Well rewrite orderings and well quasi-orderings

Pierre Lescanne

UNITE DE RECHERCHE
INRIA-LORRAINE

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P. 105
78153 Le Chesnay Cedex
France
Tél. (1) 39 63 55 11

Rapports de Recherche

N° 1385

Programme 2
Calcul symbolique, Programmation et Génie logiciel

WELL REWRITE ORDERINGS AND WELL QUASI-ORDERINGS

Pierre LESCANNE

Janvier 1991

RR_1385

Beaux ordres de réécriture et beaux préordres
Well rewrite orderings and well quasi-orderings

Pierre LESCANNE

Résumé

Cet article étudie de beaux préordres décrits comme des ordres de réécriture et propose une nouvelle famille de beaux préodres qui étend le plongement ou l'ordre de divilibilité de G. Higman. Par exemple, un beau préodre comme ceux proposés ici peut contenir des paires de la forme $f(f(x)) > f(g(f(x)))$. On donne des conditions pour lesquelles la clôture transitive d'une relation de réécriture noethérienne est un beau préordre. Enfin, une tentative d'étendre l'ordre récursif sur les chemin est entreprise.

Abstract

This paper studies well (quasi) orderings described as rewrite orderings and proposes a new family of well (quasi) orderings that extends the embedding or divisibility order of G. Higman. For instance, the well (quasi) orderings proposed in this paper may contain pairs of the form $f(f(x)) > f(g(f(x)))$. Conditions under which the transitive closure of a well-founded rewrite relation is a well-quasi-rodering are given. Finally, an attempt to extend the recursive path ordering is proposed.

Well rewrite orderings and well quasi-orderings*

Pierre LESCANNE
Centre de Recherche en Informatique de Nancy (CNRS)
and INRIA-Lorraine
Domaine Scientifique Victor Grignard, BP 239,
54506 Vandœuvre-les-Nancy, France
email: lescanne@loria.crin.fr

Abstract

This paper studies well (quasi) orderings described as rewrite orderings and proposes a new family of well (quasi) orderings that extends the embedding or divisibility order of G. Higman. For instance, the well (quasi) orderings proposed in this paper may contain pairs of the form $f(f(x)) > f(g(f(x)))$. Conditions under which the transitive closure of a well-founded rewrite relation is a well-quasi-ordering are given. Finally, an attempt to extend the recursive path ordering is proposed.

Introduction and notations

In this paper, we are interested in properties of well (quasi) orderings used in proving termination of rewrite systems, we especially propose an extension of Higman's theorem [7]. Therefore we consider the set $\mathcal{T}(\mathcal{F})$ of terms on a set $\mathcal{F}$ of symbols, that are terms where each operator $f \in \mathcal{F}$ has a fixed arity. Therefore we do not consider the set of varyadic terms as in Kruskal's theorem [11, 18].

Most notations are borrowed from [4]. $\ominus$ is the set difference and the relation difference. A (rewrite) *rule* is an oriented pair of terms. $\Sigma(E)$ is the set of substitutions with range E, i.e. the mapping from $\mathcal{X}$ to E and we write $t\sigma$ the result of applying the substitution σ to t. Therefore, $\Sigma(\mathcal{T}(\mathcal{F}))$ (resp. $\Sigma(\mathcal{T}(\mathcal{F},\mathcal{X}))$) is the set of substitutions with range $\mathcal{T}(\mathcal{F})$(resp. $\mathcal{T}(\mathcal{F},\mathcal{X})$). We say that a relation $\vdash$ satisfies the *replacement property* if

$$s \vdash t \Rightarrow f(\ldots,s,\ldots) \vdash f(\ldots,t,\ldots).$$

A relation $\vdash$ is *invariant* for a set Σ of substitutions if

$$(\forall \sigma \in \Sigma)\ s \vdash t \Rightarrow s\sigma \vdash t\sigma.$$

A relation is *fully invariant* if it is invariant for $\Sigma(\mathcal{T}(\mathcal{F}))$. A fully invariant relation which has the replacement property is called a *rewrite relation*, it is written $\rightarrow_R$ if it is generated by a set R of rewrite rules. If in addition it is transitive and reflexive it is called a *derivability* relation and written $\xrightarrow{*}_R$. If the derivability relation is well-founded it is called a *rewrite ordering*. We also consider two classical orderings on terms the *subsumption ordering* $\succeq$ and the *encompassment ordering* $\trianglerighteq$, with the following definitions: a term s is subsumed by a

*This research was sponsored by ESPRIT Basic Research Action COMPASS #3264

term t, written $s \succcurlyeq t$, if there is a substitution σ such that $s = t\sigma$ and a term s encompasses a term t, written $s \trianglerighteq t$, if t subsumes a subterm of s. The *componentwise extension* of an ordering $\succeq$ on E is the ordering $\succeq^{comp}$ defined on sequences of elements as follows. If $(s_1, \ldots, s_n)$ and $(t_1, \ldots, t_n)$ are two sequences of same length, then $(s_1, \ldots, s_n) \succeq^{comp} (t_1, \ldots, t_n)$ if, for all $i \in [1..n]$, $s_i \succeq t_i$.

Section 1 introduces the main concepts and their use in computer science. Section 2 presents known well quasi-orderings as well rewrite orderings. Section 3 gives the main theorem of this paper. Section 4 proposes direct definition of well rewrite orderings and of the main result for readers less familiar with rewrite systems terminology. Section 5 contains several examples of well rewrite orderings. Section 6 is an extension of well rewrite orderings to a king of recursive path ordering. A short version of this paper was presented at the fifth symposium on Logic in Computer Science [15].

1 The role of well quasi orderings

The paper where the concept of well quasi-ordering is probably presented for the first time is due to Janet [8] (see also [13]) in 1920 (see [3, 6, 10] for other informations). It is surprising to notice that this paper proposes also the ancestor of Gröbner bases and that the author already identifies well quasi-orderings as important tools for the termination of the algorithms he presents. Here, we are interested in an aspect of well quasi-orderings useful in completion procedures for rewrite systems, namely incrementality.

Incrementality

Actually, well quasi-orderings are important not only for theoretical reason in computer science, but because they carry solutions for building friendly completion procedures for rewrite systems. Indeed the concept of incrementality of orderings is then especially useful to avoid unnecessary failures. This concept used in REVE [5, 12] was described as a main feature of the decomposition ordering [9, 14]. A well-founded ordering is incremental if all its extensions are still well-founded. From a practical point of view, one wants to answer positively to the following question: can a well-founded ordering be extended (by adding pairs that are currently incomparable) and still remains well-founded? For this, one has to forbid the possibility of adding an infinite decreasing chain, therefore an incremental ordering has to satisfy the following properties:

1. it is well-founded,
2. there is no infinite set of pairwise uncomparable elements.

This properties are precisely the definition of well quasi-orderings. Thus we may state the following claim:

Incremental orderings
are
well quasi-orderings.

Alternate definitions of well quasi-orderings are already in Janet's paper. They are fully equivalent to the previous one and will be used freely in the sequel. They say that an ordering $\geq$ is a well quasi-ordering if and only if it satisfies one of the following conditions.

- Each infinite sequence $(t_n)_{n\in\mathrm{N}}$ contains a pair t_i, t_j such that $i > j$ and $t_i \geq t_j$.
- From each infinite sequence $(t_n)_{n\in\mathrm{N}}$, one can extract an increasing subsequence. In other words, there exists an increasing function $\varphi : \mathrm{N} \to \mathrm{N}$ such that $t_{\varphi(n+1)} \geq t_{\varphi(n)}$.

Incrementality and completion

Usually incremental (or well quasi) orderings are used as follows. If in the middle of a completion the procedure fails because an identity cannot be oriented into a rule, one tries to see whether the current ordering can be extended to allow the system orienting the identity; if this is possible, one knows that the termination of the already generated set of rules is preserved, since the ordering is incremental and the completion may continue as no failure would have happened. If the ordering is a recursive path ordering, the extension is usually done by extending the precedence or the status [5] and optimal suggestions are proposed to the users by the software. For instance, running a completion procedure on the axioms of groups:

$$\begin{array}{rcl} x * e & = & x \\ x * i(x) & = & e \\ (x * y) * z & = & x * (y * z) \end{array}$$

with a lexicographic path ordering based on a left-to-right status for $*$ and a precedence $* > e$ and $i > e$, generates the rewrite system:

$$\begin{array}{rcl} x * e & \rightarrow & x \\ e * x & \rightarrow & x \\ x * i(x) & \rightarrow & e \\ i(x) * x & \rightarrow & e \\ i(e) & \rightarrow & e \\ i(i(x)) & \rightarrow & x \\ (x * y) * z & \rightarrow & x * (y * z) \\ x * (i(x) * y) & \rightarrow & y \\ i(x) * (x * y) & \rightarrow & y \end{array}$$

and the critical pair

$$i(x * y) = i(y) * i(y)$$

that it cannot orient; however the system can continue if one increments the ordering by adding the pair $i > *$ in the precedence.

Therefore incremental (or well quasi) orderings play a really important role. However, currently the only incremental (or well quasi) orderings we can build are those that contain the embedding which is the rewrite transitive relation generated by the rules

$$\{f(x_1, ..., x_n) \rightarrow x_i \mid f \in \mathcal{F} \ \wedge \ 1 \leq i \leq n\}$$

This is a big drawback since such incremental (or well quasi) orderings will never be able to prove the termination of terminating rewrite systems like

$$f(f(x)) \rightarrow f(g(f(x)))$$

because the right-hand side is embedded in the left-hand side or, if one prefers, the right-hand side rewrites to the left-hand side with the above embedding relation. We exhibit a family of well quasi-orderings that includes the embedding as given above, but also includes rewrite relations that contains rules like $f(f(x)) \to f(g(f(x)))$. These relations can be described by sets of rewrite rules. Since these relations are both *well* quasi-orderings and *rewrite* orderings, we propose to call them *well rewrite orderings*.

2 Some well rewrite orderings

There exist many results on well quasi-orderings (see [10]). We present here only two well rewrite orderings through their underlying term rewrite systems.

2.1 Higman divisibility order

The divisibility order or embedding on $\mathcal{T}(\mathcal{F})$ is the rewrite relation $\xrightarrow{*}_{\mathcal{EMB}}$ where $\to_{\mathcal{EMB}}$ is given by the following term rewrite system $\mathcal{EMB}$

$$f(x_1, \ldots, x_n) \underset{\mathcal{EMB}}{\longrightarrow} x_i \text{ for all } i \in [1..n]$$

where the f's are all the operators in $\mathcal{F}$. In other words $\xrightarrow{*}_{\mathcal{EMB}}$ is the smallest derivability relation, generated by $\mathcal{EMB}$. Sometime a relation that contains $\mathcal{EMB}$ is said to satisfy the *subterm property*, since the relation we consider have also to satisfy the replacement property, we prefer to give the set $\mathcal{EMB}$ of rewrite rules. In this new framework, Higman's theorem asserts:

Theorem 1 ([7]) *The derivability relation* $\xrightarrow{*}_{\mathcal{EMB}}$ *is a well quasi-ordering.*

2.2 Embedding with patterns

It is possible to consider an extension of Higman's theorem based on patterns. In term of rules, one basically considers a set $\mathcal{P}$ of rules

$$p(x_1, \ldots, x_n) \underset{\mathcal{P}}{\to} x_i \text{ for all } i \in [1..n]$$

where each p is a linear term of $\mathcal{T}((\mathcal{F}, \mathcal{X}))$, i.e., a term where each variable occurs only once, also called *pattern*, and n is the number of variables occurring in this pattern. The set of patterns is written P. In order to make the derivability relation a well quasi-ordering, the set P has to be *unavoidable* which means that every ground term but a finite set A, must encompass a pattern in P; in other words

$$(\forall t \in \mathcal{T}(\mathcal{F}) \ominus A)\, (\exists p \in P)\, t \trianglerighteq p.$$

$F = \{f(x_1, \ldots, x_n) \,|\, f \in \mathcal{F}\}$ is trivially an unavoidable set of patterns. Notice that F is the set of patterns used in Higman's relation $\to_{\mathcal{EMB}}$. With general patterns, one has the following theorem:

Theorem 2 ([20]) *The relation* $\xrightarrow{*}_{\mathcal{P}}$ *is a well quasi-ordering if and only if* P *is unavoidable.*

3 An extension of Higman's theorem

In our theorems we need a property a little stronger than unavoidability. Indeed, when with an unavoidable set P, we require each term large enough to encompass a pattern in P, in a *basis* G we require every term large enough to match directly a unique pattern in G. Let us give a formal definition of a *basis*.

Definition 1 (Basis) *A* basis *G is a finite set of linear terms such that there exists a finite set A of ground terms and for every term t in $\mathcal{T}(\mathcal{F}) \ominus A$ there exists* a unique *term g in G that subsumes t,in other words*

$$(\forall t \in \mathcal{T}(\mathcal{F}) \ominus A)\,(\exists!\, g \in G)\; t \succeq g.$$

and for $t \in A$, there exists no term in G that subsumes t.

We now define what we call a *basic set of rules*.

Definition 2 (Basic sets of rewrite rules) *A set of rewrite rules is* basic *if the set of its left-hand sides is a basis.*

The uniqueness of top pattern for every term in the case of a basic set of rules is not a real restriction as shown by Corollary 1. It is important that the different left-hand form a basis, therefore the definition prevent any term even ground terms to match two different patterns on the top. For instance,

$$\begin{array}{rcl} f(x) & \rightarrow & g(x) \\ f(a) & \rightarrow & a \end{array}$$

is not a basic set of rewrite rules, since terms of the form $f(a)$ have two matches on the top. For reasons that will be made clear later on, we first define a relation generated by a set of rules which is not the derivability relation generated by R,

Definition 3 (Pseudo-Derivability) *Given a finite set R of rules, the R-*pseudo-derivability *is the relation $\overset{*}{\hookrightarrow}_R$ defined as follows.*

$$s \underset{R}{\overset{*}{\hookrightarrow}} t$$

if and only if, either

$$s = t$$

or

$$(\exists (g,d) \in R)\,(\exists \sigma \in \Sigma(\mathcal{F},\mathcal{X}))\; g\sigma = s \;\wedge\; d\sigma \underset{R}{\overset{*}{\hookrightarrow}} t \tag{1}$$

or

$$\begin{array}{c} (\exists (g,d) \in R)\,(\exists \sigma \in \Sigma(\mathcal{F},\mathcal{X}))(\exists \theta \in \Sigma(\mathcal{F},\mathcal{X})) \\ g\sigma = s \;\wedge\; g\theta = t \;\wedge\; (d_1\sigma \underset{R}{\overset{*}{\hookrightarrow}} d_1\theta, \ldots, d_{n_g}\sigma \underset{R}{\overset{*}{\hookrightarrow}} d_{n_g}\theta) \end{array} \tag{2}$$

where $\{g \rightarrow d_1, \ldots, g \rightarrow d_{n_g}\}$ are all the rules with left-hand side g.

As said in proposition 2, pseudo-derivability is under some conditions an extension of derivability. Condition (1) says that a term s pseudo-derivates to a term t if one can transform s into t by a head rewrite. Condition (2) says that s pseudo-derivates to t, if s and t have the same pattern g at the top and if for each rule $g \to d$, $d\sigma$ pseudo-derivates to $d\theta$. For instance, suppose

$$R = \{h(x,y) \to f(x); h(x,y) \to y\}$$

we can assert that $h(h(a,b),c)$ R-pseudo-derivates to $f(h(a,b))$ by (1). Furthermore, $h(s_1,s_2)$ R-pseudo-derivates to $h(t_1,t_2)$, if $f(s_1)$ R-pseudo-derivates to $f(t_1)$ and s_2 R-pseudo-derivates to t_2.

Proposition 1 (Transitivity of pseudo-derivability) *If R is basic then the pseudo-derivability $\overset{*}{\hookrightarrow}_R$ is transitive.*

Proof: Let s, t and u be three terms such that $s \overset{*}{\hookrightarrow}_R t$ and $t \overset{*}{\hookrightarrow}_R u$. The proof is done by induction on the number of uses of rules (1) and (2) and works as a "cut elimination". Consider the first rule invoked in the proof of $s \overset{*}{\hookrightarrow}_R t$ and $t \overset{*}{\hookrightarrow}_R u$. There is no problem if an equality step is used or if $s \overset{*}{\hookrightarrow}_R t$ uses (1) or if $s \overset{*}{\hookrightarrow}_R t$ and $t \overset{*}{\hookrightarrow}_R u$ use both (2). Now suppose $s = g\sigma \overset{*}{\hookrightarrow}_R t = g\theta$ uses (2) and $t = g\theta \overset{*}{\hookrightarrow}_R u$ uses (1) which means there exists a rule $g \to d_i \in R$ such that $d_i\theta \overset{*}{\hookrightarrow}_R u$, let us prove $s \overset{*}{\hookrightarrow}_R u$. Indeed by (2) for all $1 \leq j \leq n_g$, $d_j\sigma \overset{*}{\hookrightarrow}_R d_j\theta$, hence for the specific i above mentioned, $d_i\sigma \overset{*}{\hookrightarrow}_R d_i\theta$. Using now $d_i\theta \overset{*}{\hookrightarrow}_R u$ we can apply the induction and use transitivity to prove $d_i\sigma \overset{*}{\hookrightarrow}_R u$. Therefore, (1) can be applied to $d_i\sigma \overset{*}{\hookrightarrow}_R u$. □

Notice that the above proof uses the unicity of the pattern g, as it was required in the definition of a basic set of rules.

Theorem 3 (Well quasi-orderedness of pseudo-derivability) *Let R be a finite basic set of rewrite rules. If the R-pseudo-derivability is well-founded, then it is a well quasi-ordering.*

Proof: We write $\overset{+}{\hookrightarrow}_R$ the relation $\overset{*}{\hookrightarrow}_R \ominus \equiv$. The proof of the theorem is based on Nash-William's least counter-example method [16, 18] used in the proof of Kruskal's and Higman's theorems. In those proofs, "least" means "with respect to the size", here "least" is "with respect to $\overset{+}{\hookrightarrow}_R$". The proof works as follows.

Suppose the pseudo-derivability $\overset{*}{\hookrightarrow}_R$ is not a well quasi-ordering. Therefore there exists an infinite sequence

$$t_0, t_1, \ldots, t_n, \ldots$$

which is a least counter-example in the following sense. It contains no pair (t_i, t_j) with $i > j$ and $t_i \overset{*}{\hookrightarrow}_R t_j$ and there is no counter-example starting with s such that $t_0 \overset{+}{\hookrightarrow}_R s$ and more generally there exists no counter-example starting with $t_0, \ldots, t_n, s$ such that $t_{n+1} \overset{+}{\hookrightarrow}_R s$. This is possible since $\overset{+}{\hookrightarrow}_R$ is well-founded.

Since the system R is basic, there exists a left-hand side g of a rule in R and a infinite subsequence with top pattern g, which means there exists an increasing function $\varphi : \mathbf{N} \to \mathbf{N}$ and substitutions σ_n such that $g\sigma_n = t_{\varphi(n)}$.

Let $d_1 \ldots d_{n_g}$ be the right-hand sides of the rules in R with left-hand sides g. For $k \in [1..n_g]$ and $\chi : \mathbf{N} \to \mathbf{N}$ an increasing function, a sequence $(d_k\sigma_{\chi(n)})_{n\in\mathbf{N}}$ is not a counter-example. Otherwise suppose $(d_k\sigma_{\chi(n)})_{n\in\mathbf{N}}$ is a counter-example, therefore

$$t_0, \ldots, t_{\varphi\circ\chi(0)-1}, d_k\sigma_{\chi(0)}, d_k\sigma_{\chi(1)}, \ldots$$

is a counter-example less than $(t_n)_{n\in\mathbf{N}}$; it is less than $(t_n)_{n\in\mathbf{N}}$, since trivially $t_{\varphi\circ\chi(0)} = g\sigma_{\chi(0)} \overset{+}{\hookrightarrow}_R d_k\sigma_{\chi(0)}$ and it is a counter-example because one can have

- neither $t_i \overset{*}{\hookrightarrow}_R t_j$ for $\varphi \circ \chi(0) > i > j$
- nor $d_k\sigma_{\chi(m)} \overset{*}{\hookrightarrow}_R d_k\sigma_{\chi(p)}$ for $m > p \geq \varphi \circ \chi(0)$, since $(d_k\sigma_{\chi(n)})_{n\in\mathbf{N}}$ is a counter-example,
- nor $d_k\sigma_{\chi(m)} \overset{*}{\hookrightarrow}_R t_j$ for $m \geq 0$ and $\varphi \circ \chi(0) > j$, since this would imply $t_{\varphi\circ\chi(m)} = g\sigma_{\chi(m)} \overset{+}{\hookrightarrow}_R t_j$, which is impossible since $(t_n)_{n\in\mathbf{N}}$ is a counter-example.

Therefore, there exist n_g increasing functions $\psi_1, \ldots, \psi_{n_g}$ such that

$$d_1\sigma_{\psi_1(n+1)} \underset{R}{\overset{*}{\hookrightarrow}} d_1\sigma_{\psi_1(n)}$$

$$d_2\sigma_{\psi_2\circ\psi_1(n+1)} \underset{R}{\overset{*}{\hookrightarrow}} d_2\sigma_{\psi_2\circ\psi_1(n)}$$

$$\vdots$$

$$d_{n_g}\sigma_{\psi_{n_g}\circ\ldots\circ\psi_2\circ\psi_1(n+1)} \underset{R}{\overset{*}{\hookrightarrow}} d_{n_g}\sigma_{\psi_{n_g}\circ\ldots\circ\psi_2\circ\psi_1(n)}$$

Let us write $\psi = \psi_{n_g} \circ \ldots \circ \psi_1$. ψ satisfies

$$g\sigma_{\psi(n)} = t_{\varphi(\psi(n))}$$

and

$$(\forall i \in [1..n_g])\, d_i\sigma_{\psi(n+1)} \underset{R}{\overset{*}{\hookrightarrow}} d_i\sigma_{\psi(n)}$$

Using the last part of the definition of pseudo-derivability, one gets

$$t_{\varphi(\psi(n+1))} \underset{R}{\overset{*}{\hookrightarrow}} t_{\varphi(\psi(n))}$$

which is a contradiction with the fact that $(t_i)_{i\in\mathbf{N}}$ is a counter-example. □

In the previous theorem, the key condition is well-foundedness since it is rather easy to check that a set of rewrite rules is basic. To make the proof of well-foundedness easier it is often better to work on rewrite orderings instead of pseudo-derivabilities. To extend Theorem 3 to rewrite orderings, we have to require a specific property on the set of rewrite rules and the derivability relation that it generates. We call this property *projectivity*.

Definition 4 (Projectivity) *A set R of rewrite rules is* projective, *if for every left-hand side g of a rule in R and for every pair (σ, θ) of ground substitutions, one has*

$$\begin{array}{c}[d_1\sigma \overset{*}{\rightarrow}_R d_1\theta \wedge \ldots \wedge d_i\sigma \overset{*}{\rightarrow}_R d_i\theta \wedge \ldots \wedge d_{n_g}\sigma \overset{*}{\rightarrow}_R d_{n_g}\theta] \\ \Longrightarrow \\ g\sigma \overset{*}{\rightarrow}_R g\theta\end{array}$$

where $\{d_1, \ldots, d_{n_g}\}$ is the set of right-hand sides associated with g.

The relation $\rightarrow_{\mathcal{EMB}}$ that we called embedding is projective, indeed since the rules are of the form $f(x_1, \ldots, x_n) \rightarrow x_i$, projectivity means

$$[s_1 \longrightarrow^*_{\mathcal{EMB}} t_1 \wedge \ldots \wedge s_i \longrightarrow^*_{\mathcal{EMB}} t_i \wedge \ldots \wedge s_n \longrightarrow^*_{\mathcal{EMB}} t_n]$$
$$\Longrightarrow$$
$$f(s_1, \ldots, s_n) \longrightarrow^*_{\mathcal{EMB}} f(t_1, \ldots, t_n)$$

and is obvious. For the embedding with patterns projectivity means

$$[s_1 \rightarrow^*_{\mathcal{P}} t_1 \wedge \ldots \wedge s_i \rightarrow^*_{\mathcal{P}} t_i \wedge \ldots \wedge s_n \rightarrow^*_{\mathcal{P}} t_n]$$
$$\Longrightarrow$$
$$p(s_1, \ldots, s_n) \rightarrow^*_{\mathcal{P}} p(t_1, \ldots, t_n)$$

where p is a pattern and is also obvious. An intuitive interpretation of projectivity and a justification of the name is as follows. A term s rewrites to a term t with the same top pattern, if all "projections" of s rewrite to the "projections" of t. Here, "projection" means replacement of the top pattern g by d if a rule $g \rightarrow d$ exists. This also means that the knowledge of derivations $d_i\sigma \overset{*}{\rightarrow}_R d_i\theta$ can be lifted up as a derivation $g\sigma \overset{*}{\rightarrow}_R g\theta$. The situation can be complicated when the right-hand side of a rule is a non variable term and especially when this right-hand side overlaps other left-hand sides. This is a typical situation with basic term rewrite systems. It is then not trivial that what is known on derivations between instances of right-hand sides can be lifted-up to derivations between instances of left-hand sides. Projectivity has been shown by Uwe Waldman to be undecidable (see appendix). In the case of projective term rewrite systems, *derivability* always contains *pseudo-derivability*.

Proposition 2 *If R is a* projective *set of rewrite rules, then the R-pseudo-derivability $\overset{*}{\hookrightarrow}_R$ is contained in the R-derivability $\overset{*}{\rightarrow}_R$, i.e. $\overset{*}{\hookrightarrow}_R \subseteq \overset{*}{\rightarrow}_R$.*

> **Proof:** The proof is easy from a careful examination of an axiomatic definition of derivability which is the same as pseudo-derivability except property (2) that is:
>
> $$s = f(s_1, \ldots, s_m) \wedge f(t_1, \ldots, t_m) = t \wedge (s_1, \ldots, s_m)(\overset{*}{\rightarrow}_R)^{comp}(t_1, \ldots, t_m) \tag{2}$$
>
> for derivability. □

Therefore, we may state the main theorem as a corollary of Theorem 3.

Theorem 4 (Well rewrite orderings) *Let R be a* basic *and* projective *set of rewrite rules. If R-derivability is well-founded, then it is a well quasi-ordering.*

Corollary 1 *Given a set S of rewrite rules such that the derivability relation $\overset{+}{\rightarrow}_S$ has no loop, if R is a basic, projective and terminating set of rewrite rules such that $l \rightarrow r \in R$ implies $l \overset{+}{\rightarrow}_S r$ then $\overset{+}{\rightarrow}_S$ is a well-quasi ordering.*

> **Proof:** This comes from the preservation of well-foundedness by restriction of an ordering and the preservation of well-quasi orderedness by extension of an ordering (or incrementality). □

This corollary works in the specific case where $S \supseteq R$. Since the relation with $\xrightarrow{*}_{\mathcal{P}}$ is trivially well-founded, we may state a corollary about embedding with patterns.

Corollary 2 *If P is a set of pattern that contains a basic set of pattern, then the relation $\xrightarrow{*}_{\mathcal{P}}$ is a well quasi-ordering.*

and more specifically about embedding.

Corollary 3 *The relation $\longrightarrow^{*}_{\mathcal{EMB}}$ is a well quasi-ordering*

4 A direct definition of the ordering

In this section, we give a direct and axiomatic definition of the well rewrite orderings, without invoking the concept of rewriting.

R is a given set of pairs of terms. We define an ordering $>_R$ based on R as follows.

$$\begin{array}{ll} s >_R t \Rightarrow f(\ldots,s,\ldots) >_R f(\ldots,t,\ldots) & \textit{replacement} \\ s >_R t \Rightarrow s\sigma >_R t\sigma & \textit{instantiation} \\ s >_R t \ \wedge\ t >_R u \ \Rightarrow\ s >_R u & \textit{transitivity} \\ (\forall (s,s') \in R)\ s >_R s' & \textit{foundation} \end{array}$$

Properties of R. In addition R satisfies the two following properties:

- For almost each term t in $\mathcal{T}(\mathcal{F})$ i.e., each term but a finite number, there exists a pair $(s, s') \in R$ such t is an instance of s. R is said *basic*.
- Given a term s, and $\{(s,s'_1),\ldots,(s,s'_n)\}$ the set of pairs of R where s occurs as first component, if
$$s'_1\sigma >_R s'_1\theta \ \wedge\ \ldots\ \wedge\ s'_n\sigma >_R s'_n\theta$$
then
$$s\sigma >_R t\sigma.$$
R is said *projective*.

With these properties $>_R$ is a well quasi-ordering if and only if it a well founded ordering.

5 Examples

5.1 An example that contains a rule $f(f(x)) \to f(g(f(x)))$

In this section, we show that the derivability relation associated with the set of rewrite rules T:

$$\begin{aligned} f(f(x)) &\to f(g(f(x))) && (1) \\ g(g(x)) &\to S(x) && (2) \\ f(g(x)) &\to x && (3) \\ g(f(x)) &\to x && (4) \end{aligned}$$

where $S(x)$ is either x or $g(x)$ or $g(f(g(x)))$ is a well quasi-ordering.

We drop parentheses in the sequel.

Claim 1. The derivability relation $\xrightarrow{*}_T$ is well-founded. Indeed the rewrite relation $\xrightarrow{+}_T$ either decreases the number of subpatterns ff or gg or else decreases the length of the whole string.
Claim 2. T is basic. If A is the set $\{f(a), g(a)\}$, it can be easily seen that one of the patterns $f(f(x))$, $g(g(x))$, $f(g(x))$ or, $g(f(x))$ subsumes any term in $\mathcal{T}(\mathcal{F}) \ominus A$.
Claim 3. T is projective. For rules (3) and (4) and for rule (2) if $S(x)$ is either x or $g(x)$ the projectivity is obvious, let us prove it for rule (1), the proof for rule (2) when $S(x)$ is $g(f(g(x)))$ is symmetric and therefore will be omitted. In this precise case *projectivity* means:

$$fgf\alpha \underset{T}{\overset{*}{\rightarrow}} fgf\beta \;\Rightarrow\; ff\alpha \underset{T}{\overset{*}{\rightarrow}} ff\beta \tag{5}$$

The proof is done by induction on the n length of a derivation $fgf\alpha \xrightarrow{n}_T fgf\beta$, where α and β are two given strings and it never requires the right-hand side of rule (2).

$\mathbf{n = 0}$ The induction hypothesis means $fgf\alpha = fgf\beta$, therefore $\alpha = \beta$ and

$$ff\alpha \underset{T}{\overset{0}{\rightarrow}} ff\beta.$$

Assume the property is true for $n-1$ and suppose further the first rewrite takes place at the position i, i.e., in

$$fgf\alpha \underset{T}{\rightarrow} \gamma \underset{T}{\overset{n-1}{\rightarrow}} fgf\beta$$

the first letter of the pattern is the i-th symbol in the string $fgf\alpha$.

- The first rewrite takes place at $i \geq 4$, then

  $$fgf\alpha \underset{T}{\rightarrow} fgf\gamma' = \gamma \underset{T}{\overset{n-1}{\rightarrow}} fgf\beta$$

  where the first rewrite affects the α part of $fgf\alpha$, hence

  $$ff\alpha \underset{T}{\rightarrow} ff\gamma'$$

  By induction

  $$ff\gamma' \underset{T}{\overset{*}{\rightarrow}} ff\beta$$

  Therefore

  $$ff\alpha \underset{T}{\overset{*}{\rightarrow}} ff\beta$$

  q.e.d.
- The first rewrite takes place at 3. We can write

  $$f\alpha \underset{T}{\rightarrow} \gamma'$$

  and

  $$\gamma = fg\gamma' \underset{T}{\overset{n-1}{\rightarrow}} fgf\beta$$

- The first rule used is rule (1). We can write $\alpha = f\alpha'$. Therefore

$$ff\alpha' \underset{T}{\rightarrow} fgf\alpha' = \gamma'$$

or

$$f\alpha \underset{T}{\rightarrow} fg\alpha = \gamma'.$$

The induction hypothesis is then

$$fgf\alpha \underset{T}{\rightarrow} fgfg\alpha \overset{n-1}{\underset{T}{\rightarrow}} fgf\beta$$

and by induction

$$ffg\alpha \overset{*}{\underset{T}{\rightarrow}} ff\beta$$

using again

$$f\alpha \underset{T}{\rightarrow} fg\alpha$$

one gets

$$ff\alpha \underset{T}{\rightarrow} ffg\alpha \overset{*}{\underset{T}{\rightarrow}} ff\beta.$$

q.e.d.

- The first rule used is rule (3). We can write $\alpha = g\alpha'$. The induction hypothesis is then

$$fgfg\alpha' \underset{T}{\rightarrow} fg\alpha' \overset{n-1}{\underset{T}{\rightarrow}} fgf\beta$$

and by $gf\beta \rightarrow_T \beta$

$$fg\alpha' \overset{n}{\underset{T}{\rightarrow}} f\beta$$

therefore

$$ff\alpha = ffg\alpha' \overset{n}{\underset{T}{\rightarrow}} ff\beta.$$

q.e.d.

- The first rewrite takes place at 2, then

$$fgf\alpha \underset{T}{\rightarrow} f\alpha \overset{n-1}{\underset{T}{\rightarrow}} fgf\beta$$

We have

$$fgf\beta \underset{T}{\rightarrow} f\beta$$

therefore

$$f\alpha \overset{n}{\underset{T}{\rightarrow}} f\beta$$

and

$$ff\alpha \overset{n}{\underset{T}{\rightarrow}} ff\beta$$

q.e.d.

- The first rewrite takes place at 1, then

$$fgf\alpha \underset{T}{\rightarrow} f\alpha \overset{n-1}{\underset{T}{\rightarrow}} fgf\beta$$

and the proof continues just as above.

Notice that $\hookrightarrow_T^* \neq \overset{*}{\rightarrow}_T$; indeed $ffga \overset{*}{\rightarrow}_T fa$, but one does not have $ffga \hookrightarrow_T^* fa$.

5.2 A generalization

Given an ordering $>$ on a set A, an ordering $\succ$ on B is an epimorphic image of $>$, if there exists a surjective function $\phi : A \to B$ such that

$$\phi(a) \succ \phi(b) \ \Rightarrow\ a > b.$$

In this case, it is known that if $\succ$ is a well quasi-ordering, then $>$ is a well quasi-ordering. This result can be applied to the previous example to show that the following term rewrite system generates a well quasi-ordering. For this, we define the following morphism

$$\mathcal{T}(\{f_1, \ldots, f_m, g_1, \ldots, g_n\}) \to \mathcal{T}(\{f, g\})$$

that maps the f_i's onto f and the g_j's onto g and one gets a well rewrite ordering described by the following rules:
for $(i, j, k, p) \in [1..m]^4$ and $l \in [1..n]$,

$$f_i(f_j(x)) \ \to\ f_k(g_l(f_p(x)))$$

for $(i, j, k) \in [1..n]^3$,

$$g_i(g_j(x)) \ \to\ g_k(x)$$

for $i \in [1..m]$ and $j \in [1..n]$,

$$f_i(g_j(x)) \ \to\ x$$

for $j \in [1..m]$ and $i \in [1..n]$,

$$g_i(f_j(x)) \ \to\ x$$

By a simple generalization of the reasoning used for the previous example, one sees easily that this set of rules generates a well rewrite ordering.

5.3 Another example that cannot contain the embedding

In [1], Bachmair, Dershowitz, and Plaisted mention an ordering that cannot be a simplification ordering. It can be defined by the set of rewrite rules

$$\begin{aligned}
f(h(x)) &\to f(i(x)) && (1)\\
g(i(x)) &\to g(h(x)) && (2)\\
f(f(x)) &\to f(x) && (3)\\
g(g(x)) &\to g(x) && (4)\\
f(i(x)) &\to x && (5)\\
g(h(x)) &\to x && (6)\\
h(x) &\to x && (7)\\
i(x) &\to x && (8)
\end{aligned}$$

and proved to define a well rewrite ordering. This system can be easily proved basic and terminating. Again the interesting property is projectivity. Only rules (1) and (2) pose problem and because of the symetry of the definition with respect to f and g and h and i, one can only focus on rule (1). This leads to show that

$$f(i(\alpha)) \xrightarrow{*} f(i(\beta)) \ \Rightarrow\ f(h(\alpha)) \xrightarrow{*} f(h(\beta))$$

We need first a lemma.

Lemma 1 $i(\alpha) \stackrel{(8)}{\to} \alpha \stackrel{*}{\to} \beta \Rightarrow i(\alpha) \stackrel{*}{\to} i(\beta) \stackrel{(8)}{\to} \beta$

Again we prove the projectivity on the number of rewrites in $f(i(\alpha)) \stackrel{n}{\to} f(i(\beta))$, if $n = 0$, then $\alpha = \beta$. For the general case, suppose $f(i(\alpha)) \stackrel{k}{\to} f(i(\beta))$, for $k \leq n$.

- $\alpha \stackrel{n}{\to} \beta$, there is nothing to prove.
- If $f(i(\alpha)) \stackrel{n}{\to} f(i(\beta))$ there is no rewrite at the top, but

$$\alpha \stackrel{*}{\to} \gamma \wedge i(\gamma) \stackrel{(8)}{\to} \gamma \wedge \gamma \stackrel{*}{\to} i(\beta)$$

  then

$$\alpha \stackrel{*}{\to} \gamma \stackrel{*}{\to} i(\beta) \stackrel{(8)}{\to} \beta$$

  which means that the rewrite by (8) can be shifted on the right and which implies $\alpha \stackrel{*}{\to} \beta$.

- $f(i(\alpha) \stackrel{*}{\to} f(\gamma) \to_\epsilon \gamma' \stackrel{*}{\to} f(i(\beta))$, where the first rewrites from $f(i(\alpha)$ to $f(\gamma)$ are all done below the top and the rewrite from $f(\gamma)$ to γ' is done at the top. Three cases have to be considered according to the rule which is used to rewrite $f(\gamma)$ to γ'.
  - $f(\gamma) \stackrel{(1)}{\to} \gamma'$ wich means $f(h(\gamma_1)) \stackrel{(1)}{\to} f(i(\gamma_1))$, the length of $f(i(\gamma_1)) \stackrel{*}{\to} f(i(\beta))$ is smallest than the length of $f(i(\alpha)) \stackrel{*}{\to} f(i(\beta))$, therefore by the induction hypothesis $f(h(\gamma_1)) \stackrel{*}{\to} f(h(\beta))$ and $f(h(\alpha)) \stackrel{(1)}{\to} f(i(\alpha)) \stackrel{*}{\to} f(h(\gamma_1)) \stackrel{*}{\to} f(h(\beta))$.
  - $f(\gamma) \stackrel{(3)}{\to} \gamma'$ wich means $f(f(\gamma_1)) \stackrel{(3)}{\to} f(\gamma_1)$. Since the first rewrites are below the top, we know that $i(\alpha) \stackrel{*}{\to} f(\gamma_1)$, in this sequence of rewrites there is at least one use of rule (8) at the top that can shifted by Lemma 1 yielding $i(\alpha) \stackrel{*}{\to} i(f(\gamma_1)) \stackrel{(8)}{\to} f(\gamma_1)$ and we can consider that all the other rewrites are done below the top. This means $\alpha \stackrel{*}{\to} f(\gamma_1)$. On the other hand $f(\gamma_1) \stackrel{*}{\to} f(i(\beta)) \stackrel{(5)}{\to} \beta$, thus $\alpha \stackrel{*}{\to} \beta$.
  - $f(\gamma) \stackrel{(5)}{\to} \gamma'$ wich means $f(i(\gamma_1)) \stackrel{(3)}{\to} \gamma_1$. By induction, $f(i(\alpha)) \stackrel{*}{\to} f(i(\gamma_1))$ implies $f(h(\alpha)) \stackrel{*}{\to} f(h(\gamma_1))$ and since $\gamma_1 \stackrel{*}{\to} f((i(\beta)) \stackrel{(5)}{\to} \beta$, $f(h(\gamma_1)) \stackrel{*}{\to} f(h(\beta))$, then $f(h(\alpha)) \stackrel{*}{\to} f(h(\beta))$.

5.4 A well-founded pseudo-derivability

We consider now an example of a non projective set of rules on $\mathcal{T}(\{f,g,h,a\},\mathcal{X})$ which provides a well-founded pseudo-derivability

$$\begin{array}{rcl} f(f(x)) & \to & f(g(f(x))) \\ g(g(x)) & \to & g(x) \\ f(g(x)) & \to & x \\ g(f(x)) & \to & x \end{array} \qquad \begin{array}{rcl} f(h(x)) & \to & f(x) \\ g(h(x)) & \to & g(x) \\ h(x) & \to & x. \end{array}$$

It is obviously basic, but it is not projective because of $ffa \to fgfa$ and $fhfa \not\to fhgfa$. However the pseudo-derivability is well-founded and the proof uses an argument similar to this of Section 5.1. Therefore the pseudo-derivability is a well-quasi ordering.

6 An ordering based on patterns

In this section, we are going to describe an extension of the recursive path ordering [2]. We use a *precedence* that is an ordering between patterns.

Definition 5 (Pattern precedence) *A* precedence *on a set of patterns P is an ordering such that $p > p'$ implies $p' \equiv f(s_1, \ldots, s_n)$ with $f \in \mathcal{F}$ occurs only in rules of the form*

$$f(s_1, \ldots, s_n) \to s_i$$

and $p > q$ for every pattern of the form $q \equiv f(t_1, \ldots, t_n)$.

In the following we are going to use two abbreviations: $\geq_{ppo[B]}$ stands for $>_{ppo[B]} \cup \equiv$ and $(>_{ppo[B]})^{comp}$ stands for $(\geq_{ppo[B]})^{comp} \ominus \equiv$.

Definition 6 (Pattern Path Ordering) *Given a basic set of rules B and a pattern precedence $>$, the* pattern path ordering *or* ppo *w.r.t. B and $>$ is written $>_{ppo[B]}$; it is defined as follows:*

$$s = f(s_1, \ldots, s_m) = g\sigma >_{ppo[B]} t = h(t_1, \ldots, t_n) = g'\tau$$

if and only if *one of the following holds*

1. $g > g' \wedge s >_{ppo[B]} t_1 \wedge \ldots \wedge s >_{ppo[B]} t_n$
2. $(\exists(u_1, \ldots, u_m))\ (s_1, \ldots, s_m) >_{ppo[B]}^{comp} (u_1, \ldots, u_m) \wedge f(u_1, \ldots, u_m) \geq_{ppo[B]} t$
3. $(\exists[g \to d] \in B)\ d\sigma \geq_{ppo[B]} t$

Notice that the pattern path ordering uses a componentwise composition instead of a lexicographical composition as the classical lexicographical path ordering does. This is done to make the proof of the replacement property obvious. The pattern path ordering is not reflexive as seen from its definition. It is transitive.

Proposition 3 (Transitivity of the ppo) *The pattern path ordering is transitive.*

Proof: Let use the abbreviation $\vec{s}$ for $(s_1, \ldots, s_m)$ and $f\vec{s}$ for $f(s_1, \ldots, s_m)$ and let $s = f\vec{s} = g\sigma$, $t = h\vec{t} = g'\tau$ and $u = k\vec{u} = g''\theta$ be such that $s >_{ppo[B]} t$ and $t >_{ppo[B]} u$. The proof is done by induction on (i, j) where i is the number of calls to the definition for proving $s >_{ppo[B]} t$, q is the number of calls to the definition for proving $t >_{ppo[B]} u$, and $(i, j) > (i', j')$ if $j > j'$ or $j = j' \wedge i > i'$. Let us look at the way the proofs are done.

- *The first step of $s >_{ppo[B]} t$ uses* (2). Let $\vec{s} >_{ppo[B]}^{comp} \vec{v}$ and $f\vec{v} \geq_{ppo[B]} t$. By induction, $f\vec{v} \geq_{ppo[B]} t$ and $t >_{ppo[B]} u$ give $f\vec{v} \geq_{ppo[B]} u$, then by (2) $s >_{ppo[B]} u$.
- *The first step of $s >_{ppo[B]} t$ uses* (3). $d\sigma \geq_{ppo[B]} t$ and $t >_{ppo[B]} u$ give by induction $d\sigma \geq_{ppo[B]} u$ and then by (3) $s >_{ppo[B]} u$.
- *The first step of $s >_{ppo[B]} t$ uses* (1) and *the first step of $t >_{ppo[B]} u$ uses* (1). Then $g > g' > g''$ and $(\forall i \in [1..n])s >_{ppo[B]} t_i$ and $(\forall j \in [1..n'])t >_{ppo[B]} u_j$. By transitivity $g > g''$ and by induction, $(\forall j \in [1..n'])s >_{ppo[B]} u_j$. Then (1) applies.

- *The first step of* $s >_{ppo[B]} t$ *uses* (1) and *the first step of* $t >_{ppo[B]} u$ *uses* (2). One has $g > g'$, $(\forall i \in [1..n])\ s >_{ppo[B]} t_i$, $\vec{t} >^{comp}_{ppo[B]} \vec{v}$, and $h\vec{v} \equiv p\varphi \geq_{ppo[B]} u$. By induction, $(\forall i \in [1..n])\ s >_{ppo[B]} v_i$; therefore, since $g > p$ (by property on precedences) by (1) $s >_{ppo[B]} h\vec{v}$. By induction $h\vec{v} >_{ppo[B]} u$ and $s >_{ppo[B]} h\vec{v}$ gives $s >_{ppo[B]} u$.
- *The first step of* $s >_{ppo[B]} t$ *uses* (1) and *the first step of* $t >_{ppo[B]} u$ *uses* (3). From $g > g'$ we know that g' occurs only in a rule of the form $h(v_1, \ldots, v_n) \rightarrow v_i$. Therefore, the first step of $t >_{ppo[B]} u$ is $t_i \geq_{ppo[B]} u$. Since $s >_{ppo[B]} t_i$ is a part of the proof of the first step, $s >_{ppo[B]} u$ comes by induction.

□

In the definition of the pattern path ordering w.r.t. B, (2) and (3) are the definitions of the derivability associated with B. Therefore, $\xrightarrow{*}_B \subset >_{ppo[B]}$ and if the derivability is well-founded, the pattern path ordering is a well quasi-ordering, therefore it is well-founded and incremental.

Proposition 4 *If B is a basic, projective and, terminating set of rules, the* pattern path ordering *is a well-quasi ordering.*

In [19], another extension of the recursive path ordering is proposed. It is based on Theorem 2 and unavoidability; there is not always a pattern at the top and the relevant pattern is found through a hierarchy using the concept of decomposition.

7 Conclusion

This study raises a lot of open issue. Among them, are

- To apply the pattern path ordering or another extension to the proof of term rewrite systems,
- To extend the scope of our main theorem especially by relaxing the condition that the left-hand sides have to form a basic set,
- To propose a constructive proof of the main theorem like this of J. Russell and C. Murthy for Higman's lemma [17].

References

[1] L. Bachmair, N. Dershowitz, and D. Plaisted. Completion without failure. In H. Ait-Kaci and M. Nivat, editors, *Resolution of Equations in Algebraic Structures, Volume 2: Rewriting Techniques*, pages 1–30. Academic Press, 1989.

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

[3] N. Dershowitz. Termination of rewriting. *Journal of Symbolic Computation*, 3(1 & 2):69–116, 1987.

[4] N. Dershowitz and J.-P. Jouannaud. Rewrite systems. In Van Leuven, editor, *Handbook of Theoretical Computer Science.* North Holland, 1990.

[5] R. Forgaard and D. Detlefs. An incremental algorithm for proving termination of term rewriting systems. In J.-P. Jouannaud, editor, *Proceedings 1st International Conference on Rewriting Techniques and Applications*, pages 255–270. Springer Verlag, 1985.

[6] J. Gallier. What's so special about Kruskal's theorem and the ordinal Γ_0? A survey of some results in proof theory. Technical report, University of Pennsylvania, 1989.

[7] G. Higman. Ordering by divisibility in abstract algebra. *Proc. London Math. Soc.*, 3(2):326–336, 1952.

[8] M. Janet. Sur les systèmes d'équations aux dérivées partielles. *Journal de Mathématiques*, III(8):65–151, 1920.

[9] J.-P. Jouannaud, P. Lescanne, and F. Reinig. Recursive decomposition ordering. In Bjørner D., editor, *Formal Description of Programming Concepts 2*, pages 331–348, Garmisch-Partenkirchen, Germany, 1982. North Holland.

[10] J. B. Kruskal. The theory of well-quasi-ordering: A frequently discovered concept. *J. Combinatorial Theory Ser. A*, 13(3):297–305, November 1972.

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

[12] P. Lescanne. Computer experiments with the REVE term rewriting systems generator. In *Proceedings of 10th ACM Symposium on Principles of Programming Languages*, pages 99–108. ACM, 1983.

[13] P. Lescanne. Well quasi-orderings in a paper by Maurice Janet. *Bulletin of European Association for Theoretical Computer Science*, 39:185–188, October 1989.

[14] P. Lescanne. On the recursive decomposition ordering with lexicographical status and other related orderings. *Journal of Automated Reasoning*, 6:39–49, 1990.

[15] P. Lescanne. Well rewrite orderings. In J. Mitchell, editor, *5th Symp. Logic in Computer Science*, pages 239–256. IEEE, 1990.

[16] J-J. Lévy. Kruskaleries. Unpublished manuscript, 1981.

[17] C. R. Murthy and J. R. Russell. A constructive proof of Higman's lemma. In J. Mitchell, editor, *5th Symp. Logic in Computer Science*, pages 257–267. IEEE, 1990.

[18] C. St. J. A. Nash-Williams. On well-quasi-ordering finite trees. *Proc. Cambridge Phil. Soc.*, 59:833–835, 1963.

[19] L. Puel. Embedding with patterns and associated recursive path ordering. In N. Dershowitz, editor, *Proceedings 3rd Conf. on Rewriting Techniques and Applications*, pages 371–387. Springer-Verlag, April 1989. Lecture Notes in Computer Science, volume 355.

[20] L. Puel. Using unavoidable set of trees to generalize Kruskal's theorem. *Journal of Symbolic Computation*, 8:335–382, 1989.

Appendix

Undecidability of the projectivity

Uwe Waldmann
Dortmund University, Germany

In this section, one finds a proof of the undecidability of the projectivity due to Uwe Waldmann.

Theorem 5 (Undecidability of the projectivity) *The projectivity is undecidable even in the case of monadic term rewrite systems or Thue systems.*

The proof of this theorem requires a lemma on reachability in term rewrite systems whose proof is easily obtained by coding Turing machines for instance.

Lemma 2 *Given a monadic signature $\mathcal{F}$, a term rewrite system R over $\mathcal{T}(\mathcal{F})$ and two ground terms $t \in \mathcal{T}(\mathcal{F})$ and $t' \in \mathcal{T}(\mathcal{F})$, it is undecidable whether $t \xrightarrow{*}_R t'$.*

The proof of the theorem is now done by reducing the proof of reachability for two terms t ad t' to a proof of the projectivity of a term rewrite system built from R.

Now suppose given two terms t and t' of $\mathcal{T}(\mathcal{F})$ and a rewrite system R. One builds an associated rewrite system R' on $\mathcal{T}(\mathcal{F}')$ where $\mathcal{F}'$ is defined by

$$\mathcal{F}' = \mathcal{F} \biguplus \{a, b, f, h, k_{l\to r}\}$$

with a and b fresh constant, f and h monadic functions and $k_{l\to r}$ a n_l-ary function if l contains exactly n_l variables. R' is defined by

$$\begin{aligned} R' &= R \biguplus \{l \to k_{l\to r}(x_1, \ldots, x_{n_l}) \mid \text{for each } l \to r \in R\} \\ &\quad \uplus f(a) \to t \uplus t' \to f(b) \uplus h(a) \to h(b) \uplus f(x) \to h(x) \end{aligned}$$

The proof is now made of three lemmas:

Lemma 3 *$R \biguplus \{l \to k_{l\to r}(x_1, \ldots, x_{n_l})\}$ is projective with respect to R'*

> **Proof:** $k_{l\to r}(x_1, \ldots, x_{n_l})\sigma \xrightarrow{*}_{R'} k_{l\to r}(x_1, \ldots, x_{n_l})\theta$ implies $x_m\sigma \xrightarrow{*}_{R'} x_m\theta$ for all $m \in [1..n_l]$ which implies that $l\sigma \xrightarrow{*}_{R'} l\theta$. □

Lemma 4 *$\{fa(a) \to t, t' \to f(b), h(a) \to h(b)\}$ is projective with respect to R'*

> **Proof:** All rewrite rules are ground □

From Lemma 3 and Lemma 4 we see that R' is projective if and only if the only rule $f(x) \to h(x)$ is projective with respect R' if and only if $f(a) \xrightarrow{*}_{R'} f(b)$.

Lemma 5 *$f(a) \xrightarrow{*}_{R'} f(b)$ if and only $t \xrightarrow{*}_R t'$*

> **Proof:** $f(a) \xrightarrow{*}_{R'} f(b)$ if and only if $t \xrightarrow{*}_{R'} f(b)$ if and only $t \xrightarrow{*}_{R'} t'$. Let us show that this last derivation can be made using rules of R only. Suppose indeed that $t \xrightarrow{*}_{R'} t'$ using a rule $l \to k_{l\to r}(x_1, \ldots, x_{n_l})$ at position p. Then there exists a variable-deleting rule that is applied above p and the application of $l \to k_{l\to r}(x_1, \ldots, x_{n_l})$ can be omitted. A similar argument holds for $t' \to f(b)$. Hence in the derivation $t \xrightarrow{*}_{R'} t'$ none of the function symbols is generated. Thus the rules $f(a) \to t$, $h(a) \to h(b)$ and $f(x) \to h(x)$ are not applied, thus $t \xrightarrow{*}_R t'$. □

CONTEMPORARY MATHEMATICS

Logic and Combinatorics

Proceedings of a Summer Research Conference held August 4–10, 1985

AMERICAN MATHEMATICAL SOCIETY

VOLUME 65

Imprimé en France
par
l'Institut National de Recherche en Informatique et en Automatique

ISSN 0249-6399