Toward an Efficient Equality Computation in Connection Tableaux: A Modification Method without Symmetry Transformation[1] — A Preliminary Report—

Koji Iwanuma[§1], **Hidetomo Nabeshima**[§1], **and Katsumi Inoue**[§2]

[§1]**University of Yamanashi**, 4-3-11 Takeda, Kofu-shi, Yamanashi, 400-8511, Japan
Email: {iwanuma,nabesima}@yamanashi.ac.jp

[§2]**National Institute of Informatics**, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo, 101-8430, Japan. Email: ki@nii.ac.jp

Abstract

In this paper, we study an efficient equality computation in connection tableaux, and give a new variant of Brand, Bachmair-Ganzinger-Voronkov and Paskevich's modification methods, where the symmetry elimination rule is never applied. As is well known, effective equality computing is very difficult in a top-down theorem proving framework such as connection tableaux, due to a strict restriction to re-writable terms. The modification method with ordering constraints is a well-known remedy for top-down equality computation, and Paskevich adapted the method to connection tableaux. However the improved modification method still causes essentially redundant computation which originates in a symmetry elimination rule for equational clauses. The symmetry elimination may produce an exponential number of clauses from a given single clause, which inevitably causes a huge amount of redundant backtracking in connection tableaux. In this paper, we study a simple but effective remedy, that is, we abandon such symmetry elimination for clauses and instead introduce new equality inference rules into connection tableaux. These new inference rules have a possibility of achieving efficient equality computation, without losing the symmetry property of equality, which never cause redundant backtracking nor redundant contrapositive computation. We implemented the proposed methods in a sophisticated prover SOLAR which is originally designed to finding logical consequences, and show a preliminary experimental results for TPTP benchmark problems. This research is now in progress, thus the experimental results provided in this paper are tentative ones.

1 Introduction

In this paper, we study an efficient equality computation in connection tableaux, and give a variant of modification methods investigated by Brand [2], Bachmair-Ganzinger-Voronkov [1] and Paskevich [9]. We investigate a novel modification method such that a symmetry elimination rule is never applied.

[1]This research was partially supported by the Grant-in-Aid from The Ministry of Education, Science and Culture of Japan ((A) No.20240016)

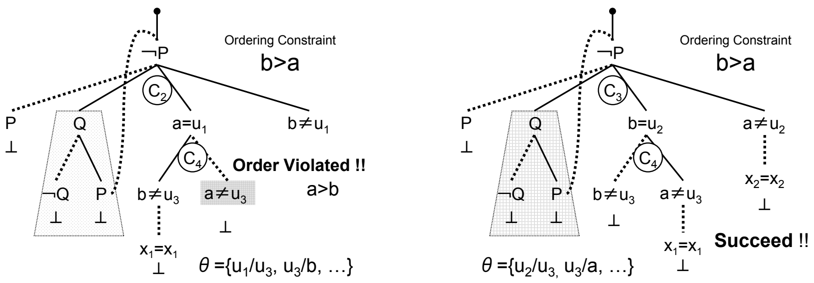

Figure 1: Connection Tableaux for Modification with ordering constraints

As is well known, effective equality [1, 3] computing is very difficult in a top-down theorem proving framework such as connection tableaux [6], due to a strict restriction to re-writable terms [12]. The modification method proposed by Brand has the great possibility for improving top-down equality computation. Bachmair, Ganzinger and Voronkov improved Brand's method with ordering constraints, and Paskevich adapted connection tableau calculus to the modification method using ordering. However the improved connection tableaux still causes redundant computation which is essentially involved by a symmetry elimination rule for equational clauses. The symmetry elimination may produce an exponential number of clauses from a given single clause, which inevitably causes a huge amount of redundant backtracking in Connection Tableaux.

Let $\mathcal{S}_1$ be a set of clauses $\{\, \neg P,\ P \vee Q \vee a \approx b,\ b \not\approx a,\ \neg Q \vee P\}$. The modification method transforms $\mathcal{S}_1$ into the following set of clauses with ordering constraints:

$$
\begin{array}{ll}
C_1: & \neg P \\
C_2: & (P \ \vee\ Q \ \vee\ a \simeq u_1 \ \vee\ b \not\simeq u_1) \cdot (a \succ u_1 \wedge b \succeq u_1) \\
C_3: & (P \ \vee\ Q \ \vee\ b \simeq u_2 \ \vee\ a \not\simeq u_2) \cdot (b \succ u_2 \wedge a \succeq u_2) \\
C_4: & (b \not\simeq u_3 \ \vee\ a \not\simeq u_3) \cdot (b \succeq u_3 \wedge a \succeq u_3) \\
C_5: & \neg Q \vee P \\
\text{Ref}: & x \simeq x \qquad \text{(Reflexivity Axiom)}
\end{array}
$$

A clause with ordering constraints takes the form of $D \cdot \delta$ where D is an ordinary clause and δ is a conjunction of ordering constraints $s \succ t$, $s \succeq t$ or $s = t$. The ordering constraint δ of $D \cdot \delta$ is expected to be satisfiable together with D. Notice that the above two clauses C_2 and C_3 are produced from the single clause $P \vee Q \vee a \approx b$ by symmetry elimination rule (more precisely, together with transitivity elimination rule). Figure 1 depicts two consecutive tableaux in a connection tableaux derivation, where we assume the ordering $b \succ a$ over constants. The left tableau fails to be closed because the goal $a \not\simeq u_2$ violates the ordering constraint $a \succeq u_3$, where the variable u_3 is substituted with b. The failure of derivation invokes backtracking, and eventually replaces the tableau clause C_2 below the top clause C_1 with the clause C_3. The right tableau in Fig. 1 succeeded in being closed, and simultaneously satisfies the ordering constraints. Notice that there are identical subtableaux below the goal Q in both left and right tableaux. Unfortunately, none of well-known pruning methods, such as folding-up/C-reduction or local failure caching, can prevent the redundant duplicated computation, because the clause C_2 containing Q in the left tableau is replaced with C_3 in the right tableau. Such a redundant computation essentially originates in the duplication of a given clause by symmetry elimination.

In this paper, we study a simple but effective remedy, that is, we abandon such symmetry elimination of clauses, and instead introduce new equality inference rules into connection tableaux. These new inference rules can achieve efficient equality computation, without losing the symmetry property of equality, which never cause redundant backtracking nor redundant contrapositive computation. Finally, we evaluate the proposed method through experiments with TPTP benchmark problems. Paskevich [9] also gave a new connection tableau calculus which uses lazy paramodulation instead of symmetry elimination. Paskevich's paramodulation-based connection calculus is very sophisticated, but seems to be a bit complicated and difficult in efficient implementation. Although the calculus proposed in this paper is superficially a little bit complicated, the underlying principle is very simple, and is easy to implement. At last, we emphasize that this research is now in progress, In this paper we show just some tentative results.

2 Preliminaries

We give some preliminaries according to Paskevich [9]. A language considered in this paper is first-order logic with equality in clausal form. A *clause* is a multi-set of literals, usually written as a disjunction $L_1 \vee \ldots \vee L_n$. The empty clause is denoted as $\perp$.

The equality predicate is denoted by the symbol $\approx$. We abbreviate the negation $\neg(s \approx t)$ as $s \not\approx t$. We consider equalities as unordered pairs of terms; that is, $a \approx b$ and $b \approx a$ stand for the same formula. As is well known, the equality is characterized by the congruence axioms $\mathcal{E}$ consisting of four axioms, i.e., *reflexivity*, *symmetry*, *transitivity* and *monotonicity*. The symbol $\simeq$ will denote "pseudo-equality", i.e., a binary predicate without any specific semantics. We utilize $\simeq$ in order to replace the symbol $\approx$ when we transform a clause set into a logic without equality. The order of arguments becomes significant here: $a \simeq b$ and $b \simeq a$ denote different formulas. The expression $s \not\simeq t$ stands for $\neg(s \simeq t)$.

We denote non-variable terms by $\mathbf{nv}$, $\mathbf{nv_1}$ and $\mathbf{nv_2}$, and also arbitrary terms by l, r, s, t, u and v. Variables are denoted by x, y and z. Substitutions are denoted by σ and θ. The result applying a substitution σ to an expression E is denoted by $E\sigma$. We write $E[s]$ to indicate that a term s occurs in E, and also write $E[t]$ to denote the expression obtained from E by replacing one occurrence of s with t.

We use an ordering constraint as defined in Bachmair et al. [1]. A *constraint* is a conjunction of *atomic constraints* $s = t$, $s \succ t$ or $s \succeq t$. The letters γ and δ denote constraints. A compound constraint $(a = b \wedge b \succ c)$ can be written in an abbreviated form $a = b \succ c$. A substitution σ *solves* an atomic constraint $s = t$ if the terms $s\sigma$ and $t\sigma$ are syntactically identical. It is a solution of an atomic constraint $s \succ t$ $(s \succeq t)$ if $s\sigma > t\sigma$ ($s\sigma \geq t\sigma$, respectively) with respect to a given term ordering $>$. Throughout this paper, we assume that a term ordering $\succ$ is a *reduction ordering* which is total over ground terms.[2] We say that σ is a solution of a constraint γ if it solves all atomic constraints in γ; γ is called *satisfiable* whenever it has a solution.

[2] A *reduction ordering* $>$ is an ordering over terms such that: (1) $>$ is well-founded; (2) for any terms s, t, u and any substitution θ, if $s > t$ then $u[s\theta] > u[t\theta]$ holds.

Expansion (Exp):
$$\frac{\mathcal{S}, (L_1 \vee \cdots \vee L_k) \ || \ \Gamma}{L_1 \quad \cdots \quad L_k}$$

Strong Connection (SC):
$$\frac{\mathcal{S} \ || \ \Gamma, \neg P(r), P(s)}{\bot \cdot (r = s)} \qquad \frac{\mathcal{S} \ || \ \Gamma, P(r), \neg P(s)}{\bot \cdot (r = s)}$$

Weak Connection (WC):
$$\frac{\mathcal{S} \ || \ \Gamma, \neg P(r), \Delta, P(s)}{\bot \cdot (r = s)} \qquad \frac{\mathcal{S} \ || \ \Gamma, P(r), \Delta, \neg P(s)}{\bot \cdot (r = s)}$$

Figure 2: Connection calculus **CT** for a set $\mathcal{S}$ of clauses

Let $\mathcal{S}$ be a set of clauses. A *constrained clause tableau* for $\mathcal{S}$ is a finite tree $\mathcal{T}$ (See Fig. 1 as an example). Each node except for a root node is a pair $L \cdot \gamma$ where L is a literal and γ is a constraint. Any branch that contains the literal $\bot$, which represents the false, is *closed*. A tableau is *closed*, whenever every branch in it is closed and the overall of constraints in it is satisfiable.

Each inference step grows some branch in the tableau by adding new leaves under the leaf of the branch in question. Initially, an inference starts from the single root node. Symbolically, we describe an inference rule as follows:

$$\frac{\mathcal{S} \ || \ \Gamma}{L_1 \cdot \gamma_1 \quad \cdots \quad L_n \cdot \gamma_n}$$

where $\mathcal{S}$ is an initial given set of clauses, Γ is the branch being augmented (with constraints not mentioned), and $(L1 \cdot \gamma_1), \ldots, (L_n \cdot \gamma_n)$ are the added nodes. Whenever we choose some clause C in $\mathcal{S}$ to participate in the inference, we implicitly rename all variable in C to some fresh variables. The standard connection tableau calculus [6, 9], denoted by **CT**, for a set $\mathcal{S}$ of clauses has inference rules depicted in Fig. 2.

Any clause tableau built by the rules of **CT** can be considered as a tree of inference steps. Every tableau of **CT** always starts with an expansion step; also that first expansion step can be followed only by another expansion, since connection step requires at least two literals in a branch. In a tableau, an *expansion clause* is the added clause in an expansion step.

Let $\mathcal{T}$ be a tableau of **CT** for a set $\mathcal{S}$ of clauses. We say that $\mathcal{T}$ is *strongly connected* whenever every strong connection step in a tableau follows an expansion step, and every expansion step except for the first (or top) one is followed by exactly one strong connection step. Moreover, $\mathcal{T}$ is said to be a *refutation* for $\mathcal{S}$ if $\mathcal{T}$ is strongly connected and closed.

Theorem 1 (Letz et.al [6]) *The* **CT** *calculus is sound and complete in first-order logic without equality.*

Ordered paramodulation is a well-known efficient equality inference rule. It is well known that top-down (or linear) deduction systems, including connection tableaux, are difficult frameworks for efficient equality computation because of hard restriction of redexes, i.e., subterms allowed to rewrite. For example, Snyder and Lynch [12] showed that: paramodulation into a variable is necessary for completeness; ordering constraints is incompatible with top-down theorem proving even if paramodulation into a variable is allowed. As a remedy, the modification proposed by Brand [2] has been investigated by many researchers.

2.1 Modification Method and Connection Tableaux

In this subsection, we firstly, show the modification method given by Bachmair, Ganzinger and Voronkov [1] which uses ordering constraints. Secondly, we show Paskevich's connection tableau calculus [9], denoted as $\mathbf{CT}^{\simeq}$,[3] for refuting a set of clauses generated by the modification method.

2.1.1 Elimination of Congruence Axioms

Given a set $\mathcal{S}$ of equational clauses, we apply three kinds of elimination rules and replace the equality predicate $\approx$ by the predicate $\simeq$ to obtain a modified clause set $\mathcal{S}$', such that $\mathcal{S}$' is satisfiable iff $\mathcal{S}$ is equationally satisfiable. If R is a set of such elimination rules, we say a constrained clause is in R-normal form if no rule in R is applicable to it. We denote by $R(\mathcal{S})$ the set of all R-normal forms of a clause in $\mathcal{S}$.

We first show S-modification rules which replaces the equality symbol $\approx$ with the pseudo-equality $\simeq$, and generates several clauses which can simulate computational effects of symmetry axiom.

- *Positive S-modification*:
$$s \approx t \vee C \quad \Rightarrow \quad s \simeq t \vee C \text{ and } t \simeq s \vee C$$

- *Negative S-modification*:
$$\begin{array}{rcl} \mathbf{nv} \not\approx t \vee C & \Rightarrow & \mathbf{nv} \not\simeq t \vee C \\ x \not\approx \mathbf{nv} \vee C & \Rightarrow & \mathbf{nv} \not\simeq x \vee C \\ x \not\approx y \vee C & \Rightarrow & C\theta \end{array}$$

  where θ is a substitution $\{x/y\}$.

Remark: Positive S-modification rule is quite problematic, because one equation is *duplicated* to two equations each of which has converse directions. We shall give a remedy for it in the next section.

Secondly we give M-modification rules which flatten clauses by abstracting subterms via introduction of new variables as follows:

$$\begin{array}{rcl} P(\ldots, \mathbf{nv}, \ldots) \vee C & \Rightarrow & \mathbf{nv} \not\simeq z \vee P(\ldots, z, \ldots) \vee C \\ \neg P(\ldots, \mathbf{nv}, \ldots) \vee C & \Rightarrow & \mathbf{nv} \not\simeq z \vee \neg P(\ldots, z, \ldots) \vee C \\ f(\ldots, \mathbf{nv}, \ldots) \simeq t \vee C & \Rightarrow & \mathbf{nv} \not\simeq z \vee f(\ldots, z, \ldots) \simeq t \vee C \\ f(\ldots, \mathbf{nv}, \ldots) \not\simeq t \vee C & \Rightarrow & \mathbf{nv} \not\simeq z \vee f(\ldots, z, \ldots) \not\simeq t \vee C \\ s \simeq f(\ldots, \mathbf{nv}, \ldots) \vee C & \Rightarrow & \mathbf{nv} \not\simeq z \vee s \simeq f(\ldots, z, \ldots) \vee C \\ s \not\simeq f(\ldots, \mathbf{nv}, \ldots) \vee C & \Rightarrow & \mathbf{nv} \not\simeq z \vee s \not\simeq f(\ldots, z, \ldots) \vee C \end{array}$$

where z is a new variable, called an *abstraction variable*.

The third one is T-modification rule for generating clauses which can simulate effects of transitivity axiom.

[3]Notice that $\mathbf{CT}^{\simeq}$ was introduced to prove the completeness of the *lazy paramodulation calculus* in [9].

Expansion (Exp):

$$\frac{\text{SMT}(\mathcal{S}), (L_1 \vee \cdots \vee L_k) \ || \ \Gamma}{L_1 \quad \cdots \quad L_k}$$

$\mathbf{Equality\ Resoution(ER)}$:

$$\frac{\text{SMT}(\mathcal{S}) \ || \ \Gamma, \ l \not\simeq r}{\bot \cdot (l = r)}$$

Strong Connection (SC):

$$\frac{\text{SMT}(\mathcal{S}) \ || \ \Gamma, \neg P(r), P(s)}{\bot \cdot (r = s)} \qquad \frac{\text{SMT}(\mathcal{S}) \ || \ \Gamma, P(r), \neg P(s)}{\bot \cdot (r = s)}$$

$$\frac{\text{SMT}(\mathcal{S}) \ || \ \Gamma, \mathbf{nv} \not\simeq r, s \simeq t}{\bot \cdot (\mathbf{nv} = s \succ t = r)} \qquad \frac{\text{SMT}(\mathcal{S}) \ || \ \Gamma, s \simeq t, \mathbf{nv} \not\simeq r}{\bot \cdot (\mathbf{nv} = s \succ t = r)}$$

Weak Connection (WC):

$$\frac{\text{SMT}(\mathcal{S}) \ || \ \Gamma, \neg P(r), \Delta, P(s)}{\bot \cdot (r = s)} \qquad \frac{\text{SMT}(\mathcal{S}) \ || \ \Gamma, P(r), \Delta, \neg P(s)}{\bot \cdot (r = s)}$$

$$\frac{\text{SMT}(\mathcal{S}) \ || \ \Gamma, \mathbf{nv} \not\simeq r, \Delta, s \simeq t}{\bot \cdot (\mathbf{nv} = s \succ t = r)} \qquad \frac{\text{SMT}(\mathcal{S}) \ || \ \Gamma, s \simeq t, \Delta, \mathbf{nv} \not\simeq r}{\bot \cdot (\mathbf{nv} = s \succ t = r)}$$

Figure 3: Connection tableaux **CT**$^{\simeq}$ for SMT$(\mathcal{S})$

- *Positive T-modification:*

$$s \simeq \mathbf{nv} \ \vee \ C \quad \Rightarrow \quad s \simeq z \vee \mathbf{nv} \not\simeq z \ \vee \ C$$

- *Negative T-modification:*

$$s \not\simeq \mathbf{nv} \ \vee \ C \quad \Rightarrow \quad s \not\simeq z \vee \mathbf{nv} \not\simeq z \ \vee \ C$$

  where z is a new variable, called a *link variable*.

Notice that if the term t in $s \simeq t$ is a variable, then T-modification does *nothing*.

Let SMT$(\mathcal{S})$ denote a set T(M(S($\mathcal{S}$))), i.e., the set of normal clauses obtained from $\mathcal{S}$ by consecutively applying S, M and T-modification. Notice that the size of SMT($\mathcal{S}$) is *exponential* to the one of S.

Theorem 2 (Bachmair et al. [1]) *$\mathcal{S} \cup \mathcal{E}$ is unsatisfiable iff* SMT($\mathcal{S}$) $\cup \{x \simeq x\}$ *is unsatisfiable, where $\simeq$ is a new symbol for simulating the equality.*

Bachmair et al. [1] studied weak ordering constraints for modification. An atomic ordering constraint $s \succ t$ ($s \succeq t$) is assigned to each positive (or respectively, negative) literal $s \simeq t$ (or respectively, $s \not\simeq t$) in SMT($\mathcal{S}$), except for the negative equality $x \not\simeq y$ for any variables x and y.

CEE($\mathcal{S}$) denote the set of clauses of SMT($\mathcal{S}$) with ordering constraints.

Theorem 3 (Bachmair et al. [1]) *$\mathcal{S} \cup \mathcal{E}$ is unsatisfiable iff* CEE($\mathcal{S}$) $\cup \{x \simeq x\}$ *is unsatisfiable, where $\simeq$ is a new symbol for simulating the equality.*

2.1.2 Connection Tableaux for Modification with Ordering Constraints

Paskevich [9] adapted the calculus **CT** for computing CEE($\mathcal{S}$), and gave the connection tableau calculus **CT**$^{\simeq}$ for modification with ordering constraints, which is described in Fig. 3. Notice that $\mathbf{nv}$ denotes a non-variable term in **CT**$^{\simeq}$.

Theorem 4 (Paskevich [9]) *The calculus* $\mathbf{CT}^{\simeq}$ *is sound and complete. That is,* $\mathcal{S} \cup \mathcal{E}$ *is unsatisfiable iff there is a closed and strongly connected tableau in* $\mathbf{CT}^{\simeq}$ *for SMT*($\mathcal{S}$)*.*

3 Connection Tableaux for Modification without S-Modification

The size of SMT($\mathcal{S}$) is unfortunately *exponential* to the one of S, which is truly problematic and causes a huge amount of redundant computation. The positive S-modification, hence, should be abandoned. We alternatively introduce new inference rules for simulating the effects of symmetry axiom and construct a new connection tableau calculus **CTwS** (Connection Tableaux for modification Without S-modification).

Definition 1 Let *P-modification* be a transformation rule of clauses, which just replaces the equality symbol $\approx$ with the pseudo symbol $\simeq$ in positive equalities. We define nSMT$(\mathcal{S})$ to be a a set of normal clauses obtained from $\mathcal{S}$ by just successively applying P-modification, negative S-modification, M-modification and negative T-modification.

Notice that the size of nSMT$(\mathcal{S})$ is *linear* to the one of $\mathcal{S}$ because positive S-modification is never applied.

Once the positive S-modification is abandoned, no symmetry formula $t \simeq s$ of an initial equality $s \simeq t$ is generated in the modification process, which means that the succeeding positive T-modification is not accomplished either. Therefore, we need a mechanism compensating such a deficit of clause transformation. In this paper, we introduce new inference rules which can simulate not only positive S-modification but also *positive T-modification* for keeping transitivity properties of a positive equality.

We propose the following new rules, called *symmetry and transitivity splitting rules*, abbreviated as *ST-splitting*, which can simultaneously simulate the computational effects of symmetry and transitivity axioms.

Naive ST-Splitting Rule:

$$\frac{\text{nSMT}(\mathcal{S}) \ \| \ \Gamma, s \simeq \mathbf{nv}}{s \simeq z \quad \mathbf{nv} \not\simeq z} \qquad \frac{\text{nSMT}(\mathcal{S}) \ \| \ \Gamma, s \simeq x}{s \simeq x}$$

$$\frac{\text{nSMT}(\mathcal{S}) \ \| \ \Gamma, \mathbf{nv} \simeq t}{t \simeq z \quad \mathbf{nv} \not\simeq z} \qquad \frac{\text{nSMT}(\mathcal{S}) \ \| \ \Gamma, x \simeq t}{t \simeq x}$$

where $\mathbf{nv}$ is a non-variable term and x is a variable.

3.1 Controlling ST-Splitting I: A Raw Equality

ST-Splitting should be applied to each positive equality *at most one time*, because more than two times applications of these rules are clearly redundant. Therefore we need a controlling mechanism.

In this paper, we firstly give a *raw positive equality*, denoted as $\boxed{s \simeq t}$, which is introduced into a tableau by the expansion rule. Some of raw positive equalities $\boxed{s \simeq t}$

are changed to *ordinary equality literals* by ST-splitting. Conversely ST-Splitting rule is restricted to apply only to a raw positive equality. Moreover, the strong connection rule for a negative equality is also restricted to apply only to raw positive equalities. Furthermore, we force every raw positive equality to be followed either by ST-Splitting or by new strong contraction rules shown below.

Given a literal L, we write $[L]$ to denote a framed literal $\boxed{s \simeq t}$, called a *raw positive literal* if L is a positive equality $s \simeq t$; otherwise $[L]$ denotes L itself. We modify the expansion rule into the one which produces a raw literal for a positive equality.

Expansion for nSMT$(\mathcal{S})$:

$$\frac{\text{nSMT}(\mathcal{S}),\ (L_1 \vee \cdots \vee L_k)\ \|\ \Gamma}{[L_1]\ \ \cdots\ [L_k]}$$

ST-Splitting Rule should be changed to treat only raw positive literals.

ST-Splitting Rule:

$$\frac{\text{nSMT}(\mathcal{S})\ \|\ \Gamma, \boxed{s \simeq \mathbf{nv}}}{s \simeq z \quad \mathbf{nv} \not\simeq z} \qquad \frac{\text{nSMT}(\mathcal{S})\ \|\ \Gamma, \boxed{s \simeq x}}{s \simeq x}$$

$$\frac{\text{nSMT}(\mathcal{S})\ \|\ \Gamma, \boxed{\mathbf{nv} \simeq t}}{t \simeq z \quad \mathbf{nv} \not\simeq z} \qquad \frac{\text{nSMT}(\mathcal{S})\ \|\ \Gamma, \boxed{x \simeq t}}{t \simeq x}$$

Example 1 Consider the set $\mathcal{S}_1$ of clauses in Section 1. The set nSMT$(\mathcal{S})$ of normal clauses is:

$$\begin{array}{ll} C_1: & \neg P. \\ C_6: & P \vee Q \vee a \simeq b \\ C_4': & (b \not\simeq u_3\ \vee\ a \not\simeq u_3) \\ C_5: & \neg Q \vee P \end{array}$$

Figure 4 shows two connection tableaux in **CTwS** for $\mathcal{S}_1$, each of which corresponds with the one in Fig. 1. Notice that no backtracking occurs for undoing the expansion introducing the clause C_6 in the derivation from the left tableau to the right one. Therefore none of duplicated computations invoked for the subgoal Q in **CT**$^{\simeq}$ occur in the calculus **CTwS**.

3.2 Controlling ST-Splitting II: Strong Connection

The original form of strong connection for negative equality is no longer appropriate, because it cannot deal with raw positive equalities nor corporate with ST-splitting rule. The new calculus **CTwS** has to simulate all valid inferences involving the strong connection in **CT**$^{\simeq}$ for SMT$(\mathcal{S})$ in order to preserve completeness. Let $C \in \mathcal{S}$ be a clause $s \simeq t \vee K_1 \vee \cdots \vee K_m$. There are four possible clauses obtained by S-modification and T-modification from C with respect to $s \simeq t$:

$$\begin{array}{llr} D_1: & s \simeq z \vee \mathbf{nv_2} \not\simeq z \vee K_1' \vee \cdots \vee K_m' & \text{if } t \text{ is a non-variable term } \mathbf{nv_2} \\ D_2: & s \simeq x \vee K_1' \vee \cdots \vee K_m' & \text{if } t \text{ is a variable } x \\ D_3: & t \simeq z \vee \mathbf{nv_2} \not\simeq z \vee K_1' \vee \cdots \vee K_m' & \text{if } s \text{ is a non-variable term } \mathbf{nv_2} \\ D_4: & t \simeq x \vee K_1' \vee \cdots \vee K_m' & \text{if } s \text{ is a variable } x \end{array}$$

where z is a fresh variable. All of these clauses have possibilities to be used as an expansion clause for the strong connection in **CT**$^{\simeq}$. Next we consider new strong connection rules for **CTwS** in order to simulate these inferences in **CT**$^{\simeq}$.

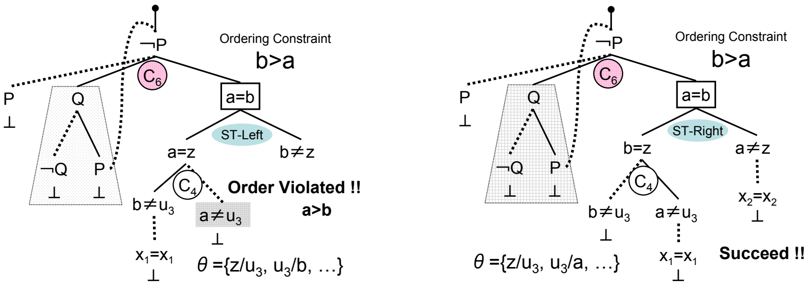

Figure 4: Two connection tableaux in **CTwS** for nSMT($\mathcal{S}_1$)

Firstly, we study a simulation of strong connection using the clause D_1 in SMT($\mathcal{S}$). Consider an expansion inference for D_1 in $\mathbf{CT}^{\simeq}$.

$$\frac{L}{s \simeq z \quad \mathbf{nv_2} \not\simeq z \quad K'_1 \quad \cdots \quad K'_m}\ (\text{Exp})$$

If L is a negative equality $\mathbf{nv_1} \not\simeq r$ such that $\mathbf{nv_1}$ is non-variable and is unifiable with s, then the following strong connection is available in $\mathbf{CT}^{\simeq}$:

(1) $$\frac{\mathbf{nv_1} \not\simeq r}{\dfrac{s \simeq z}{\bot \cdot (\mathbf{nv_1} = s \succ z = r)}\ (\text{SC}) \quad \mathbf{nv_2} \not\simeq z \quad K'_1 \ \cdots \ K'_m}\ (\text{Exp})$$

On the other hand, if L is a positive equality $u \simeq v$ such that u is unifiable with $\mathbf{nv_2}$, then we have the following strong connection in $\mathbf{CT}^{\simeq}$:

(2) $$\frac{u \simeq v}{s \simeq z \quad \dfrac{\mathbf{nv_2} \not\simeq z}{\bot \cdot (\mathbf{nv_2} = u \succ v = z)}\ (\text{SC}) \quad K'_1 \ \cdots \ K'_m}\ (\text{Exp})$$

The above first inference (1) in $\mathbf{CT}^{\simeq}$ can be simulated in nSMT($\mathcal{S}$) with the new expansion rule and ST-splitting for a raw equality $\boxed{s \simeq \mathbf{nv_2}}$ and the *weak connection* rule as follows:

$$\frac{\mathbf{nv_1} \not\simeq r}{\dfrac{\boxed{s \simeq \mathbf{nv_2}}}{\dfrac{s \simeq z}{\bot \cdot (\mathbf{nv_1} = s \succ z = r)}\ (\text{WC}) \quad \mathbf{nv_2} \not\simeq z}\ (\text{ST}) \quad K'_1 \ \cdots \ K'_m}\ (\text{new Exp})$$

However, it is definitely better to use a sort of strong connection rule instead of the weak connection, because a connection constraint for a tableau becomes much simpler and more effective to drastically reduce the search space. We, hence, introduce a new strong connection rule which can perform the above inference steps as an integrated one-step inference in **CTwS**. The following is a naive form for directly simulating the inference (1):

$$\frac{\text{nSMT}(\mathcal{S})\ \ ||\ \Gamma,\ \mathbf{nv_1} \not\simeq r,\ \boxed{s \simeq \mathbf{nv_2}}}{\bot \cdot (\mathbf{nv_1} = s \succ z = r) \quad \mathbf{nv_2} \not\simeq z}$$

where $\mathbf{nv_1}$ and $\mathbf{nv_2}$ are non-variable terms. We can eliminate the link variable z because z never occurs elsewhere in a tableau, and moreover we can add an ordering constraint. The final form of the above rule is:

$$\frac{\text{nSMT}(\mathcal{S})\ \ ||\ \Gamma,\ \mathbf{nv_1} \not\simeq r,\ \boxed{s \simeq \mathbf{nv_2}}}{\bot \cdot (\mathbf{nv_1} = s \succ r) \quad \mathbf{nv_2} \not\simeq r \cdot (\mathbf{nv_2} \succeq r)}$$

Remark: The above ordering constraint $\mathbf{nv_2} \succeq r$ is not explicitly used in the strong connection in **CT**$^{\simeq}$, as shown in the inference (1). Thus this additional constraint can reduce the alternative choices of expansion rules, compared with **CT**$^{\simeq}$. Recall the term $\mathbf{nv_2}$ initially occurs as an argument of the equality $s \simeq \mathbf{nv_2}$ of the original clause $s \simeq \mathbf{nv_2} \vee K_1 \vee \cdots \vee K_m$ in $\mathcal{S}$. Thus we can say, **CTwS** directly uses full information of the equality $s \simeq \mathbf{nv_2}$ for strong connection and thus expansion, while **CT**$^{\simeq}$ just uses this information indirectly through variable binding for a linked variable.[4] This difference is a rather important point because several state-of-arts top-down provers, such as SETHEO [6] and SOLAR [8], often reorder goals for improving the efficiency of inferences.

Similarly, the above inference (2) can also be simulated in nSMT$(\mathcal{S})$ with a raw positive equality $\boxed{s \simeq \mathbf{nv_2}}$ as follows:

$$\dfrac{u \simeq v}{\dfrac{\boxed{s \simeq \mathbf{nv_2}}}{s \simeq z \qquad \dfrac{\mathbf{nv_2} \not\simeq z}{\bot \cdot (\mathbf{nv_2} = u \succ v = z)}\ \text{(WC)}}\ \text{(ST)} \qquad K_1' \ \cdots \ K_m'}\ \text{(new Exp)}$$

This observation leads to the following rule, which can achieve the above inference steps as a single inference.

$$\frac{\text{nSMT}(\mathcal{S})\ \ ||\ \Gamma,\ u \simeq v,\ \boxed{s \simeq \mathbf{nv_2}}}{s \simeq z \quad \bot \cdot (\mathbf{nv_2} = u \succ v = z)}$$

We can also eliminate the link variable z and add an additional ordering for $s \simeq z$ without losing completeness. Finally, we obtain the following new rule:

$$\frac{\text{nSMT}(\mathcal{S})\ \ ||\ \Gamma,\ u \simeq v,\ \boxed{s \simeq \mathbf{nv_2}}}{s \simeq v \cdot (s \succ v) \quad \bot \cdot (\mathbf{nv_2} = u \succ v)}$$

Notice that this rule superficially requires a *positive* raw literal $\boxed{s \simeq \mathbf{nv_2}}$ as a partner of strong connection of a *positive* literal $u \simeq v$.

Next we study a simulation of strong connection using the clause D_2. Consider the following inference involving expansion and strong connection of D_2 in **CT**$^{\simeq}$.

(3)
$$\dfrac{\mathbf{nv_1} \not\simeq r}{\dfrac{s \simeq x}{\bot \cdot (\mathbf{nv_1} = s \succ x = r)}\ \text{(SC)} \qquad K_1' \ \cdots \ K_m'}\ \text{(Exp)},$$

[4] See the variable binding of z in the inference (1), for example.

where $\mathbf{nv_1}$ is a non-variable term. The above (3) can simply be simulated in nSMT($\mathcal{S}$) with the raw equality $\boxed{s \simeq x}$ as follows:

$$\cfrac{\cfrac{\cfrac{\mathbf{nv_1} \not\simeq r}{\boxed{s \simeq x}} \text{(new Exp)},}{s \simeq x} \text{(ST)} \quad K'_1 \ \cdots \ K'_m}{\bot \cdot (\mathbf{nv_1} = s \succ x = r)} \text{(WC)}$$

This observation derives the following strong connection rule in **CTwS**:

$$\frac{\text{nSMT}(\mathcal{S}) \ || \ \Gamma, \ \mathbf{nv_1} \not\simeq r, \ \boxed{s \simeq x}}{\bot \cdot (\mathbf{nv_1} = s \succ x = r)}$$

Moreover, we have to investigate inferences using strong connections with the clauses D_3 and D_4 of SMT($\mathcal{S}$), and can derive additional three rules for nSMT($\mathcal{S}$) by similar discussions. Eventually, we obtain the following set of strong connection rules for nSMT($\mathcal{S}$):

Strong Connection for Negative Equality in nSMT($\mathcal{S}$):

$$\frac{\text{nSMT}(\mathcal{S}) \ || \ \Gamma, \ \mathbf{nv_1} \not\simeq r, \ \boxed{s \simeq \mathbf{nv_2}}}{\bot \cdot (\mathbf{nv_1} = s \succ r) \quad \mathbf{nv_2} \not\simeq r \cdot (\mathbf{nv_2} \succeq r)} \qquad \frac{\text{nSMT}(\mathcal{S}) \ || \ \Gamma, \ \mathbf{nv_1} \not\simeq r, \ \boxed{s \simeq x}}{\bot \cdot (\mathbf{nv_1} = s \succ x = r)}$$

$$\frac{\text{nSMT}(\mathcal{S}) \ || \ \Gamma, \ \mathbf{nv_1} \not\simeq r, \ \boxed{\mathbf{nv_2} \simeq t}}{\bot \cdot (\mathbf{nv_1} = t \succ r) \quad \mathbf{nv_2} \not\simeq r \cdot (\mathbf{nv_2} \succeq r)} \qquad \frac{\text{nSMT}(\mathcal{S}) \ || \ \Gamma, \ \mathbf{nv_1} \not\simeq r, \ \boxed{x \simeq t}}{\bot \cdot (\mathbf{nv_1} = t \succ x = r)}$$

Strong Connection for Positive Equality in nSMT($\mathcal{S}$):

$$\frac{\text{nSMT}(\mathcal{S}) \ || \ \Gamma, \ u \simeq v, \ \boxed{s \simeq \mathbf{nv_2}}}{s \simeq v \cdot (s \succ v) \quad \bot \cdot (\mathbf{nv_2} = u \succ v)} \qquad \frac{\text{nSMT}(\mathcal{S}) \ || \ \Gamma, \ u \simeq v, \ \boxed{\mathbf{nv_2} \simeq t}}{t \simeq v \cdot (t \succ v) \quad \bot \cdot (\mathbf{nv_2} = u \succ v)}$$

where $\mathbf{nv_1}$ and $\mathbf{nv_2}$ denote non-variable terms, x is a variable.

We show a total view of the connection tableaux **CTwS** for nSMT($\mathcal{S}$) in Fig. 5. The following is the first main theorem of this paper:

Theorem 5 *The calculus* **CTwS** *is sound and complete. That is, $\mathcal{S} \cup \mathcal{E}$ is unsatisfiable iff there is a closed and strongly connected tableau in* **CTwS** *for nSMT($\mathcal{S}$).*

3.3 Yet another Connection Tableaux for Modification

In this section, we consider yet another connection tableaux, called **CTwST**, where the strong connection for positive equality is further improved with a more strict ordering constraint. As was shown in the previous subsection, one of the strong connection for a positive equality for nSMT($\mathcal{S}$) is:

SC-PosE-1: $$\frac{\text{nSMT}(\mathcal{S}) \ || \ \Gamma, \ s \simeq t, \ \mathbf{nv_1} \not\simeq r}{\bot \cdot (\mathbf{nv_1} = s \succ t = r)}$$

Expansion (Exp):

$$\frac{\text{nSMT}(\mathcal{S}),\ (L_1 \vee \cdots \vee L_k)\ ||\ \Gamma}{[L_1]\ \cdots\ [L_k]}$$

Equality Resolution (ER)

$$\frac{\text{nSMT}(\mathcal{S})\ ||\ \Gamma,\ l \not\simeq r}{\bot \cdot (l = r)}$$

ST Splitting (ST):

$$\frac{\text{nSMT}(\mathcal{S})\ ||\ \Gamma, \boxed{s \simeq \mathbf{nv_1}}}{s \simeq z \quad \mathbf{nv_1} \not\simeq z}$$

$$\frac{\text{nSMT}(\mathcal{S})\ ||\ \Gamma, \boxed{s \simeq x}}{s \simeq x}$$

$$\frac{\text{nSMT}(\mathcal{S})\ ||\ \Gamma, \boxed{\mathbf{nv_1} \simeq t}}{t \simeq z \quad \mathbf{nv_1} \not\simeq z}$$

$$\frac{\text{nSMT}(\mathcal{S})\ ||\ \Gamma, \boxed{x \simeq t}}{t \simeq x}$$

Strong Connection for Non-Equality

$$\frac{\text{nSMT}(\mathcal{S})\ ||\ \Gamma, \neg P(r), P(s)}{\bot \cdot (r = s)}$$

(SC–NonE):

$$\frac{\text{nSMT}(\mathcal{S})\ ||\ \Gamma, P(r), \neg P(s)}{\bot \cdot (r = s)}$$

Strong Connection for Neg. Equality

$$\frac{\text{nSMT}(\mathcal{S})\ ||\ \Gamma,\ \mathbf{nv_1} \not\simeq r,\ \boxed{s \simeq \mathbf{nv_2}}}{\bot \cdot (\mathbf{nv_1} = s \succ r) \quad \mathbf{nv_2} \not\simeq r \cdot (\mathbf{nv_2} \succeq r)}$$

(SC–NegE):

$$\frac{\text{nSMT}(\mathcal{S})\ ||\ \Gamma,\ \mathbf{nv_1} \not\simeq r,\ \boxed{s \simeq x}}{\bot \cdot (\mathbf{nv_1} = s \succ x = r)}$$

$$\frac{\text{nSMT}(\mathcal{S})\ ||\ \Gamma,\ \mathbf{nv_1} \not\simeq r,\ \boxed{\mathbf{nv_2} \simeq t}}{\bot \cdot (\mathbf{nv_1} = t \succ r) \quad \mathbf{nv_2} \not\simeq r \cdot (\mathbf{nv_2} \succeq r)}$$

$$\frac{\text{nSMT}(\mathcal{S})\ ||\ \Gamma,\ \mathbf{nv_1} \not\simeq r,\ \boxed{x \simeq t}}{\bot \cdot (\mathbf{nv_1} = t \succ x = r)}$$

Strong Connection for Pos. Equality

(SC–PosE):

$$\frac{\text{nSMT}(\mathcal{S})\ ||\ \Gamma,\ s \simeq t,\ \mathbf{nv_1} \not\simeq r}{\bot \cdot (\mathbf{nv_1} = s \succ t = r)}$$

$$\frac{\text{nSMT}(\mathcal{S})\ ||\ \Gamma,\ u \simeq v,\ \boxed{s \simeq \mathbf{nv_1}}}{s \simeq v \cdot (s \succ v) \quad \bot \cdot (\mathbf{nv_1} = u \succ v)}$$

$$\frac{\text{nSMT}(\mathcal{S})\ ||\ \Gamma,\ u \simeq v,\ \boxed{\mathbf{nv_1} \simeq t}}{t \simeq v \cdot (t \succ v) \quad \bot \cdot (\mathbf{nv_1} = u \succ v)}$$

Weak Connection (WC):

$$\frac{\text{nSMT}(\mathcal{S})\ ||\ \Gamma, \neg P(r), \Delta, P(s)}{\bot \cdot (r = s)}$$

$$\frac{\text{nSMT}(\mathcal{S})\ ||\ \Gamma, P(r), \Delta, \neg P(s)}{\bot \cdot (r = s)}$$

$$\frac{\text{nSMT}(\mathcal{S})\ ||\ \Gamma, \mathbf{nv_1} \not\simeq r, \Delta, s \simeq t}{\bot \cdot (\mathbf{nv_1} = s \succ t = r)}$$

$$\frac{\text{nSMT}(\mathcal{S})\ ||\ \Gamma, s \simeq t, \Delta, \mathbf{nv_1} \not\simeq r}{\bot \cdot (\mathbf{nv_1} = s \succ t = r)}$$

Figure 5: Connection tableaux **CTwS** for nSMT($\mathcal{S}$)

Recall that T-modification splits a given negative equality $l \not\simeq r$ into the disjunction $l \not\simeq z \vee r \not\simeq z$ if r is not a variable. Thus, the literal $l \not\simeq z$ (or $r \not\simeq z$) loses the information about the initial partner term r (or respectively, l). Thus the above strong connection rule cannot utilize full information provided by negative equalities in a clause set $\mathcal{S}$ which is initially given. As a remedy, we omit T-modification for negative equality literals as well, and instead give a set of new connection rules for preserving transitivity

Definition 2 We define nSM$(\mathcal{S})$ to be a a set of normal clauses obtained from $\mathcal{S}$ by just applying negative S-modification and M-modification.

The calculus **CTwST** differs from **CTwS** in the following points; firstly **CTwST** accepts nSM$(\mathcal{S})$ as an input set of clauses, not nSMT$(\mathcal{S})$; secondly we add a new expansion rule and T-splitting rules for treating a *raw negative* equality; thirdly we replace the strong connection **SC-PosE-1** with new three rules with *raw negative* equalities. We modify the expansion rule to the one which produces raw literals both for positive and negative equalities. Given a literal L, we write $[[L]]$ to denote the framed literal $\boxed{L}$, called a *raw literal* if L is a positive equality $s \simeq t$ or a negative equality $s \not\simeq t$; otherwise $[[L]]$ denotes L itself.

Expansion Rule for nSM$(\mathcal{S})$:

$$\frac{\text{nSM}(\mathcal{S}),\ (L_1 \vee \cdots \vee L_k)\ \ ||\ \ \Gamma}{[[L_1]]\ \ \cdots\ \ [[L_k]]}$$

We add the following T-splitting rules in order to treating raw negative equalities, which naturally correspond with T-modification.

T-Splitting for Negative Equality for nSM$(\mathcal{S})$:[-0.5ex]

$$\frac{\text{nSM}(\mathcal{S})\ \ ||\ \ \Gamma,\ \boxed{s \not\simeq \mathbf{nv_1}}}{s \not\simeq z \quad \mathbf{nv_1} \not\simeq z} \qquad \frac{\text{nSM}(\mathcal{S})\ \ ||\ \ \Gamma,\ \boxed{s \not\simeq y}}{s \not\simeq y}$$

At last, we replace the rule **SC-PosE-1** by the following three rules:

Strong Connection for Positive Equality for nSM$(\mathcal{S})$:

$$\frac{\text{nSM}(\mathcal{S})\ \ ||\ \ \Gamma, l \simeq r,\ \boxed{\mathbf{nv_1} \not\simeq \mathbf{nv_2}}}{\bot \cdot (\mathbf{nv_1} = l \succ r) \quad \mathbf{nv_2} \not\simeq r \cdot (\mathbf{nv_2} \succeq r)} \qquad \frac{\text{nSM}(\mathcal{S})\ \ ||\ \ \Gamma, l \simeq r,\ \boxed{s \not\simeq \mathbf{nv_2}}}{s \not\simeq r \cdot (s \succeq r) \quad \bot \cdot (\mathbf{nv_2} = l \succ r)}$$

$$\frac{\text{nSM}(\mathcal{S})\ \ ||\ \ \Gamma, l \simeq r,\ \boxed{\mathbf{nv_1} \not\simeq y}}{\bot \cdot (\mathbf{nv_1} = l \succ r = y)}$$

4 Extended SOLAR and Experimental Evaluation

In this section, we show some tentative experimental results with **SOLAR** [8], which is an efficient consequence finding program based on Skipping Ordered Linear Resolution [4] by using Connection Tableaux technology [6, 5]. At first we show the basic

Table 1: Basic performance comparison of theorem provers

	SOLAR	Otter	E 1.0	E 1.0 (A)*
# of solved unit EQ.	170	474	589	630
# of solved non-unit EQ.	676	727	907	2013
# of solved non-EQ.	1163	1044	1131	1640

* Note: (A) means that E system uses the option"-xAuto -tAuto".

performance of SOLAR compared with state-of-the-art theorem provers Otter 3.0 [7] and E 1.0 [11]. Table 1 shows the numbers of problems of TPTP library v.3.5.0 which each theorem prover can solve within the time limit of 60 CPU seconds. The first row is for unit equation problems; the second is non-unit equational ones; the third is for non-equational ones. SOLAR is competitive for the class of non-equational problems, but is not for equational problems.[5]

Table 2 shows the performances of several kinds of equality computation methods in connection tableaux.[6] The first "Axioms" indicates a naive use of the congruence axioms, and the second "M-mod" represents a method for using just M-modification together with reflexivity, symmetry and transitivity axioms. Each row denoted by "infer." is the sum total of the numbers of inferences needed for equational problems which can commonly be solved by all of $\mathbf{CT}^{\simeq}$, **CTwS** and **CTwST**. The upper half of Table 2 shows the results obtained by using ordinary M-modification, while the lower half is for the ones obtained by using a semi-optimized M-modification, given in [1], such that the flattening never applies to any occurrences of an ordering-minimal constant symbol. Regretfully, the best performance is provided by the naive use method of the congruence axioms. Modification methods commonly inherit a disadvantage caused by M-modification which increases the length of each clause by flattening. $\mathbf{CT}^{\simeq}$ and **CTwS**, however, significantly decrease the number of inference steps from M-modification method. With the semi-optimized M-modification, **CTwS** is superior to $\mathbf{CT}^{\simeq}$. Certainly, **CTwS** decreases the amount of inference steps compared with $\mathbf{CT}^{\simeq}$, which means that **CTwS** succeeds to prevent redundant computations originating in S-modification. By comparison between the upper part and the lower one in Table 2, we can understand the importance of optimization of M-modification for avoiding redundant computations, which are invoked by long disjunctions of *thin* negative equalities produced by flattening operations.

5 Conclusion and Future Work

We investigated Paskevich's connection tableaux for equality computation, and pointed out that a naive use of S-modification is problematic. We proposed, as some remedies, improved connection tableau calculi for efficient equality computation. We also showed tentative experimental results of evaluating the proposed methods using SOLAR. This research is now in progress. For example, we are still studying a further

[5]TPTP library v.3.5.0 has 2,175 non-equational problems and 4,171 equational problems, where there are 863 unit equational problems.

[6]Throughout experiments, we used non-recursive Knuth-Bendix ordering given by Riazanov and Voronkov [10]. as a reduction ordering.

Table 2: Comparison of equality computation methods in connection tableaux

	Axioms	M-mod	**CT**$^{\simeq}$	**CTwS**	**CTwST**
# of unit EQ.	170	161	180	183	179
# of non-unit EQ.	636	490	507	499	489
# of infer. of unit EQ.	4,883K	12,900K	8,903K	1,403K	2,367K
# of infer. of non-unit EQ.	38,621K	251,244K	86,837K	78,339K	119,094K
# of unit EQ.	—	—	183	185	183
# of non-unit EQ.	—	—	518	540	512
# of infer. of unit EQ.	—	—	5,545K	5,212K	8,397K
# of infer. of non-unit EQ.	—	—	66,529K	58,588K	86,253K

improvement of M-modification. Moreover, we found that the dynamic term-binding to variables in derivations frequently gives ill effects on the behaviors of **CTwS** and **CTwST**. In order to improve this situation, we will re-formalize our methods in the context of the basic method and the closure mechanism in the near future. Furthermore, one of anonymous referees suggested that the effects of ST-splitting can be achieved by the following clause transformation:

$$s \approx t \vee C \quad \Rightarrow \quad P_{new}(\vec{x}) \vee C,\ \neg P_{new}(\vec{x}) \vee s \simeq t \text{ and } \neg P_{new}(\vec{x}) \vee t \simeq s$$

where P_{new} is a new predicate symbol and $\vec{x}$ denotes the list of variables occurring in s and t. Notice that the literal $P_{new}(\vec{x})$ corresponds to a raw equality in our framework. This rule can be used for simulating ST-splitting instead of positive S-modification rule. This method seems to have a great possibility in several aspects. We are now conducting some theoretical studies and experimental evaluations.

References

[1] L. Bachmair, H. Ganzinger and A. Voronkov: Elimination of equality via transformation with ordering constraints. *Proc. CADE-15, LNCS*, Vol.1421, pp.175-190 (1998)

[2] D. Brand: Proving theorems with the modification method. *SIAM Journal of Computing*, Vol.4, pp.412-430 (1975)

[3] P. Baumgartner and C. Tinelli: The model evolution calculus with equality. *Proc. CADE-20, LNAI.* Vol.3632, pp.392–408 (2005)

[4] K. Inoue: Linear resolution for consequence finding. *Artificial Intelligence* Vol.56 pp.301–353 (1992)

[5] K. Iwanuma, K. Inoue and K. Satoh: Completeness of pruning methods for consequence finding procedure SOL. *Proc. Int. Workshop on First-order Theorem Proving (FTP2000)*, pp.89-100 (2000).

[6] R. Letz, C. Goller and K. Mayr: Controlled integration of the cut rule into connection tableau calculi. *J. Automated Reasoning*, Vol.13, pp.297-338 (1994).

[7] W. McCune: Skolem functions and equality in automated deduction. *Proc. AAAI-90*, pp.246–251 (1990)

[8] H. Nabeshima, K. Iwanuma and K. Inoue: SOLAR: a consequence finding system for advanced reasoning. *Proc. Tableaux'03. LNAI* Vol.2796, pp.257-263 (2003)

[9] A. Paskevich: Connection tableaux with lazy paramodulation. *J. Automated Reasoning* (2008).

[10] A. Riazanov: Implementing an efficient theorem prover. PhD thesis, University of Manchester, 2002.

[11] St. Schulz: System description E 0.81. *Proc IJCAR-2004, LNAI*, Vol.3097, pp.223-228 (2004)

[12] W. Snyder and C. Lynch: Goal directed strategies for paramodulation, *Proc. RTA-91, LNCS* Vol.448, pp.150–111 (1991).