Termination of Two Variable Homogeneous Linear Loops

Yi Li, Chuancan Li, Wenyuan Wu, Yong Feng
Chongqing Key Laboratory of Automated Reasoning and Cognition CIGIT,CAS.
{*liyi@cigit.ac.cn, lichuancan@cigit.ac.cn, wuwenyuan@cigit.ac.cn, yongfeng@cigit.ac.cn*}

***Abstract*—Tiwari gave the quantified formulas to check whether the two variable homogeneous linear loop terminates in 2004. But we find that a certain quantified formula is neglected in Tiwari's result, which corresponds to a possible case of the set NT of nonterminating points. In this paper, through analysing all possible cases of NT, we reconstruct a group of complete quantified formulas to determine if the two variable homogeneous linear loop terminates.**

***Keywords*-Program Verification; Loop Termination; QEP-CAD; Quantifier Elimination;**

I. Introduction

In the 21st, software systems are pervasive in all aspects of society from online shopping to control system of rocket. As the soul of computers, software mediates between the significant parts of our life, any fault happening in them may result in disastrous consequences. For instances, Ariane 5's maiden flight (Ariane 5 Flight 501) on 4 June 1996 failed with the rocket self-destructing in 37 seconds. How to guarantee software systems trustworthy, that is, to design the reliable software systems, is a grand challenge and has received much attention. As one of the bases for designing reliable software, the verification of loop's termination is a difficult problem and unable to always be proved [1].

The dominant method for establishing termination is the use of so-called ranking functions that assign a value from the well-founded domains to each program state. The principle is that, through demonstrating each step in the loop reduces the value assigned by the ranking function and there can be no infinite descending chain of elements of a well-founded domain, we can get the loop must eventually terminate. Hence, how to automatically generate such ranking function have been studied by [2–6].

However, the method of synthesising ranking functions has its own limitations. This is because that the ranking functions of loops may not be easy to be discovered. Thus, if we can not synthesize the ranking function of a given loop, we can not assert the loop is nonterminating. In contrast to the method of using ranking function, [7, 8] prove the decidability of the termination problem for a special class of linear loops over the reals and the integers respectively. [9] further develops the work of [7] by symbolic computing. Following the work of [7, 8], [10] considers the termination problems of more general classes of programs and provides an algorithm to decide the termination of linear programs with nonlinear constraints over the reals.

In addition, some other methods [11–15] are also available to analyse the termination of loop programs. For instance, through analysing finite difference trees, [11] presents a sound but not complete method for proving termination of multipath loops with polynomial guards and assignments. [12] proposes the first known automatic counterexample-guided abstraction refinement algorithm for termination proofs. [13] describes an automatic termination tool TERMINATOR which provides a capacity for large program fragments together with accurate support for programming language features. [14] describes an algorithm for proving the termination of the loops with nonlinear assignments by divergence testing for each variable in the cone-of-influence of the loops termination condition. Basing on transitivity of ranking relations to prove program termination, [15] presents an alternative algorithm that uses a light-weight check.

In this paper, we determine the termination of two variable homogeneous linear loops by analysing all possible cases of its nonterminating set NT.

In 2004, Tiwari presented the corresponding quantified formulas (see Theorem 3 in [7]) to check the termination of two variable homogeneous linear loops. Namely, this kind of loops is terminating iff none of these formulas holds. But we find that a certain quantified formula is neglected in Tiwari's result. Namely, we have found an example which does not terminate, but all of the quantified formulas in Theorem 3 of [7] are false. In contrast to [7], we enumerate all the possible cases of the set NT firstly and then construct the corresponding quantified formulas. This enables us to check whether two variable homogeneous linear loops terminate completely.

The rest of this paper is organized as follows: Section II introduces the loop abstraction and basic concepts. Section III analyses all possible cases of the set NT of two variable homogeneous linear loops in $\Re^2$ and reconstructs a group of complete quantified formulas for determining the termination of two variable homogeneous linear loops. Section IV gives a practical example. Section V concludes.

II. Preliminaries

This section introduces our loop abstraction and basic concepts. Moreover, we restate some of definitions from [7].

Definition 1. The general structure of the homogeneous linear loop over the tuple of program variables $x = (x_1, x_2)$

978-1-4799-4777-5/14 $31.00 © 2014 IEEE
DOI 10.1109/BIFE.2013.3

is the following form (denoted by L2):

$$L2 : while(Bx > 0)\{x := Ax\}$$

where $B = \left\{c_1^T, c_2^T\right\}$ and each c_i^T denotes an 2-dimensional row vector. Namely, B represents an $m \times 2$ matrix, where $m \in \mathbb{Z}^+$. The symbol A represents a 2×2 matrix and the assignment $x := Ax$ is interpreted as being done simultaneously. In addition, the program variables and the matrices A, B are interpreted over the set $\Re$ of reals.

Definition 2. If B of L2 satisfies either of the following conditions,

- The matrix B consists of two linearly independent row vectors.
- B is a non-zero row vector.

then we call this kind of loops the standard form of L2.

Example 1. The following example E1 is the standard form of L2.

$E1 : while(3x + 2y > 0 \wedge y > 0)\{x := 3x - 2y; y := y\}$

Remark 1. It is not difficult to see that any a program L2 can be simplified to its standard form. Therefore, we just need to consider the termination of the standard form of L2. In what following, L2 refer to its standard forms.

Example 2. Given the following E2, we can below get its standard form E3.

$E2 : while(3x + 2y > 0 \wedge y > 0 \wedge x > 0 \wedge 3x - 2y > 0)\{x := 3x - 2y, y := y\}$

$E3 : while(y > 0 \wedge 3x - 2y > 0)\{x := 3x - 2y, y := y\}$

Definition 3. We define Ω as the set consisting of all points making L2 execute,

$$\Omega = \left\{x \in \Re^2 : Bx > 0\right\}$$

where $Bx > 0$ is the loop guard of loop L2. Define $\overline{\Omega} = \Re^2 - \Omega$ to be the set of all points which do not make L2 execute.

Remark 2. Obviously, in 2-dimensional space, if $\Omega \neq \emptyset$, then the region Ω will be a sector.

In following section, we restate some notations for linear programs and some basic properties we require later on from [7].

We denote by the set NT all points on which the loop does not terminate. If program L2 is nonterminating, then

$$NT = \left\{x \in \Re^2 : Bx > 0, BAx > 0, ..., BA^i x > 0, ...\right\}.$$

Tiwari in [7] describes some properties about the set NT of linear programs. Clearly, NT of L2 also has these properties. We conclude them below:

Property 1 (NT of L2).

- $NT \subseteq \Omega$.
- NT is A-invariant, that is, if $v \in NT$, then $Av \in NT$.
- NT is a convex cone, that is, it is closed under addition and scalar multiplication by positive reals.

we can get these properties easily by means of the definition of the set NT. In addition, it is difficult to see that given a set Φ if $\Phi \subseteq \Omega$ and Φ is A-invariant, then $\Phi \subseteq NT$.

Remark 3. Generally, in $\Re^2$, if $NT \neq \emptyset$, the region NT of L2 will be a sector and be specified by its two boundary rays. Especially, if the two boundary rays of the sector coincide, then the region NT of L2 will be a ray.

Let $T = \Re^2 - NT$ denote the set of all points on which the loop terminates. Obviously, any a point in T will fall into the set $\overline{\Omega}$ ($\overline{\Omega} = \Re^2 - \Omega$) after finite iterations. Define the boundary, ∂NT, of NT and T to be the set of all v such that (for all ϵ) there exists a point in the ϵ-neighborhood of v that belongs to T and another that belongs to NT.

Remark 4. Clearly, in $\Re^2$, when the region NT is a sector, all points on the two boundaries of the sector corresponding to NT constitute the set ∂NT. Especially, if the two boundary rays of the sector coincide, i.e., NT is a ray, then $\partial NT = NT \cup \{(0,0)\}$. In addition, Since NT is closed under scalar multiplication by positive reals, so, if any point on one boundary ray of the region NT lies in NT, then all points (excluding (0,0)) on this boundary also lie in NT. Oppositely, we can get if there exists one point of such boundary in T, then all points (excluding (0,0)) of this boundary also lie in T.

III. Termination of L2

A. Analysis of NT of L2 in $\Re^2$

In $\Re^2$, if $NT \neq \emptyset$, we already know the region NT will be a sector or a ray (when the two boundary rays of NT coincide). However, when the region NT is a sector, we can still find some differences between such A-invariant sectors according to the number of the boundaries of the sector NT included in NT. Hence, we can classify all possible cases of the set NT of L2 as follows. For expository convenience, we respectively denote the two boundaries of NT as β_1, β_2 with $(0,0) \notin \beta_i$, $i = 1, 2$, when NT is a sector.

- $NT = \emptyset$.
- $NT \neq \emptyset$.
  - NT is a ray.
  - NT is a sector.
    - $\beta_1 \subseteq T$, $\beta_2 \subseteq T$.
    - $\beta_1 \subseteq NT$, $\beta_2 \subseteq T$ or $\beta_1 \subseteq T$, $\beta_2 \subseteq NT$.
    - $\beta_1 \subseteq NT$, $\beta_2 \subseteq NT$.

B. Quantified Formulas for L2

In Section III-A, we know the region NT of L2 will be a ray or a sector which may include its boundaries. Thus, if $NT \neq \emptyset$, we can respectively describe them by $a^T x = 0 \wedge Bx > 0$ and $a^T x \rhd 0 \wedge b^T x \rhd 0 \wedge x \neq 0$ on which the loop guard always evaluates to true, where $\rhd \in \{>, \geqslant\}$ and a^T, b^T are 2-dimensional row vectors. Let $a = (a_1, a_2)^T$, $b = (b_1, b_2)^T$. The formulas can be expressed as quantified formulas over the theory of linear arithmetic and we get the following complete theorem for checking the termination of L2.

Theorem 1. With the above notions. Loop L2 is non-terminating iff any one of the following quantified formulas is true in the theory of reals.

$$\begin{aligned}&\exists a, b.[\exists x.\phi_1(a, b, x) \wedge \\ &\forall x.(\phi_1(a, b, x) \Rightarrow (Bx > 0 \wedge \phi_1(a, b, Ax)))].\end{aligned} \tag{3.1}$$

where $\phi_1(a, b, x)$ denotes $a^T x > 0 \wedge b^T x > 0$.

$$\begin{aligned}&\exists a, b.[\exists x.\phi_2(a, b, x) \wedge \\ &\forall x.(\phi_2(a, b, x) \Rightarrow (Bx > 0 \wedge \phi_2(a, b, Ax)))].\end{aligned} \tag{3.2}$$

where $\phi_2(a, b, x)$ denotes $a^T x \geq 0 \wedge b^T x > 0$.

$$\begin{aligned}&\exists a, b.[\exists x.\phi_3(a, b, x) \wedge \\ &\forall x.(\phi_3(a, b, x) \Rightarrow (Bx > 0 \wedge \phi_3(a, b, Ax)))].\end{aligned} \tag{3.3}$$

where $\phi_3(a, b, x)$ denotes $a^T x \geq 0 \wedge b^T x \geq 0 \wedge x \neq 0$.

$$\exists a.[\exists x.\phi_4(a, x) \wedge \forall x.(\phi_4(a, x) \Rightarrow (\phi_4(a, Ax)))]. \tag{3.4}$$

where $\phi_4(a, x)$ denotes $a^T x = 0 \wedge Bx > 0$.

Proof: (Sketch) In the case of soundness, as for (3.1)-(3.4), it is not difficult to see that the solution sets Φ of $\phi_i(a, b, x)$, $i \in \{1, 2, 3\}$, and $\phi_4(a, x)$ respectively correspond to a sector and a ray in $\Re^2$. Simultaneously, $\exists x.\phi_i(a, b, x)$ and $\exists x.\phi_4(a, x)$ guarantee $\Phi \neq \emptyset$. Furthermore, the formulas $\forall x.(\phi_i(a, b, x) \Rightarrow (Bx > 0 \wedge \phi_i(a, b, Ax)))$ and $\forall x.(\phi_4(a, x) \Rightarrow (\phi_4(a, Ax)))$ imply that such solution sets Φ is A-invariant and included in Ω. It follows that if any one of (3.1)-(3.4) holds, then there at least exists an A-invariant solution set Φ corresponding to a sector or a ray in Ω. By Property 1, we know that $\Phi \subseteq NT$. Thus, we know if any one of (3.1)-(3.4) holds, then the set NT of L2 must exist.

For completeness, we just need to show if none of (3.1)-(3.4) holds, then $NT = \emptyset$, i.e., L2 terminates. In $\Re^2$, if $NT \neq \emptyset$, there are four possible cases of NT: the ray, the sector with $\beta_1 \subseteq T$, $\beta_2 \subseteq T$, the sector with $\beta_1 \subseteq NT$, $\beta_2 \subseteq T$ or $\beta_1 \subseteq T$, $\beta_2 \subseteq NT$ and the sector with $\beta_1 \subseteq NT$, $\beta_2 \subseteq NT$.

As for (3.1), we can see if it is false, then there does not exist an A-invariant set Φ corresponding to a sector without its two boundaries in Ω. Clearly, NT is an A-invariant set. Hence, we know if (3.1) is false, then the set NT like Φ does not exist in $\Re^2$, i.e., the case of NT as a sector with $\beta_1 \subseteq T$, $\beta_2 \subseteq T$ does not exist.

In the same manner, we can deduce that if (3.2) is false, since NT is an A-invariant set, then the case of NT as a sector with $\beta_1 \subseteq NT$, $\beta_2 \subseteq T$ or $\beta_1 \subseteq T$, $\beta_2 \subseteq NT$ does not exist. Simultaneously, if (3.3) is false, then the case of NT as a sector with $\beta_1 \subseteq NT$, $\beta_2 \subseteq NT$ does not exist. So far if none of (3.1)-(3.3) is true, we can conclude that NT is not a sector.

Next, we claims that if (3.4) does not hold, then the region NT can not be a ray. in (3.4), we use $a^T x = 0 \wedge Bx > 0$ represent a ray included in Ω and guarantee this ray is A-invariant by $\forall x.(\phi_4(a, x) \Rightarrow (\phi_4(a, Ax)))$. Hence, if (3.4) is false, then such a ray does not exist, i.e., the region NT can not be a ray.

Now, we get if none of (3.1)-(3.4) holds, then $NT = \emptyset$. Thus, loop L2 is terminating. This completes the proof. □

In the Theorem 3 of [7], Tiwari presents some quantified formulas that can be used to check whether L2 terminates. But through the verification, we find his result neglects the fact that the region NT may be a ray. In other words, the formula (3.4) is omitted in his Theorem 3. The Theorem 3 of [7] is as follow:

Theorem 2 (Theorem 3 in [7]). A two variable homogeneous linear loop program (i.e. Loop L2).

$while(Bx > 0)\,\{x := Ax\}$,

is non-terminating iff the following sentence is true in the theory of reals

$$\begin{aligned}&\exists a, b.[\exists x.\phi(a, b, x) \wedge \\ &\forall x.(\phi(a, b, x) \Rightarrow (Bx > 0 \wedge \phi(a, b, Ax)))]\end{aligned} \tag{3.5}$$

where $\phi(a, b, x)$ denotes $a^T x \rhd 0 \wedge b^T x \rhd 0 \wedge x \neq 0$ and $\rhd \in \{>, \geq\}$.

Obviously, (3.5) $\Leftrightarrow$ ((3.1) $\vee$ (3.2) $\vee$ (3.3)). We will take the following example to illustrate the formulas in Theorem 3 of [7] is incomplete.

Example 3. We reconsider the following example from [7].

$Q1 : while(x > 0 \wedge y > 0)\{x := -2x + 10y; y := y\}$.

$$B = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad A = \begin{pmatrix} -2 & 10 \\ 0 & 1 \end{pmatrix}$$

As for $Q1$, we use the quantifier elimination tool QEPCAD to eliminate the quantifiers of formulas (3.1)-(3.3). Then, we find that none of these formulas is true. Hence, according to Theorem 3 in [7], $Q1$ should terminate. However, it contradicts with the fact that $Q1$ does not terminate on point (10,3), which is a fixed point.

The reason of the fault is that $\phi(a, b, x)$ in (3.5) can not describe a ray without (0,0) (Even if $a = -b$ in $\phi_3(a, b, x)$, $\phi_3(a, b, x)$ can only describe a straight line excluding (0,0)).

In fact, NT may be a such ray. This induces the case of NT which is a ray to be neglected in Theorem 3 of [7].

IV. Example

Let us take the following example to illustrate how to apply Theorem 1 to check the termination of L2.

Example 4. Reconsider the following L2 from [7]:
$Q2 : while(x > 0 \wedge y > 0)\{x := x - y; y := y\}$.

$$B = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad A = \begin{pmatrix} 1 & -1 \\ 0 & 1 \end{pmatrix}$$

By Theorem 1, we get the corresponding quantified formulas (3.1)-(3.4) as follows:

$$\exists a_1, a_2, b_1, b_2.[[\exists x, y.[a_1x + a_2y > 0 \wedge b_1x + b_2y > 0] \wedge \\ \forall x, y.[[a_1x + a_2y > 0 \wedge b_1x + b_2y > 0] \Rightarrow \\ [x > 0 \wedge y > 0 \wedge a_1(x - y) + a_2y > 0 \wedge \\ b_1(x - y) + b_2y > 0]]].$$

$$\exists a_1, a_2, b_1, b_2.[[\exists x, y.[a_1x + a_2y \geqslant 0 \wedge b_1x + b_2y > 0] \wedge \\ \forall x, y.[[a_1x + a_2y \geqslant 0 \wedge b_1x + b_2y > 0] \Rightarrow \\ [x > 0 \wedge y > 0 \wedge a_1(x - y) + a_2y \geqslant 0 \wedge \\ b_1(x - y) + b_2y > 0]]].$$

$$\exists a_1, a_2, b_1, b_2.[[\exists x, y.[a_1x + a_2y \geqslant 0 \wedge b_1x + b_2y \geqslant 0] \wedge \\ \forall x, y.[[a_1x + a_2y \geqslant 0 \wedge b_1x + b_2y \geqslant 0] \Rightarrow \\ [x > 0 \wedge y > 0 \wedge a_1(x - y) + a_2y \geqslant 0 \wedge \\ b_1(x - y) + b_2y \geqslant 0]]].$$

$$\exists a_1, a_2.[[\exists x, y.[a_1x + a_2y = 0 \wedge x > 0 \wedge y > 0] \wedge \\ \forall x, y.[[a_1x + a_2y = 0 \wedge x > 0 \wedge y > 0] \Rightarrow \\ [a_1(x - y) + a_2y = 0 \wedge x - y > 0 \wedge y > 0]]].$$

Then, we eliminate the quantifiers in above formulas by the tool QEPCAD. Finally, we find that none of these is true. It immediately follows that $Q2$ terminates.

V. Conclusion

In this paper, through analysing all possible cases of the set NT in $\Re^2$, we reconstruct the quantified formulas to determine if the loop L2 terminates. This makes up the results in Theorem 3 of [7].

In addition, by analysing the positional relationship between NT and Ω, we already get some results about simplifying the quantified formulas in Theorem 1. More to the point, we can determine termination of the initialized L2 basing on analysing all possible cases of NT. In general, the initialized loop widely exists in practical code. However, it is difficult and different from termination of uninitialized one. Generally, a uninitialized loop does not terminates iff the set NT is not empty. So, for the initialized one, we must check whether the initial value of such loop lies in the set NT. If not, we can say this initialized loop terminates. We will present the relative results in another paper.

Acknowledgment

This work was partially supported by NNSFC(61103110), NSF of Chongqing (cstc2012ggB40004, cstc2012pt-sy40001, cstc2013jjys40001).

References

[1] B. Cook, "Principles of program termination," *Engineering Methods and Tools for Software Safety and Security: Volume 22 NATO Science for Peace and Security Series-D: Information and Communication Security*, vol. 22, p. 161, 2009.

[2] D. Dams, R. Gerth, and O. Grumberg, "A heuristic for the automatic generation of ranking functions," in *Workshop on advances in verification*. Citeseer, 2000.

[3] M. A. Colóon and H. B. Sipma, "Synthesis of linear ranking functions," in *Tools and Algorithms for the Construction and Analysis of Systems*. Springer, 2001, pp. 67–81.

[4] M. A. Colón and H. B. Sipma, "Practical methods for proving program termination," in *Computer Aided Verification*. Springer, 2002, pp. 442–454.

[5] A. Podelski and A. Rybalchenko, "A complete method for the synthesis of linear ranking functions," in *Verification, Model Checking, and Abstract Interpretation*. Springer, 2004, pp. 239–251.

[6] Y. Chen, B. Xia, L. Yang, N. Zhan, and C. Zhou, "Discovering non-linear ranking functions by solving semi-algebraic systems," in *Theoretical Aspects of Computing–ICTAC 2007*. Springer, 2007, pp. 34–49.

[7] A. Tiwari, "Termination of linear programs," in *Computer Aided Verification*. Springer, 2004, pp. 70–82.

[8] M. Braverman, "Termination of integer linear programs," in *Computer aided verification*. Springer, 2006, pp. 372–385.

[9] B. Xia, L. Yang, N. Zhan, and Z. Zhang, "Symbolic decision procedure for termination of linear programs," *Formal Aspects of Computing*, vol. 23, no. 2, pp. 171–190, 2011.

[10] B. Xia and Z. Zhang, "Termination of linear programs with nonlinear constraints," *Journal of Symbolic Computation*, vol. 45, no. 11, pp. 1234–1249, 2010.

[11] A. R. Bradley, Z. Manna, and H. B. Sipma, "Termination of polynomial programs," in *Verification, Model Checking, and Abstract Interpretation*. Springer, 2005, pp. 113–129.

[12] B. Cook, A. Podelski, and A. Rybalchenko, "Abstraction refinement for termination," in *Static Analysis*. Springer, 2005, pp. 87–101.

[13] B. Cook, A. Podelski, and Rybalchenko, "Termination proofs for systems code," in *ACM SIGPLAN Notices*, vol. 41, no. 6. ACM, 2006, pp. 415–426.

[14] D. Babic, B. Cook, A. Hu, and Z. Rakamaric, "Proving termination of nonlinear command sequences," *Formal Aspects of Computing*, vol. 25, no. 3, pp. 389–403, 2013.

[15] D. Kroening, N. Sharygina, A. Tsitovich, and C. M. Wintersteiger, "Termination analysis with compositional transition invariants," in *Computer Aided Verification*. Springer, 2010, pp. 89–103.