3 实现与实验结果

3.1 原型系统的实现

本文在传统抽象解释框架下实现了我们提出的改进方法.具体地,对支持 APRON[17]库的基于抽象解释的静态分析工具 Interproc[6]进行扩展,加入了含有条件分支语句循环和单变量线性赋值循环的不变式生成模块.其中,修改了Interproc的程序分析流程:识别含有条件分支语句的循环,调用循环分解函数将其分解为简单循环,根据可达的分路径的幂集扩展,生成析取形式的不变式并将其融入原流程中继续分析;识别单变量线性赋值循环,提取参数,直接生成指数型循环不变式.由于指数型循环不变式不是传统数值抽象域的域元素形式,因此在保留精确形式的同时,求得每个指数型循环不变式的最大值和最小值,替换为区间的形式,随后将其带入原流程中继续分析.由第 2 节的讨论可知,每个指数型不变式都是单调的,因此这种转换是可行且易于实现的.

此外,修改了 APRON 库中区间抽象域和凸多面体抽象域的实现,增加了支持幂集扩展的计算规则:

$$(l_1\vee l_2)\odot l_3=(l_1\odot l_3)\vee(l_2\odot l_3)=\{l_1\odot l_3,l_2\odot l_3\},$$

其中,l_1,l_2,l_3 为抽象域的域元素;$\odot$表示抽象域中原有的运算符,包括逻辑运算中的交与接合、算术操作中的加减乘除、加宽算子、变窄算子.$l_1\vee l_2$ 为由改进方法获得的析取形式的不变式,由于它是通过语义分析所得,由前面的分析可知,l_1,l_2 不存在交集.

3.2 实验评估

为了评估本文方法的有效性,我们把基于循环分解的改进方法和基于归纳推理的改进方法分别嵌入传统数值抽象域中,并与没有嵌入本文改进方法的传统数值抽象域在生成不变式的精度和性能等方面进行了比较.随后,将两种改进方法同时嵌入到传统数值抽象域中,以说明同时考虑两种方法对不变式生成的精度有更大程度的提高.本文采用的实验平台为 Ubuntu 12.04 Linux 操作系统,4G 物理内存,Intel i5 3.2GHz 四核 CPU 处理器.

3.2.1 基于循环分解的改进方法

为了说明基于状态可达迁移图的循环分解方法在处理条件分支循环上的优势,将其与同样处理该类循环的文献[12]中的阶段切分谓词方法进行对比.表 2 中给出这两种方法分别嵌入凸多面体抽象域的对比实验结果.

Table 2 Experimental results of the method based on loop decomposition

表 2 基于循环分解改进方法的对比实验结果

程序名	变量数目	凸多面体抽象域+循环分解方法		凸多面体抽象域+阶段切分谓词方法		精度比较
		计算时间(s)	安全性质(Y/N)	计算时间(s)	安全性质(Y/N)	
prog1	2	0.021	NS	0.013	NS	>
prog1′	2	0.020	Y	0.013	N	>
prog1″	2	0.024	NS	0.014	NS	>
rajamani	5	0.033	Y	0.020	N	>
phase	3	0.023	NS	0.012	NS	=
tacs08	5	0.025	NS	0.016	NS	=
tacs08′	5	0.027	NS	0.017	NS	>

程序 prog1 为图 1 的代码实例.prog1′和 prog1″为 prog1 程序稍作修改的新程序,在第 1.3 节给出了描述.并且,以上 3 个程序作为例子在第 1 节中给出了基于循环分解方法的不变式生成过程以及与凸多面体抽象域分析结果的对比(由本文开始部分可知,该程序用文献[12]中方法无法处理).rajamani 是 InvGen 工具的测试用例.phase 是 Frama-c[18]工具的测试用例.tacs08 是文献[19]的例子,tacs08′是 tacs08 的一个变种,把其标量值的初始状态放大为区间值.

表 2 中:“变量数目”栏给出了程序中的变量数目;“精度比较”栏比较了两种方法在程序结尾处生成的不变式的精度,其中,>表示本文改进方法生成的不变式的精度大于所对比的方法,=表示本文改进方法生成的不变式精度和所对比的方法一样;“安全性质”栏比较了两种方法生成的不变式能否证明程序中安全性质相关的断言,其中,Y 表示能够证明,N 表示不能证明,NS 表示程序中没有和安全性质相关的断言.

表 2 中的实验结果显示:针对条件分支循环,与文献[12]中的阶段切分谓词方法相比,基于循环分解的方法

能够优化某些文献[12]中方法所不能处理的循环,并且生成的不变式比原抽象域的分析结果更精确.当文献[12]中的方法能够改进时,本文方法生成的不变式精度不低于文献[12].具体地,由表 2 可知:

- 前 4 个测试程序均不满足文献[12]中的判断规则,因此文献[12]中的方法无法处理;而基于循环分解的改进方法能够进行优化,进而生成比原抽象域分析结果更精确的不变式;
- 后 3 个程序用本文方法和文献[12]中方法均能处理,但当程序变量初始值不存在确定值时,程序执行会存在多种路径可能,本文方法能够更细粒度地分析,生成比文献[12]中方法更精确的不变式(如程序 tacs08′);当存在确定的单一路径时,本文方法和文献[12]中方法生成的不变式相同;而且本文方法能够识别程序中可能的不终止路径,并给出警告.

3.2.2 基于归纳推理的改进方法

为了说明归纳推理方法在处理单变量线性赋值循环上的优势,将其生成的指数型不变式和未借助归纳推理方法的区间抽象域的分析结果进行对比.需要指出的是:目前实现的此类改进只是针对非关系型的单变量不变式,而区间抽象域正是针对单变量的抽象域.表 3 中给出了对比实验结果.

Table 3 Experimental results of the method based on deduction

表 3 基于归纳推理改进方法的对比实验结果

程序名	变量数目	区间抽象域+归纳推理方法		区间抽象域		精度比较
		计算时间(s)	安全性质(Y/N)	计算时间(s)	安全性质(Y/N)	
prog2	1	0.008	Y	0.006	N	>
prog2′	1	0.008	Y	0.005	N	>
PADO01	1	0.006	NS	0.004	NS	>
SAS09	2	0.010	NS	0.006	NS	>
rec_lin2	6	0.028	Y	0.019	N	>
10-Filter	8	0.031	NS	0.022	NS	>

程序 prog2 和 prog2′为图 2 的代码实例,已在第 2 节作为例子分析了利用归纳推理方法的不变式生成过程以及与区间抽象域分析结果的差别.PADO01 和 SAS09 分别是文献[20]和文献[16]中的程序片段.rec_lin2 和 10-Filter 分别是文献[21]和文献[22]中的例子.这些程序均是含有或者经过化简含有单变量线性赋值循环的程序.

表 3 中的各栏说明同表 2.表 3 中的实验结果显示:针对单变量线性赋值循环,相比于区间抽象域的分析结果,本文的基于归纳推理的改进方法嵌入区间抽象域后能够生成更精确的不变式.并且,根据指数型循环不变式的性质(规则 A~规则 C),能够分析出程序中导致不安全状态的变量取值.

3.2.3 本文改进方法的总体效果

由于凸多面体抽象域是在区间抽象域的基础上构造的,从单变量扩展为表达变量间关系的区间性质,因此,凸多面体抽象域的分析结果包含了区间抽象域的分析结果.于是,本文在验证改进方法总体效果的有效性时,将同时嵌入了两种改进方法的凸多面体抽象域的分析结果与只嵌入了其中一种的分析结果进行比较.表 4 给出了对比实验结果.

Table 4 Experimental results of the integral methods

表 4 改进方法的总体对比实验结果

程序名	变量数目	凸多面体抽象域+循环分解方法+归纳推理方法	凸多面体抽象域+循环分解方法	凸多面体抽象域+归纳推理方法	精度比较	
		计算时间(s)	计算时间(s)	计算时间(s)		
rabin	4	0.062	0.045	0.039	>	>
md5	7	0.081	0.059	0.052	>	>
ecdsa	11	0.213	0.154	0.146	>	>

据我们所知,目前的研究工作中并没有同时研究这两种程序结构的不变式生成问题,因此没有现成的测试案例.但是,现实的开源程序中不乏同时满足两个程序特征的片段,比如含有较多数值计算的加密算法.我们选取了有代表性的 3 个作为测试案例.表格中的 rabin,md5,ecdsa 分别为 rabin 加密算法、md5 加密算法、椭圆曲线签名算法的实现程序片段.由于本文改进方法实现的工具只支持 C 语言语法的子集,因此在使用之前,我们对这些现实程序进行了一些人工的修改.

表 4 中:“精度比较”分为两个子栏,分别表示同时使用了两种改进方法和只使用其中一种在生成不变式精度方面的比较;其余各栏说明同表 2.表 4 中的实验结果显示:同时使用两种改进方法应用在凸多面体抽象域上,相比于只用了一种改进方法,能够生成更精确的不变式.由于待分析程序中存在的循环同时满足含有条件分支语句和单变量线性赋值语句,说明这个程序同时含有显式析取语义和隐式析取语义.改进方法的处理过程是叠加的:条件分支循环经过分解,成为不含条件判断语句的简单循环序列后,基于归纳推理的改进方法才会去寻找单变量线性赋值循环作进一步的处理.因此不会产生冲突或相互抑制,生成的不变式精度比单用其中一种改进方法更高.由表 2~表 4 的“计算时间”栏可以看出:本文的改进方法由于需要识别特殊的程序结构,并且作细粒度的处理,会额外消耗更多的时间,但是都在可接受的范围之内(一般不会超过传统数值抽象域的两倍).

4 相关工作

不变式在程序的形式化分析和验证中发挥着重要的作用,因此,程序不变式的自动化生成是学术界一直关注的问题.Cousot[11]最先提出了抽象解释方法用于不变式的推导,该方法是在抽象域上进行符号执行,通过过近似(over-approximate)的语义分析得到不变式.近些年,Cousot 把抽象解释框架应用到终止性证明[23]等领域.Miné 等人[24]提出的八面体抽象域在实际工业界代码的分析中取得了成功应用.Chen 等人[16]提出的区间多面体抽象域对传统凸多面体抽象域进行了扩展,具有更强的表达能力.Jeannet 等人[25,26]提出了利用 Jordan 标准型的抽象加速技术来解决抽象解释对程序状态空间的放大问题.

对于含有析取语义的程序结构而言,抽象解释作为一种静态的分析方法为了加速不动点计算会得到过于保守的结果.一些近期的工作对该问题进行了改进:Sharma 等人[12]利用切分谓词对部分简单的多阶段循环进行了循环分解,但其所能解决的场景比较有限,需要满足严格的逻辑表达式;Mauborgne 等人[27]关于迹划分的工作不是完全自动化的,它需要用户输入对于划分的指导;Gulwani 等人[28]使用了控制流精化的方法获得循环迭代次数的上界,但是由于其属于更高层次的抽象表达,对于不变式的精化帮助有限.相比于以上的工作,本文引入了状态可达迁移图,能够处理的循环场景比文献[12]更多样,并且是完全自动化的,能够表达复杂程序结构的变量取值特性.

对于隐式含有析取语义的单变量线性赋值循环,现有的改进方法需要借助 Grobner 基[29]计算、一阶量词消去[30]等复杂度较高的方法来生成非线性的不变式.Mastroeni 等人[31]提出了表达代数幂性质的抽象域,主要用于随机程序的静态分析和因式分解.它能够处理本文提出的含有 $x=a\times x+b$ 的循环,但其计算效率相比本文归纳推理的方法要低很多.Yang 等人[32,33]将符号计算方法应用在程序验证领域,将不变式生成转化为半代数系统的求解,取得了较好的研究成果.该方法不仅对线性程序,而且对含有析取语义的非线性程序同样能够生成较为精确的不变式.但是该方法的计算复杂度较高,本文基于归纳推理的改进方法在处理含有 $x=a\times x+b$ 的循环时只要根据参数就可以直接生成不变式,计算复杂度更低.

5 总结和展望

本文提出了含有析取语义的循环不变式生成的改进方法.主要针对包含条件分支语句的循环和单变量线性赋值循环这两类程序,运用基于状态可达迁移图的循环分解技术和归纳推理,能够更精确地表达程序语义,从而生成更精确的不变式.实现了嵌入改进方法的传统数值抽象域的原型系统.实验结果表明,嵌入了改进方法的传统数值抽象域能够生成更精确的不变式.进一步的工作包括:结合动态方法,分析含有条件分支语句的循环和区间线性化技术的研究.

References:

[1] Cousot P, Cousot R. Abstract interpretation frameworks. Journal of Logic and Computation, 1992,2(4):511−547. [doi: 10.1093/logcom/2.4.511]

[2] Cousot P, Cousot R. Static determination of dynamic properties of programs. In: Proc. of the 2nd Int'l Symp. on Programming. Paris, 1976. 106−130. https://nyu.pure.elsevier.com/en/publications/static-determination-of-dynamic-properties-of-programs

[3] Granger P. Static analysis of arithmetical congruences. Int'l Journal of Computer Mathematics, 1989,30(3-4):165−190. [doi: 10.1080/00207168908803778]

[4] Cousot P, Halbwachs N. Automatic discovery of linear restraints among variables of a program. In: Proc. of the 5th ACM SIGACT-SIGPLAN Symp. on Principles of Programming Languages (POPL). New York: ACM Press, 1978. 84−96. [doi: 10.1145/512760.512770]

[5] Cousot P, Cousot R, Feret J, Mauborgne L, Miné A, Monniaux D, Rival X. The ASTREE analyzer, programming languages and systems. In: Proc. of the 14th European Symp. on Programming, ESOP. Edinburgh: Springer-Verlag, 2005. 21−30. [doi: 10.1007/978-3-540-31987-0_3]

[6] Jeannet B. Interproc analyzer for recursive programs with numerical variables. INRIA. 2010. http://pop-art.inrialpes.fr/interproc/interprocweb.cgi

[7] Sankaranarayanan S, Sipma HB, Manna Z. Constraint-Based linear-relations analysis. In: Proc. of the 11th Int'l Symp. (SAS). Verona: Springer-Verlag, 2004. 53−68. [doi: 10.1007/978-3-540-27864-1_7]

[8] Gupta A, Rybalchenko A. Invgen: An efficient invariant generator. In: Proc. of the 21st Int'l Conf. (CAV). Grenoble: Springer-Verlag, 2009. 634−640. [doi: 10.1007/978-3-642-02658-4_48]

[9] Chen L, Wang J, Hou S. An abstract domain of one-variable interval linear inequalities. Chinese Journal of Computers, 2010,33(3):427−439 (in Chinese with English abstract). http://cjc.ict.ac.cn/qwjs/view.asp?id=3050 [doi: 10.3724/SP.J.1016.2010.00427]

[10] Tarski A. A lattice-theoretical fixpoint theorem and its applications. Pacific journal of Mathematics, 1955,5(2):285−309. [doi: 10.2140/pjm.1955.5.285]

[11] 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 ACM SIGACT-SIGPLAN Symp. on Principles of Programming Languages (POPL). New York: ACM Press, 1977. 238−252. [doi: 10.1145/512950.512973]

[12] Sharma R, Dillig I, Dillig T, Aiken A. Simplifying loop invariant generation using splitter predicates. In: Proc. of the 23rd Int'l Conf. (CAV). Snowbird: Springer-Verlag, 2011. 703−719. [doi: 10.1007/978-3-642-22110-1_57]

[13] Cousot P, Cousot R. Systematic design of program analysis frameworks. In: Proc. of the 6th ACM SIGACT-SIGPLAN Symp. on Principles of Programming Languages (POPL). New York: ACM Press, 1979. 269−282. [doi: 10.1145/567752.567778]

[14] Cormen TH, Leiserson CE, Rivest RL, Stein C. Introduction to Algorithms. Cambridge: MIT Press, 2001.

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

[16] Chen L, Miné A, Wang J, Cousot P. Interval polyhedra: An abstract domain to infer interval linear relationships. In: Proc. of the 16th Int'l Symp. (SAS). Los Angeles: Springer-Verlag, 2009. 309−325. [doi: 10.1007/978-3-642-03237-0_21]

[17] Jeannet B, Miné A. Apron: A library of numerical abstract domains for static analysis. In: Proc. of the 21st Int'l Conf. (CAV). Grenoble: Springer-Verlag, 2009. 661−667. [doi: 10.1007/978-3-642-02658-4_52]

[18] Cuoq P, Kirchner F, Kosmatov N, Prevosto V, Signoles J, Yakobowski B. Frama-c. In: Proc. of the 10th Int'l Conf. (SEFM). Greece: Springer-Verlag, 2012. 233−247. [doi: 10.1007/978-3-642-33826-7_16].

[19] Gulavani BS, Chakraborty S, Nori AV, Rajamani SK. Automatically refining abstract interpretations. In: Proc. of the 14th Int'l Conf. (TACAS). Budapest: Springer-Verlag, 2008. 443−458. [doi: 10.1007/978-3-540-78800-3_33]

[20] Mastroeni I. Numerical power analysis. In: Proc. of the 2nd Int'l Conf. (PADO). Aarhus: Springer-Verlag, 2001. 117−137. [doi: 10.1007/3-540-44978-7_8]

[21] Boldo S. Floats and ropes: A case study for formal numerical program verification. In: Proc. of the 36th Int'l Collogquium (ICALP). Rhodes: Springer-Verlag, 2009. 91−102. [doi: 10.1007/978-3-642-02930-1_8]

[22] Taylor DE, Turner JS. Classbench: A packet classification benchmark. In: Proc. of the 24th Annual Joint Conf. of the IEEE Computer and Communications Societies (INFOCOM). IEEE, 2005. 2068−2079. [doi: 10.1109/INFCOM.2005.1498483]

[23] Cousot P, Cousot R. An abstract interpretation framework for termination. In: Proc. of the 39th ACM SIGACT-SIGPLAN Symp. on Principles of Programming Languages (POPL). New York: ACM Press, 2012. 245−258. [doi: 10.1145/2103621.2103687]

[24] Miné A. The octagon abstract domain. Higher-Order and Symbolic Computation, 2006,19(1):31−100. [doi: 10.1007/s10990-006-8609-1]

[25] Schrammel P, Jeannet B. Logico-Numerical abstract acceleration and application to the verification of data-flow programs. In: Proc. of the 18th Int'l Symp. (SAS). Venice: Springer-Verlag, 2011. 233−248. [doi: 10.1007/978-3-642-23702-7_19]

[26] Jeannet B, Schrammel P, Sankaranarayanan S. Abstract acceleration of general linear loops. In: Proc. of the 41st Annual ACM SIGPLAN-SIGACT Symp. on Principles of Programming Languages (POPL). New York: ACM Press, 2014. 529−540. [doi: 10.1145/2535838.2535843]

[27] Mauborgne L, Rival X. Trace partitioning in abstract interpretation based static analyzers. In: Proc. of the 14th European Symp. on Programming (ESOP). Edinburgh: Springer-Verlag, 2005. 5−20. [doi: 10.1007/978-3-540-31987-0_2]

[28] Gulwani S, Jain S, Koskinen E. Control-Flow refinement and progress invariants for bound analysis. In: Proc. of the 2009 ACM SIGPLAN Conf. on Programming Language Design and Implementation (PLDI). New York: ACM Press, 2009. 375−385. [doi: 10.1145/1542476.1542518]

[29] Sankaranarayanan S, Sipma HB, Manna Z. Non-Linear loop invariant generation using Gröbner bases. In: Proc. of the 31st Annual ACM SIGPLAN-SIGACT Symp. on Principles of Programming Languages (POPL). New York: ACM Press, 2004. 318−329. [doi: 10.1145/964001.964028]

[30] Kovács L. Reasoning algebraically about p-solvable loops, tools and algorithms for the construction and analysis of systems. In: Proc. of the 14th Int'l Conf. (TACAS). Budapest: Springer-Verlag, 2008. 249−264. [doi: 10.1007/978-3-540-78800-3_18]

[31] Mastroeni I. Algebraic power analysis by abstract interpretation. Higher-Order and Symbolic Computation, 2004,6300:297−345. [doi: 10.1007/s10990-004-4867-y]

[32] Yang L, Zhan N, Xia B, Zhou C. Program verification by using DISCOVERER. In: Proc. of the 1st IFIP TC 2/WG 2.3 Conf. (VSTTE). Zurich: Springer-Verlag, 2005. 528−538. [doi: 10.1007/978-3-540-69149-5_58]

[33] Yang L, Zhou C, Zhan N, Xia B. Recent advances in program verification through computer algebra. Frontiers of Computer Science in China, 2010,4(1):1−16. [doi: 10.1007/s11704-009-0074-7]

附中文参考文献:

[9] 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. http://cjc.ict.ac.cn/qwjs/view.asp?id=3050 [doi: 10.3724/SP.J.1016.2010.00427]

附录

定理 2. 单变量线性赋值循环 while (){…;$x=a\times x+b$;…},当 $a>0$ 且 $a\neq 1$ 时,存在指数型循环不变式 $x=t_1\times t_2^n+t_3$.其中,$t_1=x_0+b/(a-1)$,$t_2=a$,$t_3=-b/(a-1)$,x_0 为进入循环前 x 的初始值.

证明:根据数学归纳法和循环不变式的定义,当 $x=x_0$,即,x 等于初始值时:

$$x=\left(x_0+\frac{b}{a-1}\right)\times a^n-\frac{b}{a-1}=x_0,$$

其中,a,b 为常量,$n\in\mathbb{N}$.

可以解得:当 $n=0$ 时,等式成立.故存在 n,使得等式成立.

假设当 $x_k=t$ 时($k\in\mathbb{N}$)等式成立,即 $x_k=\left(x_0+\frac{b}{a-1}\right)\times a^{n_k}-\frac{b}{a-1}=t$, 则当 $x_{k+1}=a\times t+b$ 时:

$$x_{k+1}=a\times x_k+b=a\times\left(\left(x_0+\frac{b}{a-1}\right)\times a^{n_k}-\frac{b}{a-1}\right)+b=\left(x_0+\frac{b}{a-1}\right)\times a^{n_k+1}-\frac{b}{a-1},$$

其中,当 $n=n_k+1$ 时,满足指数型循环不变式的形式.故存在 n,使得等式成立. □

引理 1. $\forall n_1,n_2\in\mathbb{N}$,等式 $\left(x_0+\frac{b}{a-1}\right)\times a^{2\times n_1}-\frac{b}{a-1}=\left(a\times x_0+a\times\frac{b}{a-1}\right)\times a^{2\times n_2}-\frac{b}{a-1}$ 不成立,其中,$x_0\neq b/(1-a)$.

证明:假设存在 $n_1,n_2\in\mathbb{N}$,使得 $\left(x_0+\frac{b}{a-1}\right)\times a^{2\times n_1}-\frac{b}{a-1}=\left(a\times x_0+a\times\frac{b}{a-1}\right)\times a^{2\times n_2}-\frac{b}{a-1}$ 成立.

那么有 $\left(x_0+\frac{b}{a-1}\right)\times a^{2\times n_1}=\left(x_0+\frac{b}{a-1}\right)\times a^{1+2\times n_2}$,

化简得 $2\times n_1=1+2\times n_2$,即 $n_1=0.5+n_2$,在 $n_1,n_2\in\mathbb{N}$的条件下,找不到满足条件的 n_1,n_2.

因此,$\left(x_0+\frac{b}{a-1}\right)\times a^{2\times n_1}-\frac{b}{a-1}\wedge\left(a\times x_0+a\times\frac{b}{a-1}\right)\times a^{2\times n_2}-\frac{b}{a-1}=\varnothing(n_1,n_2\in\mathbb{N})$. □

定理 3. 单变量线性赋值循环 while (){…;$x=a\times x+b$;…},当 $a<0$ 时,存在指数型循环不变式:

$$x=(t_1'\times t_2^{n_1}+t_3)\vee(t_1''\times t_2^{n_2}+t_3),$$

其中, $t_1'=x_0+b/(a-1)$,$t_1''=a\times x_0+a\times b/(a-1)$, $t_2=a$,$t_3=-b/(a-1)$,x_0 为进入循环前 x 的初始值,$n_1=n/2$(其中,n mod

2=0),$n_2=(n-1)/2$(其中,$n \bmod 2=1$).

证明:根据数学归纳法和循环不变式的定义,当 $x=x_0$,即,x 等于初始值时:

$$x=\left(x_0+\frac{b}{a-1}\right)\times a^n-\frac{b}{a-1}=\left(x_0+\frac{b}{a-1}\right)\times a^{2\times n_1}-\frac{b}{a-1}\vee\left(a\times x_0+a\times\frac{b}{a-1}\right)\times a^{2\times n_2}-\frac{b}{a-1}=x_0,$$

其中,a,b 为常量.可以解得:当 $n=0$ 时,等式成立.故存在 n,使得等式成立.

假设当 $x_k=t$ 时($k\in\mathbb{N}$),等式成立,即,公式(13)成立:

$$x_k=\left(x_0+\frac{b}{a-1}\right)\times a^{2\times n_1}-\frac{b}{a-1}\vee\left(a\times x_0+a\times\frac{b}{a-1}\right)\times a^{2\times n_2}-\frac{b}{a-1} \tag{13}$$

根据引理 1 可知,公式(13)的析取符号两边的表达式无交集,则公式(13)可化简为

$$x_k=\left(x_0+\frac{b}{a-1}\right)\times a^{2\times n_1}-\frac{b}{a-1}\vee x_k=\left(a\times x_0+a\times\frac{b}{a-1}\right)\times a^{2\times n_2}-\frac{b}{a-1}.$$

当 $x_{k+1}=a\times t+b$ 时,要证明 $x_{k+1}=\left(x_0+\frac{b}{a-1}\right)\times a^{2\times n_1}-\frac{b}{a-1}\vee\left(a\times x_0+a\times\frac{b}{a-1}\right)\times a^{2\times n_2}-\frac{b}{a-1}$,即证明:

$$x_{k+1}=\left(x_0+\frac{b}{a-1}\right)\times a^{2\times n_1}-\frac{b}{a-1}\vee x_{k+1}=\left(a\times x_0+a\times\frac{b}{a-1}\right)\times a^{2\times n_2}-\frac{b}{a-1}.$$

分以下两种情况讨论:

i. 当 $x_k=\left(x_0+\frac{b}{a-1}\right)\times a^{2\times n_1}-\frac{b}{a-1}$ 时:

$$x_{k+1}=a\times x_k+b=a\times\left(\left(x_0+\frac{b}{a-1}\right)\times a^{2\times n_1}-\frac{b}{a-1}\right)+b=\left(a\times x_0+a\times\frac{b}{a-1}\right)\times a^{2\times n_1}-\frac{b}{a-1}.$$

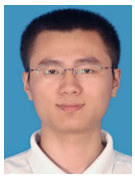
当 $n_1=n_2$ 时,得证.

ii. 当 $x_k=\left(a\times x_0+a\times\frac{b}{a-1}\right)\times a^{2\times n_2}-\frac{b}{a-1}$ 时:

$$x_{k+1}=a\times x_k+b=a\times\left(\left(a\times x_0+a\times\frac{b}{a-1}\right)\times a^{2\times n_2}-\frac{b}{a-1}\right)+b=\left(x_0+\frac{b}{a-1}\right)\times a^{2\times n_2+2}-\frac{b}{a-1}.$$

当 $n_1=n_2+1$ 时,得证.

综合情形 i、情形 ii,得到 $x_{k+1}=\left(x_0+\frac{b}{a-1}\right)\times a^{2\times n_1}-\frac{b}{a-1}\vee\left(a\times x_0+a\times\frac{b}{a-1}\right)\times a^{2\times n_2}-\frac{b}{a-1}$ 成立. □



潘建东(1989－),男,江苏南京人,硕士生,主要研究领域为程序分析与验证,抽象解释,信息安全.

孙浩(1987－),男,博士生,主要研究领域为信息安全,程序分析.

陈立前(1982－),男,博士,助理研究员,CCF 会员,主要研究领域为程序分析与验证,抽象解释.

曾庆凯(1963－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为信息安全,分布计算.

黄达明(1976－),男,博士生,讲师,CCF 会员,主要研究领域为形式化验证和测试,安全评估.