Journal of Zhejiang University SCIENCE A
ISSN 1673-565X (Print); ISSN 1862-1775 (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Equality detection for linear arithmetic constraints*

Li LI[†1,3], Kai-duo HE[1,3], Ming GU[2,3], Xiao-yu SONG[4]

([1]Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China)

([2]School of Software, Tsinghua University, Beijing 100084, China)

([3]MOE Key Laboratory for Information System Security, Beijing 100084, China)

([4]Department of Electrical and Computer Engineering, Portland State University, Oregon 97207, USA)

[†]E-mail: l-li-05@mails.tsinghua.edu.cn

Received Nov. 12, 2008; Revision accepted May 4, 2009; Crosschecked Aug. 14, 2009; Published online Oct. 21, 2009

Abstract: Satisfiability modulo theories (SMT) play a key role in verification applications. A crucial SMT problem is to combine separate theory solvers for the union of theories. In previous work, the simplex method is used to determine the solvability of constraint systems and the equalities implied by constraint systems are detected by a multitude of applications of the dual simplex method. We present an effective simplex tableau-based method to identify all implicit equalities such that the simplex method is harnessed to an irreducible minimum. Experimental results show that the method is feasible and effective.

Key words: Model checking, Satisfiability modulo theories (SMT), Linear arithmetic
doi:10.1631/jzus.A0820812 **Document code:** A **CLC number:** TP31

INTRODUCTION

Satisfiability modulo theories (SMT) are widely used in model checking, artificial intelligence (AI) planning, and scheduling (Barrett *et al.*, 2005). The SMT problem is a decision problem for logic formulas in the combinations of background theories expressed in first-order logic with equality. Quantifier-free linear arithmetic inequality (LI) is an important theory in SMT. Decision procedures for LI determine whether a Boolean combination of linear equalities and inequalities is satisfiable.

Among many solvers, Simplify (Detlefs *et al.*, 2005) and CVC-Lite (Barrett and Berezin, 2004) are Nelson-Oppen style decision procedures using the 'equality-sharing' method to prove satisfiability for combined theories (Nelson, 1981). For general linear arithmetic, existing tools rely either on Fourier-Motzkin elimination (Dantzig, 1973) (e.g., CVC-Lite (Barrett and Berezin, 2004), CVC (Stump *et al.*, 2002) or SVC (Barrett *et al.*, 1996)) or on simplex methods (Chvatal, 1983) (e.g., MathSat (Bozzano *et al.*, 2005), ICS (Filliatre *et al.*, 2001), Simplify, Yices, or ARIO (Sheini and Sakallah, 2005)). Several tools, e.g., Barcelogic (Nieuwenhuis and Oliveras, 2005) and Slice (Wang *et al.*, 2005), are specialized for the difference-logic fragment of linear arithmetic and rely on graph algorithms.

Detecting implicit equalities is a fundamental operation in solving generalized linear constraints (Lassez and McAloon, 1992). When using the simplex method to decide the satisfiability of linear constraints, the method of finding equalities is simplex correlative. An implicit equality is an inequality $a_1x_1+a_2x_2+\ldots+a_nx_n\le b_0$ from a set of constraints M that can be converted into the equation $a_1x_1+a_2x_2+\ldots+a_nx_n=b_0$ without changing the set of solutions of M. For instance, in the set $\{x_1\le x_2,\ x_1-x_3\ge x_2,\ x_3\ge 0\}$ all inequalities are implicit. This set can be rewritten as $\{x_1=x_2,\ x_1-x_3=x_2,\ x_3=0\}$ without changing its set of solutions.

This paper presents a variant simplex method to check the satisfiability of linear constraints based on

* Project supported by the National Natural Science Foundation of China (Nos. 60635020 and 90718039) and the National Basic Research Program (973) of China (No. 2004CB719406)

the work addressed in Kroening and Strichman (2008). This is a novel method for detecting equality of which both soundness and completeness are proved.

SATISFIABILITY OF LINEAR ARITHMETIC INEQUALITIES

Consider a system M of linear arithmetic inequalities (LI):

$$\sum_{j=1}^{n} a_{ij}x_j \nabla b_i, \quad x_j, b_i \geq 0, \tag{1}$$

where $\nabla \in \{\leq, =, \geq\}$, $a_{ij}, x_j, b_i \in \mathbb{R}$, $1 \leq i \leq m$, and $1 \leq j \leq n$.

M is satisfiable if there exists a solution $\boldsymbol{x}=[x_1, x_2, \ldots, x_n] \in \mathbb{R}^n$ which satisfies Eq.(1). The matrix formulation of Eq.(1) is given as follows:

$$\boldsymbol{Ax} \nabla \boldsymbol{b}, \ \boldsymbol{x}, \boldsymbol{b} \geq \boldsymbol{0}, \ \boldsymbol{A} = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix}, \tag{2}$$

where $\boldsymbol{A}$ is an $m \times n$-coefficient matrix, $\boldsymbol{x}$ is an n-dimensional column vector of variables, and $\boldsymbol{b}$ is an m-dimensional column vector of constants.

The satisfiability problem of M can be converted to a constrained programming problem (Kroening and Strichman, 2008). A variant method of Kroening and Strichman (2008) is described as follows.

Step 1: By introducing a slack variable $\boldsymbol{y}=[y_1, y_2, \ldots, y_k, \ldots, y_m]$ (Vanderbei, 2001), we transform the system M to a satisfiability equivalent system of equalities M' (Kroening and Strichman, 2008):

$$\boldsymbol{A\Gamma} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{bmatrix} = \boldsymbol{b}, \quad \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{b} \geq \boldsymbol{0}, \tag{3}$$

where $\boldsymbol{\Gamma}=\text{diag}\{c_{11}, \ldots, c_{ii}, \ldots, c_{mm}\}$ is a quasi-identity matrix and $c_{ii} \in \{-1, 0, 1\}$, $1 \leq i \leq m$.

Step 2: Introduce an artificial variable $\boldsymbol{z}=[z_1, z_2, \ldots, z_h, \ldots, z_m]$ (Vanderbei, 2001):

$$\boldsymbol{IA\Gamma} \begin{bmatrix} \boldsymbol{z} \\ \boldsymbol{x} \\ \boldsymbol{y} \end{bmatrix} = \boldsymbol{b}, \quad \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{b} \geq \boldsymbol{0}, \tag{4}$$

where $\boldsymbol{I}=\text{diag}\{d_{11}, \ldots, d_{ii}, \ldots, d_{mm}\}$ ($d_{ii}=1 \ \forall 1 \leq i \leq m$) is an identity matrix. Eq.(4) is called a restricted normal form where each equality is of the form $z_h+c_{ik}y_k + \sum_{j=1}^{n} a_{ij}x_j = b_i$, where $i=k=h$.

Step 3: By introducing the goal function $\max \sum_{i=1}^{m} \sum_{h=1}^{m} (-d_{ih}z_h)$ in the constraints of Eq.(4), we obtain a constrained optimization problem M'':

$$\max \sum_{i=1}^{m} \sum_{h=1}^{m} (-d_{ih}z_h) = \sum_{i=1}^{m} \left(-b_i + \sum_{j=1}^{n} a_{ij}x_j + \sum_{k=1}^{m} c_{ik}y_k \right)$$
$$\text{subject to} \quad \boldsymbol{IA\Gamma} \begin{bmatrix} \boldsymbol{z} \\ \boldsymbol{x} \\ \boldsymbol{y} \end{bmatrix} = \boldsymbol{b}, \quad \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{b} \geq \boldsymbol{0}. \tag{5}$$

Step 4: By using the simplex tableau method, we determine if Eq.(5) has an optimal solution 0. If it is true, then the system M is satisfiable. The initial simplex tableau is shown in Table 1, where the obj column is the basic variable and the B column is the constant item.

Table 1 The initial simplex tableau

obj	$x_1, \ldots, x_j, \ldots, x_n$	$y_1, \ldots, y_k, \ldots, y_m$	$z_1, \ldots, z_h, \ldots, z_m$	B
max	$\sum_{i=1}^{m} a_{i1}, \cdots, \sum_{i=1}^{m} a_{ij}, \cdots, \sum_{i=1}^{m} a_{in}$	$\sum_{i=1}^{m} c_{i1}, \cdots, \sum_{i=1}^{m} c_{ik}, \cdots, \sum_{i=1}^{m} c_{im}$	**0**	$\sum_{i=1}^{m} b_i$
$\boldsymbol{z}$	$\boldsymbol{A}$	$\boldsymbol{\Gamma}$	$\boldsymbol{I}$	$\boldsymbol{b}$

Theorem 1 If M is satisfiable, M'' has an optimal solution 0.

Proof Notice that, M has a solution$\Leftrightarrow M''$ has an optional solution 0.

First prove '$\Rightarrow$'. Assume that $\{x_1, x_2, \ldots, x_n\}$ is a feasible solution of M and $\{x_1, x_2, \ldots, x_n; y_1, y_2, \ldots, y_m\}$ is the corresponding feasible solution for M'. When substituting the system M by Eq.(3) and adding up each row, we have

$$\sum_{i=1}^{m} \left(\sum_{j=1}^{n} a_{ij}x_j + \sum_{k=1}^{m} c_{ik}y_k \right) = \sum_{i=1}^{m} b_i. \tag{6}$$

Therefore if using the valuations $\{\boldsymbol{z}=\boldsymbol{0}; \boldsymbol{x}=[x_1, x_2, \ldots, x_n]; \boldsymbol{y}=[y_1, y_2, \ldots, y_m]\}$ for system M'', we reach an optimal solution $\sum_{i=1}^{m}\sum_{h=1}^{m}(-d_{ih}z_h)=0.$

Then prove '⇐'. If there is an optimal solution [$\boldsymbol{z}$; $\boldsymbol{x}$; $\boldsymbol{y}$] making goal function $\sum_{i=1}^{m}\sum_{h=1}^{m}(-d_{ih}z_h)=0,$ by $z_h \ge 0$ ($1 \le h \le m$) in constraints of the system M'', we believe that the optimal solution must be [$\boldsymbol{z}$=**0**; $\boldsymbol{x}$; $\boldsymbol{y}$]. The reduced [$\boldsymbol{x}$; $\boldsymbol{y}$] is thus a feasible solution for the system M' and corresponding to M.

Example 1 We determine the satisfiability of the following set of LI constraints:

$$\begin{cases} x_1 - x_2 \le 8, \\ x_1 + x_2 - x_3 \ge 8, \\ x_1, x_2, x_3 \ge 0. \end{cases}$$

By introducing slack variables y_1, y_2 and artificial variables z_1, z_2, we have

$$\max(-(z_1+z_2)) = -16 + (2x_1 - x_3 + y_1 - y_2),$$
$$\begin{cases} z_1 + x_1 - x_2 + y_1 = 8, \\ z_2 + x_1 + x_2 - x_3 - y_2 = 8, \\ x_1, x_2, x_3, y_1, y_2, z_1, z_2 \ge 0. \end{cases}$$

We construct the constraint programming with the following objectives. By considering the initial simplex tableau, we have

obj	x_1	x_2	x_3	y_1	y_2	z_1	z_2	B
max	2	0	−1	1	−1	0	0	16
z_1	1	−1	0	1	0	1	0	8
z_2	1	1	−1	0	−1	0	1	8

Using the standard simplex method (Chvatal, 1983), we attain the following final tableau, and the constraint programming acquires an optimal solution 0. Thus, the original problem is satisfiable.

obj	x_1	x_2	x_3	y_1	y_2	z_1	z_2	B
max	0	0	0	0	0	−1	−1	0
z_1	1	0	−0.5	0.5	−0.5	0.5	0.5	8
z_2	0	1	−0.5	−0.5	−0.5	−0.5	0.5	0

In the final simplex tableau, column B contains only constant items, variables in obj-column are basic variables, and other variables are non-basic ones (Chvatal, 1983).

Theorem 2 If M'' has an optimal solution 0 (or M is satisfiable), then $\sum_{h=1}^{m} d_{ih}z_h = 0$ for each row of the final simplex tableau.

Proof M'' has an optimal solution 0 such that the goal function $\sum_{i=1}^{m}\sum_{h=1}^{m}(-d_{ih}z_h)=0,$ whereas $z_h \ge 0$ ($1 \le h \le m$) in M'' and the final simplex tableau is an elementary operation of rows and columns of M''. Hence, $z_h=0$ ($1 \le h \le m$) for each row in the final simplex tableau. We obtain $d_{ih}z_h=0$ ($1 \le h \le m$) and $\sum_{h=1}^{m} d_{ih}z_h = 0.$

EQUALITY DETECTION

A key problem occurring in SMT is to combine separate theory solvers for the union of theories. The Nelson-Oppen method (Nelson, 1981) chooses a sufficient condition to combine two theories over disjoint signatures. Disjunctions of equalities over shared variables that are implied by one of the theories are propagated. If the system of constraints M is satisfiable, we have to find the implied equalities. The simplex is usually adopted to determine the solvability of linear arithmetic constraints M and to decide if M has an implicit equality.

In Stuckey (1991), the dual simplex was used many times to find all equalities. A similar method was employed in Refalo (1998) where each time the equality was identified by revoking the simplex once. Both methods are time consuming due to the complexity of the simplex. Dutertre and de Moura (2006) translated the original constraints into a series of equalities and bounded inequalities by introducing new variables, and then adopted a variant of simplex to decide satisfiability. Since dealing only with the linear arithmetic theory, they need not detect all equalities between variables. Detlefs *et al.*(2005) presented an incremental constraint solver for linear arithmetic inequalities constraints based on the simplex algorithm. Four cases were provided for detecting equalities. Rueß and Shankar (2004) extended this method using heuristic-based static analysis to avoid unwanted pivoting for detecting equalities.

Here, we present a similar but more effective simplex tableau-based method to identify all implicit equalities such that the simplex is employed only once.

Process of detection equality

For a satisfiable LI system M'' and its final simplex tableau with coefficients a_{ij}, c_{ik}, d_{ih} of x_j, y_k, z_h ($1\le i, k, h\le m$; $1\le j\le n$), respectively, we partition the tableau into two parts by the constant item b_i: 0-rows with $b_i=0$ and positive-rows with $b_i>0$. Let m_0 be the number of 0-rows, $0\le m_0\le m$. The following procedure finds all the implied equalities.

Step 1: Deletion of the $\boldsymbol{z}$-columns.

Delete the $\boldsymbol{z}$-columns from the simplex tableau because $\sum_{h=1}^{m} d_{ih}z_h = 0$ by Theorem 2, and then delete the max-row.

Step 2: Detection of $x_j=0$ or $y_k=0$, where $1\le j\le n$, $1\le k\le m$. For the 0-rows part with $b_i=0$ in the simplex tableau:

Step 2.1: Scan each row. If all the coefficients of non-basic variables are zero, then the basic variable $x_i=0$. Delete the row and columns located.

Step 2.2: Scan each row. If all non-zero coefficients are positive at the same time, we have $\sum_{j=1}^{n} a_{ij}x_j + \sum_{k=1}^{m} c_{ik}y_k = 0$. For $\boldsymbol{x}$, $\boldsymbol{y}\ge\boldsymbol{0}$, we obtain $x_j=y_k=0$ and delete the row and columns located.

Step 2.3: Scan every n'-row, $n'\in\{2, 3, \ldots, m_0\}$. If $\sum_{i=1}^{n'} a_{ij} = \sum_{i=1}^{n'} c_{ik} = 0$ for every non-basic variable, then all basic variables $x_i=0$ in these n'-rows. Delete the columns located.

Go back to Step 2.1 and scan again, till having no deletion in the simplex tableau.

Step 3: Detection of $x_j=y_k$, where $1\le j\le n$, $1\le k\le m$. For the 0-rows part with $b_i=0$ in the residue simplex tableau:

Step 3.1: Scan each row in the residue tableau. If there are only two variables v_1, $v_2\in\{x_j, y_k|1\le j\le n, 1\le k\le m\}$ with non-zero coefficients a'_{v_1}, a'_{v_2}, respectively, and if $|a'_{v_1}| = |a'_{v_2}|$ such that one is positive and the other is negative, then $v_1=v_2$.

Step 3.2: Scan every two rows. If all the coefficients of non-basic variables are equal, then the corresponding basic variables $x_{i_1} = x_{i_2}$, where i_1, $i_2\in\{1, 2, \ldots, m\}$.

Example 2 Given a set of LI constraints M

$$\begin{cases} x_1 \le x_2, \\ x_2 + x_3 \le x_1, \\ x_1, x_2, x_3 \ge 0, \end{cases}$$

by transforming it into the programming objective form, we obtain the final simplex tableau as follows:

obj	x_1	x_2	x_3	y_1	y_2	z_1	z_2	B
max	0	0	0	0	0	−1	−1	0
x_1	1	−1	0	1	0	1	0	0
x_3	0	0	1	1	1	1	1	0

At Step 1, we obtain a simplified simplex tableau as follows:

obj	x_1	x_2	x_3	y_1	y_2	B
x_1	1	−1	0	1	0	0
x_3	0	0	1	1	1	0

At Step 2, by considering row 3, we obtain

$$\begin{cases} x_3 + y_1 + y_2 = 0, \\ \boldsymbol{x}, \boldsymbol{y} \ge \boldsymbol{0}. \end{cases}$$

Therefore $x_3=y_1=y_2=0$. By deleting the corresponding rows and columns, we have

obj	x_1	x_2	B
x_1	1	−1	0

By Step 3, we have $x_1=x_2$.

Step 4: Detection of $x_i=b$ or $x_{i_1} = x_{i_2}$, where i, i_1, $i_2\in\{1, 2, \ldots, m\}$. For the positive-rows part with $b_i>0$ in the residue simplex tableau:

Step 4.1: Scan each row. If $b_i=b$ and there is only one non-zero coefficient in this row, then we attain the equality $x_i=b$.

Step 4.2: Scan every two rows with $b_{i_1} = b_{i_2} = b$. If all the coefficients of non-basic variables are equal, then the corresponding basic variables $x_{i_1} = x_{i_2}$.

Example 3 For a set of LI constraints

$$\begin{cases} x_1 - x_3 = 3, \\ x_2 - x_3 = 3, \\ x_1, x_2, x_3 \ge 0, \end{cases}$$

by transforming it into objective programming, we attain the final simplex tableau through Steps 1~3:

obj	x_1	x_2	x_3	B
x_1	1	0	−1	3
x_2	0	1	−1	3

By Step 4, at the last two rows of the tableau, the coefficients of the non-basic variable x_3 are equal and the constant items are equal, thus $x_1=x_2$.

Correctness of the process

Theorem 3 (Soundness) For a system M of linear arithmetic inequalities, the equality derived by the proposed procedure is an equality implied by M.

Proof Clear evidence has been shown in the above procedure.

The procedure of equality detection is implemented by Steps 1~4. After Step 1, and renaming the variables by basic variables and B-column (Step 1), we obtain a simplified simplex tableau as follows (b>0):

obj	$\boldsymbol{v}_{\text{basic}}$	$\boldsymbol{v}_{\text{non-basic}}$	B
$\boldsymbol{v}_{\text{basic}}$	$\boldsymbol{I}$	$\boldsymbol{a}_{\text{non-basic}}$	$\boldsymbol{0}$
			$\boldsymbol{b}$

We enumerate all the possibilities. For v_{basic}, v'_{basic}, $v_{\text{non-basic}}$, $v'_{\text{non-basic}}\in\{x_j, y_k|1\le j\le n, 1\le k\le m\}$, and b>0, all the equalities implied by M belong to one of the following forms:

1. $v_{\text{basic}}=0$ or $v_{\text{non-basic}}=0$, i.e., basic or non-basic variables equal 0.
2. $v_{\text{basic}}=b$, i.e., basic variables are positive constants.
3. $v_{\text{non-basic}}=b$, i.e., non-basic variables are positive constants.
4. $v_{\text{basic}}=v'_{\text{basic}}$, i.e., a basic variable equals another basic variable.
5. $v_{\text{non-basic}}=v'_{\text{non-basic}}$, i.e., a non-basic variable equals another non-basic variable.
6. $v_{\text{basic}}=v_{\text{non-basic}}$, i.e., a basic variable equals a non-basic variable.

Since Steps 2 and 3 are implemented by examining 0-rows and Step 4 is implemented by examining positive rows, we introduce Lemma 1 first:

Lemma 1 For the final simplex tableau, it is impossible to obtain the equalities formed as equality 3 ($v_{\text{non-basic}}=b$, b>0) by any elementary transformation on rows.

Proof The simplex method maintains as invariant that each row in the tableau contains exactly one basic variable with coefficient 1. The coefficient to the variable is 0 in all other rows. Therefore, no row can be used to imply equality 3. Since the basic variables are different in each row, equality 3 also cannot be derived by combining two rows.

Lemma 2 After Steps 1~4, it is impossible to obtain more equalities formed as equalities 2 ($v_{\text{basic}}=b$), 4 ($v_{\text{basic}}=v'_{\text{basic}}$), 5 ($v_{\text{non-basic}}=v'_{\text{non-basic}}$), and 6 ($v_{\text{basic}}=v_{\text{non-basic}}$) by any elementary operations on more than two rows.

Proof Using more than two rows, there exist three or more basic variables whose coefficients are not equal to 0. By means of elementary operations on rows, we cannot remove any one of basic variables. Therefore, it is impossible to obtain equalities in the form of equality 2, nor can we obtain equalities between variables in the form of equalities 4~6.

Theorem 4 (Completeness) For a system M of linear arithmetic inequalities, the proposed procedure can find all the equalities implied by M.

Proof The equality detection procedure consists of two parts: (a) find all the equalities 1 by Step 2, and (b) locate the other equalities 2~6 in the residue tableau by Steps 3 and 4.

Case (a): As Step 2.1 is a limited matching on each 0-row, and Steps 2.2 and 2.3 are both limited searching and pattern matching on 0-rows, we can find all the equalities formed as equality 1.

Case (b): By Lemma 1, the equalities formed as equality 3 do not appear. Step 3 works on the residue simplex tableau with b_i=0, while Step 4 with b_i>0. Both of them are also a finite pattern matching process, and thus we can detect the remaining cases equalities 2 and 4~6. Moreover, operating on three or more rows, we cannot obtain more equalities using Lemma 2. Therefore, the derived equalities are all equalities implied by M, and the detection procedure is complete.

CONCLUDING REMARKS

As a part of a software model checking tool Jchecker1 (http://code.google.com/p/fopic/), we developed a decision procedure named Fopic (http://code.google.com/p/jchecker/) for linear inequality and equality with un-interpreted function (UIF) symbols theories. The Fopic can also be used alone as a decision procedure. To test the efficiency, we extract formulas from actual C programs (http://mtc.epfl.ch/software-tools/blast/) (for practical results of the experiment, see Table 1). To state the performance, we ran these benchmarks with a similar tool Foci (http://www.kenmcmil.com/foci.html) (McMillan, 2004). The results are listed in Table 2.

Table 2 Results of the experiment

Tool	sat	unsat	Time (s)
Foci	36	14	768
Fopic	35	15	542

In this paper, we presented a variant simplex-based solver for solving SMT problems dedicated to linear arithmetic. The main features of this approach include the completeness to detect the implied equalities by introducing artificial variables, which can also be generally applied to the up-to-date SMT technology such as fast backtracking and graph-based solvers. Experimental results show that the method is feasible and efficient.

References

Barrett, C., Berezin, S., 2004. CVC Lite: a new implementation of the cooperating validity checker. *LNCS*, **3114**:515-518.

Barrett, C., Dill, D., Levitt, J., 1996. Validity checking for combinations of theories with equality. *LNCS*, **1166**:187-201. [doi:10.1007/BFb0031808]

Barrett, C., de Moura, L., Stump, A., 2005. SMT-COMP: satisfiability modulo theories competition. *LNCS*, **3576**: 20-23. [doi:10.1007/11513988_4]

Bozzano, M., Bruttomesso, R., Cimatti, A., Junttila, T., Rossum, P.V., Schulz, S., Sebastiani, R., 2005. The MathSAT 3 System. *LNCS*, **3632**:315-321. [doi:10.1007/11532231_23]

Chvatal, V., 1983. Linear Programming. Springer-Verlag New York, LLC, p.34-45.

Dantzig, G.B., 1973. Fourier-Motzkin elimination and its dual. *J. Comb. Theory Ser. A*, **14**(3):288-297. [doi:10.1016/0097-3165(73)90004-6]

Detlefs, D., Nelson, G., Saxe, J.B., 2005. Simplify: a theorem prover for program checking. *J. ACM*, **52**(3):365-473. [doi:10.1145/1066100.1066102]

Dutertre, B., de Moura, L., 2006. A fast linear-arithmetic solver for DPLL(T). *LNCS*, **4144**:81-94. [doi:10.1007/11817963_11]

Filliatre, J.C., Owre, S., Rueb, H., Shankar, N., 2001. ICS: Integrated Canonization and Solving. Int. Conf. on Computer Aided Verification, p.246-249.

Kroening, D., Strichman, O., 2008. Decision Procedures: An Algorithmic Point of View. Springer-Verlag New York Inc., p.113-122.

Lassez, J.L., McAloon, K., 1992. A canonical form for generalized linear constraints. *J. Symb. Comput.*, **13**(1):1-24. [doi:10.1016/0747-7171(92)90002-L]

McMillan, K.L., 2004. An Interpolating Theorem Prover. TACAS 2004: Tools and Algorithms for Construction and Analysis of Systems, p.102-121.

Nelson, G., 1981. Techniques for Program Verification. Technical Report, CSL-81-10, Xerox Palo Alto Research Center, Palo Alto, CA.

Nieuwenhuis, R., Oliveras, A., 2005. DPLL(T) with Exhaustive Theory Propagation and Its Application to Difference Logic. Int. Conf. on Computer Aided Verification, p.321-324.

Refalo, P., 1998. Approaches to the Incremental Detection of Implicit Equalities with the Revised Simplex Method. Proc. 10th Int. Symp. on Principles of Declarative Programming, p.481-496. [doi:10.1007/BFb0056634]

Rueß, H., Shankar, N., 2004. Solving Linear Arithmetic Constraints. Technical Report, CSL-SRI-04-01, SRI International, Menlo Park, CA, USA.

Sheini, H.M., Sakallah, K.A., 2005. A scalable method for solving satisfiability of integer linear arithmetic logic. *LNCS*, **3569**:241-256. [doi:10.1007/11499107_18]

Stuckey, P.J., 1991. Incremental linear constraint solving and detection of implicit equalities. *ORSA J. Comput.*, **3**(4): 269-271.

Stump, A., Barrett, C.W., Dill, D.L., 2002. CVC: a cooperating validity checker. *LNCS*, **2404**:500-504. [doi:10.1007/3-540-45657-0_40]

Vanderbei, R.J., 2001. Linear Programming: Foundations and Extensions (2nd Ed.). Springer-Verlag New York, LLC, p.16-22.

Wang, C., Ivancic, F., Ganai, M., Gupta, A., 2005. Deciding separation logic formulae by SAT and incremental negative cycle elimination. *LNCS*, **3835**:322-336. [doi:10.1007/11591191_23]