ICOT Technical Report: TR- 062

TR-062

An Ordering Method for Term Rewriting Systems

by

Kô Sakai

April, 1984

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03) 456-3191 ~ 5
Telex ICOT J32964

Institute for New Generation Computer Technology

An Ordering Method for Term Rewriting Systems

Kō Sakai

ICOT Research Center
Institute for New Generation Computer Technology
Tokyo, Japan

ABSTRACT

The properties that a partial order on terms should possess for being applied to the Knuth-Bendix algorithm is discussed. A well-founded order having the desired properties is introduced and its efficacy is demonstrated by means of several examples.

1. Preliminaries

In this section, we introduce the terminology and notation used in this paper and briefly summarize some well-known results.

1.1 Ordered sets

Definition 1.1

Let X be a set and $\preceq$ be a relation on X. Then, the pair $(X, \preceq)$ is said to be an *ordered set* if it satisfies the following conditions:

(1) $x \preceq x$ for all x in X

(2) If $x \preceq y$ and $y \preceq z$, then $x \preceq z$

(3) If $x \preceq y$ and $y \preceq x$, then $x = y$

Definition 1.2

An ordered set $(X, \preceq)$ is said to be *totally ordered* if $x \preceq y$ or $y \preceq x$ for all x and y in X.

Definition 1.3

Let $(X, \preceq)$ be an ordered set.

(1) An element x is said to be *minimum* in a subset S of X if $x \preceq y$ for all y in S.

(2) An element x is said to be *minimal* in a subset S of X if there exists no y in S such that $y \prec x$. (The notation $x \prec y$ or $y \succ x$ means that $x \preceq y$ and $x \neq y$.)

If there exists a minimum element, then it is unique.

Definition 1.4

An ordered set $(X, \preceq)$ is said to be *well-ordered* if every non-empty subset of X has a minimum element.

Definition 1.5

An ordered set $(X, \preceq)$ is said to be *semi-well-ordered* if for every infinite sequence $x_1, x_2, \cdots$ of elements of X there exist i and j such that $x_i \preceq x_j$ and $i < j$.

Definition 1.6

An ordered set $(X, \preceq)$ is said to be *well-founded* if there exists no infinite sequence $x_1 \succ x_2 \succ \cdots$ of elements of X.

Theorem 1.7

(1) An ordered set $(X, \preceq)$ is well-ordered if and only if every non-empty subset of X has exactly one minimal element.

(2) An ordered set $(X, \preceq)$ is semi-well-ordered if and only if every non-empty

subset of x has a finite number of minimal elements.

(3) An ordered set $(X, \preceq)$ is well-founded if and only if every non-empty subset of X has (possibly infinite) minimal elements.

Corollary 1.8

(1) A well-ordered set is totally ordered and semi-well-ordered.

(2) A semi-well-ordered set is well-founded.

(3) A totally ordered well-founded set is well-ordered.

Remark 1.9

Let $(X, \preceq)$ be an ordered set. Then, the relation $\prec$ satisfies the following condition:

if $x \prec y$ and $y \prec z$, then $x \prec z$ and $x \neq z$.

On the other hand, let $\prec$ be an arbitrary relation on X satisfying the above condition and $\preceq$ be defined by:

$x \preceq y$ if and only if $x = y$ or $x \prec y$.

Then, $(X, \preceq)$ is an ordered set.

1.2 Term rewriting systems

In this section, we will deal with finite sequences of the following three kinds of symbols (and parentheses and commas in order to make reading easy):

(1) a finite set F of *function symbols* in which each symbol has a fixed arity of arguments

(2) a denumerable set V of *variables*

(3) a special symbol Ω called a *slot*.

Definition 1.10

The *terms* and the *contexts* on F and V are defined recursively as follows:

(1) Every variable in V is a term and a context.

(2) A slot Ω is a context.

(3) If f is an n-ary function symbol in F and $t_1, \cdots, t_n$ are terms (contexts), then $f(t_1, \cdots, t_n)$ is a term (context). (We allow the case that $n=0$, and call such a function symbol a constant.)

The set of all terms on F and V is denoted by $\mathfrak{T}(F, V)$. A term without variables is called a ground term. The subset of $\mathfrak{T}(F, V)$ consisting of all ground terms is denoted by $\mathfrak{T}(F)$.

Definition 1.11

Let $c[\Omega_1, \cdots, \Omega_n]$ denote a context with n slots, where Ω_i indicates the i'th slot form the left. Let $t_1, \cdots, t_n$ be terms (contexts). Then $c[t_1, \cdots, t_n]$ denotes the term (context) obtained by replacing the slot Ω_i with t_i. In particular, we will use the notation $c[\Omega]$ for representing a context containing precisely one slot. A term s is said to be a *subterm* of t if there is a context $c[\Omega]$ such that $t = c[s]$. If $c[\Omega] \neq \Omega$, s is called a *proper subterm* of $c[s]$.

Definition 1.12

A function σ from V to $\mathfrak{T}(F, V)$ is said to be a (finite) *substitution* if $\sigma(v) = v$ for all but a finite number of v's in V.

A substitution can be extended homomorphically to a function σ^* from $\mathfrak{T}(F, V)$ to $\mathfrak{T}(F, V)$. This is defined recursively as follows:

(1) $\sigma^*(v) = \sigma(v)$ for all v in V.

(2) $\sigma^*(f(t_1, \cdots, t_n)) = f(\sigma^*(t_1), \cdots, \sigma^*(t_n))$.

A substitution can be extended also to a function from contexts to contexts. These extended functions will be called a substitution and denoted by σ as well. Note that $\sigma(c[s]) = \sigma(c)[\sigma(s)]$.

Definition 1.13

A *term rewriting system* (*TRS* for short) is a finite set of pairs $l \rightarrow r$ of terms. An element $l \rightarrow r$ of a TRS is called a *rewrite rule*.

In order to avoid renaming variables in the following discussion, we will assume that no two rewrite rules have common variables.

Definition 1.14

Let R be a TRS. A pair $t \Rightarrow u$ of terms is said to be a *derivation* with respect to R if there exist a rewrite rule $l \rightarrow r$, a context $c[\Omega]$, and a substitution σ such that $c[\sigma(l)] = t$ and $c[\sigma(r)] = u$. Let us denote the reflexive transitive closure of $\Rightarrow$ by $\overset{*}{\Rightarrow}$ and the transitive closure $\overset{+}{\Rightarrow}$.

Definition 1.15

Let R be a TRS. Two terms u and v are said to be *confluent* (with respect to R) if there exists a term t such that $u \overset{*}{\Rightarrow} t$ and $v \overset{*}{\Rightarrow} t$. A TRS is said to be *confluent* if for any two derivation sequences $t \overset{*}{\Rightarrow} t_1$ and $t \overset{*}{\Rightarrow} t_2$, t_1 and t_2 are confluent.

Definition 1.16

A TRS is said to *terminate* if there exists no infinite derivation sequence $t_1 \Rightarrow t_2 \Rightarrow \cdots$

Proposition 1.17

A TRS terminates if and only if $(\mathfrak{T}(F, V), \overset{+}{\Rightarrow})$ is a well-founded set.

Theorem 1.18

A terminating TRS is confluent if and only if for any two derivations $t \Rightarrow t_1$ and $t \Rightarrow t_2$ there exists a term u such that $t_1 \overset{*}{\Rightarrow} u$ and $t_2 \overset{*}{\Rightarrow} u$.

Definition 1.19

A term t is said to be *irreducible* if there exists no term u such that $t \Rightarrow u$.

Theorem 1.20

Let R be a terminating TRS. For every term t, there exists an irreducible term u such that $t \overset{*}{\Rightarrow} u$. Moreover, R is confluent if and only if the irreducible term u is unique. In this case, the term u is called the normal form of t (with respect to R) and denoted by $\mathcal{N}(t)$.

Definition 1.21

A relation ρ on $\mathfrak{T}(F, V)$ is said to have the *substitution property* if $t \;\rho\; u$ implies that $\sigma(t) \;\rho\; \sigma(u)$ for any substitution σ.

Definition 1.22

A relation ρ on $\mathfrak{T}(F, V)$ is said to have the *replacement property* if, for any context $c[\Omega_1, \cdots \Omega_n]$, $t_1 \rho u_1, \cdots, t_n \rho u_n$ implies that $c[t_1, \cdots, t_n] \;\rho\; c[u_1, \cdots, u_n]$.

Definition 1.23

An *equational theory* is a set of pairs $t_1 \sim t_2$ of terms satisfying the following conditions. (We will use the symbol $\sim$ instead of $=$, because the symbol $=$ means identity in this paper.)

(1) $t \sim t$ for all term t.

(2) if $t_1 \sim t_2$, then $t_2 \sim t_1$.

(3) if $t_1 \sim t_2$, $t_2 \sim t_3$, then $t_1 \sim t_3$.

(4) $\sim$ has the substitution property.

(5) $\sim$ has the replacement property.

The equality problem in an equational theory T involves the determination of whether $t_1 \sim t_2$ for two arbitrary terms t_1 and t_2.

Any set E of pairs $l \sim r$ of terms can be extended to an equational theory $T(E)$ by considering the closure of E with respect to the above conditions (1)-(5). An equational theory T is said to be (finitely) *axiomatizable* if there exists a finite set E such that $T = T(E)$. In this case, E is called an *axiom system* for T, and an element of E is called an *axiom*.

Decidability Theorem

If R is a confluent and terminating TRS, then the equality problem on $T(R)$ is decidable.

Outline of Proof :

Let t_1 and t_2 be given two terms. Find $\mathcal{N}(t_1)$ and $\mathcal{N}(t_2)$. Then, $\mathcal{N}(t_1)=\mathcal{N}(t_2)$ if and only if $t_1 \sim t_2$.

2. TRS Equivalence

In this section we will discuss the relation between two TRSs. Let E_1 and E_2 be two equational theories. It is natural to say that E_2 implies E_1 if $E_1 \subseteq E_2$. Since, in this paper, TRSs are considered to be mechanical methods for solving the equality problem in an equational theory, implication between TRSs should agree with that between equational theories.

Definition 2.1

Let R_1 and R_2 be TRSs. R_2 is said to *weakly imply* R_1, if any two terms confluent with respect to R_2 are also confluent with respect to R_1.

If R_2 weakly implies R_1, then it is obvious that $T(R_1)$ implies $T(R_2)$.

Lemma 2.2

Let R_2 be a confluent TRS. R_2 weakly implies R_1 if and only if for any u and v such that $u \overset{*}{\to}_1 v$ there exists a term t such that $u \overset{*}{\to}_2 t$ and $v \overset{*}{\to}_2 t$. (The symbol $\overset{*}{\to}_i$ represents a derivation sequence with respect to R_i).

Proof :

The only-if-part is obvious. We prove the if-part. Let u and v are confluent with respect to R_1, i.e., there exists a term t such that $u \overset{*}{\to}_1 t$ and $v \overset{*}{\to}_1 t$. From the condition of the lemma, there exist u_1 and v_1 such that $u \overset{*}{\to}_2 u_1$, $t \overset{*}{\to}_2 u_1$, $v \overset{*}{\to}_2 v_1$, and $t \overset{*}{\to}_2 v_1$. Since R_2 is confluent, u_1 and v_1 are confluent with respect to R_2, and therefore, u and v are also confluent with respect to R_2.

Proposition 2.3

Let R_1 and R_2 be confluent and terminating TRSs. The following conditions are equivalent:

(1) R_2 weakly implies R_1

(2) if $\mathcal{N}_1(t)=\mathcal{N}_1(s)$, then $\mathcal{N}_2(t)=\mathcal{N}_2(s)$

(3) $\mathcal{N}_2(\mathcal{N}_1(t))=\mathcal{N}_2(t)$

where $\mathcal{N}_i(t)$ represents the normal form of t with respect to R_i.

Proof :

(1)→(3): Since $t \overset{*}{\to}_1 \mathcal{N}_1(t)$, there exists a term s such that $t \overset{*}{\to}_2 s$ and $\mathcal{N}_1(t) \overset{*}{\to}_2 s$. Therefore,

$$\mathcal{N}_2(t)=\mathcal{N}_2(s)=\mathcal{N}_2(\mathcal{N}_1(t))$$

(3)→(2): If $\mathcal{N}_1(t)=\mathcal{N}_1(s)$, then

$$\mathcal{N}_2(t)=\mathcal{N}_2(\mathcal{N}_1(t))=\mathcal{N}_2(\mathcal{N}_1(s))=\mathcal{N}_2(s)$$

(2)→(1): If $t \overset{*}{\to}_1 s$, then $\mathcal{N}_1(t)=\mathcal{N}_1(s)$. Therefore, $\mathcal{N}_2(t)=\mathcal{N}_2(s)=u$. Thus $t \overset{*}{\to}_2 u$ and $s \overset{*}{\to}_2 u$.

Corollary 2.4

Let R_1 and R_2 be confluent and terminating TRSs. Then R_1 weakly implies R_2 if and only if $T(R_1)$ implies $T(R_2)$.

Definition 2.5

Two TRSs R_1 and R_2 are said to be *weakly equivalent*, if they weakly imply each other.

Corollary 2.6

Let R_1 and R_2 be confluent and terminating TRSs. R_1 and R_2 are weakly equivalent if and only if $T(R_1)=T(R_2)$.

Even if two confluent and terminating TRSs are weakly equivalent, their normal forms are not necessarily the same. We will now define strong equivalence, which makes the normal forms the same.

Definition 2.7

Let R_1 and R_2 be TRSs. R_1 is said to *strongly imply* R_2, if R_2 weakly implies R_2 and any irreducible term with respect to R_2 is also irreducible with respect to R_1.

Lemma 2.8

Let R_1 and R_2 be confluent and terminating TRSs. Then, R_2 strongly implies R_1 if and only if

$$\mathcal{N}_1(\mathcal{N}_2(t))=\mathcal{N}_2(\mathcal{N}_1(t)) =\mathcal{N}_2(t)$$

for any term t.

Proof :

Immediate form Proposition 2.3 and Definition 2.7.

Definition 2.9

Two TRSs R_1 and R_2 are said to be *strongly equivalent* if they strongly imply each other.

Theorem 2.10

Let R_1 be a confluent and terminating TRS and R_2 be a terminating TRS. Then the following conditions are equivalent:

(1) R_1 and R_2 are strongly equivalent.

(2) R_1 weakly implies R_2 and any term irreducible with respect to R_2 is also irreducible with respect to R_1.

(3) R_2 is confluent and $\mathcal{N}_1(t)=\mathcal{N}_2(t)$ for any term t.

Proof :

(1)→(2): Clear.

(2)→(3): Let $t\overset{*}{\rightarrow}_2 u$ and $t\overset{*}{\rightarrow}_2 v$. Since R_2 is terminating, there exist terms u_0 and v_0 that are irreducible with respect to R_2 such that $u\overset{*}{\rightarrow}_2 u_0$ and $v\overset{*}{\rightarrow}_2 v_0$. Since R_1 weakly implies R_2, there exist terms v_1 and u_1 such that $t\overset{*}{\rightarrow}_1 u_1$, $u_0\overset{*}{\rightarrow}_1 u_1$, $t\overset{*}{\rightarrow}_1 v_1$ and $v_0\overset{*}{\rightarrow}_1 v_1$. Since R_1 is confluent there exists a term t_0 such that $u_1\overset{*}{\rightarrow}_1 t_0$ and $v_1\overset{*}{\rightarrow} t_0$. However, u_0 and v_0 is also irreducible with respect to R_1 and, therefore, $t_0=v_0=u_0$. Hence u and v are confluent. Therefore $\mathcal{N}_1(t)=\mathcal{N}_1(\mathcal{N}_2(t))=\mathcal{N}_2(t)$.

(3)→(1): Since $\mathcal{N}_1(\mathcal{N}_2(t))=\mathcal{N}_2(\mathcal{N}_2(t))=\mathcal{N}_2(t)$ and $\mathcal{N}_2(\mathcal{N}_1(t))=\mathcal{N}_2(\mathcal{N}_2(t)) =\mathcal{N}_2(t)$, R_2 strongly implies R_1. The symmetric discussion proves that R_1 strongly implies R_2.

Definition 2.11

R_2 is said to *trace* R_1, if $u\overset{*}{\rightarrow}_2 v$ for any u and v such that $u\overset{*}{\rightarrow}_1 v$.

Clearly, if R_2 traces R_1, then R_2 weakly implies R_1. The following corollary of Theorem 2.10 is convenient.

Corollary 2.12

Let R_1 be a confluent and terminating TRS, and R_2 be a terminating TRS. R_2 is confluent and strongly equivalent to R_1, if R_1 traces R_2, and any term irreducible with respect to R_2 is also irreducible with respect to R_1.

3. Knuth-Bendix Algorithm

Definition 3.1

A relation ρ on $\mathfrak{T}(F,V)$ is said to be *stable* if ρ has both the substitution property and the replacement property.

Proposition 3.2

Let $(\mathfrak{T}(F,V),\preceq)$ be an ordered set such that $\preceq$ is stable, and R be a TRS such that $l\succ r$ for all rewrite rules $l\rightarrow r$. Then, $t\succ u$ for all $t\Rightarrow u$.

Corollary 3.3

Let $(\mathfrak{T}(F,V),\preceq)$ be a well-founded set such that $\preceq$ is stable, and R be a TRS such that $l \succ r$ for all rewrite rules $l \rightarrow r$. Then, R terminates.

Definition 3.4

The two terms t_1 and t_2 are said to be *unifiable* if there exists a substitution σ such that $\sigma(t_1)=\sigma(t_2)$. The substitution σ is called a *unifier* of t_1 and t_2. A unifier μ of t_1 and t_2 is said to be *most general* if for any unifier ν of t_1 and t_2 there exists a substitution ϕ such that $\phi \circ \mu = \nu$. (The composition of functions f and g is represented by $f \circ g$, i.e., $f \circ g(x) = f(g(x))$.)

Unification Theorem [Robinson 65]

There exists an algorithm that determines whether two given terms are unifiable, and that finds a most general unifier when they are.

Superposition Theorem [Knuth 70]

A terminating TRS is confluent if and only if the following condition is satisfied for all pairs of rewriting rules $l_1 \rightarrow r_1$, $l_2 \rightarrow r_2$, and all non-trivial subterms s of l_2 such that l_1 and s have a most general unifier μ:

Let $l_2 = c[s]$. If $\mu(c[r_1]) \stackrel{*}{\rightarrow} t$, $\mu(r_2) \stackrel{*}{\rightarrow} u$ for irreducible terms t and u, then $t = u$.

The term $\mu(l_2)$ is called the *superposition* of l_1 on s in l_2. The pair $\mu(c[r_1]) \sim \mu(r_2)$ is called a *critical pair* generated by $l_1 \rightarrow r_1$ and $l_2 \rightarrow r_2$. (A term is said to be non-trivial if it is not a variable.)

Let $(\mathfrak{T}(F,V),\prec)$ be a well-founded set such that $\preceq$ is stable. If it is decidable whether $t_1 \preceq t_2$ for any two terms t_1 and t_2, the superposition theorem, together with Corollary 3.3, suggests that there is an algorithm (possibly non-terminating and possibly unsuccessful) for constructing a terminating and confluent TRS that solves the equality problem of $T(E)$ for a given axiom system E.

Knuth-Bendix Algorithm [Knuth 70]

Step 0: Set E to be the initally given set of equations. Set R to be empty. Go to *Step 1*.

Step 1: If E is empty, the current value of R is the desired TRS. Otherwise, go to *Step 2*.

Step 2: Remove a pair $t \sim u$ from E, and find irreducible terms t_1 and u_1 such that $t \stackrel{*}{\rightarrow} t_1$, $u \stackrel{*}{\rightarrow} u_1$ with respect to R. If $t_1 = u_1$, go to *Step 1*. If $t_1 \prec u_1$ or $t_1 \succ u_1$, go to *Step 3*. Otherwise, stop; the procedure is unsuccessful.

Step 3: We can assume $t_1 \succ u_1$ without loss of generality. Remove all the rewrite rules $l \rightarrow r$ from R such that either l or r is reducible by the rewrite rule $t_1 \rightarrow u_1$, and append $l \sim r$ to E instead. Append the new rule $t_1 \rightarrow u_1$ to R. Construct all the critical pairs generated by each two rules in R and append them to E. Go to *Step 1*.

It is easy to verify that R is always a terminating TRS, and that $T(E \bigcup R)$ is invariant for each step in the above algorithm. Moreover, if the algorithm completes successfully, the resulting TRS R satisfies the condition stated in the superposition theorem. Therefore, R is confluent and $T(E)=T(R)$ for the initially given E and the resulting R. Thus, the partial correctness of the algorithm follows from the decidability theorem.

Definition 3.5

Rewrite rules $l_1 \rightarrow r_1$ and $l_2 \rightarrow r_2$ are said to be *mutually irreducible* if neither l_1 nor r_1 can be rewritten by $l_2 \rightarrow r_2$ and neither l_2 nor r_2 can be rewritten by $l_1 \rightarrow r_1$. A TRS is said to be irreducible if any two distinct rules are mutually irreducible.

Let R be a confluent and terminating TRS and the left-hand side l of a rewrite rule $l \to r$ in R be rewritable by another rule in R. Consider the TRS S to have been obtained by removing the rule $l \to r$ from R. Then, S satisfies the following conditions:

(1) S is terminating

(2) R traces S

(3) An irreducible term in S is also irreducible with respect to R

Therefore, S is confluent and strongly equivalent to R by Corollary 2.12.

If the right-hand side r is rewritable, consider the TRS S to have been obtained by replacing the rule $l \to r$ with $l \to \mathcal{N}(r)$. Then, S again satisfies the above three conditions and, therefore, is confluent and strongly equivalent to R.

Thus, by removing the reducible rules one after another, any confluent and terminating TRS can be transformed into its strong equivalent which is, moreover, irreducible. Therefore, in order to search for a confluent and terminating TRS, we can restrict the search for an irreducible one. In fact, the Knuth-Bendix algorithm stated above works in such a way that the rewrite rules are mutually irreducible and, hence, the resulting TRS is always irreducible.

4. Well-founded Orders on $\mathfrak{T}(F, V)$

As shown in the previous section, the key point of the Knuth-Bendix algorithm is the existence of a stable and well-founded ordering of $\mathfrak{T}(F, V)$. We will discuss here a sufficient condition for a stably ordered set $(\mathfrak{T}(F, V), \preceq)$ to be well-founded.

Definition 4.1

A relation ρ on $\mathfrak{T}(F, V)$ is said to have the *subterm property* if $s \rho t$ for any term t and any subterm s of t.

Theorem 4.2

An ordered set $(\mathfrak{T}(F), \preceq)$ such that $\preceq$ has the replacement property and the subterm property is semi-well-ordered.

As shown by Dershowitz [Dershowitz 82], this theorem is easily obtained as a special case of Kruskal's tree theorem [Kruskal 60]. However, we prove this theorem directly using the technique introduced by Nash-Williams for his shorter proof of the tree theorem [Nash-Williams 63]. First, we show the following lemma.

Lemma 4.3

Let $(X, \preceq)$ be an ordered set. An infinite sequence $x_1, x_2, \cdots$ of elements of X is said to be nowhere-ascending if there are no i and j such that $x_i \preceq x_j$ and $i < j$. If a sequence $x_1, x_2, \cdots$ does not contain nowhere-ascending subsequences, it contains an ascending subsequence $x_{k_1} \preceq x_{k_2} \preceq \cdots$.

Proof :

Assume that a sequence $x_1, x_2, \cdots$ is given. Let us call an index i non-ascending if there is no j such that $x_i \preceq x_j$ and $i < j$. If a sequence contains infinitely many non-ascending indexes, the subsequence indexed by all and only non-ascending indexes clearly forms a nowhere-ascending subsequence. Therefore, the sequence $x_1, x_2, \cdots$ can have only a finite number of non-ascending indexes. Select an index k_1 to be larger than any non-ascending index. Since k_1 is ascending (not non-ascending), select an index k_2 such that $x_{k_1} \preceq x_{k_2}$ and $k_1 < k_2$. Since k_2 is ascending again, by repeating this process we can obtain an ascending subsequence $x_{k_1} \preceq x_{k_2} \preceq x_{k_3} \preceq \cdots$.

Lemma 4.3 is very useful. For example, it is obvious from the lemma that an ordered set is semi-well-ordered if and only if every infinite sequence contains an infinite ascending subsequence, and therefore, the Cartesian product of a finite number of semi-well-ordered sets is a semi-well-ordered set.

Proof of Theorem 4.2 :

Assume that there is a nowhere-ascending sequence. Select a term s_1 such that s_1 is the first term of a nowhere-ascending sequence and no proper subterm of s_1 can be such a term. Then, select a term s_2 such that s_1 and s_2 are the first two terms of a nowhere-ascending sequence and no proper subterm of s_2 can be such a term. Proceeding in this way, we can obtain a nowhere-ascending sequence $s_1, s_2, \cdots$.

Since F is finite, there exists an f in F and a subsequence $s_{k_1}, s_{k_2}, \cdots$ such that all s_{k_i} have the form $f(s^1_{k_i}, \cdots, s^n_{k_i})$. If the sequence $s^1_{k_1}, s^1_{k_2}, \cdots$ has a nowhere-ascending subsequence $s^1_{m_1}, s^1_{m_2}, \cdots$, then

$$s_1, s_2, \cdots, s_{m_1-1}, s^1_{m_1}, s^1_{m_2}, \cdots$$

forms a nowhere-ascending sequence, since, if $s_i \preceq s^1_{m_j}$, then $s_i \preceq s_{m_j}$ by the subterm property. This, however, contradicts the definition of s_{m_j}. Therefore, $s^1_{k_1}, s^1_{k_2}, \cdots$ cannot contain nowhere-ascending subsequences and, from Lemma 4.3, must contain an ascending subsequence $s^1_{p_1} \preceq s^1_{p_2} \preceq \cdots$.

Let us now consider the sequence $s^2_{p_1}, s^2_{p_2}, \cdots$. The method discussed above again constructs an ascending subsequence $s^2_{q_1} \preceq s^2_{q_2} \preceq \cdots$. Repeating this process n times, we finally arrive at a subsequence $s_{r_1}, s_{r_2}, \cdots$, such that $s^j_{r_1} \preceq s^j_{r_2} \preceq \cdots$ for all $j=1, \cdots, n$. Since $\preceq$ has the replacement property, it follows that $s_{r_1} \preceq s_{r_2} \preceq \cdots$. This, however, is inconsistent with the sequence $s_1, s_2, \cdots$ being nowhere-ascending.

Theorem 4.4

Let $(\mathfrak{T}(F), \preceq)$ be an ordered set such that $\preceq$ has the replacement property. Then, $(\mathfrak{T}(F), \preceq)$ is well-ordered if and only if $\preceq$ has the subterm property.

Proof :

The if-part of the theorem is a special case of Theorem 4.2. Here we prove the only-if-part. Assume that there exists a context $c[\Omega]$ and a term s such that $s \not\preceq c[\Omega]$. Since $\preceq$ is a total ordering, $c[\Omega] \prec s$. Let $t_1 = s$, $t_{i+1} = c[t_i]$. Then, the replacement property shows that the sequence $t_1 \succ t_2 \succ t_3 \succ \cdots$ is descending.

5. Lexicographic Subterm Ordering

In this section, we present a method of well-founded ordering based on the discussion in the previous section. For the purpose of obtaining a terminating and confluent TRS, at *Step 2* in the Knuth-Bendix algorithm, the two terms t_1 and u_1 are desired to be comparable. Therefore, the stronger ordering is considered to be better for the above purpose.

On the other hand, Theorem 4.4 says that a stable ordering of $\mathfrak{T}(F,V)$, which is strong enough to become total when restricted to $\mathfrak{T}(F)$, is well-founded if and only if it has the subterm property. Thus, possesion of the subterm property is a good criterion for well-foundedness. In fact, the ordering method introduced by Knuth and Bendix [Knuth 70] defines a stable and total ordering of $\mathfrak{T}(F)$ with the subterm property. Their ordering, however, is predicated on the somewhat arbitrary concept of the "weight" of function symbols.

Various methods for proving that an ordered set of terms is well-founded or that a TRS terminates have been suggested in recent years. Among these, the recursive path ordering [Dershowitz 81] is one of the best, but it is not total on $\mathfrak{T}(F)$.

We here define a stable ordering on $\mathfrak{T}(F,V)$ that is total on $\mathfrak{T}(F)$ and stronger than the recursive path ordering, without assigning "weights" to function symbols.

Definition (lexicographic subterm ordering)

Let F be a finite set of function symbols, and $(F, \leq)$ be a totally ordered set. The lexicographic subterm ordering $\preceq$ of $\mathfrak{T}(F,V)$ is then defined recursively as follows:

(1) For a trivial term (i.e., a variable) v, there are no terms t such that $t \prec v$.

(2) For a non-trivial term $t = g(t_1, \cdots, t_n)$ and a term s, $s \prec t$ if and only if

(2-1) there exists j such that $s \preceq t_j$ or

(2-2) $s = f(s_1, \cdots, s_m)$ and $s_i \prec t$ for all i and

(2-2-1) $f < g$ or

(2-2-2) $f = g$ and there exist i such that $s_1 = t_1, \cdots, s_{i-1} = t_{i-1}$ and $s_i \prec t_i$.

For the reader's reference, we define recursive path ordering below. It was originally defined for the ground terms constructed from possibly infinite function symbols without fixed arity of arguments [Dershowitz 81]. However, for the purpose of applying it to the Knuth-Bendix algorithm, it is enough to consider only a finite set of function symbols with fixed arity. We will, therefore, modify it for the terms constructed from such a set of function symbols.

Definition (recursive path ordering)

Let F be a finite set of function symbols totally ordered as above. The recursive path ordering $\preceq\preceq$ on $\mathfrak{T}(F,V)$ is defined recursively as follows:

(1) For a trivial term (i.e., a variable) v, there are no terms t such that $t \prec\prec v$.

(2) For a non-trivial term $t = g(t_1, \cdots, t_n)$ and a term s, $s \preceq\preceq t$ if and only if

(2-1) there exists j such that $s \preceq\preceq t_j$ or

(2-2) $s = f(s_1, \cdots, s_m)$ and

(2-2-1) $f < g$ and $s_i \prec\prec t$ for all i or

(2-2-2) $f = g$ and $s_i \preceq\preceq t_i$ for all i.

It is easy to verify by induction that lexicographic subterm ordering is stronger than recursive path ordering.

We hope to show that lexicographic subterm ordering possesses the desired properties discussed in the previous section. The proofs mainly involve induction on the construction of terms; the symbol ** is used to represent the induction hypothesis.

Lemma 5.1

Let $\preceq$ represent lexicographic subterm ordering. Let $s = f(s_1, \cdots, s_m)$ and t be terms such that $s \preceq t$. Then, $s_i \prec t$ for all i.

Proof:

If $s = t$, then $s_i \prec t$ is straightforward by definition. If $s \prec t$, then, since t cannot be a variable, let $t = g(t_1, \cdots, t_n)$. In the case of (2-1), $s_i \prec t_j$ from ** and $s_i \prec t$ for all i by definition. In the case of (2-2), $s_i \prec t$ is itself the condition of the case.

Theorem 5.2

Let $\preceq$ represent lexicographic subterm ordering. Then, $(\mathfrak{T}(F,V), \preceq)$ is an ordered set.

Proof:

We show that if $s \prec t$, and $t \prec u$, then $s \prec u$ and $s \neq u$. Since neither t nor u can be a variable, let $t = g(t_1, \cdots, t_n)$ and $u = h(u_1, \cdots, u_p)$. In the case that $s \preceq t_j$ or $t \preceq u_k$, it is easy to show that $s \prec u$ by definition, Lemma 5.1 and **. Moreover, if $s = u$, from Lemma 5.1, it follows that $t_j \prec s$ and $u_k \prec t$, and contradiction follows from **. Let us consider the case of (2-2), both for $s \prec t$ and $t \prec u$. Let $s = f(s_1, \cdots, s_m)$, $s_i \prec t$ for all i and $t_j \prec u$ for all j. From **, it follows that $s_i \prec u$ for all i. If $f < g$ or $g < h$, then

$f<h$ and, therefore, $s \prec u$ by definition and $s \not\approx u$. Assume that $f=g=h$ and there exist i and j such that

$s_1=t_1, \cdots, s_{i-1}=t_{i-1}, s_i \prec t_i$, and

$t_1=u_1, \cdots, t_{j-1}=u_{j-1}, t_j \prec u_j$.

If we let k be the minimum of i and j, then

$s_1=u_1, \cdots, s_{k-1}=u_{k-1}, s_k \prec u_k$,

and $s_k \not\approx t_k$ from $**$. Thus, we conclude that $s \prec u$ and $s \not\approx u$, for any case.

Theorem 5.3

Let $\preceq$ represent lexicographic subterm ordering. Then, $\preceq$ is stable and has the subterm property.

Proof :

We show here only that $\preceq$ has the subterm property. That it has the other two properties, namely, the substitution property and the replacement property, is also proved easily by induction. If $c[\Omega]=\Omega$, then $s \preceq s=c[s]$. Let $c[\Omega]=f(\cdots, d[\Omega], \cdots)$ for a context $d[s]$. Since $c[s]=f(\cdots, d[s], \cdots)$, $**$ and the condition (2-1) of the definition assures that $s \prec c[s]$.

Theorem 5.4

Let $\preceq$ represent lexicographic subterm ordering. Then, $(\mathfrak{T}(F), \preceq)$ is a totally ordered set.

Proof :

Let s and t be in $\mathfrak{T}(F)$. We will prove that if $s \not\approx t$, then $s \prec t$, or $t \prec s$. Let $s=f(s_1, \cdots, s_m)$ and $t=g(t_1, \cdots, t_n)$. If there exists t_j such that $s \preceq t_j$, or s_i such that $t \preceq s_i$, then $s \prec t$, or $t \prec s$, respectively. Therefore we can assume that $s \not\preceq t_j$ for all j and $t \not\preceq s_i$ for all i. From $**$, it follows that $s \succ t_j$, and $t \succ s_i$ for all i and j. Therefore, if $f<g$, or $g>f$, then $s \prec t$, or $t \prec s$, respectively. Let us assume $f=g$. Since $s \not\approx t$, there exists the least i such that $s_i \not\approx t_i$. Then $s_i \prec t_i$, or $s_i \succ t_i$ from $**$. Therefore, $s \prec t$, or $s \succ t$, respectively, by (2-2-2) of the definition.

6. Applications

In this section we report the results of some computer experiments. Here, we represent variables by character strings begining with an upper case letter, and function symbols by those begining with a lower case letter. We also sometimes use infix notation, such as $A+B$, in place of prefix notation, such as $+(A, B)$.

The Knuth-Bendix algorithm using lexicographic subterm ordering was programed in Prolog on a DEC 2060 computer. In the implementation, we adopted the following additional strategies:

(1) Choice of Minimal Term

At *Step 2* of the Knuth-Bendix algorithm, choose a pair t $\sim$u to minimize $max(\|t_1\|, \|u_1\|)$, where the notation $\|t\|$ represents the number of function symbols in t or, in other words, the number of non-trivial subterms of t.

There are three reasons for the above strategy. The first is, roughly speaking, that the fewer a term's function symbols, the more general it is and the stronger its rewriting power. Therefore, we can expect that the algorithm will construct a more efficient TRS (in other words, one with fewer rewrite rules) in a shorter time. The second reason is that the fewer a term's non-trivial subterms, the fewer the critical pairs that will be generated. Thus, at *Step 3*, we can mitigate the explosion of pairs in E. The last reason is that the strategy is fair [Huet 81] in the sense that it will ultimately choose any pair in E if the pair remains in E without being reduced, because there exist only a finite number of terms having fewer function symbols than a given number.

(2) Generation of a New Function Symbol

According to the original algorithm, if neither $t_1 \prec u_1$ nor $t_1 \succ u_1$ at *Step 2*, we cannot go any farther. Let $V_1, \cdots, V_n$ represent all the common variables in t_1 and u_1. We modify the algorithm such that, in this case, it generates a new function symbol $\mathcal{F}$ and appends the new pairs $t_1 \sim \mathcal{F}(V_1, \cdots, V_n)$ and $u_1 \sim \mathcal{F}(V_1, \cdots, V_n)$ in E instead of $t_1 \sim u_1$.

For example, let $t_1 = f(A, f(B, C))$ and $u_1 = f(A, f(B, D))$. Then, neither $t_1 \prec u_1$ nor $t_1 \succ u_1$ are true. However, it is clear that $f(A, f(B, C))$ is really a binary function in spite of its appearance, and it is very natural to introduce a new function symbol $\mathcal{F}$ for expressing both t_1 and u_1 as $\mathcal{F}(A, B)$.

Example 6.1

The first example is taken from Knuth and Bendix [Knuth 70]. The program was given the three axioms of group theory:

(1) $0+A=A$
(2) $(-A)+A=0$
(3) $(A+B)+C=A+(B+C)$

The ordering on the function symbols was given as $0<+<-$. The program stopped after the following output:

1: $0+A=A \Leftarrow 0$
2: $(-A)+A=0 \Leftarrow 0$
3: $(A+B)+C=A+(B+C) \Leftarrow 0$
4: $(-A)+(A+B)=B \Leftarrow 2/3$
5: $(-0)+A=A \Leftarrow 1/4$
6: $(-(-A))+B=A+B \Leftarrow 4/4$
7: $A+0=A \Leftarrow 2/4$
8: $-(-A)=A \Leftarrow 7/6$
delete 6
9: $A+(-A)=0 \Leftarrow 8/2$
10: $-0=0 \Leftarrow 9/1$
delete 5
11: $A+((-A)+B)=B \Leftarrow 9/3$
12: $A+(B+(-(A+B)))=0 \Leftarrow 9/3$
13: $A+(-(B+A))=-B \Leftarrow 12/4$
delete 12
14: $(-(A+B))+A=-B \Leftarrow 13/13$
15: $-((-A)+B)=(-B)+A \Leftarrow 11/14$
16: $-(A+B)=(-B)+(-A) \Leftarrow 8/15$
delete 15
delete 14
delete 13

1: $0+A=A$
2: $(-A)+A=0$
3: $(A+B)+C=A+(B+C)$
4: $(-A)+(A+B)=B$
7: $A+0=A$
8: $-(-A)=A$
9: $A+(-A)=0$
10: $-0=0$
11: $A+((-A)+B)=B$
16: $-(A+B)=(-B)+(-A)$

Each equation in the output should be interpreted from left to right as a rewrite rule. The symbol $\Leftarrow 0$ means that the equation was obtained from the given axiom set. The symbol $\Leftarrow n/m$ means that the equation was obtained from a critical pair generated by the previous rules n and m. The line "delete n" shows that the rule n was removed at *Step 3* because the left or the right side of the rule was reducible by the newly obtained rule. The set of equations under the horizontal line is the final TRS, which is terminating, confluent, and irreducible.

Though the resulting set of rewrite rules is the same as Knuth and Bendix's, it seems that the strategy of choice of minimal term is efficient, because the algorithm generated only six superfluous rules, many fewer than the ten reported by Knuth and Bendix.

Example 6.2

This example is the same as Knuth and Bendix's Example 3. We used a symmetric axiom system having right identity and right inverse, namely:

(1) $A+0=A$
(2) $A+(-A)=0$
(3) $(A+B)+C=A+(B+C)$

The program obtained the same set of rules as in Example 6.1 after outputing a list of rules, including nine superfluous rules, again, fewer than the 14 superfluous rules obtained by Knuth and Bendix.

Example 6.3

This example is the same as Knuth and Bendix's Example 11. Group theory can be defined with weaker axioms than the axioms given in Example 6.2. Besides the associative law, we postulate the existence of an idempotent element 0. Furthermore, each element has as least one right inverse with respect to 0. Finally, we postulate that each element has at most one left inverse with respect to 0. Knuth and Bendix axiomatized these conditions as follows:

(1) $(A+B)+C=A+(B+C)$
(2) $0+0=0$
(3) $A+(-A)=0$
(4) $f(0,A,B)=A$
(5) $f(A+B,A,B)=g(A+B,B)$

As noted in their paper, a binary function $f(A,B)$ could have been used in place of $f(A,B,C)$. A ternary operator was used because the terms $f(A+B,A)$ and $g(A+B,B)$ were not comparable with respect to their ordering. In lexicographic subterm ordering, however, we do not encounter this difficulty. We gave the program the following axioms instead of (4) and (5), and specified the ordering as $0<+<-<g<f$.

(4a) $f(0,A)=A$
(5a) $f(A+B,A)=g(A+B,B)$

The program terminated after finding the following 12 rules, including the 10 rules found in Examples 6.1 and 6.2.

2: $A+(-A)=0$
3: $(A+B)+C=A+(B+C)$
14: $-0=0$
17: $0+A=A$
20: $A+((-A)+B)=B$
21: $(-A)+(A+B)=B$
24: $(-A)+A=0$
25: $A+0=A$
26: $-(-A)=A$
27: $g(0,A)=-A$
30: $f(A,B)=g(A,(-B)+A)$
33: $-(A+B)=(-B)+(-A)$

As shown by Knuth and Bendix, if axioms (1) through (5) had been given, the computation would have continued to generate new rules forever after the 10 rules had been derived.

Example 6.4

This is the same as Knuth and Bendix's Example 12. The axioms of (l,r)-systems were given together with the ordering specified as $0<+<-$.

(1) $(A+B)+C=A+(B+C)$
(2) $0+A=A$
(3) $A+(-A)=0$

The program output is as follows:

1: $0+A=A \Leftarrow 0$
2: $A+(-A)=0 \Leftarrow 0$
3: $-0=0 \Leftarrow 2/1$
4: $(A+B)+C=A+(B+C) \Leftarrow 0$
5: $A+((-A)+B)=B \Leftarrow 2/4$
6: $-(-A)=A+0 \Leftarrow 2/5$
7: $(-A)+(A+B)=B \Leftarrow 6/5$
8: $(-A)+0=-A \Leftarrow 2/7$
9: $-(A+0)=-A \Leftarrow 6/6$
10: $A+(B+(-(A+B)))=0 \Leftarrow 4/2$
11: $-((-A)+A)=0 \Leftarrow 10/7$
12: $A+(-(B+A))=-B \Leftarrow 10/7$
delete 10

13: $-(A+B)=(-B)+(-A)$ ⇐12/7
delete 12
delete 11
delete 9

1: $0+A=A$
2: $A+(-A)=0$
3: $-0=0$
4: $(A+B)+C=A+(B+C)$
5: $A+((-A)+B)=B$
6: $-(-A)=A+0$
7: $(-A)+(A+B)=B$
8: $(-A)+0=-A$
13: $-(A+B)=(-B)+(-A)$

This computation contains only four superfluous rules, and the resulting set consists of nine rules; some of which are different from Knuth and Bendix's. That is, their rule, $A+0\rightarrow-(-A)$, was replaced with rule 6, which has a different orientation, and therefore, the rule $-(-(-A)))\rightarrow-A$ was replaced with rule 8, and the rule $-(-(A))+B\rightarrow A+B$ became unnecessary. The other seven rules are the same.

We also experimented with Knuth and Bendix's Examples 13 and 14 and obtained a set of 12 rules for each. Comparing our set with Knuth and Bendix's for each example, we again found they have rules with different orientations. Knuth and Bendix's set in Example 14 consisted of 21 rules. This shows that the orientation of rewrite rules considerably affects the number of rules.

Example 6.5

This is the same as Knuth and Bendix's Example 16, in which they define two unary functions, l and r, as follows:

(1) $(A+A)+A=l(A)$
(2) $A+(A+A)=r(A)$.

These were given to the program together with the basic axiom

(3) $(A+B)+(B+C)=B$,

which defines a central groupoid, together with the further axiom,

(4) $r(A)+B=A+B$

in order to determine whether these axioms would define a "natural" central groupoid. The resulting rules were

$$l(l(A))\rightarrow l(A), \quad l(r(A))\rightarrow l(A),$$
$$r(l(A))\rightarrow r(A), \quad r(r(A))\rightarrow r(A),$$
$$l(A+B)\rightarrow r(A), \quad r(A+B)\rightarrow l(A),$$
$$A+(B+C)\rightarrow A+r(B),$$
$$(A+B)+C\rightarrow l(B)+C,$$
$$r(A)+B\rightarrow A+B, \quad A+l(B)\rightarrow A+B,$$
$$A+r(A)\rightarrow r(A), \quad l(A)+A\rightarrow l(A),$$
$$l(A)+r(A)\rightarrow A.$$

We experimented with the set consisting of the basic axiom (3) and the axiom

(5) $(A+(A+A))+B=A+B$,

which is equivalent to (4), without using definitions (1) and (2). After generating the first four rules, the program stopped with the following message:

5: $\mathcal{F}(A)=(A+(A+B))\not\gtrless(A+(A+A))$ ⇐4/2

Since the newly introduced function $\mathcal{F}$ was the same as r, we let the program continue. After a short while, it stopped again. To our surprise, the message was

17: $\mathcal{G}(A,B)=((C+A)+B)\not\gtrless((D+A)+B)$ ⇐16/13

The function $\mathcal{G}$ was not equivalent to the unary function l, but a binary operator. Nevertheless we let the program continue. It then completed the computation without any more stops and the resulting set consisted of the following seven rules:

8: $\mathcal{F}(\mathcal{F}(A))=\mathcal{F}(A)$
19: $\mathcal{G}(A,\mathcal{F}(A))=A$
21: $\mathcal{G}(A,\mathcal{G}(B,C))=\mathcal{G}(A,B)$
22: $\mathcal{G}(\mathcal{G}(A,B),C)=\mathcal{G}(A,C)$
23: $A+B=\mathcal{G}(\mathcal{F}(A),B)$
24: $\mathcal{G}(\mathcal{F}(A),\mathcal{F}(A))=\mathcal{F}(A)$

29: $\mathcal{F}(\mathcal{G}(A,B))=\mathcal{G}(B,B)$

It can easily be shown that the two sets of rules are equivalent to each other by defining:

$l(A)=\mathcal{G}(A,A), r(A)=\mathcal{F}(A)$ or

$\mathcal{F}(A)=r(A), \mathcal{G}(A,B)=l(A)+B$.

In fact, examining either of the above two sets, we find that the free system on n generators has 4^n elements.

ACKNOWLEDGEMENTS

The author wishes to express his thanks to K. Fuchi, Director, and T. Yokoi, Third Laboratory Chief, of the Institute for New Generation Computer Technology (ICOT) for providing the opportunity to conduct this research. Thanks are also due to T. Kurokawa of ICOT, Y. Toyama of Nippon Telegraph and Telephone Public Corporation and T. Matsuda of Science University of Tokyo for their helpful suggestions.

REFERENCES

[Dershowitz 82] Dershowitz, N.: *Orderings for term-rewriting systems*, Theoretical Computer Science 17 (1982) 279-301.

[Huet 81] Huet, G.: *A complete proof of correctness of the Knuth-Bendix completion algorithm*, J. of Computer and System Science 23 (1981) 11-21.

[Knuth 70] Knuth, D. E., Bendix, P. B.: *Simple word problems in universal algebras*, Computational problems in abstract algebra, J. Leech (ed), Pergamon Press, Oxford, (1970) 263-297. also in: Automated Reasoning 2 (Siekmann and Wrightson eds.), Springer (1983)

[Kruskal 60] Kruskal, J. B.: *Well-quasi-ordering, the tree theorem, and Vazsonyi's conjecture*, Trans. Amer. Math. Soc. 95 (1960) 210-225.

[Nash-Williams 63] Nash-Williams, C. St. J. A.: On well-quasi-ordering finite trees, Proc. Cambridge Philos. Soc. 59 (1963) 833-835.

[Robinson 65] Robinson, J. A.: *A machine-oriented logic based on the resolution principle*, J. ACM 12,(1965) 23-41.