Dealing with Non-Orientable Equations in Rewriting Induction

Takahito Aoto

Research Institute of Electrical Communication, Tohoku University, Japan
aoto@nue.riec.tohoku.ac.jp

Abstract. Rewriting induction (Reddy, 1990) is an automated proof method for inductive theorems of term rewriting systems. Reasoning by the rewriting induction is based on the noetherian induction on some reduction order. Thus, when the given conjecture is not orientable by the reduction order in use, any proof attempts for that conjecture fails; also conjectures such as a commutativity equation are out of the scope of the rewriting induction because they can not be oriented by any reduction order. In this paper, we give an enhanced rewriting induction which can deal with non-orientable conjectures. We also present an extension which intends an incremental use of our enhanced rewriting induction.

1 Introduction

Properties of programs are often proved by induction on the data structures such as natural numbers or lists. Such properties are called inductive properties of programs. Inductive properties are indispensable in formal treatments of programs. Thus automated reasoning of inductive properties is appreciated in techniques such as the program verification and the program transformation.

Term rewriting systems (TRSs) is a computational model based on equational logic. Equational inductive properties of TRSs are called inductive theorems, and automated reasoning methods for inductive theorems have been investigated many years [3, 4, 7–12, 15]. In this paper, we extend rewriting induction proposed by Reddy [12], which is one of such inductive theorem proving methods.

The rewriting induction falls in a category of implicit induction methods; in implicit induction, induction scheme is not specified explicitly—such methods are different from explicit induction methods that stem from [5]. Historically, the implicit induction method has been investigated mainly in the context of inductionless induction [7–9, 11, 15]. Usually inductionless induction methods require (kinds of) the Church-Rosser property; while in the rewriting induction, the termination property is needed instead—Koike and Toyama [10] revealed that the rewriting induction[1] and the inductionless induction have different underlying principles. In this context, the underlying principle of (the inductive theorem

[1] Renaming the original "term rewriting induction" [6, 12] to "rewriting induction" is proposed by them.

proving part of) the inductive theorem prover SPIKE [3, 4] can be also classified as a rewriting induction method. The rewriting induction is also useful as a program synthesis [6, 13].

Inductive proofs by the rewriting induction are based on the noetherian induction on some reduction order. Thus, when the given conjecture is not orientable by the reduction order in use, any proof attempt for that conjecture fails; also conjectures such as a commutativity equation are out of the scope of the rewriting induction because they can not be oriented by any reduction order.

To overcome this defect, several approaches have been proposed. One is to use rewriting modulo equations [12]. Another is to use ordered rewriting technique [2, 6] which rewrites a term by possibly non-oriented equations when it simplifies (w.r.t. some ordering). The former appears only in a short remark in [12], and, as far as the author knows, the idea is not explored since then. The latter approach has been embodied in the inductive theorem prover SPIKE. In this paper, we present an enhanced rewriting induction designed following the first approach.

In our enhanced rewriting induction, a reduction order whose equational classes are "coarser" is more suitable to prove non-oriented conjectures. On the other hand, such a reduction order may fail to handle some equations orientable by other reduction orders. This observation leads us to introduce incremental rewriting induction in which already-proved lemmas can be applied more easily.

The rest of the paper is organized as follows. After fixing basic notations (Section 2), we review the principle and the procedure of the rewriting induction (Section 3). In Section 4, we give an enhanced rewriting induction that can deal with non-orientable conjectures and show its correctness. In Section 5, we introduce incremental rewriting induction which intends an incremental use of the enhanced rewriting induction. In Section 6, we conclude our result and compare our approach and the ordered rewriting approach.

2 Preliminaries

Let us fix some notations in *abstract reduction systems (ARSs)*. Let $\rightarrow$ be a binary relation on a set A. The reflexive transitive closure (transitive closure, symmetric closure, equivalence closure) of $\rightarrow$ is denoted by $\xrightarrow{*}$ ($\xrightarrow{+}$, $\leftrightarrow$, $\xleftrightarrow{*}$, respectively). The relation $\rightarrow$ is well-founded (denoted by $\mathrm{SN}(\rightarrow)$) when there exists no infinite chain $a_0 \rightarrow a_1 \rightarrow \cdots$. An element $a \in A$ is said to be *normal* when there is no $b \in A$ such that $a \rightarrow b$. The set of normal elements is denoted by $\mathrm{NF}(\rightarrow)$. The union $\rightarrow_i \cup \rightarrow_j$ of two binary relations $\rightarrow_i$ and $\rightarrow_j$ is abbreviated as $\rightarrow_{i \cup j}$. The composition is denoted by $\circ$. We denote by $\rightarrow_i / \rightarrow_j$ the relation defined by $\xleftrightarrow{*}_j \circ \rightarrow_i \circ \xleftrightarrow{*}_j$. The relation $\rightarrow_i / \rightarrow_j$ is abbreviated as $\rightarrow_{i/j}$. We assume $/$ associates stronger than $\cup$; $/, \cup$ associate stronger than closure operations so that, for example, $\xleftrightarrow{*}_{1 \cup 2}$ stands for the equivalence closure of $\rightarrow_1 \cup \rightarrow_2$.

We next introduce notations on term rewriting used in this paper. (See [1, 14] for details.) The sets of (arity-fixed) function symbols and variables are denoted by $\mathcal{F}$ and V, respectively. $\mathrm{T}(\mathcal{F}, V)$ is the set of terms over $\mathcal{F}, V$. We use $\equiv$ to denote the syntactical equality on terms. The set of variables contained in

t is denoted by $V(t)$. $\mathrm{root}(t)$ is the *root symbol* of a term t. The *domain* of a substitution σ is denoted by $\mathrm{dom}(\sigma)$. A term $\sigma(t)$ is called an *instance* of the term t; $\sigma(t)$ is also written as $t\sigma$. We denote by $\mathrm{mgu}(s,t)$ the *most general unifier* of terms s,t. A pair $l \to r$ of terms satisfying conditions (1) $\mathrm{root}(l) \in \mathcal{F}$; (2) $V(r) \subseteq V(l)$ is said to be a *rewrite rule*. A *term rewriting system (TRS)* is a set of rewrite rules. When the underlying set of function symbols is not clear, we refer to a pair $\langle \mathcal{F}, \mathcal{R} \rangle$ as a TRS—however, we assume that the set of function symbols are those appearing in rewrite rules in this paper. The *rewrite relation* of a TRS $\mathcal{R}$ is denoted by $s \to_{\mathcal{R}} t$. An *equation* $l \doteq r$ is just a pair $\langle l, r \rangle$ of terms in $\mathrm{T}(\mathcal{F}, V)$. When we write $l \doteq r$, we do not distinguish $\langle l, r \rangle$ and $\langle r, l \rangle$.

Function symbols that are roots of some lhs of rewrite rules are called *defined function symbols*; we write $\mathcal{D}_{\mathcal{R}}$ the set $\{\mathrm{root}(l) \mid l \to r \in \mathcal{R}\}$ of defined function symbols (of a TRS $\mathcal{R}$). When $\mathcal{R}$ is obvious from its context, we omit the subscript $_{\mathcal{R}}$. The set of defined symbols appearing in a term t is denoted by $\mathcal{D}(t)$. The set $\mathcal{C} = \mathcal{F} \setminus \mathcal{D}$ of function symbols is the set of *constructor symbols*. Terms in $\mathrm{T}(\mathcal{C}, V)$ are said to be *constructor terms*; substitution σ such that $\sigma(x) \in \mathrm{T}(\mathcal{C}, V)$ for any $x \in \mathrm{dom}(\sigma)$ is called a *constructor substitution*. A term of the form $f(c_1, \ldots, c_n)$ for some $f \in \mathcal{D}$ and $c_1, \ldots, c_n \in \mathrm{T}(\mathcal{C}, V)$ is said to be *basic*. We write $u \trianglelefteq s$ to express that u is a subterm of s. The set $\{u \trianglelefteq s \mid \exists f \in \mathcal{D}.\ \exists c_1, \ldots, c_n \in \mathrm{T}(\mathcal{C}, V).\ u \equiv f(c_1, \ldots, c_n)\}$ of basic subterms of s is written as $\mathcal{B}(s)$.

A term t is said to be *ground* when $V(t) = \emptyset$. $\mathrm{T}(\mathcal{F})$ is the set of ground terms. When $t\sigma \in \mathrm{T}(\mathcal{F})$, $t\sigma$ is called a *ground instance* of t. Ground instances of rewrite rules, equations, etc. are defined similarly. A *ground substitution* is a substitution σ_g such that $\sigma_g(x) \in \mathrm{T}(\mathcal{F})$ for any $x \in \mathrm{dom}(\sigma_g)$. A TRS $\mathcal{R}$ is said to be *quasi-reducible* if no ground basic term is normal. In this paper, we assume w.l.o.g. that $t\sigma_g$ is ground (i.e. $V(t) \subseteq \mathrm{dom}(\sigma_g)$) when we speak of an instance $t\sigma_g$ of t by a ground substitution σ_g. An *inductive theorem* of a TRS $\mathcal{R}$ is an equation that is valid on $\mathrm{T}(\mathcal{F})$, that is, $s \doteq t$ is an inductive theorem when $s\sigma_g \overset{*}{\leftrightarrow}_{\mathcal{R}} t\sigma_g$ holds for any ground instance $s\sigma_g \doteq t\sigma_g$.

A relation R on $\mathrm{T}(\mathcal{F}, V)$ is said to be *closed under substitution* when $s\,R\,t \Rightarrow s\sigma\,R\,t\sigma$ for any substitution σ; *closed under context* when $s\,R\,t \Rightarrow C[s]\,R\,C[t]$ for any context C. A *reduction order* is a well-founded partial order that is closed under substitution and context. A quasi-order $\succsim$ is a *reduction quasi-order* when it is closed under substitution and context and its strict part $\succ\ =\ \succsim \setminus \precsim$ is a reduction order. We write the relation $\succsim \cap \precsim$ as $\approx$.

3 Rewriting Induction

Rewriting induction (RI, for short) is an automated inductive theorem proving method proposed by Reddy [12]. The inference system of rewriting induction deals with a set E of equations and a set H of rewrite rules. Intuitively, E is a set of equations to be proved and H is a set of induction hypotheses and theorems already proved. In Figure 1, we list the (downward) inference rules of the rewriting induction. Here $\uplus$ denotes the disjoint union. We note that the direction of each equation is not distinguished. $\mathcal{R}$ and $>$ are a TRS and a

Simplify

$$\frac{\langle E \uplus \{s \doteq t\},\ H\rangle}{\langle E \cup \{s' \doteq t\},\ H\rangle} \quad s \to_{\mathcal{R}\cup H} s'$$

Delete

$$\frac{\langle E \uplus \{s \doteq s\},\ H\rangle}{\langle E,\ H\rangle}$$

Expand

$$\frac{\langle E \uplus \{s \doteq t\},\ H\rangle}{\langle E \cup \mathrm{Expd}_u(s,t),\ H \cup \{s \to t\}\rangle} \ u \in \mathcal{B}(s), s > t$$

Fig. 1. Inference rules of the rewriting induction

reduction order given as inputs. The set $\mathrm{Expd}_u(s,t)$ of equations is defined like this:

$$\mathrm{Expd}_u(s,t) = \{C[r]\sigma \doteq t\sigma \mid s \equiv C[u],\ \sigma = \mathrm{mgu}(u,l),\ l \to r \in \mathcal{R},\ l\text{:basic}\}$$

The following property of Expd will be used later.

Lemma 1 (property of Expd). *Let $\mathcal{R}$ be a quasi-reducible TRS and $u \in \mathcal{B}(s)$. Then*
(1) $s\sigma_g \to_{\mathcal{R}} \circ \leftrightarrow_{\mathrm{Expd}_u(s,t)} t\sigma_g$ for any ground constructor substitution σ_g;
(2) $v \leftrightarrow_{\mathrm{Expd}_u(s,t)} w \Rightarrow v \overset{*}{\leftrightarrow}_{\mathcal{R}\cup\{s\doteq t\}} w$.

Proof. (1) Since u is basic and σ_g is a ground constructor substitution, $u\sigma_g$ is a basic ground term. Thus, by the quasi-reducibility of $\mathcal{R}$, there exists $l \to r \in \mathcal{R}$ such that $u\sigma_g$ is an instance of l. W.l.o.g. we may assume $V(l) \cap V(s) = \emptyset$ and thus by extending σ_g one can let $u\sigma_g \equiv l\sigma_g$. Then σ_g is a constructor unifier of u and l and thus we have $\sigma_g = \theta_g \circ \sigma$ for some constructor substitution θ_g, where $\sigma = \mathrm{mgu}(u,l)$. Then by letting $s \equiv C[u]$, we have $s\sigma_g \equiv C[u]\sigma_g \equiv C\sigma_g[u\sigma\theta_g] \equiv C\sigma_g[l\sigma\theta_g] \to_{\mathcal{R}} C\sigma_g[r\sigma\theta_g] \equiv C[r]\sigma\theta_g \leftrightarrow_{\mathrm{Expd}_u(s,t)} t\sigma\theta_g \equiv t\sigma_g$. (2) Let $v \leftrightarrow_{\mathrm{Expd}_u(s,t)} w$. Then $v \equiv \hat{C}[C[r]\sigma\hat{\sigma}]$, $w \equiv \hat{C}[t\sigma\hat{\sigma}]$ (or $w \equiv \hat{C}[C[r]\sigma\hat{\sigma}]$, $v \equiv \hat{C}[t\sigma\hat{\sigma}]$) for some context $\hat{C}$ and substitution $\hat{\sigma}$, where $\sigma = \mathrm{mgu}(u,l)$, $s \equiv C[u]$, $l \to r \in \mathcal{R}$. Then we have $v \equiv \hat{C}[C[r]\sigma\hat{\sigma}] \leftarrow_{\mathcal{R}} \hat{C}[C[l]\sigma\hat{\sigma}] \equiv \hat{C}[C\sigma[l\sigma]\hat{\sigma}] \equiv \hat{C}[C\sigma[u\sigma]\hat{\sigma}] \equiv \hat{C}[C[u]\sigma\hat{\sigma}] \equiv \hat{C}[s\sigma\hat{\sigma}] \leftrightarrow_{\{s\doteq t\}} \equiv \hat{C}[t\sigma\hat{\sigma}] \equiv w$. □

Definition 1 (rewriting induction). *We write $\langle E, H\rangle \rightsquigarrow_{\mathrm{RI}} \langle E', H'\rangle$ when $\langle E', H'\rangle$ is obtained from $\langle E, H\rangle$ by applying one of the inference rules of Figure 1. The reflexive transitive closure of $\rightsquigarrow_{\mathrm{RI}}$ is denoted by $\overset{*}{\rightsquigarrow}_{\mathrm{RI}}$. We sometimes put superscripts ^{s,d,e} to indicate which inference rule is used.*

The rewriting induction procedure starts by putting conjectures into E and letting $H = \emptyset$. Then the procedure rewrites $\langle E, H\rangle$ by applying one of the inference rules. If it eventually becomes of the form $\langle \emptyset, H'\rangle$ then the procedure returns "success"—this means that the conjectures are inductive theorems of $\mathcal{R}$. On the other hand, when none of the rules are applicable, it reports "failure", or it also may run forever ("divergence"), which means the rewriting induction fails to prove that the conjectures are inductive theorems.

Koike and Toyama [10] revealed that the underlying principle of rewriting induction can be formulated in terms of ARSs as below. The proof is by the noetherian induction on $>$. Later, we will give a proof of a more general theorem.

Proposition 1 (principle of rewriting induction [10]). *Let $\to_1$, $\to_2$ be binary relations on a set A. Let $>$ be a well-founded partial order on A. Suppose*

$$\begin{array}{ll} (i) & \to_{1\cup 2} \subseteq > \\ (ii) & \to_2 \subseteq \to_1 \circ \xrightarrow{*}_{1\cup 2} \circ \xleftarrow{*}_{1\cup 2}. \end{array}$$

Then $\xleftrightarrow{*}_1 = \xleftrightarrow{*}_{1\cup 2}$.

The following proposition states the correctness of the rewriting induction. The proof basically proceeds by applying Proposition 1 to binary relations $\to_{\mathcal{R}}$ and $\to_H$ on the set $\mathrm{T}(\mathcal{F})$ of ground terms. Later, we will give a proof of a more general theorem.

Proposition 2 (correctness of rewriting induction [12]). *Let $\mathcal{R}$ be a quasi-reducible TRS, E a set of equations, $>$ a reduction order satisfying $\mathcal{R} \subseteq >$. If there exists a set H such that $\langle E, \emptyset\rangle \overset{*}{\rightsquigarrow}_{\mathrm{RI}} \langle \emptyset, H\rangle$ then equations in E are inductive theorems of $\mathcal{R}$.*

Example 1 (rewriting induction). Let $\mathcal{R}$ and E be a TRS and a set of equations given as below.

$$\mathcal{R} = \left\{ \begin{array}{ll} \mathsf{plus}(\mathsf{0}, y) & \to y \\ \mathsf{plus}(\mathsf{s}(x), y) & \to \mathsf{s}(\mathsf{plus}(x, y)) \end{array} \right\}$$

$$E = \left\{ \mathsf{plus}(\mathsf{plus}(x, y), z) \doteq \mathsf{plus}(x, \mathsf{plus}(y, z)) \right\}$$

Let $>$ be a lexicographic path order [1] based on precedence $\mathsf{plus} > \mathsf{s} > \mathsf{0}$. Below we show how the rewriting induction for proving E proceeds based on the TRS $\mathcal{R}$ and the reduction order $>$.

$$\left\langle \left\{ \mathsf{plus}(\mathsf{plus}(x, y), z) \doteq \mathsf{plus}(x, \mathsf{plus}(y, z)) \right\}, \left\{\right\} \right\rangle$$

$$\rightsquigarrow^{e}_{\mathrm{RI}} \quad \left\langle \begin{array}{l} \left\{ \begin{array}{l} \mathsf{plus}(y_0, z) \doteq \mathsf{plus}(\mathsf{0}, \mathsf{plus}(y_0, z)) \\ \mathsf{plus}(\mathsf{s}(\mathsf{plus}(x_1, y_1)), z) \doteq \mathsf{plus}(\mathsf{s}(x_1), \mathsf{plus}(y_1, z)) \end{array} \right\} \\ \left\{ \mathsf{plus}(\mathsf{plus}(x, y), z) \to \mathsf{plus}(x, \mathsf{plus}(y, z)) \right\} \end{array} \right\rangle$$

$$\rightsquigarrow^{s}_{\mathrm{RI}} \rightsquigarrow^{s}_{\mathrm{RI}} \rightsquigarrow^{s}_{\mathrm{RI}} \left\langle \begin{array}{l} \left\{ \begin{array}{l} \mathsf{plus}(y_0, z) \doteq \mathsf{plus}(y_0, z) \\ \mathsf{s}(\mathsf{plus}(\mathsf{plus}(x_1, y_1), z)) \doteq \mathsf{s}(\mathsf{plus}(x_1, \mathsf{plus}(y_1, z))) \end{array} \right\} \\ \left\{ \mathsf{plus}(\mathsf{plus}(x, y), z) \to \mathsf{plus}(x, \mathsf{plus}(y, z)) \right\} \end{array} \right\rangle$$

$$\rightsquigarrow^{s}_{\mathrm{RI}} \rightsquigarrow^{d}_{\mathrm{RI}} \rightsquigarrow^{d}_{\mathrm{RI}} \left\langle \left\{\right\}, \left\{ \mathsf{plus}(\mathsf{plus}(x, y), z) \to \mathsf{plus}(x, \mathsf{plus}(y, z)) \right\} \right\rangle$$

The procedure ends in the form $\langle \emptyset, H\rangle$. Thus from Proposition 2 it follows that the equation in E is an inductive theorem of $\mathcal{R}$.

4 Proving Non-Orientable Conjectures

A key of the rewriting induction is the *Expand* rule. But *Expand* rule is applicable only to the equation that can be oriented by the input reduction order. Thus, when the given conjecture is not orientable by the given reduction order, the proof of that conjecture always fails.

Example 2 (failure of rewriting induction). Let $\mathcal{R}$ be a TRS for the addition of natural numbers.

$$\mathcal{R} = \left\{ \begin{array}{ll} \mathsf{plus}(0, y) & \to y \\ \mathsf{plus}(\mathsf{s}(x), y) & \to \mathsf{s}(\mathsf{plus}(x, y)) \end{array} \right\}$$

The following equation e expresses the commutativity of addition.

$$e = \quad \mathsf{plus}(x, y) \doteq \mathsf{plus}(y, x)$$

The equation e is an inductive theorem of $\mathcal{R}$. However, because neither $\mathsf{plus}(x, y) > \mathsf{plus}(y, x)$ nor $\mathsf{plus}(y, x) > \mathsf{plus}(x, y)$ holds, the rewriting induction procedure starting with $\langle \{e\}, \emptyset \rangle$ stops immediately having no rules to apply.

To deal with non-orientable equations, Reddy proposed to use $\to_{\mathcal{R}}/\to_H$ instead of $\to_{\mathcal{R}} \cup \to_H$ (Remark 14 in [12]); however, he does not seem to elaborate on this. In fact, a naive extension seems to lead unsound reasoning—this is illustrated by the following proposition obtained by modifying Propsition 1 suitably for $\to_{\mathcal{R}}/\to_H$.

Conjecture 1 (incorrect conjecture). Let $\to_1$, $\to_2$ be binary relations on a set A. Let $\succsim$ be a well-founded quasi-order on A. Suppose

$$\begin{array}{rl} (i) & \to_1 \subseteq \succ \\ (ii) & \to_2 \subseteq \approx \\ (iii) & \to_2 \subseteq \to_1 \circ \xrightarrow{*}_{1/2} \circ \xleftarrow{*}_{1/2}. \end{array}$$

Then $\xleftrightarrow{*}_1 = \xleftrightarrow{*}_{1\cup 2}$.

Example 3 (a counterexample to the Conjecture 1). Consider a set $A = \{a, b, c\}$ and relations $\to_1 = \{\langle a, b \rangle\}$ and $\to_2 = \{\langle a, c \rangle\}$ on A. Let $\succsim$ be a quasi-order such that $c \approx a \succ b$. Then conditions (i),(ii) clearly hold. Since $a \to_1 b \leftarrow_1 a \leftrightarrow_2 c$, condition (iii) holds also. But we have $c \xleftrightarrow{*}_{1\cup 2} b$ and $c \not\xleftrightarrow{*}_1 b$.

In Figure 2, we list the inference rules in which $\mathcal{R} \cup H$ is replaced by $\mathcal{R}/H$. This inference system is not sound as the following example shows.

Example 4 (incorrect inference). Let $\mathcal{R}$ be a TRS for the append of two lists:

$$\mathcal{R} = \left\{ \begin{array}{ll} \mathsf{app}(\mathsf{nil}, ys) & \to ys \\ \mathsf{app}(\mathsf{cons}(x, xs), ys) & \to \mathsf{cons}(x, \mathsf{app}(xs, ys)) \end{array} \right\}.$$

The append operation is not commutative, hence

$$\mathsf{app}(xs, ys) \doteq \mathsf{app}(ys, xs) \tag{1}$$

Simplify

$$\frac{\langle E \uplus \{s \doteq t\},\ H\rangle}{\langle E \cup \{s' \doteq t\},\ H\rangle} \quad s \rightarrow_{\mathcal{R}/H} s'$$

Delete

$$\frac{\langle E \uplus \{s \doteq t\},\ H\rangle}{\langle E,\ H\rangle} \quad s \overset{*}{\leftrightarrow}_H t$$

Expand

$$\frac{\langle E \uplus \{s \doteq t\},\ H\rangle}{\langle E \cup \mathrm{Expd}_u(s,t),\ H \cup \{s \doteq t\}\rangle} \ u \in \mathcal{B}(s), s \approx t$$

Fig. 2. Inference rules with rewriting modulo equations(not sound)

is not an inductive theorem of $\mathcal{R}$. However, by taking $\succsim$ as a recursive path order [1] based on the precedence $\mathsf{app} \succ \mathsf{cons} \succ \mathsf{nil}$, the inference of modified rewriting induction successfully proves the conjecture (1).

In *Expand* rule in Figure 2, for $v \doteq w$ in $\mathrm{Expd}_u(s,t)$, only v is "smaller" than $s\sigma$ while w is "just as big" as $s\sigma$. Hence, application of the inductive hypothesis to w is unsound. This observation suggests a new kind of *Expand* rule for non-orientable equations (*Expand2*, below), which expands both lhs and rhs of the equation.

Simplify

$$\frac{\langle E \uplus \{s \doteq t\},\ H,\ G\rangle}{\langle E \cup \{s' \doteq t\},\ H,\ G\rangle} \quad s \rightarrow_{(\mathcal{R}\cup H)/G} s'$$

Delete

$$\frac{\langle E \uplus \{s \doteq t\},\ H,\ G\rangle}{\langle E,\ H,\ G\rangle} \quad s \overset{*}{\leftrightarrow}_G t$$

Expand

$$\frac{\langle E \uplus \{s \doteq t\},\ H,\ G\rangle}{\langle E \cup \mathrm{Expd}_u(s,t),\ H \cup \{s \rightarrow t\},\ G\rangle} \ u \in \mathcal{B}(s), s \succ t$$

Expand2

$$\frac{\langle E \uplus \{s \doteq t\}, H, G\rangle}{\langle E \cup \mathrm{Expd2}_{u,v}(s,t), H, G \cup \{s \doteq t\}\rangle} \ u \in \mathcal{B}(s), v \in \mathcal{B}(t), s \approx t$$

Fig. 3. Inference rules of eRI

Here, $\mathrm{Expd2}_{u,v}(s,t)$ is defined like this:

$$\mathrm{Expd2}_{u,v}(s,t) = \bigcup \left\{\mathrm{Expd}_{v\sigma}(t\sigma, s') \mid \langle s', t\sigma\rangle \in \mathrm{Expd}_u(s,t)\right\}$$

Definition 2 (enhanced rewriting induction). *We write $\langle E, H, G\rangle \rightsquigarrow_{\mathrm{eRI}} \langle E', H', G'\rangle$ when $\langle E', H', G'\rangle$ is obtained from $\langle E, H, G\rangle$ by applying one of the inference rules of Figure 3. The reflexive transitive closure of $\rightsquigarrow_{\mathrm{eRI}}$ is denoted by $\overset{*}{\rightsquigarrow}_{\mathrm{eRI}}$. We sometimes put superscripts ^{s,d,e,e2} to indicate which inference rule is used.*

Example 5 (application of Expand2 rule). Let

$$\mathcal{R} = \left\{ \begin{array}{ll} \mathsf{plus}(\mathsf{0}, y) & \to y \\ \mathsf{plus}(\mathsf{s}(x), y) & \to \mathsf{s}(\mathsf{plus}(x, y)) \end{array} \right\},$$

$s \doteq t \;=\; \mathsf{plus}(x, y) \doteq \mathsf{plus}(y, x)$. Then we have

$$\mathrm{Expd2}_{s,t}(s, t) = \left\{ \begin{array}{l} \mathsf{0} \doteq \mathsf{0} \\ \mathsf{s}(x_1) \doteq \mathsf{s}(\mathsf{plus}(x_1, \mathsf{0})) \\ \mathsf{s}(\mathsf{plus}(x_3, \mathsf{s}(y_3))) \doteq \mathsf{s}(\mathsf{plus}(y_3, \mathsf{s}(x_3))) \end{array} \right\}.$$

Note that an equation $\mathsf{s}(\mathsf{plus}(x_2, \mathsf{0})) \doteq \mathsf{s}(x_2)$ which is also included in $\mathrm{Expd2}_{s,t}(s, t)$ is omitted, since this equation is same as the second one (as an equation).

Lemma 2 (property of Expd2). *Let $\mathcal{R}$ be a quasi-reducible TRS and let $u \in \mathcal{B}(s)$ and $v \in \mathcal{B}(t)$. Then*
(1) $s\sigma_g \to_{\mathcal{R}} \circ \leftrightarrow_{\mathrm{Expd2}_{u,v}(s,t)} \circ \leftarrow_{\mathcal{R}} t\sigma_g$ for any ground constructor substitution σ_g;
(2) $q \leftrightarrow_{\mathrm{Expd2}_{u,v}(s,t)} w \Rightarrow q \overset{*}{\leftrightarrow}_{\mathcal{R} \cup \{s \doteq t\}} w$.

Proof. (1) Let $l \to r \in \mathcal{R}$, l: basic, $\sigma = \mathrm{mgu}(u, l)$, $s \equiv C[u]$. Then, by Lemma 1 (1), $s\sigma_g \to_{\mathcal{R}} C[r]\sigma\theta_g \leftrightarrow_{\{C[r]\sigma \doteq t\sigma\}} t\sigma\theta_g \equiv t\sigma_g$ for some $\langle C[r]\sigma, t\sigma \rangle \in \mathrm{Expd}_u(s, t)$ and constructor substitution θ_g. Since σ is a constructor substitution and $v \in \mathcal{B}(t)$, we know $v\sigma \in \mathcal{B}(t\sigma)$. Thus applying Lemma 1 (1) once again, we have $t\sigma\theta_g \to_{\mathcal{R}} \circ \leftrightarrow_{\mathrm{Expd}_{v\sigma}(t\sigma, C[r]\sigma)} C[r]\sigma\theta_g$. Thus $s\sigma_g \to_{\mathcal{R}} C[r]\sigma\theta_g \leftrightarrow_{\mathrm{Expd2}_{u,v}(s,t)} \circ \leftarrow_{\mathcal{R}} t\sigma\theta_g \equiv t\sigma_g$. (2) Suppose $q \leftrightarrow_{\mathrm{Expd2}_{u,v}(s,t)} w$. Then, by definition, there exists $\langle s', t\sigma \rangle \in \mathrm{Expd}_u(s, t)$ such that $q \leftrightarrow_{\mathrm{Expd}_{v\sigma}(t\sigma, s')} w$, where $\sigma = \mathrm{mgu}(l, u)$, $l \to r \in \mathcal{R}$, l: basic. Then σ is a constructor substitution and thus $v\sigma \in \mathcal{B}(t\sigma)$. Then by Lemma 1 (2) $q \overset{*}{\leftrightarrow}_{\mathcal{R} \cup \{s' \doteq t\sigma\}} w$. Therefore we have $q \overset{*}{\leftrightarrow}_{\mathcal{R} \cup \mathrm{Expd}_u(s,t)} w$. By applying Lemma 1 (2) once again, we have $q \overset{*}{\leftrightarrow}_{\mathcal{R} \cup \{s \doteq t\}} w$. $\square$

The soundness of the enhanced rewriting induction is based on the following alternative principle.

Lemma 3 (principle of enhanced rewriting induction). *Let $\to_i$ $(1 \le i \le 3)$ be binary relations on a set A, and $\succsim$ be a well-founded quasi-order on A. Suppose*

$$\begin{array}{rl} (i) & \to_{1 \cup 2} \subseteq \succ \\ (ii) & \to_3 \subseteq \approx \\ (iii) & \to_2 \subseteq \to_1 \circ \overset{*}{\to}_{(1\cup 2)/3} \circ \overset{*}{\leftrightarrow}_3 \circ \overset{*}{\leftarrow}_{(1\cup 2)/3} \\ (iv) & \to_3 \subseteq \to_1 \circ \overset{*}{\to}_{(1\cup 2)/3} \circ \overset{*}{\leftrightarrow}_3 \circ \overset{*}{\leftarrow}_{(1\cup 2)/3} \circ \leftarrow_1 \\ (v) & \forall x, y \in \mathrm{NF}(\to_{(1\cup 2)/3}).\ (x \overset{*}{\leftrightarrow}_3 y \Rightarrow x = y). \end{array}$$

Then $\overset{*}{\leftrightarrow}_1 = \overset{*}{\leftrightarrow}_{1\cup 2\cup 3}$.

Proof. It suffices to show $\supseteq$. For this, we first show by noetherian induction on $\succ$ that

$$\text{for any } x \in A\ [\forall y \in A.\ (x \overset{*}{\to}_{(1\cup 2)/3} y \Rightarrow x \overset{*}{\leftrightarrow}_1 y)] \tag{2}$$

holds.
(Base Step) Then $x = y$ and thus $x \overset{*}{\leftrightarrow}_1 y$ trivially holds.
(Induction Step) The case when $x \in \mathrm{NF}(\rightarrow_{(1\cup2)/3})$ follows immediately; so, suppose $x \overset{*}{\leftrightarrow}_3 u \rightarrow_{1\cup2} v \overset{*}{\leftrightarrow}_3 z \overset{*}{\rightarrow}_{(1\cup2)/3} y$. Since $x \succ z$, it follows

$$z \overset{*}{\leftrightarrow}_1 y \tag{3}$$

by induction hypothesis.

We now claim that

$$a \rightarrow_3 b \Rightarrow a \overset{*}{\leftrightarrow}_1 b \text{ for any } a, b \precsim x \tag{4}$$

We note that it immediately follows from this that

$$x \overset{*}{\leftrightarrow}_1 u \text{ and } v \overset{*}{\leftrightarrow}_1 z \tag{5}$$

Suppose $a \rightarrow_3 b$. Then by condition (iv) we have

$$a \rightarrow_1 c \overset{*}{\rightarrow}_{(1\cup2)/3} c' \overset{*}{\leftrightarrow}_3 d' \overset{*}{\leftarrow}_{(1\cup2)/3} d \leftarrow_1 b$$

for some c, c', d, d'. By induction hypothesis we have $c \overset{*}{\leftrightarrow}_1 c'$ and $d' \overset{*}{\leftrightarrow}_1 d$. If $c', d' \in \mathrm{NF}(\rightarrow_{(1\cup2)/3})$ then the claim follows from the condition (v). So, suppose $c' \notin \mathrm{NF}(\rightarrow_{(1\cup2)/3})$. Then by conditions (i),(ii), $c' \overset{+}{\rightarrow}_{(1\cup2)/3} n$ for some $n \in \mathrm{NF}(\rightarrow_{(1\cup2)/3})$. Then we have $d' \overset{*}{\leftrightarrow}_3 c' \overset{+}{\rightarrow}_{(1\cup2)/3} n$ and hence $d \overset{*}{\rightarrow}_{(1\cup2)/3} n$. Thus we have

$$a \rightarrow_1 c \overset{*}{\rightarrow}_{(1\cup2)/3} n \overset{*}{\leftarrow}_{(1\cup2)/3} d \leftarrow_1 b$$

Then by the induction hypothesis $a \rightarrow_1 c \overset{*}{\leftrightarrow}_1 n \overset{*}{\leftrightarrow}_1 d \leftarrow_1 b$ follows. Since the case $d' \notin \mathrm{NF}(\rightarrow_{(1\cup2)/3})$ is shown similarly, the claim (4) has been shown.

It remains to show

$$u \overset{*}{\leftrightarrow}_1 v \tag{6}$$

The case when $u \rightarrow_1 v$ is trivial. So, suppose $u \rightarrow_2 v$. By condition (iii), we have

$$u \rightarrow_1 w \overset{*}{\rightarrow}_{(1\cup2)/3} w' \overset{*}{\leftrightarrow}_3 v' \overset{*}{\leftarrow}_{(1\cup2)/3} v$$

Since $x \succ w, v$, we have $w \overset{*}{\leftrightarrow}_1 w'$ and $v' \overset{*}{\leftrightarrow}_1 v$ by induction hypothesis. Then, as above, one can suppose w.l.o.g. $w', v' \in \mathrm{NF}(\rightarrow_{(1\cup2)/3})$. Thus the claim follows from the condition (v). Thus by (3), (5), (6), the proof of the claim (2) has been completed.

Next we show $\rightarrow_3 \subseteq \overset{*}{\leftrightarrow}_1$. This can be proved same as the proof of the claim (4), except that this time we use already proved claim (2) instead of the induction hypothesis.

Finally, the statement of the lemma follows from the fact $\overset{*}{\rightarrow}_{1\cup2\cup3} \subseteq \overset{*}{\rightarrow}_{(1\cup2)/3} \cup \overset{*}{\leftrightarrow}_3$. □

Below we prove the correctness of the enhanced rewriting induction. In remaining lemmata in this section, we assume that the TRS $\mathcal{R}$ is quasi-reducible and that $\succsim$ is a reduction quasi-order satisfying $\mathcal{R} \subseteq \succ$.

Lemma 4 (invariance). *Let* $\langle E_n, H_n, G_n\rangle \rightsquigarrow_{\mathrm{eRI}} \langle E_{n+1}, H_{n+1}, G_{n+1}\rangle$. *Then* $\stackrel{*}{\leftrightarrow}_{\mathcal{R}\cup E_n\cup H_n\cup G_n} = \stackrel{*}{\leftrightarrow}_{\mathcal{R}\cup E_{n+1}\cup H_{n+1}\cup G_{n+1}}$ *on* $\mathrm{T}(\mathcal{F})$.

Proof. Use Lemma 1 (2) and Lemma 2 (2). □

Lemma 5 (property of E_n). *Let* $\langle E_n, H_n, G_n\rangle \stackrel{*}{\rightsquigarrow}_{\mathrm{eRI}} \langle \emptyset, H^\sharp, G^\sharp\rangle$. *Then* $\leftrightarrow_{E_n} \subseteq \stackrel{*}{\rightarrow}_{(\mathcal{R}\cup H^\sharp)/G^\sharp} \circ \stackrel{*}{\leftrightarrow}_{G^\sharp} \circ \stackrel{*}{\leftarrow}_{(\mathcal{R}\cup H^\sharp)/G^\sharp}$ *on* $\mathrm{T}(\mathcal{F})$.

Proof. By induction on the length of $\langle E_n, H_n, G_n\rangle \stackrel{*}{\rightsquigarrow}_{\mathrm{eRI}} \langle \emptyset, H^\sharp, G^\sharp\rangle$. □

Lemma 6 (property of $H^\sharp$). *Let* $\langle E_n, H_n, G_n\rangle \stackrel{*}{\rightsquigarrow}_{\mathrm{eRI}} \langle \emptyset, H^\sharp, G^\sharp\rangle$. *Then* $\rightarrow_{H^\sharp} \subseteq \rightarrow_{\mathcal{R}} \circ \stackrel{*}{\rightarrow}_{(\mathcal{R}\cup H^\sharp)/G^\sharp} \circ \stackrel{*}{\leftrightarrow}_{G^\sharp} \circ \stackrel{*}{\leftarrow}_{(\mathcal{R}\cup H^\sharp)/G^\sharp}$ *on* $\mathrm{T}(\mathcal{F})$.

Proof. Suppose $s \rightarrow t \in H^\sharp$, $s_g \rightarrow_{\{s\rightarrow t\}} t_g$, and let $s_g \equiv C_g[s\sigma_g]$, $t_g \equiv C_g[t\sigma_g]$.

It suffices to consider the case when $\sigma_g(x) \in \mathrm{NF}(\rightarrow_{\mathcal{R}})$ for any $x \in V(s)$. For, by $\mathrm{SN}(\rightarrow_{\mathcal{R}})$, there exists a substitution $\hat{\sigma}_g$ such that $\sigma_g(x) \stackrel{*}{\rightarrow}_{\mathcal{R}} \hat{\sigma}_g(x) \in \mathrm{NF}(\rightarrow_{\mathcal{R}})$ for any $x \in V(s)$. Thus if once we have shown $C_g[s\hat{\sigma}_g] \rightarrow_{\mathcal{R}} \circ \stackrel{*}{\rightarrow}_{(\mathcal{R}\cup H^\sharp)/G^\sharp} \circ \stackrel{*}{\leftrightarrow}_{G^\sharp} \circ \stackrel{*}{\leftarrow}_{(\mathcal{R}\cup H^\sharp)/G^\sharp} C_g[t\hat{\sigma}_g]$, then by $C_g[s\sigma_g] \stackrel{*}{\rightarrow}_{\mathcal{R}} C_g[s\hat{\sigma}_g]$, $C_g[t\sigma_g] \stackrel{*}{\rightarrow}_{\mathcal{R}} C_g[t\hat{\sigma}_g]$, we would have $C_g[s\sigma_g] \rightarrow_{\mathcal{R}} \circ \stackrel{*}{\rightarrow}_{(\mathcal{R}\cup H^\sharp)/G^\sharp} \circ \stackrel{*}{\leftrightarrow}_{G^\sharp} \circ \stackrel{*}{\leftarrow}_{(\mathcal{R}\cup H^\sharp)/G^\sharp} C_g[t\sigma_g]$.

Thus let us suppose that $\sigma_g(x) \in \mathrm{NF}(\rightarrow_{\mathcal{R}})$ for any $x \in V(s)$. Then by the quasi-reducibility of $\mathcal{R}$, σ_g is a constructor substitution. For some n, we have $\langle E_0, \emptyset, \emptyset\rangle \stackrel{*}{\rightsquigarrow}_{\mathrm{eRI}} \langle E_n, H_n, G_n\rangle \rightsquigarrow_{\mathrm{eRI}} \langle E_{n+1}, H_{n+1}, G_{n+1}\rangle \stackrel{*}{\rightsquigarrow}_{\mathrm{eRI}} \langle \emptyset, H^\sharp, G^\sharp\rangle$, $H_{n+1} = H_n \cup \{s \rightarrow t\}$ where $E_n = E \uplus \{s \doteq t\}$, $E_{n+1} = E \cup \mathrm{Expd}_u(s,t)$, $u \in \mathcal{B}(s)$, $s \succ t$. Then by Lemma 1 (1) we have $s\sigma_g \rightarrow_{\mathcal{R}} \circ \leftrightarrow_{\mathrm{Expd}_u(s,t)} t\sigma_g$. Thus $s_g \equiv C_g[s\sigma_g] \rightarrow_{\mathcal{R}} \circ \leftrightarrow_{E_{n+1}} C_g[t\sigma_g] \equiv t_g$. Therefore, by Lemma 5, $s_g \rightarrow_{\mathcal{R}} \circ \stackrel{*}{\rightarrow}_{(\mathcal{R}\cup H^\sharp)/G^\sharp} \circ \stackrel{*}{\leftrightarrow}_{G^\sharp} \circ \stackrel{*}{\leftarrow}_{(\mathcal{R}\cup H^\sharp)/G^\sharp} t_g$. □

Lemma 7 (property of $G^\sharp$). *Let* $\langle E_n, H_n, G_n\rangle \stackrel{*}{\rightsquigarrow}_{\mathrm{eRI}} \langle \emptyset, H^\sharp, G^\sharp\rangle$. *Then* $\leftrightarrow_{G^\sharp} \subseteq \rightarrow_{\mathcal{R}} \circ \stackrel{*}{\rightarrow}_{(\mathcal{R}\cup H^\sharp)/G^\sharp} \circ \stackrel{*}{\leftrightarrow}_{G^\sharp} \circ \stackrel{*}{\leftarrow}_{(\mathcal{R}\cup H^\sharp)/G^\sharp} \circ \leftarrow_{\mathcal{R}}$ *on* $\mathrm{T}(\mathcal{F})$.

Proof. Similar to Lemma 6 using Lemma 2 (1) instead of Lemma 1 (1).

Theorem 1 (correctness of enhanced rewriting induction). *Let $\mathcal{R}$ be a quasi-reducible TRS, E a set of equations, $\succsim$ a reduction quasi-order satisfying $\mathcal{R} \subseteq \succ$. If there exist sets H, G (of rewrite rules and of equations, respectively) such that $\langle E, \emptyset, \emptyset\rangle \stackrel{*}{\rightsquigarrow}_{\mathrm{eRI}} \langle \emptyset, H, G\rangle$, then equations in E are inductive theorems of $\mathcal{R}$.*

Proof. By repeatedly applying Lemma 4, from $\langle E, \emptyset, \emptyset\rangle \stackrel{*}{\rightsquigarrow}_{\mathrm{eRI}} \langle \emptyset, H, G\rangle$, it follows that $\stackrel{*}{\leftrightarrow}_{E\cup\mathcal{R}} = \stackrel{*}{\leftrightarrow}_{\mathcal{R}\cup H\cup G}$ on $\mathrm{T}(\mathcal{F})$. Therefore it suffices to show $\stackrel{*}{\leftrightarrow}_{\mathcal{R}\cup H\cup G} = \stackrel{*}{\leftrightarrow}_{\mathcal{R}}$ holds on $\mathrm{T}(\mathcal{F})$. We apply Lemma 3 for $A = \mathrm{T}(\mathcal{F})$, $\rightarrow_1 = \rightarrow_{\mathcal{R}}$, $\rightarrow_2 = \rightarrow_H$, and $\rightarrow_3 = \rightarrow_G$.

By $\mathcal{R} \cup H \subseteq \succ$, $G \subseteq \approx$, the conditions (i),(ii) of Lemma 3 hold. The condition (iii) holds by Lemma 6 and (iv) by Lemma 7. Also by the side conditions of *Expand2* rule, we have $\mathcal{B}(s) \neq \emptyset$ and $\mathcal{B}(t) \neq \emptyset$ for any $s \doteq t \in G$. Thus by the quasi-reducibility of $\mathcal{R}$, $s_g \stackrel{+}{\leftrightarrow}_G t_g \Rightarrow s_g, t_g \notin \mathrm{NF}(\rightarrow_{\mathcal{R}})$ for any s_g, t_g. Hence the condition (v) holds. Therefore, by Lemma 3, $\stackrel{*}{\leftrightarrow}_{\mathcal{R}} = \stackrel{*}{\leftrightarrow}_{\mathcal{R}\cup H\cup G}$. □

Example 6 (enhanced rewriting induction). Let $\mathcal{R}$ and E be as follows:

$$\mathcal{R} = \left\{ \begin{array}{ll} \mathsf{plus}(0,y) & \to y \\ \mathsf{plus}(\mathsf{s}(x),y) & \to \mathsf{s}(\mathsf{plus}(x,y)) \end{array} \right\}$$

$$E = \left\{ \mathsf{plus}(x,y) \doteq \mathsf{plus}(y,x) \right\}$$

Let $\succsim$ be a recursive path order based on the precedence $\mathsf{plus} \succ \mathsf{s} \succ \mathsf{0}$. Then the eRI works as follows.

$$\begin{array}{ll}
& \langle \{ \mathsf{plus}(x,y) \doteq \mathsf{plus}(y,x) \}, \{\}, \{\} \rangle \\
\rightsquigarrow^{e2} & \left\langle \begin{array}{l} \left\{ \begin{array}{l} \mathsf{0} \doteq \mathsf{0},\ \mathsf{s}(x_1) \doteq \mathsf{s}(\mathsf{plus}(x_1,\mathsf{0})) \\ \mathsf{s}(\mathsf{plus}(x_2,\mathsf{s}(y_2))) \doteq \mathsf{s}(\mathsf{plus}(y_2,\mathsf{s}(x_2))) \end{array} \right\} \\ \{\}, \{ \mathsf{plus}(x,y) \doteq \mathsf{plus}(y,x) \} \end{array} \right\rangle \\
\rightsquigarrow^{d}\rightsquigarrow^{s}\rightsquigarrow^{d} & \left\langle \begin{array}{l} \{ \mathsf{s}(\mathsf{plus}(x_2,\mathsf{s}(y_2))) \doteq \mathsf{s}(\mathsf{plus}(y_2,\mathsf{s}(x_2))) \} \\ \{\}, \{ \mathsf{plus}(x,y) \doteq \mathsf{plus}(y,x) \} \end{array} \right\rangle \\
\rightsquigarrow^{s}\rightsquigarrow^{s}\rightsquigarrow^{d} & \langle \{\}, \{\}, \{ \mathsf{plus}(x,y) \doteq \mathsf{plus}(y,x) \} \rangle
\end{array}$$

By Theorem 1, the equation in E is an inductive theorem of $\mathcal{R}$.

5 Incremental proofs by rewriting induction

In the enhanced rewriting induction, a reduction order whose equational classes are "coarser" is more suitable to prove non-oriented conjectures. On the other hand, such a reduction order may fail to handle some equations orientable by other reduction orders.

Example 7 (handling multiple conjectures). Let $\mathcal{R}$ and E be as follows:

$$\mathcal{R} = \left\{ \begin{array}{ll} \mathsf{plus}(0,y) & \to y \\ \mathsf{plus}(\mathsf{s}(x),y) & \to \mathsf{s}(\mathsf{plus}(x,y)) \\ \mathsf{times}(0,y) & \to 0 \\ \mathsf{times}(\mathsf{s}(x),y) & \to \mathsf{plus}(\mathsf{times}(x,y),y) \end{array} \right\}$$

$$E = \left\{ \begin{array}{ll} \mathsf{plus}(x,y) & \doteq \mathsf{plus}(y,x) \\ \mathsf{plus}(x,\mathsf{plus}(y,z)) & \doteq \mathsf{plus}(\mathsf{plus}(x,y),z) \\ \mathsf{times}(x,y) & \doteq \mathsf{times}(y,x) \end{array} \right\}$$

The recursive path order can handle the commutativity equations but not the associativity equation. To the contrary, the lexicographic path order can handle the associativity equations but not the commutativity equations. As we will see, we need both commutativity and associativity of plus to prove the commutativity of times and thus eRI can not handle the commutativity of times.

This observation leads us to introduce *incremental rewriting induction* in which already-proved lemmas can be applied more easily. The incremental rewriting induction can employ different reduction orders in each phase so that it can be also benefited from variations of reduction orders.

We first formulate abstract principle of incremental rewriting induction. The proof is similar to that of Lemma 3.

Lemma 8 (principle of incremental rewriting induction). *Let* $\to_i$ $(1 \leq i \leq 4)$ *be a relation on a set* A*, and* $\succsim$ *be a well-founded quasi-order on* A*. Suppose*

$$\begin{array}{rl}
(i) & \to_{1\cup 2} \subseteq \succ \\
(ii) & \to_3 \subseteq \approx \\
(iii) & \to_4 \subseteq \overset{*}{\leftrightarrow}_1 \cap \succsim \\
(iv) & \to_2 \subseteq \to_1 \circ \overset{*}{\to}_{((1\cup 2)/3)\cup 4} \circ (\overset{*}{\leftrightarrow}_3 \cup \overset{*}{\leftrightarrow}_1) \circ \overset{*}{\leftarrow}_{((1\cup 2)/3)\cup 4} \\
(v) & \to_3 \subseteq \to_1 \circ \overset{*}{\to}_{((1\cup 2)/3)\cup 4} \circ (\overset{*}{\leftrightarrow}_3 \cup \overset{*}{\leftrightarrow}_1) \circ \overset{*}{\leftarrow}_{((1\cup 2)/3)\cup 4} \circ \leftarrow_1 \\
(vi) & \forall x, y \in \mathrm{NF}(\to_{(1\cup 2)/3}).\ (x \overset{*}{\leftrightarrow}_3 y \Rightarrow x = y).
\end{array}$$

Then $\overset{*}{\leftrightarrow}_1 = \overset{*}{\leftrightarrow}_{1\cup 2\cup 3\cup 4}$.

In Figure 4, we list inference rules designed based on this abstract principle.

Definition 3 (incremental rewriting induction). *We write* $\langle E, H, G\rangle \rightsquigarrow_{\mathrm{iRI}} \langle E', H', G'\rangle$ *when* $\langle E', H', G'\rangle$ *is obtained from* $\langle E, H, G\rangle$ *by applying one of the inference rules of Figure 4. The reflexive transitive closure of* $\rightsquigarrow_{\mathrm{iRI}}$ *is denoted by* $\overset{*}{\rightsquigarrow}_{\mathrm{iRI}}$*. We put superscripts* ^{s,s2,d,d2,e,e2} *to indicate which inference rule is used.*

Simplify

$$\frac{\langle E \uplus \{s \doteq t\},\ H,\ G\rangle}{\langle E \cup \{s' \doteq t\},\ H,\ G\rangle}\quad s \to_{(\mathcal{R}\cup H)/G} s'$$

Simplify2

$$\frac{\langle E \uplus \{s \doteq t\},\ H,\ G\rangle}{\langle E \cup \{s' \doteq t\},\ H,\ G\rangle}\quad s \overset{*}{\leftrightarrow}_{\mathcal{R}\cup\mathcal{E}} s',\ s \succsim s'$$

Delete

$$\frac{\langle E \uplus \{s \doteq t\},\ H,\ G\rangle}{\langle E,\ H,\ G\rangle}\quad s \overset{*}{\leftrightarrow}_G t$$

Delete2

$$\frac{\langle E \uplus \{s \doteq t\},\ H,\ G\rangle}{\langle E,\ H,\ G\rangle}\quad s \overset{*}{\leftrightarrow}_{\mathcal{R}\cup\mathcal{E}} t$$

Expand

$$\frac{\langle E \uplus \{s \doteq t\},\ H,\ G\rangle}{\langle E \cup \mathrm{Expd}_u(s,t),\ H \cup \{s \to t\},\ G\rangle}\quad u \in \mathcal{B}(s), s \succ t$$

Expand2

$$\frac{\langle E \uplus \{s \doteq t\}, H, G\rangle}{\langle E \cup \mathrm{Expd2}_{u,v}(s,t), H, G \cup \{s \doteq t\}\rangle}\quad u \in \mathcal{B}(s), v \in \mathcal{B}(t), s \approx t$$

Fig. 4. Inference rules of iRI

The correctness of the incremental rewriting induction is proved similarly to the enhanced rewriting induction by putting $\to_1 = \to_{\mathcal{R}}$, $\to_2 = \to_H$, $\to_3 = \to_G$, and $\to_4 = \overset{*}{\leftrightarrow}_{\mathcal{R}\cup\mathcal{E}} \cap \succsim$.

Theorem 2 (correctness of incremental rewriting induction). *Let $\mathcal{R}$ be a quasi-reducible TRS, $E, \mathcal{E}$ sets of equations, $\succsim$ a reduction quasi-order satisfying $\mathcal{R} \subseteq \succ$. Suppose equations in $\mathcal{E}$ are inductive theorems of $\mathcal{R}$. If there exist sets H, G (of rewrite rules and of equations, respectively) such that $\langle E, \emptyset, \emptyset\rangle \overset{*}{\rightsquigarrow}_{\mathrm{iRI}} \langle \emptyset, H, G\rangle$, then equations in E are inductive theorems of $\mathcal{R}$.*

Example 8 (incremental rewriting induction). Let $\mathcal{R}$, $\mathcal{E}$ and E be as below. We may suppose that equations in $\mathcal{E}$ has been already proved (Examples 1, 6).

$$\mathcal{R} = \left\{ \begin{array}{ll} \mathsf{plus}(\mathsf{0}, y) & \to y \\ \mathsf{plus}(\mathsf{s}(x), y) & \to \mathsf{s}(\mathsf{plus}(x, y)) \\ \mathsf{times}(\mathsf{0}, y) & \to \mathsf{0} \\ \mathsf{times}(\mathsf{s}(x), y) & \to \mathsf{plus}(\mathsf{times}(x, y), y) \end{array} \right\}$$

$$\mathcal{E} = \left\{ \begin{array}{ll} \mathsf{plus}(x, \mathsf{plus}(y, z)) & \doteq \mathsf{plus}(\mathsf{plus}(x, y), z) \\ \mathsf{plus}(x, y) & \doteq \mathsf{plus}(y, x) \end{array} \right\}$$

$$E = \left\{ \mathsf{times}(x, y) \doteq \mathsf{times}(y, x) \right\}$$

Then the incremental rewriting induction by the recursive path order based on precedence $\mathsf{times} \succ \mathsf{plus} \succ \mathsf{s} \succ \mathsf{0}$ proceeds as follows:

$$\begin{array}{l}
\left\langle \left\{ \mathsf{times}(x, y) \doteq \mathsf{times}(y, x) \right\}, \{\}, \{\} \right\rangle \\
\rightsquigarrow^{e2}_{\mathrm{iRI}} \\
\left\langle \begin{array}{l} \left\{ \begin{array}{l} \mathsf{0} \doteq \mathsf{0} \\ \mathsf{plus}(\mathsf{times}(x_1, \mathsf{0}), \mathsf{0}) \doteq \mathsf{0} \\ \mathsf{plus}(\mathsf{times}(x_1, \mathsf{s}(y_1)), \mathsf{s}(y_1)) \doteq \mathsf{plus}(\mathsf{times}(y_1, \mathsf{s}(x_1)), \mathsf{s}(x_1)) \end{array} \right\} \\ \{\}, \left\{ \mathsf{times}(x, y) \doteq \mathsf{times}(y, x) \right\} \end{array} \right\rangle \\
\rightsquigarrow^{d}_{\mathrm{iRI}} \rightsquigarrow^{s}_{\mathrm{iRI}} \rightsquigarrow^{s}_{\mathrm{iRI}} \rightsquigarrow^{d}_{\mathrm{iRI}} \\
\left\langle \begin{array}{l} \left\{ \mathsf{plus}(\mathsf{times}(x_1, \mathsf{s}(y_1)), \mathsf{s}(y_1)) \doteq \mathsf{plus}(\mathsf{times}(y_1, \mathsf{s}(x_1)), \mathsf{s}(x_1)) \right\} \\ \{\}, \left\{ \mathsf{times}(x, y) \doteq \mathsf{times}(y, x) \right\} \end{array} \right\rangle \\
\rightsquigarrow^{s}_{\mathrm{iRI}} \rightsquigarrow^{s2}_{\mathrm{iRI}} \rightsquigarrow^{s}_{\mathrm{iRI}} \rightsquigarrow^{s}_{\mathrm{iRI}} \rightsquigarrow^{s2}_{\mathrm{iRI}} \rightsquigarrow^{s}_{\mathrm{iRI}} \\
\left\langle \begin{array}{l} \left\{ \mathsf{s}(\mathsf{plus}(y_1, \mathsf{plus}(\mathsf{times}(x_1, y_1), x_1))) \doteq \mathsf{s}(\mathsf{plus}(x_1, \mathsf{plus}(\mathsf{times}(x_1, y_1), y_1))) \right\} \\ \{\}, \left\{ \mathsf{times}(x, y) \doteq \mathsf{times}(y, x) \right\} \end{array} \right\rangle \\
\rightsquigarrow^{d2}_{\mathrm{iRI}} \\
\left\langle \{\}, \{\}, \left\{ \mathsf{times}(x, y) \doteq \mathsf{times}(y, x) \right\} \right\rangle
\end{array}$$

Thus the commutativity of times has been proved.

6 Conclusion

We have presented an extension of the rewriting induction that can deal with conjectures not orientable by the given reduction order. We gave inference rules of the enhanced rewriting induction and proved its correctness. We have also present incremental rewriting induction in which already-proved lemmas can be applied more easily.

Our approach to deal with non-orientable equations is based on the rewriting modulo equations originally suggested in [12]. Another approach is to use ordered rewriting technique [2, 6]. The latter approach is embodied in the inductive theorem prover SPIKE [3, 4]. Below we list some results of inductive theorem proving of non-orientable conjectures (in purely equational setting) by SPIKE and our inference systems. It appears that results of these two approaches are quite different even in simple examples. In particular, our system can not directly deal with conjectures that are incomparable in the given reduction quasi-order, although SPIKE can deal with such conjectures directly. On the other hand, our system successfully handle commutativity equations that are hard for SPIKE.

(many-sorted) conjectures	SPIKE	Enhanced RI	Incremental RI
$\mathsf{max}(x, y) \doteq \mathsf{max}(y, x)$	$\surd$	$\surd$	$\surd$
$\mathsf{minus}(\mathsf{minus}(x, y), z) \doteq \mathsf{minus}(\mathsf{minus}(x, z), y)$	$\surd$	$\surd$	$\surd$
$\mathsf{len}(\mathsf{app}(xs, ys)) \doteq \mathsf{len}(\mathsf{app}(ys, xs))$	$\surd$	$\surd$	$\surd$
$\mathsf{len}(\mathsf{qrev}(xs, ys)) \doteq \mathsf{len}(\mathsf{qrev}(ys, xs))$	$\times$	$\surd$	$\surd$
$\mathsf{plus}(x, \mathsf{plus}(y, z)) \doteq \mathsf{plus}(y, \mathsf{plus}(x, z))$	$\surd$	$\times$	$\times$
$\left\{\begin{array}{r@{}l} \mathsf{plus}(x, y) &{} \doteq \mathsf{plus}(y, x) \\ \mathsf{plus}(x, \mathsf{plus}(y, z)) &{} \doteq \mathsf{plus}(\mathsf{plus}(x, y), z) \\ \mathsf{plus}(x, \mathsf{plus}(y, z)) &{} \doteq \mathsf{plus}(y, \mathsf{plus}(x, z)) \end{array}\right\}$	$\surd$	$\times$	$\surd$
$\left\{\begin{array}{r@{}l} \mathsf{plus}(x, \mathsf{plus}(y, z)) &{} \doteq \mathsf{plus}(\mathsf{plus}(x, y), z) \\ \mathsf{plus}(x, y) &{} \doteq \mathsf{plus}(y, x) \\ \mathsf{times}(x, y) &{} \doteq \mathsf{times}(y, x) \end{array}\right\}$	$\times$	$\times$	$\surd$
$\left\{\begin{array}{r@{}l} \mathsf{plus}(x, \mathsf{plus}(y, z)) &{} \doteq \mathsf{plus}(\mathsf{plus}(x, y), z) \\ \mathsf{plus}(x, y) &{} \doteq \mathsf{plus}(y, x) \\ \mathsf{sum}(\mathsf{app}(xs, ys)) &{} \doteq \mathsf{sum}(\mathsf{app}(ys, xs)) \end{array}\right\}$	$\times$	$\times$	$\surd$

To see the difference, we show how the process proving the commutativity of times proceeds in SPIKE. First by expansion rule, it produces

$$0 \doteq \mathsf{times}(0, 0) \tag{7}$$

$$\mathsf{plus}(\mathsf{times}(x_1, 0), 0) \doteq \mathsf{times}(0, \mathsf{s}(x_1)) \tag{8}$$

$$0 \doteq \mathsf{times}(\mathsf{s}(x_1), 0) \tag{9}$$

$$\mathsf{plus}(\mathsf{times}(x_2, \mathsf{s}(x_1)), \mathsf{s}(x_1)) \doteq \mathsf{times}(\mathsf{s}(x_1), \mathsf{s}(x_2)) \tag{10}$$

In the presence of commutativity and associativity equations for plus (as proved lemmas), the successive simplification procedure works as follows:

(7) $\Rightarrow 0 \doteq 0 \Rightarrow$ *deleted*
(8) $\Rightarrow \mathsf{plus}(\mathsf{times}(0, x_1), 0) \doteq \mathsf{times}(0, \mathsf{s}(x_1)) \Rightarrow 0 \doteq 0 \Rightarrow$ *deleted*

$$
\begin{aligned}
(9) &\Rightarrow \mathsf{0} \doteq \mathsf{plus}(\mathsf{times}(x_1, \mathsf{0}), \mathsf{0}) \\
(10) &\Rightarrow \mathsf{plus}(\mathsf{times}(\mathsf{s}(x_1), x_2), \mathsf{s}(x_1)) \doteq \mathsf{times}(\mathsf{s}(x_1), \mathsf{s}(x_1)) \\
&\Rightarrow \mathsf{plus}(\mathsf{plus}(\mathsf{times}(x_1, x_2), x_2), \mathsf{s}(x_1)) \doteq \mathsf{plus}(\mathsf{times}(x_1, \mathsf{s}(x_2)), \mathsf{s}(x_2)) \\
&\Rightarrow \mathsf{plus}(\mathsf{times}(x_1, x_2), \mathsf{plus}(x_2, \mathsf{s}(x_1))) \doteq \mathsf{plus}(\mathsf{times}(x_1, \mathsf{s}(x_2)), \mathsf{s}(x_2))
\end{aligned}
$$

Thus, it results in a two elements set of remaining conjectures. On the other hand, in our procedure, as shown in Example 8, one expansion and successive simplifications successfully eliminate all equations.

Acknowledgments

Thanks are due to anonymous referees for helpful comments and suggestions. This work was partially supported by a grant from Japan Society for the Promotion of Science, No. 17700002.

References

1. F. Baader and T. Nipkow. *Term Rewriting and All That*. Cambridge University Press, 1998.
2. L. Bachmair, N. Dershowitz, and D. A. Plaisted. Completion without failure. In *Resolution of Equations in Algebraic Structure*, volume 2, pages 1–30. Academic Press, 1989.
3. A. Bouhoula. Automated theorem proving by test set induction. *Journal of Symbolic Computation*, 23:47–77, 1997.
4. A. Bouhoula, E. Kounalis, and M. Rusinowitch. Automated mathematical induction. *Journal of Logic and Computation*, 5(5):631–668, 1995.
5. R. S. Boyer and J. S. Moore. *A Computational Logic*. Academic Press, 1979.
6. N. Dershowitz and U. S. Reddy. Deductive and inductive synthesis of equational programs. *Journal of Symbolic Computation*, 15:467–494, 1993.
7. G. Huet and J.-M. Hullot. Proof by induction in equational theories with constructors. *Journal of Computer and System Sciences*, 25(2):239–266, 1982.
8. J.-P. Jouannaud and E. Kounalis. Automatic proofs by induction in theories without constructors. *Information and Computation*, 82:1–33, 1989.
9. D. Kapur, P. Narendran, and H. Zhang. Automating inductionless induction using test sets. *Journal of Symbolic Computation*, 11(1–2):81–111, 1991.
10. H. Koike and Y. Toyama. Inductionless induction and rewriting induction. *Computer Software*, 17(6):1–12, 2000. In Japanese.
11. D. R. Musser. On proving inductive properties of abstract data types. In *Proc. of the 7th Annual ACM Symposium on Principles of Programming Languages*, pages 154–162. ACM Press, 1980.
12. U. S. Reddy. Term rewriting induction. In *Proc. of the 10th International Conference on Automated Deduction*, volume 449 of *LNAI*, pages 162–177. Springer-Verlag, 1990.
13. K. Sakamoto, T. Aoto, and Y. Toyama. Fusion transformation based on rewriting induction. In *Proc. of the JSSST 21th Annual Conference,* 2B-3, 2004. In Japanese.
14. Terese. *Term Rewriting Systems*. Cambridge University Press, 2003.
15. Y. Toyama. How to prove equivalence of term rewriting systems without induction. *Theoretical Computer Science*, 90(2):369–390, 1991.