区间线性规划 $\max\left(\left(\sum_k[\underline{A}_{ik},\overline{A}_{ik}]x_k\right)[x_j'/x_j]\right)$, s. t. $(\boldsymbol{IA}\cdot\boldsymbol{x}\leq\boldsymbol{c}\wedge x_j'-exp=0)$第 i 行求得新的上界.

例 10. 设当前的区间模版多面体为

$$itvTCM_1=\begin{bmatrix}[-1,1] & [1,1] & [1,1]\\ [1,1] & [2,2] & [3,3]\\ [-1,1] & [2,2] & [3,3]\end{bmatrix}\begin{bmatrix}x_1\\x_2\\x_3\end{bmatrix}\leq\begin{bmatrix}5\\10\\8\end{bmatrix},$$

赋值迁移语句为 $x_1=x_2+1$,则 $itvTCM_1$ 在该赋值语句作用后的结果为

$$\begin{bmatrix}[-1,1] & [1,1] & [1,1]\\ [1,1] & [2,2] & [3,3]\\ [-1,1] & [2,2] & [3,3]\end{bmatrix}\begin{bmatrix}x_1\\x_2\\x_3\end{bmatrix}\leq\begin{bmatrix}3\\10\\8\end{bmatrix}.$$

4 区间线性模版约束抽象域的模版生成策略

在分析前,我们通过预分析或者人工设定等方法提供区间线性模版.本节我们将讨论模版中区间系数的生成策略,区间线性模版约束抽象域的系数矩阵可以在模版多面体抽象域的系数矩阵的基础上生成.

系数矩阵的设定是基于模版多面体抽象域程序分析的关键.在模版多面体的系数矩阵设定中有 3 个重要的来源:(1)赋值语句;(2)条件判断语句;(3)目标性质中包含的表达式.若将赋值语句作为一条模版内的约束,基于该模版可以将该条语句的迁移语义完全刻画出来.

如基于模版多面体分析图 1 中的程序 program1 时,根据语句"$y=2*x+1$",我们将"$2x-y$"添加到模版中;根据赋值语句"$y=3*x+2$",我们将"$3x-y$"添加到模版中.因此我们选取的模版为"$2x-y$"和"$3x-y$",令变量构成的向量为$[x,y]^{\mathrm{T}}$,则模版多面体的系数矩阵为$\begin{bmatrix}2 & -1\\ -2 & 1\\ 3 & -1\\ -3 & 1\end{bmatrix}$.根据该系数矩阵,可分析出程序在第 2 行 then 分支结束处的抽象状态为"$2x-y=-1$";在第 4 行 else 分支结束处的抽象状态为"$3x-y=-2$";而在第 5 行分支语句结合处的抽象状态为"top".

基于区间线性模版约束抽象域分析分析语句时,我们可以将不同分支中的模版进行组合,生成相

应的区间模版. 在实际分析时，由于赋值语句的数目可能过多，我们考虑只将不同分支中出现的类似却有区别的模版进行区间组合. 这里本文先介绍区间组合操作的定义. 给定两条区间线性线性不等式：$\varphi' \triangleq \sum_k [\underline{A}'_{ik}, \overline{A}'_{ik}] x_k \leq c'_i$ 和 $\varphi'' \triangleq \sum_k [\underline{A}''_{ik}, \overline{A}''_{ik}] x_k \leq c''_i$（可看作区间线性模版的第 i 行约束），则 φ' 与 φ'' 区间组合的结果为 $\varphi''' \triangleq [\underline{A}'''_{ik}, \overline{A}'''_{ik}] x_k \leq c'''_i$，其中 $c'''_i = \max\{c'_i, c''_i\}$，$[\underline{A}'''_{ik}, \overline{A}'''_{ik}] = [\min\{\underline{A}'_{ik}, \underline{A}''_{ik}\}, \max\{\overline{A}'_{ik}, \overline{A}''_{ik}\}]$. 通过区间组合，我们可以对不同分支上的区间模版多面体进行抽象，得到新的区间模版多面体. 以图 1 中程序 program1 为例，将 then 分支与 else 分支对应的两个模版“$[2,2]x+[-1,-1]y$”和“$[3,3]x+[-1,-1]y$”进行组合操作，得到区间模版“$[2,3]x+[-1,-1]y$”，即系数矩阵为$[[2,3],[-1,-1]]$. 基于该区间系数矩阵，可分析出程序在第 2 行 then 分支结束处的抽象状态为“$[2,3]x+[-1,-1]y \leq -1$”；在第 4 行 else 分支结束处的抽象状态为“$[2,3]x+[-1,-1]y \geq -2$”；在第 5 行分支语句结合处的抽象状态为“$-2 \leq [2,3]x+[-1,-1]\ y \leq -1$”.

区间线性模版约束抽象域与 $\mathbb{R}^n$（n 为变量的维度）上的每一个象限的交都是一个凸多面体抽象域. 区间线性模版约束抽象域的域表示可以看作是各个象限内的多面体抽象域域表示的析取，根据各个变量 $x_i (1 \leq i \leq n)$ 的正负值确立不同象限多面体的形态. 基于区间线性模版约束抽象域的这个特点，我们在遇到形如“$x_i \geq 0$”的条件判断语句时，考虑为 x_i 引入区间系数（若条件判断语句形如“$ax_i \geq b$”，我们可以通过平移转换操作将其归一化到形如“$x_i \geq 0$”）.

如图 9 所示的示例程序 program2 是从文献[10]中修改得到（将程序沿 x 轴向左平移到关于 y 轴对称）. 在基于模版多面体抽象域对 program2 进行分析时，假定变量构成的向量为$[x,y]^{\mathrm{T}}$，若经典模版抽象域的矩阵设置为$\begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$，在第 12 行行头（即循环结束处）得到 x 的值范围为$[-49,51]$，y 的值范围为$[-1,-1]$，因此会在第 12 行产生除零错误报（x 的实际值范围为$[51,51]$）. 若直接将不同分支中的模版进行合并，将区间系数矩阵设置为$\begin{bmatrix} [1,1] & [-1,1] \\ [1,1] & [-1,-1] \\ [1,1] & [1,1] \end{bmatrix}$，则在第 11 行循环结束处得到 x 的值范围为$[-50,+\infty]$，并不能消除第 12 行处的除零错误报. 我们注意到第 4 行判断语句为“$x<0$”，因此考虑将变量 x 的系数设为区间，以刻画程序的析取行为. 若将区间系数矩阵$\begin{bmatrix} [-1,1] & [1,1] \\ [1,1] & [-1,-1] \\ [1,1] & [1,1] \end{bmatrix}$，在第 11 行循环结束处，我们得到 x 的值范围为$[51,51]$，值范围更加精确，从而消除掉第 12 行处的除零错误报.

```
1.  x=-50;
2.  y=0;
3.  while(y<=0){
4.    if(x<0){
5.      x++;
6.      y++;
7.    }else{
8.      x++;
9.      y--;
10.   }
11. }
12. z=1/x;
```

图 9　示例程序 program2

区间线性规划是实现区间线性模版约束抽象域的核心操作，直接影响着抽象域域操作的复杂度. 如第 2 节所述，区间线性规划中，调用线性规划的次数与规划中变量含区间系数的数目相关. 本文中通过讨论变量的正负号来确定系数取区间的左值或右值，调用线性规划的次数是跟含区间的变量的个数成指数级关系的，即若变量系数含区间的个数为 n，则调用线性规划的次数为 2^n. 因此若变量系数含区间的个数过多，会使得分析效率很慢. 为了提升分析效率，在实际应用中，我们将考虑把部分变量的区间系数设置为常数，以减少分类讨论的次数，从而从一定程度上保证分析效率. 但是，将部分变量的区间系数设为常数，可能使得分析精度有所下降. 本质上，本文通过限定变量系数为常数或区间，在分析精度和分析效率之间进行权衡.

5　实现和实验

本文基于抽象解释框架实现了一个面向 C 程序的静态分析工具原型 CAI，在 CAI 的框架下对模版多面体和区间线性模版约束抽象域进行实现. 工具原型 CAI 的结构如图 10 所示，包括开源前端 CIL[11]、开源数值抽象域库 Apron[12] 以及不动点求解模块. 本文首先通过 CIL 对源文件进行解析，得到其控制流图，再使用 Apron 库提供的数据类型建立程序相应的语义方程，最后基于不动点求解模版

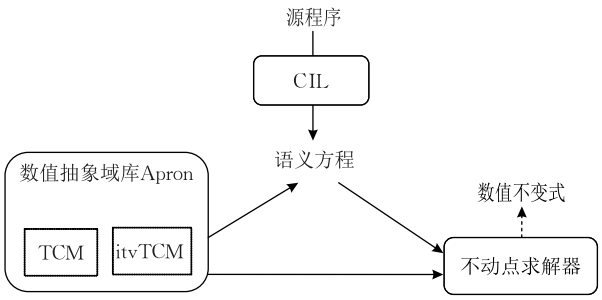

图 10 静态分析工具原型 CAI 的框架图

对语义方程进行迭代得到源程序的数值不变式.

在 Apron 框架下,本文实现了模版多面体和区间线性模版约束抽象域,并进行一系列的对比实验.本文按照文献[8]中的方法,基于 GNU 线性规划工具包 GLPK(GNU Linear Programming Kit)①实现了区间线性规划方法,并将其应用到区间线性模版约束抽象域域操作的实现中.本文采用的实验平台是:Ubuntu 14.04 操作系统,4GB 物理内存,AMD Athlon 3.1GHz 四核 CPU 处理器.实验所用的测试用例分别来自于本文的例子,相关工作[13]以及 SV-COMP 2016②.其中 program1 和 program2 是本文的中使用的两个示例程序;itv_pol1、itv_pol2 和 itv_pol3 来自参考文献[13];gr2006_true-unreach、gj2007_true-unreach、gsv2008_true-unreach 来自 SV-COMP 2016 的 loop-lit 目录,test_locks_5、test_locks_6、test_locks_7 来自 SV-COMP 2016 的 locks 目录.如表 1 所示,列"#VAR"给出了各个程序中变量的个数,列"LOC"给出了各个程序的代码行数,其中最大的程序来自于 SV-COMP 2016 的 locks 目录.本文所选的测试用例均含有析取行为(包含分支语句,循环语句,分支条件为析取表达式等).

表 1 TCM 与 itvTCM 实验结果对比表

程序名	#VAR	LOC	TCM			itvTCM			itvTCM 与 itvPol 精度比较
			时间/s	#LP	是否验证	时间/s	#LP	是否验证	
program1	2	8	0.090	150	否	0.090	202	是	=
program2	2	12	0.187	3030	否	0.251	7404	是	>
itvPol_1	1	8	0.080	134	否	0.088	208	是	=
itvPol_2	3	11	0.105	596	否	0.113	858	是	=
itvPol_3	2	12	0.297	1270	否	0.337	19832	是	>
gr2006_true-unreach	2	16	0.227	4130	否	0.301	9122	是	>
gj2007_true-unreach	2	14	0.159	2488	否	0.184	4722	是	>
gsv2008_true-unreach	2	11	0.113	624	否	0.116	1137	是	>
test_locks_5	11	93	0.785	24372	否	2.902	52466	否	>
test_locks_6	13	107	0.943	25724	否	3.226	57624	否	>
test_locks_7	15	121	1.117	26836	否	3.861	60442	否	>

表 1 给出了 CAI 基于模版多面体抽象域与区间模版多面体抽象域对测试用例的分析结果.其中,"时间(s)"列给出 CAI 基于不同抽象域分析所花的时间,"#LP"列给出分析过程中对线性规划求解器 GLPK 的调用总次数,"是否验证"列给出基于不同抽象域能否验证给定性质,"itvTCM 与 itvPol 精度比较"列将本文的区间线性模版约束抽象域与文献[13]的抽象域的分析精度作对比.

经典模版多面体抽象域中使用的模版均是常数系数(系数矩阵从赋值语句中提出,或者人工设定),区间线性模版约束抽象域中使用的模版是将不同分支下经典模版多面体体抽象域中的系数组合后生成的区间(即基于第 4 节的区间组合操作生成区间系数矩阵).

经典模版多面体抽象域的系数矩阵与区间线性模版约束抽象域相对应,如区间线性模版约束抽象域设定的系数为"[[2,3],[−1,−1]]",则经典模版多面体抽象域对应的系数为"$2x-y$"和"$3x-y$".

从表 1 中的分析结果可以看出区间线性模版约束抽象域相对于模版多面体抽象域往往可以更好地刻画程序析取行为(实验中列出的程序均带有析取行为),从而验证测试用例中的给定性质.但由于区间线性模版约束抽象域的域操作是基于区间线性规划实现的,需要调用更多次线性规划,因此分析时间往往更长.此外,当选定适当的区间模板时,itvTCM 的分析结果往往会较 itvPol 的分析结果更为精确.

6 相关工作

在抽象解释框架下,大部分数值抽象域只能基于一系列线性约束的合取来表达凸性质,比如区间

① http://www.gnu.org/software/glpk/
② http://sv-comp.sosy-lab.org

抽象域[2]、八边形抽象域[5]、凸多面体抽象域[2-3]等. Sankaranarayanan 等人提出了模版多面体抽象域，基于线性规划设计了模版多面体抽象域的域表示和域操作[3-4]. 经典的模版多面体抽象域是一个通用的凸抽象域，通过设置不同的系数矩阵，模版多面体抽象域可以表达很多常见的数值抽象域(如区间抽象域、八边形抽象域、等式抽象域等). 但经典模版多面体抽象域中变量系数为固定常数，不能表达非凸性质，而本文提出的区间线性模版约束抽象域，基于区间线性不等式系统的弱解能够表达某些非凸性质.

此外，许多学者也在非凸抽象域方面开展了一系列工作[14-15]. Sankaranarayanan 等人研究含析取的抽象域的域表示和域操作，提出基本的抽象域的幂集拓展，以及相应的不动点迭代策略，并指出幂集拓展会带来指数爆炸问题[14]. Axel 等人将 boolean flag 引入多面体抽象域，布尔变量的不同取值对应于不同的多面体表示，使整个抽象域具有一定的表达非凸性质的能力[16]；Khalil 等则使用集合的减法运算来表达非凸性质，即两个集合之间余集来作为程序的状态表示(一个集合是程序状态集合的上近似，另一个集合是程序状态集合补集的下近似)，从而具有一定的非凸性质表达能力[15]. Gurfinkel 等人对基于线性决策图对经典区间抽象域做拓展，提出了一个 BOXES 抽象域，通过多个区间的析取来表达非凸变量值范围，线性决策图与二叉决策图相近，但中间结点均是线性表达式，作者还给出基于线性决策树的抽象域域操作的高效算法[17]. Xavier 等人提出了迹划分抽象域，该抽象域基于历史控制流结构信息对程序抽象状态进行划分从而表达非凸性质，作者还讨论了迹划分策略以及迹合并的时机等问题[18]. Chen 等人提出了一个通用的基于二叉决策树的抽象域算子，二叉决策树的中间节点由程序的分支结点处的判断条件决定，二叉决策树的叶子结点为经典的凸抽象域(如区间抽象域、八边形抽象域等)，该算子可以通过刻画程序的迹语义来表达非凸性质[19]. Urban 等人在基于抽象解释的终止性分析中，将分支条件作为抽象状态划分的算子，提出了基于决策树的程序有条件终止的求解算法[20].

线性模版抽象域的模版选择与设定这一开放性问题也吸引了很多学者的关注. Sankaranarayanan 等人讨论了基于源程序的模版生成策略，即从程序的赋值语句、待验证性质中提取模版，此外他们还考虑一些带实际应用背景的模版生成策略，如对数组程序常常考虑基于区间抽象域和八边形抽象域生成模版[3-4]. Deepak 等人使用量词消去技术求解程序的循环不变式的形态，即确定程序对应的等式模版[21]. 他们首先假定程序的模版为一个含参数的形式，再基于量词消去技术，消减掉程序变量，求得不变式中的变量系数. Karpenkov 等人在基于策略迭代的程序分析中，也研究了模版生成的策略[22]，他们提出的模版生成策略在区间和八边形抽象域的基础上添加了形如“$x+2y$”、“$x+y+z$”等常见的表达式形式作为模版候选.

本文的作者在抽象解释领域特别是抽象域的设计上做了一系列的工作，本文的主要工作是设计一个能刻画区间线性约束的抽象域，与所在团队先前设计的抽象域既有联系也有区别. Chen 等人将区间线性代数应用到经典多面体抽象域中，提出了区间多面体抽象域，区间多面体抽象域可以天然地表达某些非凸性质，其区间系数是任意形式的，因而复杂度是指数级的，可扩展性有局限性[13]. 而本文提出的区间线性模版约束抽象域的域表示中变量的系数均是固定的，因此域操作的复杂度由模版中区间系数的数目决定，可以通过限定模版中区间个数降低分析的复杂度. Chen 等人提出了单变量区间线性不等式抽象域，其主要思想是使用单变量区间线性不等式约束作为域元素的约束表示方法，可以表达某类非凸、非连通性质[23]. 单变量区间线性不等式抽象域可以看作本文区间线性模版抽象域的一种特殊形式. Chen 等人将绝对值算子与经典八边形抽象域相结合，可以刻画变量的值与其绝对值之间的八边形关系，具有一定的析取表达能力[24]. 如果设置合适的模版(如模版中将变量系数设置为[−1,1]、[0,1]等)，绝对值八边形抽象域可以看作区间线性模版约束抽象域的一个特殊情形. Jiang 等人提出了区间幂集抽象域，使用有限个区间的析取来刻画变量的取值范围，从而表达非凸性质[25]. 为了降低操作的复杂度，作者限定了幂集集合中元素的个数. 为了将指针分析与数值分析相结合，Yin 等人提出了一种新的指针内存模型，然后基于该模型设计了一个刻画指针指向关系和指针偏移量的抽象域，并实现了一个面向 C 程序的静态分析工具原型，对带指针算术的 C 程序进行分析，求解源程序中指针的指向信息和偏移量信息的不变式[26].

7　总结与下一步工作

本文对抽象解释领域具有代表性的模版多面体

抽象域进行扩展,提出了一个新的数值抽象域——区间线性模版约束抽象域.该抽象域能够表达和处理带区间线性表达式,可以用来分析程序中变量之间的区间线性关系(形如$\sum_k [a_k, b_k] x_k \leq c$).与经典模版多面体抽象域类似,区间线性模版约束抽象域的模版是固定的(可通过预分析或者人工设定等方法获得).区间线性模版约束抽象域基于区间线性约束的弱解语义,能够天然地表达某类非凸性质.区间线性模版约束抽象域与 $\mathbb{R}^n$(n 为变量的维度)上的每一个象限的交都是一个凸模版多面体抽象域,其域表示是每个象限内的模版多面体抽象域的析取.区间线性模版约束抽象域的域操作可以通过区间线性规划来构造.此外,我们还讨论了如何基于经典模版多面体抽象域的模版构造区间模版多面体抽象域所需要的模版.初步的实验结果表明,区间线性模版约束抽象域可以较好地刻画程序的非凸性质,区间线性模版多面体抽象域相比于区间多面体抽象域具有更好的可扩展性.

模版的设置是基于区间线性模版约束抽象域分析程序的关键,也是一个开放性的问题.下一步,我们将研究更为通用有效的模版自动生成方法(比如面向待分析性质的后向分析来生成模版等),以提升分析方法的可用性.

参 考 文 献

[1] Cousot P, Cousot R. Abstract interpretation: A unified lattice model for static analysis of programs by construction or approximation of fix points//Proceedings of the 4th ACM Symposium on Principles of Programming Languages. New York, USA, 1977: 234-252

[2] Cousot P, Halbwachs N. Automatic discovery of linear restraints among variables of a program//Proceedings of the 5th ACM Symposium on Principles of Programming Languages. New York, USA, 1978: 84-96

[3] Colón M A, Sankaranarayanan S. Generalizing the template polyhedral domain//Proceedings of the European Conference on Programming Languages and Systems. Heidelberg, Germany, 2011: 176-195

[4] Sankaranarayanan S, Sipma H B, Manna Z. Scalable analysis of linear systems using mathematical programming//Proceedings of the Conference of Verification, Model Checking and Abstract Interpretation. Heidelberg, Germany, 2005: 25-41

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

[6] Clariso R, Cortadellab J. The octahedron abstract domain. Science of Computer Programming, 2007, 64(1): 115-139

[7] Rohn J. Solvability of systems of linear interval equations. SIAM Journal on Matrix Analysis & Applications, 2004, 25(1): 237-245

[8] Chineck J W, Ramadan K. Linear programming with interval coefficients. Journal of the Operational Research Society, 2000, 51(2): 209-220

[9] Rohn J. A handbook of results on interval linear problems. Czech Academy of Sciences, Prague, Czech Republic: Technical Report 1164, 2005

[10] Gopan D, Reps T W. Guided static analysis//Proceedings of the Static Analysis Symposium. Kongens Lyngby, Denmark, 2007: 349-365

[11] Necula G, McPeak S, Rahul S, et al. CIL: Intermediate language and tools for analysis and transformation of C programs//Proceedings of the Compiler Construction. Berlin, Germany, 2002: 209-265

[12] Jeannet B, Mine A, et al. The APRON library for numerical abstract domains//Proceedings of the 21st International Conference on Computer Aided Verification. Grenoble, France, 2009: 661-667

[13] Chen L, Miné A, Wang J, et al. Interval polyhedra: An abstract domain to infer interval linear relationships//Proceedings of the 16th International Symposium on Static Analysis. Heidelberg, Germany, 2009: 309-325

[14] Sankaranarayanan S, Ivančić F, Shlyakhter I, et al. Static analysis in disjunctive numerical domains//Proceedings of the Static Analysis Symposium. Heidelberg, Germany, 2006: 3-17

[15] Ghorbal K, Ivančić F, Balakrishnan G, et al. Donut domains: Efficient non-convex domains for abstract interpretation//Proceedings of the Conference of Verification, Model Checking and Abstract Interpretation. Heidelberg, Germany, 2012: 235-250

[16] Simon A. Splitting the control flow with Boolean flags//Proceedings of the Static Analysis Symposium. Valencia, Spain, 2008: 315-331

[17] Gurfinkel A, Chaki S. Boxes: A symbolic abstract domain of boxes//Proceedings of the Static Analysis Symposium. Perpignan, France, 2010: 287-303

[18] Rival X, Mauborgne L. The trace partitioning abstract domain. ACM Transactions on Programming Languages and Systems, 2007, 29(5): 26

[19] Chen Junjie, Cousot P. A binary decision tree abstract domain functor//Proceedings of the Static Analysis Symposium. Berlin, Germany, 2015: 36-53

[20] Urban C, Miné A. A decision tree abstract domain for proving conditional termination//Proceedings of the International Static Analysis Symposium. Munich, Germany, 2014: 302-318

[21] Kapur D. Program analysis using quantifier-elimination heuristics //Proceedings of the International Conference on Theory and Applications of MODELS of Computation. Albuquerque, USA, 2012: 94-108

[22] Karpenkov E G, Monniaux D, Wendler P. Program analysis with local policy iteration//Proceedings of the Conference of Verification, Model Checking, and Abstract Interpretation. Heidelberg, Germany, 2015: 127-146

[23] Chen Li-Qian, Wang Ji, Hou Su-Ning. An abstract domain of one-variable interval linear inequalities. Chinese Journal of Computers, 2010, 33(3): 427-439(in Chinese)
(陈立前，王戟，侯苏宁. 单变量区间线性不等式抽象域. 计算机学报，2010，33(3)：427-439)

[24] Chen L, Liu J, Miné A, et al. An abstract domain to infer octagonal constraints with absolute value//Proceedings of the Static Analysis Symposium. Munich, Germany, 2014: 101-117

[25] Jiang Jia-Hong, Chen Li-Qian, Wang Ji. Floating-point program analysis based on floating-point powerset of intervals abstract domain. Journal of Frontiers of Computer Science and Technology, 2013, 7(3): 209-217(in Chinese)
(姜加红，陈立前，王戟. 基于浮点区间幂集抽象域的浮点程序分析. 计算机科学与探索，2013，7(3)：209-217)

[26] Yin Bang-Hu, Chen Li-Qian, Wang Ji. Analysis of programs with pointer arithmetic by combining points-to and numerical abstractions. Computer Science, 2015, 42(7): 32-37(in Chinese)
(尹帮虎，陈立前，王戟. 基于指向与数值抽象的带指针算术程序的分析方法. 计算机科学，2015，42(7)：32-37)

JIANG Jia-Hong, born in 1989, Ph. D. candidate. His research interests include abstract interpretation and SMT.

YIN Bang-Hu, born in 1989, Ph. D. candidate. His research interests include abstract interpretation and program analysis.

CHEN Li-Qian, born in 1982, Ph.D., assistant professor. His research interests include program analysis and verification, formal methods.

Background

The problem of automatically inferring numerical invariants in a program has received wide attention in the analysis and verification of programs. Abstract interpretation is a general theory to soundly approximate program semantics. This theory provides a general framework to analyze value ranges of program variables, guaranteeing the soundness of the analysis. Abstract domain is key to the framework of abstract interpretation, and many numerical abstract domains have been proposed under this framework. In particular, the expressiveness of the template constraint matrix domain (TCM) subsumes most weakly relational abstract domains that are commonly used in practical program analysis, for example, interval abstract domain, octagon abstract domain, etc. This paper extends the classical template constraint matrix domain which is based on linear template constraints, to support interval linear template constraints, and proposes a new numerical domain—interval template constraint matrix domain(itvTCM), which could infer interval linear inequality relations among variables in a program in the form of "$\sum_k [a_k, b_k] x_k \leq c$"(where the interval $[a_k, b_k]$ is determined before analysis). itvTCM is more expressive than TCM, and could infer non-convex properties. This paper provides domain representation and domain operations of itvTCM, and most domain operations of itvTCM are implemented based on interval linear programming. Furthermore, we discuss how to generate templates for itvTCM. The preliminary experimental results show that itvTCM is useful to capture disjunctive behaviors of a program.

This work is supported by the National Basic Research Program of China under Grant No. 2014CB340703, the National Natural Science Foundation of China under Grant No. 61532007 and the Open Project of Shanghai Key Laboratory of Trustworthy Computing under Grant No. 07dz22304201504. These projects aim to build a systematic framework for analysis framework for analysis and verification of software system, and to improve the depend ability of software, through techniques including testing, analysis, verification, etc.