Centrum voor Wiskunde en Informatica

REPORT RAPPORT

Comparing curried and uncurried rewriting

J.R. Kennaway, J.W. Klop, M.R. Sleep, F.-J. de Vries

Computer Science/Department of Software Technology

Report CS-R9350 July 1993

P.O. Box 94079, 1090 GB Amsterdam (NL)
Kruislaan 413, 1098 SJ Amsterdam (NL)
Telephone +31 20 592 9333
Telefax +31 20 592 4199

Comparing Curried and Uncurried Rewriting

Richard Kennaway[1], Jan Willem Klop[2], Ronan Sleep[3] and Fer-Jan de Vries[4]

[1,3]*School of Information Systems,*
University of East Anglia,
Norwich NR4 7TJ, UK

[2,4]*CWI,*
P.O. Box 4079,
1009 AB Amsterdam, The Netherlands

[1]jrk@sys.uea.ac.uk, [2]jwk@cwi.nl, [3]mrs@sys.uea.ac.uk, [4]ferjan@cwi.nl

Abstract

The properties WN, SN, WCR and completeness of functional term rewriting systems are preserved if we curry these systems into applicative term rewriting systems. Under the condition of left-linearity the properties CR, NF, UN, $\text{UN}^{\rightarrow}$ and semi-completeness are also preserved by currying. By a counterexample we show that the latter set of properties, with the possible exception of semi-completeness, are in general not preserved for non-left-linear systems.

1991 AMS Mathematics Subject Classification: 68Q42.
1991 CR Categories: F4.2, F4.1.
Keywords & Phrases: term rewriting, preservation by currying, Church-Rosser property, confluence, strong normalisation, weak normalisation, termination.
Note. The authors are partially sponsored by SEMAGRAPH II, ESPRIT working group 6345. The first author was also partially supported by a SERC Advanced Fellowship and by SERC grant no. GR/F 91582.

Dedicated to Dirk van Dalen
on the occasion of his 60th anniversary.

1 Introduction

Rewriting of various kinds is used as both a theoretical and a practical computational mechanism[1]. Term rewriting is an attractive, simple form of rewriting: only terms from a first order language without bound variables are rewritten.

The modern theory of term rewriting is mainly concerned with functional rewriting (cf. [DJ89, Klo92]), where terms are constructed from function symbols of various arities, and variables. However, Curry and Feys originally introduced term rewriting in [CF58] in the applicative

[1] This is a revision of a paper that appeared in the *Dirk van Dalen Festschrift* [KKSdV93].

Report CS-R9350
ISSN 0169-118X
CWI
P.O. Box 94079, 1090 GB Amsterdam, The Netherlands

form (or in their terminology, quasi-applicative). In this form, terms are built from variables, nullary function symbols, and a binary application (often suppressed in notation).

From the modern perspective, applicative term rewriting is just a special case of functional term rewriting. The main example of an applicative term rewrite system, Combinatory Logic (CL), was developed by Schönfinkel and rediscovered by Curry ([CF58]). Applicative term rewriting systems related to CL play an important role in the design and implementation of functional programming languages such as Miranda (cf. [FH88]).

Any functional term rewrite system can be transformed into an applicative term rewrite system by the well-known method of *currying* credited to Schönfinkel (e.g. in [CF58]). So, it seems natural to ask *which properties of term rewrite systems are preserved under currying.* This question has not been studied before.

In this paper we show that strong normalisation (SN), weak normalisation (WN), weak Church-Rosser (WCR) and completeness are preserved by currying. For left-linear term rewrite systems we show that currying also preserves the Church-Rosser property, or confluence (CR), the normal form property (NF), semi-completeness and two properties UN and UN$^{\rightarrow}$ concerning unique normal forms. The properties CR, NF, UN and UN$^{\rightarrow}$ are not always preserved for non-left-linear systems, as demonstrated by a counterexample.

2 Preliminaries

2.1 Term Rewriting

In this section we give a brief description of term rewriting. Ample introductions to the subject are [DJ89, Klo92]. A *Term Rewriting System* (TRS) is a pair (Σ, R) of a signature Σ and a set of rewrite rules R. A *signature* Σ is a pair $< |\Sigma|, arity >$, where $|\Sigma|$ is a set of function symbols whose arity is given by the function $arity : |\Sigma| \longrightarrow \mathcal{N}$ which maps function symbols into natural numbers. Often we will leave the signature implicit in examples. The set $\text{Ter}(\Sigma)$ of *terms* over Σ is built in the usual way from $|\Sigma|$ and a countably infinite set of *variables* V, disjoint from $|\Sigma|$: $\text{Ter}(\Sigma)$ is the smallest set such that

- $V \subseteq \text{Ter}(\Sigma)$,
- for any $f \in |\Sigma|$ if $t_1 \in \text{Ter}(\Sigma), \ldots, t_{arity(f)} \in \text{Ter}(\Sigma)$ then $f(t_1, \ldots, t_{arity(f)}) \in \text{Ter}(\Sigma)$.

We will write F instead of $F()$ if the arity of F happens to be 0.

A *rewrite rule* is a pair (l, r) of terms $\in \text{Ter}(\Sigma)$ denoted by $l \rightarrow r$ and subject to the following two conditions:

- the left hand side l is not a variable,
- the variables in the right-hand side r are already contained in l.

The non-variable part of the left or right hand side of a rule is sometimes called *pattern.* A rewrite rule $l \rightarrow r$ is called *collapsing* if r is a variable. The rule $l \rightarrow r$ is *duplicating* if the right hand side r contains more occurrences of some variable than the left hand side l.

Contexts are 'terms' containing one occurrence of a special symbol $\Box$, denoting an empty place. A context is generally denoted by C[ ]. If $t \in \text{Ter}(\Sigma)$ and t is substituted in $\Box$, the result is $C[t] \in \text{Ter}(\Sigma)$; t is said to be a *subterm* of $C[t]$, notation $t \trianglelefteq C[t]$. Since $\Box$ is itself a context, the trivial context, we also have $t \trianglelefteq t$.

A *substitution* $\sigma : \mathrm{Ter}(\Sigma) \to \mathrm{Ter}(\Sigma)$ is a map satisfying $\sigma(F(t_1,\ldots,t_n)) = F(\sigma(t_1),\ldots,\sigma(t_n))$ for every n-ary function symbol F (here $n \geq 0$). So, σ is determined by its restriction to the set of variables. We also write t^σ instead of $\sigma(t)$.

A term s is *contained* in (or encompassed by) a term t (notation $s \trianglelefteq t$) if there is a context $C[\,]$ and a substitution σ such that $t = C[s^\sigma]$. (Examples: $f(x) \trianglelefteq g(f(a))$ and $x \trianglelefteq y$.)

The set of rewrite rules determines a *reduction relation* $\to$ on $\mathrm{Ter}(\Sigma)$. The term t reduces to s by *contracting* redex l^σ (notation $t \to s$) if t is of the form $C[l^\sigma]$ and s is of the form $C[r^\sigma]$ for some rule $l \to r$, context $C[\,]$ and substitution σ. Concatenating reduction steps we obtain (possibly infinite) *reduction sequences* $t_0 \to t_1 \to t_2 \to \cdots$ or *reductions* for short. If $t_0 \to \cdots \to t_n$ we also write $t_0 \to^* t_n$.

Two terms t, s are *convertible* (notation $t = s$) if they belong to the transitive, reflexive closure of $\to \cup \leftarrow$.

A term t in $\mathrm{Ter}(\Sigma)$ *is a normal form* if it does not contain a redex. t *has a normal form* n, if $t \to^* n$ and n is a normal form.

We will often consider pairs of reduction sequences having the same initial or final term. A *fork* is a pair of reduction sequences of the form $t_1 \leftarrow^* t \to^* t_2$. A *join* is a pair of reduction sequences of the form $t_1' \to^* t' \leftarrow^* t_2'$. It is a join of the above fork if $t_1' = t_1$ and $t_2' = t_2$, and the four reduction sequences together are then called a *tile.*

TRSs can be joined together provided the union of their alphabets is a well-formed alphabet (that is, provided each symbol common to both systems has the same arity in both). If so, then we denote by $R_1 + R_2$ the TRS obtained by taking the union of the alphabets and the sets of rewrite rules of R_1 and R_2. If the alphabets are disjoint we write $R_1 \oplus R_2$.

The set of *positions* $O(t)$ of a term t in a TRS R is defined inductively by:

- $O(t) = \lambda$, if t is a variable,
- $O(t) = \{\lambda\} \cup \{i \cdot u \mid 1 \leq i \leq n \text{ and } u \in O(t_i)\}$, if $t = f(t_1,\ldots,t_n)$.

Positions are *partially ordered by prefix order*, i.e., $u \leq v$ if there exists a (necessarily unique) w such that $u \cdot w = v$. The subterm of t at position $u \in O(t)$ is denoted by $t_{|u}$ and defined inductively by:

- $t_{|\lambda} = t$,
- $t_{|i \cdot u} = (t_i)_{|u}$, if $t = f(t_1,\ldots,t_n)$.

2.2 Applicative Term Rewriting

An *Applicative Term Rewriting System* (ATRS) is a TRS (Σ, R) where $\Sigma =< |\Sigma| \cup \{Ap\}, arity >$ in which $arity : |\Sigma| \cup \{Ap\} \longrightarrow \mathcal{N}$ is defined by

$$arity(x) = \begin{cases} 0 & \text{if } x \in |\Sigma|, \\ 2 & \text{if } x = Ap. \end{cases}$$

The set of terms we denote by $\mathrm{ATer}(\Sigma)$. The standard example of an ATRS is Combinatory Logic based on the combinators S, K, I with the following rewrite rules:

$$\begin{array}{lcl} Ap(Ap(Ap(S,x),y),z) & \to & Ap(Ap(x,z),Ap(y,z)) \\ Ap(Ap(K,x),y) & \to & x \\ Ap(I,x) & \to & x \end{array}$$

The presence of only one binary operator allows for the usual notational conventions:

- use infix notation $t \cdot s$ for $Ap(t, s)$
- as in ordinary algebra, suppress the dot
- associate to the left in order to use as few brackets as possible.

Following these notational conventions the above rewrite rules for Combinatory Logic become:

$$CL = \begin{cases} Sxyz & \rightarrow & xz(yz) \\ Kxy & \rightarrow & x \\ Ix & \rightarrow & x \end{cases}$$

It is a convenient fiction to view the S, K, I in the last three equations as "operators with variable arity".

3 Currying

Currying is a well-known construction that given a TRS R produces a corresponding *applicative* TRS R^{cur}. It is usually credited to Schönfinkel, cf. [CF58]. Let us demonstrate currying with an easy example. Consider the TRS R given by the following rules:

$$M = \begin{cases} M(x, x) & \rightarrow & 0 \\ M(Succ(x), x) & \rightarrow & 1 \end{cases}$$

Its currying M^{cur} will be the ATRS given by the rules:

$$M^{cur} = \begin{cases} Mxx & \rightarrow & 0 \\ M(Succ\ x)x & \rightarrow & 1 \end{cases}$$

So the curried TRS is constructed with the same set of function symbols, together with application, but we have "forgotten" about their arity and treat the former function symbols as constants.

The definition of currying is as follows:

DEFINITION 3.1

- The function cur:Ter(Σ) →ATer(Σ) is defined by induction on the structure of the terms in Ter(Σ) by the clauses:
  - $cur(x) = x$
  - $cur(f(t_1, \ldots, t_n)) = (f\ cur(t_1)\ \ldots\ cur(t_n))$
- Let R be a TRS. The *currying* R^{cur} of R is the applicative TRS with alphabet Σ and set of rewrite rules $\{cur(l) \rightarrow cur(r) \mid l \rightarrow r \in R\}$.

The function cur :Ter(Σ) →ATer(Σ) is not surjective: e.g., in the example M^{cur} the terms xx, M, Mx and $Mxyz$ are not in the image of cur: $M \rightarrow M^{cur}$.

The following observation is important, its proof trivial:

LEMMA 3.2 *Currying preserves redexes and reductions: let R be a TRS.*

- *l^σ is a redex for R in t, if and only if $cur(l)^{cur \circ \sigma}$ is a redex for R^{cur} in t^{cur}.*

- *$t \rightarrow s$ in R, if and only if $cur(t) \rightarrow cur(s)$ in R^{cur}.*

- *if t is a normal form in R, then t^{cur} is a normal form,*

- *R is isomorphic to a subTRS of R^{cur}, where a morphism $f : R_1 \rightarrow R_2$ between TRSs is a reduction preserving mapping ($t \rightarrow s$ implies $f(t) \rightarrow f(s)$), and a subTRS of R is a set of terms of R closed under reduction. In other words: R^{cur} is a conservative extension of this subTRS.* □

Of importance are the fragments of a term in a curried TRS in which all function symbols receive exactly the right number of arguments.

Definition 3.3 Let t be a term of a curried TRS R^{cur}.

- A *balanced* position of t is
  - either the position of a variable
  - or a position of the form $w\underbrace{1\cdots1}_{k}$ such that $t_{|w\underbrace{1\cdots1}_{n}}$ is a function symbol in R of arity n and $0 \leq k \leq n$.

- The set of *balanced positions $Bal(t)$* of t is the set of balanced positions of t.

The set of balanced positions in a term falls into zero or more patches: maximal connected non-empty subsets of balanced positions.

Definition 3.4 Let t be a term in a curried TRS R^{cur}.

- The *maximal common prefix* of two positions $u, v \in O(t)$ is denoted by $u \wedge v$.

- A subset $P \subseteq O(t)$ is *connected* if $u \wedge v \in P$ for all $u, v \in P$.

- A subset $P \subseteq O(t)$ is called a *patch* of t if it is a maximal connected non-empty subset of balanced positions.

In general a patch P of a term t of R^{cur} determines a partial subterm of t. If we complete it with distinct variables, we get a completely balanced term, which corresponds uniquely (modulo a bijective renaming of variables) with an uncurried term in R. By abuse of terminology we will identify the patch P with this term in the original TRS R. The possibility of non-left-linear rules makes the completion procedure a bit subtle: we use as many variables as there are different principal subterms of the patch, taking care that we replace identical principal subterms of the patch by identical variables.

Lemma 3.5 *There is a one-one correspondence between patches of a term t in a curried TRS R^{cur} and equivalence classes of terms p which are maximal for the property $p^{cur} \trianglelefteq t$, the equivalence being a bijective renaming of variables.* □

We need some more terminology. An example will follow the definitions.

Definition 3.6

- A term in R^{cur} is called *balanced* if all its positions are balanced.

- If $t = C[P[t_1, \ldots, t_n]]$ for some $t_1, \ldots, t_n$ in R^{cur}, context $C[]$ and patch P, then the subterms t_i are called the *principal subterms* of the patch P in t.

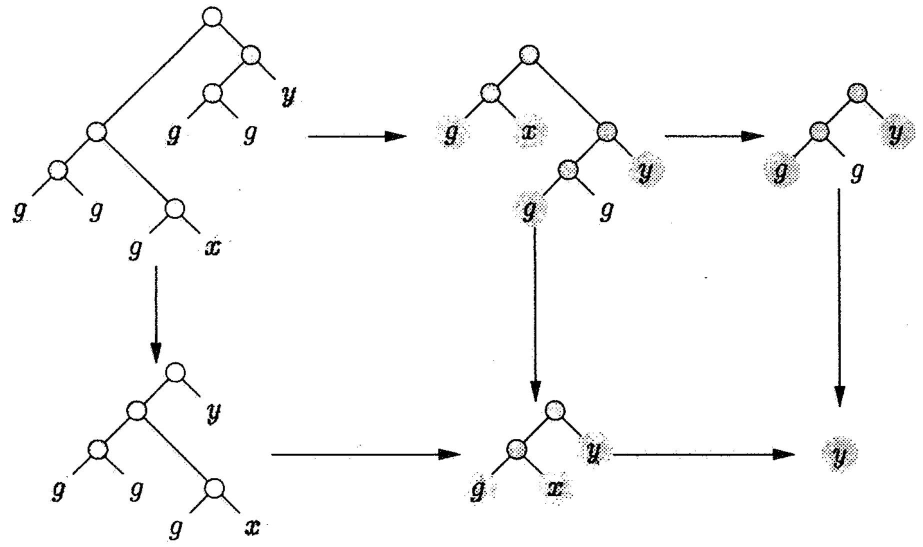
DEFINITION 3.7 Let $t_{|u} = f$ for some $u \in O(t)$. Let $n = arity(f)$.

- f at u is *underbalanced* if $\underbrace{1 \cdots 1}_{n}$ is not a suffix of u,
- f at u is *overbalanced* if $\underbrace{1 \cdots 1}_{n+1}$ is a suffix of u,
- otherwise, f at u is *balanced*, i.e., when the maximal suffix of u of the form $1 \ldots 1$ has length n.

As an example consider the ATRS constructed from the TRS consisting of the single rule $g(x, y) \to y$. Consider the term $gg(gx)(ggy)$, or if we make all application symbols explicit: $Ap(Ap(Ap(g, g), Ap(g, x)), Ap(Ap(g, g), y))$. Balanced positions are: $\{1, 11, 111, 122, 2, 21, 211, 22\}$. The following set of positions are patches: $\{\{1, 11, 111\}, \{122\}, \{2, 21, 211, 22\}\}$. The function symbol g at 111 is overbalanced, the ones at 112 and 121 are underbalanced, whereas g at 211 is balanced. We picture all possible derivations of the term $gg(gx)(ggy)$:

$$\begin{tikzcd}
gg(gx)(ggy) \arrow[r] \arrow[d] & gx(ggy) \arrow[r] \arrow[d] & ggy \arrow[d] \\
gg(gx)y \arrow[r] & gxy \arrow[r] & y
\end{tikzcd}$$

If we colour the balanced positions, we can observe the dynamics of the patches during reduction. The terms in the left column contain three patches, the others one.



Finally it is useful to observe that the pattern of any rewrite rule in any term t of a curried TRS R^{cur} is entirely contained in a patch, since then $l^{cur} \trianglelefteq C[(l^{cur})^{\sigma}] \equiv t$ for suitable $C[]$, l and σ.

4 Properties preserved by currying arbitrary term rewriting systems

In this section we will show that currying preserves the following properties of TRS: weak normalisation, strong normalisation, the weak Church-Rosser property and completeness. Let us first recall the definitions of the properties for TRSs which play a role in this section.

4.1 Preliminary facts and definitions

DEFINITION 4.1

- A TRS R is *weakly normalising* (WN) if every term of R has a normal form.
- A TRS R is *strongly normalising* (SN) if there are no infinite reduction sequences $t_1 \to t_2 \to t_3 \to \cdots$ of elements of R.
- A TRS R is *locally confluent* or *weakly Church-Rosser* (WCR) if every fork of the form $t_1 \leftarrow t \to t_2$ (that is, where the two reduction sequences each contain exactly one step) has a join $t_1 \to^* s \leftarrow^* t_2$.

$$\begin{tikzcd}
t \arrow[r] \arrow[d] & t_1 \arrow[d, dotted, "*"] \\
t_2 \arrow[r, dotted, "*"] & s
\end{tikzcd}$$

- A TRS R is *confluent* or *Church-Rosser* (CR) if every fork $t_1 \leftarrow^* t \to^* t_2$ has a join $t_1 \to^* s \leftarrow^* t_2$.

$$\begin{tikzcd}
t \arrow[r, "*"] \arrow[d, "*"'] & t_1 \arrow[d, dotted, "*"] \\
t_2 \arrow[r, dotted, "*"] & s
\end{tikzcd}$$

- A TRS is *complete* if it is both SN and CR.

Actually, completeness is equivalent to the combined properties SN and WCR by the well-known lemma of Newman:

LEMMA 4.2 **(Newman's Lemma)** *If a TRS is SN and WCR, then it is CR (e.g. see [Klo92]).*

We will use the following modularity result of the property SN, conjectured by Rusinowitch and proved by Middeldorp.

THEOREM 4.3 *The disjoint sum of two strongly normalising TRSs is strongly normalising, when one of them contains neither collapsing rules nor duplicating rules [Mid90].*

4.2 Preservation of WN by currying

THEOREM 4.4 *If R is WN, then R^{cur} is WN.*

PROOF. Let the TRS R be WN. By induction on the structure of terms we will prove that R^{cur} is WN. There are three cases to distinguish for a term t in R^{cur}.

- t is a single variable. Then it is already in normal form.
- Suppose t is a function symbol. Then either it is a normal form, or balanced. If it is balanced then it is a 0-ary term also occurring in R. Hence it has a reduction to normal form. If we curry this reduction we get a reduction to normal form in R^{cur}.
- Otherwise, $t = t_1 t_2$. By induction both t_1 and t_2 have normal forms, say T_1 and T_2.

  If $T_1 T_2$ is a normal form, then t reduces to normal form. If $T_1 T_2$ is not a normal form, then it contains a single redex, necessarily at the root. Therefore it is balanced at the root, and $T_1 T_2$ is of the form $P[s_1, \ldots, s_n]$, where P is a patch, and the s_i are normal forms which are unbalanced at their roots.

  We now replace each s_i by a new identifier x_i, choosing the identifiers such that $x_i = x_j$ if and only if $s_i = s_j$. The term $P[x_1, \ldots, x_n]$ corresponds to a term in R, which has a normal form. Hence $P[x_1, \ldots, x_n]$ reduces to normal form $Q[x_1, \ldots, x_n]$. Likewise $P[s_1, \ldots, s_n]$ reduces to $Q[s_1, \ldots, s_n]$ which is now a normal form, whether or not P has collapsed. □

4.3 Preservation of SN by currying

In this section we will prove that if a TRS R is strongly normalising, then its currying R^{cur} is strongly normalising as well.

The natural way of proving this seems to be via a minimal counterexample argument based on size. However, the possibility that the minimal counterexample has the form $P[t_1, \ldots, t_n]$ for some patch P can not so easily be refuted.

Let us see what happens. For t_i it holds that they all are SN. So if $P[t_1, \ldots, t_n]$ can perform an infinite reduction, it must be caused by the patch P. But, the patch P is just a curried term from R, which is strongly normalising, even if we take all variables of the patch the same (to ensure that non-left-linear rules applied to P in t can also be applied when the subterms $t_1, \ldots, t_n$ are omitted). Now we would like to conclude that the compound $P[t_1, \ldots, t_n]$ is strongly normalising because all its parts are. Unfortunately this conclusion is flawed: some of the t_i might despite initially being overbalanced terms reduce by collapse steps to balanced terms, thus extending the area of the root patch with fresh symbols. For SN we have to consider all possible reductions of the subterms t_i, not just one normalising reduction as for WN.

The next idea is to modify unbalanced terms of the curried TRS into balanced terms that correspond to terms of the original uncurried TRS. The modification should be such that reduction is preserved: an infinite reduction in the curried TRS can then be modified into an infinite reduction in the original uncurried TRS.

In case of TRSs which do not contain collapsing rules this is straightforward—but note that for these the above proof attempt would work as well. The general situation is harder and requires an additional construction.

4.3.1 The non-collapsing case

The simplest way of balancing an unbalanced term is the following. To the alphabet of the original TRS R we add two new function symbols: a nullary function symbol A and a binary function symbol B. We do not add new rules. Let us denote the new TRS by $R \oplus \{A, B\}$.

LEMMA 4.5 *A TRS R is strongly normalising if and only $R \oplus \{A, B\}$ is.*

PROOF. If R is SN, the extension $R \oplus \{A, B\}$ is also SN by an appeal to Theorem 4.3. □

The added function symbols A and B take care that residuals of patches present in the original term cannot be glued together.

DEFINITION 4.6 Let t be a term of R^{cur}. From this term we construct a term $\alpha(t)$ by *balancing* the subterms occurring at maximal unbalanced positions ($k > 0$ in all following clauses) depending on their form:

- replace $ft_1 \cdots t_{arity(f)+k}$ in t by $B(\cdots B(ft_1 \cdots t_{arity(f)})t_{arity(f)+1} \cdots)t_{arity(f)+k}$,
- replace $ft_1 \cdots t_{arity(f)-k}$ in t by $ft_1 \cdots t_{arity(f)-k} \underbrace{A \cdots A}_{k}$,
- replace $xt_1 \cdots t_k$ in t by $B(\cdots B(xt_1)t_2 \cdots)t_k$.

It is trivial to see that

LEMMA 4.7 *Let R be a TRS. For any t in R^{cur} the term $\alpha(t)$ is balanced.*

In the presence of collapse rules the α-construction does not preserve reduction. Consider for example the following collapse rule:

$$K(x, y) \rightarrow x.$$

Then $K(Kx)CC \rightarrow KxC \rightarrow x$, but $\alpha(K(Kx)CC) = B(K(KxA)C)C \rightarrow B(KxA)C \not\rightarrow x = \alpha(x)$. Without collapse rules the α-construction is easily seen to be reduction preserving, since unbalanced positions can only become balanced by a collapsing reduction.

LEMMA 4.8 *Let R be a TRS without collapsing rules. If $t \rightarrow s$ in R^{cur} then $\alpha(t) \rightarrow \alpha(s)$ in $(R \oplus \{A, B\})^{cur}$.*

PROOF. Observe that when a patch P is contained in a term t, then the patch P is also contained in t^{α}. Observe also that the pattern l of a redex l^{σ} is always contained in a patch. Non-left-linearity is no problem: if $t \rightarrow s$ via a non-left-linear rule, then the subterms substituted at identical variables at different positions in the left hand side of the rule are identical, and this identity is preserved by the α-construction. □

COROLLARY 4.9 *Let R be a TRS without collapsing rules. Then if R is SN, then R^{cur} is SN.*

PROOF. Suppose R^{cur} is not SN, that is, suppose t_0 in R^{cur} has an infinite reduction. Applying the α-construction we obtain an infinite reduction of balanced terms $\alpha(t_0) \rightarrow \alpha(t_1) \rightarrow \ldots$ in $(R \oplus \{A, B\})^{cur}$ by the previous Lemma 4.8. We can reinterpret this infinite reduction as an infinite reduction in $R \oplus \{A, B\}$. Hence $R \oplus \{A, B\}$ is not SN. By Lemma 4.5 R is not SN either. □

In the general case we will apply the α-construction only to terms without overbalanced nodes (so, it would have been sufficient if we had defined α only to balance the underbalanced function symbols). For such terms it holds that reduction can not make underbalanced function symbols balanced. Hence:

LEMMA 4.10 .*Let R be a TRS. Let t be a term of R^{cur} without any overbalanced positions. If $t \rightarrow s$ in R^{cur}, then $\alpha(t) \rightarrow \alpha(s)$ in R^{cur}.* □

4.3.2 The general case

In the presence of collapsing rules it is possible that an underbalanced function symbol becomes balanced such that lemma 4.8 no longer holds. Thus we need to modify this balancing transformation α. The modification is based on two constructions, which we will explain first. First we will define a second balancing operation β, which will remove all overbalanced positions by pushing them down as far as possible. This is a rather rough procedure, in which information can get lost. However, with help of a third construction, essentially a variant of nested multisets, we will define the final modified balancing transformation γ which is suited to prove the preservation of SN by currying.

We will now define a new balancing operation β :Ter(R^{cur}) $\rightarrow$Ter(R^{cur}). Given a term this balancing operation will push the "excess" arguments of each patch down to its unbalanced principal subterms. This mimics the possibility that by a total collapse of the patch the excess arguments are added to an underbalanced principal subterm.

Given a term t of the curried TRS we determine an outermost overbalanced subterm. Such an overbalanced term is of the form $P[t_1, \dots, t_n]s_1 \cdots s_k$ where P[,...,] is a patch, $k > 0$ and $n \geq 0$, or of the form $xs_1 \dots s_k$ where $k > 0$. Then we "push the excess arguments down" by replacing these subterms respectively by $P[t_1 s_1 \cdots s_k, \dots, t_n s_1 \cdots s_k]$ and x. Note that the excess arguments are "pushed over the edge" if the patch does not contain unbalanced subterms, i.e. when $n = 0$. Now repeat this rewrite procedure until no subterms at overbalanced positions are left. This reduction process is both WCR and SN[2], hence CR by Newman's Lemma.

DEFINITION 4.11 On the set of terms Ter(R^{cur}) of a curried TRS we define the rewrite relation $\rightarrow_\beta$: any term of the form $C(P[t_1, \dots, t_n]s) \rightarrow_\beta$-rewrites to $C(P[t_1 s, \dots, t_n s])$, and any term of the form $C(xs) \rightarrow_\beta$-rewrites to $C(x)$, where $C[\,]$ is a context, $P[x_1, \dots, x_n]$ is a patch and $t_1, \dots, t_n, s$ are terms in Ter(R^{cur}). By $\beta(t)$ we denote the unique normal form of the term t under this β-rewriting.

By the previous remarks the outcome, $\beta(t)$, of the "pushing excess terms down" procedure is well defined. The following properties hold.

LEMMA 4.12 *Let* $t, s \in R^{cur}$.

- *if* $t \rightarrow s$ *in* R^{cur} *then* $\beta(t) \rightarrow^* \beta(s)$ *in* $(R \oplus \{A, B\})^{cur}$,
- *if* $t \rightarrow s$ *in* R^{cur} *then* $\alpha(\beta(t)) \rightarrow^* \alpha(\beta(s))$ *in* $(R \oplus \{A, B\})^{cur}$.

PROOF. Pushing down excess terms only copies or extends patches, hence it leaves redexes intact, or removes them entirely. Non-left-linearity is not a problem.

By construction $\beta(t)$ removes all overbalanced nodes. So we can apply α and appeal to 4.10 to find that $\alpha \circ \beta$ preserves the transitive reflexive closure of reduction. □

Note that the actual number of steps in $\beta(t) \rightarrow^* \beta(s)$ may be 0. A simple example of this phenomenon is found in the TRS consisting of the single collapse rule $K(y, z) \rightarrow y$. Here, e.g., we see that $\beta(x(Kyz)) = \beta(xy) = x$. Hence, the β-construction is not strong enough to conclude the preservation of SN by currying. This is where we need the R^{TLT}-construction: we will collect copies of excess arguments in a tree, so that no information gets lost by the pushing down process.

The next construction is a variant of nested multisets (cf. references in [DM79, DJ89]).

[2] Use the well-founded multiset extension of the subterm embedding relation (cf. [DM79, DJ89]): observe that the rewrite procedure replaces an overbalanced principal subterm of the form $P[t_1, \dots, t_n]s$ by various "smaller" unbalanced principal subterms of the form $t_i s$.

DEFINITION 4.13 Given an TRS R, we will define a new rewrite system R^{TLT}. The elements of R^{TLT} are finite trees labelled by terms of R. Let T, S be term-labelled trees. We define that T rewrites to S (notation: $T \rightarrow S$) if S is obtained from T in either of the following ways:

- Delete a proper subtree.
- Choose a node labelled by t such that $t \rightarrow s$. Relabel the node with s, and replace each of the immediate subtrees of the node by any finite number of copies of themselves.

LEMMA 4.14 *If the TRS R is SN, then so is R^{TLT}.*

PROOF. Either one recognises that we have defined a disguised nested multiset order over R, or one observes that this reduction relation on trees is contained in the recursive path order[3] on the trees of R^{TLT}. In any case, if the TRS R we started with is SN, then so is R^{TLT} by classical results (cf. [DM79, DJ89] and [Gal92]). □

We are now ready for the final construction. We wil transform a term t from the curried TRS into a term-labelled tree $\gamma(t)$, in such a way that when $t \rightarrow s$ then $\gamma(t) \rightarrow^+ \gamma(s)$. The transformation of t into $\gamma(t)$ will take some steps: We start from a tree having one node labelled with $\gamma(t)$. At each stage in the transformation we choose a node (it will always be a leaf, by construction) labelled with a term $\gamma(s)$. We reconstruct the tree by adding for each subterm s' of s occurring at an excess position a new node labeled by $\gamma(s')$, at descendant position of the chosen node. Replace the label $\gamma(s)$ by $cur^{-1} \circ \alpha \circ \beta(s)$.

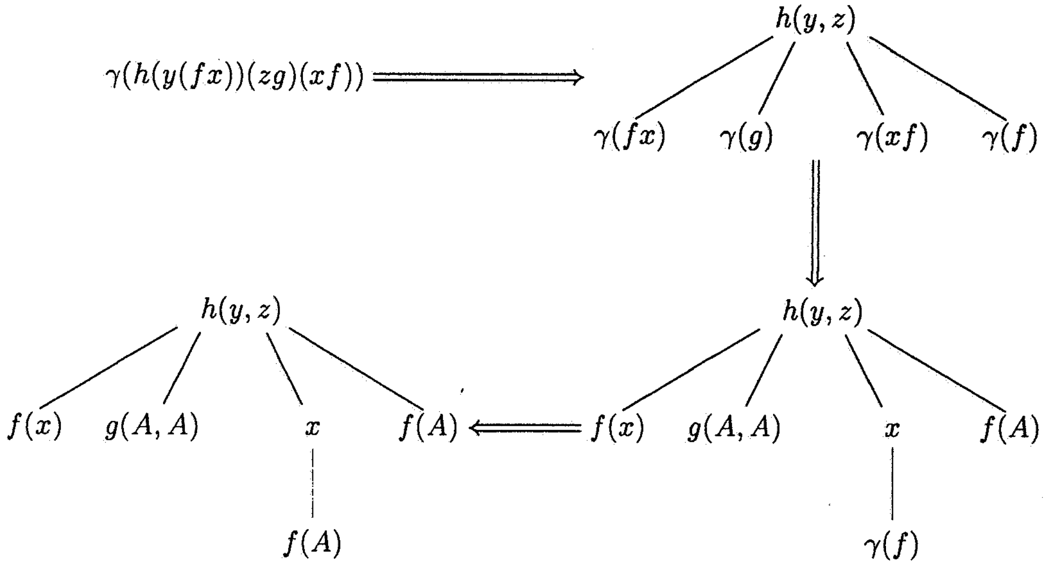
Eventually, by a similar argument as for β, this procedure terminates in a unique result. Call this result $\gamma(t)$.

EXAMPLE 4.15 Consider a signature of a TRS R containing $\{f, g, h\}$ being respectively unary, binary and binary function symbols. We will calculate $\gamma(t)$ where $t = h(y(fx))(zg)(xf)$ in R. There are four excess subterms of t: (fx), g, (xf), and f. $cur^{-1} \circ \alpha \circ \beta(t)$ is the term $h(y, z)$. This gives the first step of the diagram below.

[3] Consider finite trees labelled by terms of some strictly ordered (i.e., transitive, irreflexive order), well-founded set R. A tree T is larger in the recursive path order than S if either of the following holds:

- S can be obtained from T by one of the following rules:
  - replace a proper subtree by any number of smaller trees (including zero),
  - choose a node labelled by t such that t is greater than s in R. Relabel the node with s, and replace each of the immediate subtrees of the node by any finite number of trees smaller than T.
- a proper subtree of T is greater than or equal to S.

This is a "verbal" version of [Gal92]'s definition 11.21; a friendly version of the recursive path order in a slightly different form can be found in [Klo92].



LEMMA 4.16 *Let $t \rightarrow s$ in R^{cur}. Then $\gamma(t) \rightarrow^{+} \gamma(s)$ in $(R \oplus \{A, B\})^{TLT}$.*

PROOF. By induction on the level of occurrence of the redex in the nesting of excess terms (e.g., the redex occurs at level 2 in an excess term which occurs in some other excess term in the original term) we will describe a reduction from $\gamma(t)$ to $\gamma(s)$ in R^{TLT}.

Level 0: The redex is in the root patch of t, and must be contained in the term labelling the top node of the tree. Relabel the top node by reducing the redex there. This takes one step of R^{TLT}. By already proved properties of α and $\beta(t)$, this transforms $cur^{-1} \circ \alpha \circ \beta(t)$ into $cur^{-1} \circ \alpha \circ \beta(s)$, and hence gives the tree the same root label as $\gamma(s)$. In order to transform the rest of the tree into $\gamma(s)$, it is sufficient to duplicate subtrees arising from subterms of t which the redex duplicated, and erase those arising from subterms which were erased. This takes zero or more steps of R^{TLT}.

Level n+1: Locate every node of the excess term at level 1 which contains the redex at level n. There must be at least one. By induction, we can transform each of these subtrees of $\gamma(t)$ as required to obtain $\gamma(s)$. □

The proof of the preservation of SN by currying now follows easily:

THEOREM 4.17 *If the TRS R is SN, then its currying R^{cur} is SN.*

PROOF. Let R^{cur} be not SN. Let $t \in R^{cur}$ have an infinite reduction sequence. Then, by Lemma 4.16, the term $\gamma(t)$ has an infinite reduction sequence in $(R \oplus \{A, B\})^{TLT}$, implying that $(R \oplus \{A, B\})^{TLT}$ is not SN. Therefore $R \oplus \{A, B\}$ is not SN by Lemma 4.14. Hence R is not SN by Lemma 4.5. □

Let us finish this section with a minor application of Theorem 4.17. The reader might have thought about a possibility of rescuing the α-construction by extending $R \oplus \{A, B\}$ with some rules. For each function symbol f in the signature of R we add n rules to $R \oplus \{A, B\}$, where n is the arity of f. E.g., if the arity of f is 3, we add:

$$B(B(B(f(A, A, A), x), y), z) \rightarrow f(x, y, z)$$

$$B(B(f(x, A, A), y), z) \rightarrow f(x, y, z)$$

$$B(f(x,y,A),z) \to f(x,y,z)$$

Let us denote this extension of R by R_{AB}. One can easily verify the following improvement of Lemma 4.8:

LEMMA 4.18 *Let R be a TRS. If $t \to s$ in R^{cur} then $\alpha(t) \to^{+} \alpha(s)$ in $(R_{AB})^{cur}$.* □

It now holds that if R is strongly normalising then R_{AB} is strongly normalising. This fact does not follow straightforwardly from any known modularity result for SN, as these modularity results for SN typically require disjointness of the two systems. It does follow however as a direct corollary of Theorem 4.17 and the next lemma:

LEMMA 4.19 *Let R be some TRS. Then R_{AB} is strongly normalising if and only if R^{cur} is strongly normalising.*

PROOF. Suppose R^{cur} has an infinite reduction $t_0 \to t_1 \to \cdots$. Applying the α-construction and Lemma 4.18 we obtain an infinite derivation $\alpha(t_0) \to^{+} \alpha(t_1) \to^{+} \cdots$ in R_{AB}.

If on the other hand R_{AB} allows for an infinite reduction $t_0 \to t_1 \to \cdots$ then we can curry it into an infinite reduction $t_0^{cur} \to t_1^{cur} \to \cdots$ in R^{cur}, with a slightly modified curry-procedure: we curry A and B by deleting them, e.g., $(B(f(x,y,A),z))^{cur}$ becomes $fxyz$. That this procedure is sound follows from the observations that an infinite reduction in R_{AB} must contain an infinite number of applications of rules from R which are preserved by currying, and that an infinite reduction in R_{AB} does not contain a term consisting of A's and B's only. □

4.4 Preservation of WCR by currying

The preservation of WCR by currying can be proved in a straightforward and easy way:

THEOREM 4.20 *If R is WCR, then R^{cur} is WCR.*

PROOF. Let $t_1 \leftarrow t \to t_2$ be a fork in R^{cur} by respectively reducing redexes R_1 and R_2. Now there are three cases:

- If R_1 and R_2 belong to the same patch, then using WCR for R there exists a term s such that $t_1 \to^{*} s \leftarrow^{*} t_2$ in R^{cur}.
- If R_1 and R_2 belong to different patches (hence the redexes are non-overlapping) and one is below the other. Say R_2 is below R_1. If R is left-linear then there contracting R_1 first and then the copies of R_2 or contracting R_2 first and then R_1 results in the same term. The variant of this argument in case of non-left-linear rules is straightforward.
- If neither of the previous cases hold, let s be the term obtained by contracting both R_1 and R_2. Then clearly $t_1 \to s \leftarrow t_2$. □

4.5 Preservation of completeness by currying

The preservation of completeness is a corollary of the two preceding preservation theorems.

COROLLARY 4.21 *If R is complete, then so is R^{cur}.*

PROOF. A TRS is complete if and only if it is both WCR and SN. By Theorems 4.20 and 4.17, both properties are preserved by Currying, therefore so is completeness. □

5 Properties preserved by currying left-linear term rewriting systems

Throughout this section we will consider the currying R^{cur} of a left-linear TRS R. We will prove that the currying of a left-linear TRS preserves the Church-Rosser property and the Normal Form property as well as two versions of the unique normal form property. A counterexample will demonstrate that these properties may not be preserved for non-left-linear systems. The preservation of semi-completeness follows for left-linear systems from the preservation results of WN and CR. It remains an open question whether semi-completeness is preserved for non-left-linear systems as well.

We start with a recollection of definitions and facts.

5.1 Preliminary facts and definitions

DEFINITION 5.1

- A TRS R satisfies the *normal form property* (NF) if for any term t in R with a reduction to normal form $t \rightarrow^* n$ in R and any reduction $t \rightarrow s$ in R, there exist a reduction $s \rightarrow^* n$ also in R:

$$\begin{tikzcd}
t \arrow[rr, "*"] \arrow[dr] & & n \\
& s \arrow[ur, dotted, "*"'] &
\end{tikzcd}$$

  Note that $t \rightarrow s$ is a single step. The definition would be equivalent if the reduction of t to s were allowed to be of any finite length, but the formulation above will be more technically convenient.

- A TRS R has *unique normal forms* (UN) if convertible normal forms are identical.

- A TRS R has *unique normal forms with respect to reduction* (UN$^{\rightarrow}$) if all normal forms of a term are identical.

$$\begin{tikzcd}
& t \arrow[dl, "*"'] \arrow[dr, "*"] & \\
n_1 \arrow[rr, phantom, "\equiv"] & & n_2
\end{tikzcd}$$

- A TRS is *semi-complete* if it is WN and CR.

DEFINITION 5.2 A TRS (Σ_1, R_1) is a *conservative extension* of a TRS (Σ_2, R_2) if

- $\Sigma_1 \subseteq \Sigma_2$

- $R_1 \subseteq R_2$

- For any $t, s \in$Ter(Σ_1), if $t =_{R_2} s$ then $t =_{R_1} s$. (or just → instead of =)

Our definition of the normal form property differs from the usual one as presented, e.g., in [Klo92]:

LEMMA 5.3 *A TRS R is NF if and only if for all $s, t \in R$ whenever s is a normal form and t is convertible to s, then t is reducible to s.*

PROOF. By induction on the length of the convertibility sequence from s to t. □

We will use the following facts:

LEMMA 5.4 *$CR \Rightarrow NF \Rightarrow UN \Rightarrow UN^{\rightarrow}$ (cf. e.g. [Klo92]).*

THEOREM 5.5 *The disjoint sum of two confluent TRSs is confluent. [Toy87]*

5.2 Preservation of CR by currying for left-linear TRSs

We will show that the Church-Rosser property will be preserved by currying for left-linear TRSs.

As a tool in the proof we will need coloured versions R_{colour} and R^{cur}_{colour} of a given TRS R and its currying R^{cur}. The alphabet of R_{colour} is $\Sigma_{colour} = \Sigma \cup \Sigma \times Colours$, where Σ is the alphabet of R. The rules of R_{colour} are all uniformly coloured versions of rules of R: i.e., for each rule $l \rightarrow r$ of R we now have as many coloured versions as there are colours, in such a way that all function symbols of the patterns of the left and the right hand side of a rule are labelled with the same colour.

Let us denote by R^{cur}_{colour} the subset of $(R_{colour})^{cur}$ consisting of well-coloured terms only: a term is *well-coloured* if only balanced positions are coloured and moreover two positions have the same colour if they belong to the same patch. It is clear that coloured reduction preserves well-colouredness.

Terms in R^{cur}_{colour} can be thought of as terms of R^{cur} provided with a colouring, a partial function from the set of positions of the term to the set of colours. A *patch-colouring* is a colouring of a term such that all balanced function symbols are coloured and function symbols have the same colour if and only if they belong to the same patch. Reductions starting from a patch coloured term will be called *patch-coloured* reductions. Every patch-colouring is a well-colouring, and so a patch-coloured reduction contains only well-coloured terms, but only the starting term need be patch-coloured.

LEMMA 5.6 *Let t be a term of a curried TRS R^{cur}_{colour}. If R has the Church-Rosser property, then all reductions from a given patch-colouring of t are confluent.*

PROOF. Sketch. We use the balancing operation α again. If R is confluent then both $R \oplus \{A, B\}$ and R_{colour} are confluent by Toyama's modularity theorem for confluence. Hence, given a fork in R^{cur}_{colour} we translate it using α into a fork in $R_{colour} \oplus \{A, B\}$. The latter TRS inherits CR from R, and so contains a join of that fork. That join translates back again into a join of the given fork in R^{cur}_{colour}. □

If we now start with a reduction $t_0 \rightarrow^* t_n$ then we will replace it by a sequence of patch colour reductions. We construct such a sequence as follows. Starting from a patch-colouring of t_0 we redo in colour a longest possible initial segment of the given reduction $t_0 \rightarrow^* t_n$. There is a strictly positive number of initial reduction steps which can be lifted to a patch reduction (note that this lifting requires left-linearity). Say, the last term of the initial segment is t_k, for some $k > 0$. Now either $n = k$ or the step $t_k \rightarrow t_{k+1}$ could not be lifted to into a coloured reduction. This can happen when somewhere in the reduction a collapse has taken place such that an initially uncoloured (because unbalanced) function symbol became part of a patch. Now we repeat the process on t_k, and so on, and so on.

Here we need a definition and a lemma:

DEFINITION 5.7 Let t, s be terms of R^{cur}_{colour}. We say that t is *better* coloured than s (notation $t \geq s$)

- if s and t derive from the same uncoloured term,
- every position which is coloured in s is coloured in t,
- whenever two positions in s have the same colouring and belong to the same patch then they have the same colouring in t as well.

LEMMA 5.8 *Let R be a left-linear TRS.*

- *Let t_1, s_1 be terms of R^{cur}_{colour}. If t_1 is better coloured than s_1, then, whenever s_1 contains a redex at some position u, so does t_1 at the same position. Contraction of these redexes results in terms t_2 and s_2, respectively, such that t_2 is again better coloured than s_2.*

$$\begin{tikzcd}
s_1 \arrow[d, "u"] \arrow[r, phantom, "\leq"] & t_1 \arrow[d, "u"] \\
s_2 \arrow[r, phantom, "\leq"] & t_2
\end{tikzcd}$$

- *Let $t_0 \longrightarrow^* t_{n+k}$ be some patch reduction for some $n, k > 0$ in R^{cur}_{colour}. Then $t_n \longrightarrow^* t_{n+k}$ can be recoloured into a patch reduction. The recolouring makes t_n better coloured.*

PROOF.

- Trivial. But note that left-linearity is essential!
- With a fresh patch colouring the intermediate term t_n becomes better coloured. Now apply the previous item. □

We are now equipped to prove the preservation of CR by currying for left-linear TRSs.

THEOREM 5.9 *If a left-linear TRS R is CR, then so is its currying R^{cur}.*

PROOF. The proof of the preservation of CR by currying can be summarised by the diagram in figure 1. In this figure, each tile represents a commuting reduction diagram in the patch-colouring of the term in left-upper corner of the tile.

Let a fork $t_1 \longleftarrow^* t_0 \longrightarrow^* t_2$ in R^{cur} be given. If we lift both reductions to a sequence of patch-colourings (starting with the same patch colouring for t_0) in R^{cur}_{colour}, we obtain the left edge and top row of the figure 1. From left to right and from top to bottom we construct the tiles the figure consists of. Given the left hand side and the upper side of an unfinished tile we complete it using CR of R via lemma 5.6. We construct the left hand side for the next tile on the right from the right hand side of the just found tile, using lemma 5.8. In a similar way we construct a new upper edge for the next tile below from the just constructed bottom edge.

When in the so constructed diagram we finally bleach the colours then the constructed bottom reduction and right hand reduction are the sought for join. Hence R^{cur} has the Church-Rosser property. □

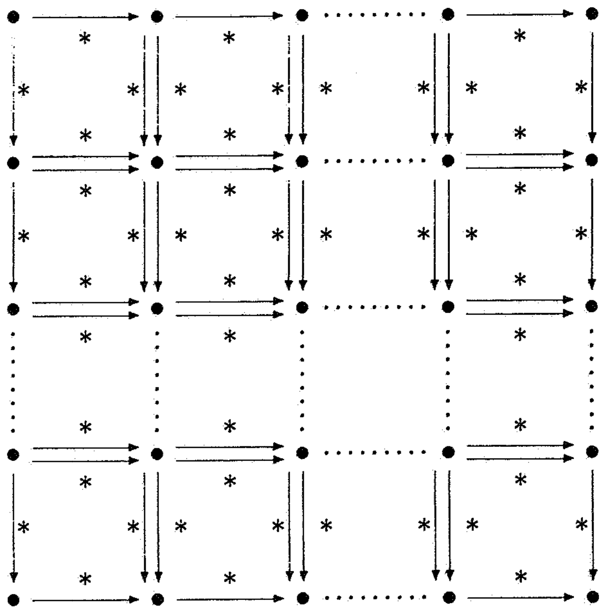


Figure 1:

5.3 Preservation of NF by currying for left-linear TRSs

By a minimal counterexample argument we will prove the preservation of NF by currying for left-linear TRSs.

Before we give this proof we will introduce *root-colouring.* Assume we have some term t in some left-linear curried TRS R^{cur}. Suppose this term has a patch at the root: i.e. suppose the head symbol is a balanced function symbol. We colour t with two colours: one, say *root*, for the root patch P and another colour, say *transparent*, for the remaining function symbols outside P. Let $t \rightarrow^* s$ be some reduction in R^{cur}. Observe first that this reduction is a valid reduction under the proposed colour scheme. One proves this by induction on the length of the reduction. The result is that we have coloured only the steps that take place in the root patch with the colour "root".

Secondly observe using left-linearity that all root-coloured reduction steps can be moved to the front, because whenever $t_1 \rightarrow_{transparent} t_2 \rightarrow_{root} t_3$ then $t_1 \rightarrow_{root} t_4 \rightarrow^*_{transparent} t_3$, since a transparent redex never overlaps with and always is below a root redex. Summarising:

LEMMA 5.10 *Let R be a left-linear TRS. Let t be a term in R^{cur}. Suppose the head symbol of t is a balanced function symbol. Then any reduction $t \rightarrow^* s$ factors into a root-reduction sequence followed by a reduction sequence in which no reductions take place in a root patch descending from the root patch in t.*

THEOREM 5.11 *If a left-linear TRS R is NF, then so is its currying R^{cur}.*

PROOF. Suppose NF holds for R but does not hold for R^{cur}. Then there exists a term t in R^{cur} that contradicts NF and is minimal wrt the number of symbols it consists of. Let us inspect as in the case of WN the four possible forms t can have. In each case we will derive a contradiction.

- t is variable or a function symbol. Then t is either a normal form or a balanced term. Since R is NF, in neither case can t be a counterexample to NF for R^{cur}.

- t is of the form $ft_1 \cdots t_k$ with $0 < k < arity(f)$. The only possible normal forms for t have the form $ft'_1 \cdots t'_k$, where $t'_1 \cdots t'_k$ are normal forms of $t_1 \cdots t_k$. Therefore if t were a counterexample to NF, at least one of the t_i would have to also be a counterexample to NF, contradicting the minimality of t.
- Suppose t is of the form $P[t_1, \ldots, t_k]$, where $P[\cdots]$ is the root patch of t.. Being a counterexample to NF, $P[t_1, \ldots, t_k]$ has a reduction to normal form, say n, and reduces in one step to a term s which cannot reduce to n. The subterms t_i are smaller than t, and therefore satisfy NF. Every variable instance of the patch P is a curried term of R, and therefore satisfies NF.

  Using 5.10 we can factor the reduction of t to n into a root reduction entirely taking place in P followed by a tail containing no reduction steps from P. The last term m of the initial root reduction contains the residue of P, which is either nothing (in case the context P has reduced to one of its holes) or some patch in normal form. In either case the root reduction reduces P to normal form. Consider now the place of the redex in the reduction step to s. If this redex is in P, then using the NF-property of R for the patch P, we can reduce s to this intermediate term m, and then to n. If the redex is in one of the t_i, then in s we copy the reduction of the patch P in $t \to^* m$ to a term m' which differs only in the leaf corresponding to t_i of the residue of P from m. In the reduction $m \to^* n$ this leaf gets reduced to normal form, so we can apply the NF property of t_i, to obtain a reduction from m' to n.
- Suppose that the minimal counterexample t to NF is of the form $P[t_1, \ldots, t_k]s_1 \cdots s_l$ for some patch P and $l > 0$. By hypothesis, t can be reduced to a normal form n, and can also be reduced in a single step to a term s not reducible to n.

  Now there are two possibilities: in the reduction of t to n the root patch P collapses, or it does not. If it collapses, then t can be reduced to a term $t_i s_1 \cdots s_l$. By an argument similar to the previous case, s can also be reduced either to the same term, or (if the step from t to s was performed within either t_i or one of $s_1 \cdots s_l$) to a term s' such that $t_i s_1 \cdots s_l$ reduces to s' in one step. But then $t_i s_1 \cdots s_l$ is a counterexample to the NF property but is smaller than t, contradiction. If P does not collapse, then n must have the form $Qs'_1 \cdots s'_l$, where Q is a normal form of $P[t_1, \ldots, t_k]$ and each s'_i is a normal form of s_i. But then either $P[t_1, \ldots, t_k]$ or one of the s_i must be a counterexample to NF which is smaller than t.

Thus all possible forms are impossible. Hence R^{cur} is NF. □

5.4 Preservation of UN by currying for left-linear TRSs

At first sight it is not clear how the preservation of UN by currying can be proved directly. We can reduce however the preservation of UN to the preservation of CR or NF. The key idea is a construction of Middeldorp (cf. [Mid90]), which given a TRS with the UN-property constructs a conservative extension which is CR. It is rather easy to see that such conservative extensions are preserved by currying.

LEMMA 5.12 *Currying preserves conservative extensions.*

PROOF. Trivial. Let R_2 be a conservative extension of R_1. Suppose $t \to_{R_2^{cur}} s$ for $t, s \in R_1^{cur}$. Consider the patch in which the redex of this reduction gets contracted. In this patch we can find subterms $t', s' \in R_1$ such that $t' \to s'$ in R_2. Hence also in R_1. Thus we see that $t \to_{R_1^{cur}} s$.

Thus R_2^{cur} is a conservative extension of R_1^{cur}. □

Now instead of verifying that Middeldorp's construction can be refined so that when presented with a left-linear TRS it produces another left-linear TRS, we will consider a simpler variant: a construction which extends a left-linear TRS which is UN into a left-linear TRS which is NF.

This extension comes down to: for each equivalence class with respect to conversion which contains a necessarily unique normal form n add rules of the form $t \to n$ for each t in the equivalence class.

In [Mid90] Middeldorp discusses this method, and rejects it as being too weak to result in a confluent TRS. It is however strong enough to give conservative extensions which are NF, as we will show now.

LEMMA 5.13 *Any left-linear TRS which is UN can be conservatively extended to a left-linear TRS which is NF.*

PROOF. Let the left-linear TRS R be given. Suppose it is UN. We extend it to a TRS R^{NF} by adding some extra rules, as follows. Consider the equivalence classes of convertible terms containing a (necessarily unique by UN) normal form. For each linear term t in such a class $[n]$, where n is the normal form, add the rule $t \to n$ provided t is not identical to n.

First we have to show that the added rules are well formed rules: the free variables of the left hand side of a rule must include the free variables of the right hand side. If n contained a variable x not occurring in t, then in the conversion of t to n one could replace all occurrences of x by any other variable y not occurring in the conversion sequence, obtaining a conversion of t to a normal form n' distinct from n. But then n and n' would be distinct but convertible normal forms, contraducting the hypothesis of UN.

The next step is the observation that R^{NF} is indeed a conservative extension of R with respect to convertibility. This follows trivially from the construction.

Finally we observe that if R is UN then this extension R^{NF} is also NF. This is easy as well: if $t \to^* n$ in R^{NF} then clearly $t \in [n]$. If for some s we have $t \to s$ in R^{NF}, then $s \in [t] = [n]$. Now we would like to conclude that $s \to n$ is an instance of a rule we have added to R. This is trivial if s is linear, but in general requires the observation that for any left-linear TRS R, whenever $t_1 \to t_2$ and t_1 is non-linear, then are terms s_1 and s_2 and a substitution σ such that s_1 is linear, $s_1 \to s_2$ and $s_i^\sigma = t_i$ for $i = 1, 2$. □

THEOREM 5.14 *If a left-linear TRS R is UN, then so is its currying R^{cur}.*

PROOF. Let the TRS R be UN and left-linear. By Lemma 5.13 we can construct a conservative extension R^{NF} of R which is left-linear and NF. By Theorem 5.11 we see that the currying of R^{NF} is also NF. Now suppose we have two convertible normal forms in R^{cur}. These are convertible normal forms in $(R^{NF})^{cur}$ as well. By the NF-property of $(R^{NF})^{cur}$ we obtain a reduction sequence from one to the other. Since the extension of R^{NF} by $(R^{NF})^{cur}$ is conservative, the reduction takes place inside is valid in R^{cur} as well. Being a reduction between normal forms it has length 0, implying that the two normal forms are identical. Thus R^{cur} is UN. □

5.5 Preservation of $\mathrm{UN}^{\rightarrow}$ by currying for left-linear TRSs

By proving that a minimal counterexample (i.e., minimal in number of symbols) does not exist we show that the property $\mathrm{UN}^{\rightarrow}$ is preserved if we curry a left-linear TRS satisfying it. This proof is based on the observation that such a minimal counter example must contain an innermost overbalanced subterm. We will show this first.

LEMMA 5.15 *Let R be a left-linear TRS satisfying $UN^{\rightarrow}$. Let t be a minimal counterexample that shows that R^{cur} does not satisfy $UN^{\rightarrow}$. Then t contains an innermost overbalanced subterm. This overbalanced subterm can reduce to a balanced term.*

PROOF. The term t must contain an unbalanced subterm s. Otherwise it is balanced, in which case it is the currying of a term in R, and hence must have the $\mathrm{UN}^{\rightarrow}$ property. So, let $t = C[s]$. Inspecting the two given reductions to different normal forms of t, we note that in one of these reductions a descendant of the root of s has to become part of a balanced patch in a derivative of $C[\,]$. (If not, with an appeal to left-linearity, we can factor the reductions into a reduction of $C[\,]$ into normal form $D[\,]$ and a reduction of s into normal form r. Both $C[\,]$ and s are smaller than t, hence their normal forms are unique. So that the resulting compound $D[r]$ is normal form of both reductions, contradiction.) But this can happen only if s is overbalanced, as reduction preserves underbalancedness of a headsymbol.

So t contains an overbalanced subterm. Hence, t being a finite term, there is an innermost overbalanced subterm in t. It can reduce to a balanced term. □

THEOREM 5.16 *If a left-linear TRS R is $UN^{\rightarrow}$, then so is its currying R^{cur}.*

PROOF. Let R be a TRS satisfying $\mathrm{UN}^{\rightarrow}$. Suppose R^{cur} is not $\mathrm{UN}^{\rightarrow}$. Then there is a minimal counterexample t with an innermost overbalanced subterm s of t. So, s is of the form $P[t_1, \ldots, t_k]s_1 \ldots s_m$ for some patch P and terms t_i, s_j in R^{cur}. s can become balanced, so $P[t_1, \ldots, t_k]$ must be able to collapse to one uniquely determined subterm, say t_i.

We now claim that if $t = C[P[t_1, \ldots, t_k]s_1 \ldots s_m]$ can reduce to a normal form n then, the smaller term $t' = C[t_i s_1 \ldots s_m]$ reduces to n as well. From this claim follows the existence of a smaller counterexample, hence a contradiction to the existence of a minimal counterexample to $UN^{\rightarrow}$.

The claim follows from the fate of derivatives of P: the normal form n does not contain any positions descending from a position of the original patch P. So either such a derivative just gets reduced to another derative, or it is removed by some action above it, or it collapses to a derivative of t_i. □

5.6 Preservation of semi-completeness by currying for left-linear TRSs

Preservation of semi-completeness has become an easy corollary of previous results:

COROLLARY 5.17 *If a left-linear TRS is semi-complete, then so is its currying.*

PROOF. A TRS is semi-complete if and only if it is WN and CR. Both these properties are preserved by currying for left-linear systems, therefore so is semi-completeness. □

In contrast to the properties CR, NF, UN and $\mathrm{UN}^{\rightarrow}$, it is open whether this result holds for all non-left-linear systems.

6 The counter example $CL^{fun} \oplus M$ for non-left-linearity

For non-left-linear systems, currying does not always preserve the properties CR, NF, UN or UN$^{\rightarrow}$. We will show this by exhibiting a TRS which satisfies CR (and hence also NF, UN and UN$^{\rightarrow}$), but whose currying does not satisfy UN$^{\rightarrow}$ (and hence not UN, NF nor CR).

The counterexample will be derived from an extension of combinatory logic with a functional TRS M such that $CL \oplus M$ is CR and $CL + M^{cur}$ is not UN$^{\rightarrow}$ (cf. [Klo80, Klo92]). This would be easy if CL could be interpreted as the currying of some functional TRS. However, the right hand side of the rule $Sxyz \rightarrow xz(yz)$ makes this impossible. The next best we can do is to construct a functional TRS CL^{fun} whose currying contains an isomorphic image of CL. For this TRS we will prove:

- $CL^{fun} \oplus M$ is CR (hence not NF, UN and UN$^{\rightarrow}$),
- $CL + M^{cur}$ is not UN$^{\rightarrow}$.

Then we will conclude that $(CL^{fun} \oplus M)^{cur}$ is not UN$^{\rightarrow}$ (hence not UN, NF or CR).

First some preliminaries.

DEFINITION 6.1 A TRS R is an *embedding* of a TRS S if there exists a injective function ι :Ter(R) $\rightarrow$Ter(S) such that $t \rightarrow s$ in R if and only if $\iota(t) \rightarrow \iota(s)$ in S for all t, s in R.

The counterexample will be derived from the extension of CL with the following TRS M used by Huet [Hue80] and Breazu-Tannen [BT88]:

$$M = \begin{cases} M(x,x) & \rightarrow & 0 \\ M(Succ(x),x) & \rightarrow & 1 \end{cases}.$$

Its currying is

$$M^{cur} = \begin{cases} Mxx & \rightarrow & 0 \\ M(Succ\,x)x & \rightarrow & 1 \end{cases}.$$

First we note that M and M^{cur} have in combination with CL different properties.

LEMMA 6.2

1. $CL \oplus M$ is CR (hence NF, UN and UN$^{\rightarrow}$),

2. $CL + M^{cur}$ is not UN$^{\rightarrow}$ (hence not UN, NF nor CR).

PROOF.

1. Combinatory logic is CR (cf. e.g. [Bar84]). It is easy to see that M is both SN and WCR, hence by Newmans Lemma 4.2 we find that M is CR. Toyama's modularity theorem 5.5 for confluent TRSs implies that $CL \oplus M$ is CR.

2. Using the fixed point combinator[4] $Yx \rightarrow x(Yx)$ definable in CL we see that the term $M(Y\,Succ)(Y\,Succ)$ has two different normal forms:

$$M(Y\,Succ)(Y\,Succ) \rightarrow 0$$

[4] Define $B = S(KS)K$, $E = B(SI)(SII)$ and $Y = EE$. Then $Bxyz \rightarrow^* x(yz)$ and $Yx \rightarrow^* x(Yx)$.

$$M(Y\,Succ)(Y\,Succ) \to^* M(Succ(Y\,Succ))(Y\,Succ) \to 1$$

Hence $CL + M^{cur}$ is not UN$^{\to}$. □

We now construct a functional TRS whose currying contains CL. To explain the construction, let us ignore the formalities for a while.

Consider the following reduction in CL:

$$Ix \to x$$

Clearly this rule could be the currying of the following functional reduction:

$$I(x) \to x$$

However, the I-rule has instances such as:

$$Ixy \to xy$$

and

$$I(xy)z \to xyz$$

Their uncurried forms (ignoring problematic notation) would be:

$$I(x,y) \to x(y)$$

and

$$I(x(y),z) \to x(y,x)$$

Clearly, this attempt to uncurry CL goes wrong, in two ways. The function symbol I occurs with different arities, and the variables of the functional TRS are treated as if they have variable arity.

We correct these defects by introducing for each arity $k \geq 0$ a special symbol I_k, and for each variable x a set of function symbols $\{X_i \mid i \geq 0\}$, where the index indicates the arity. We now transcribe the uncurried forms as:

$$I_1(x) \to x$$

$$I_2(X_0, y) \to X_1(y)$$

$$I_2(X_1(y), z) \to X_2(y, z)$$

By transforming the rules for S and K in an analogous way we are led to the following definition of CL^{fun}.

Definition 6.3 The alphabet of CL^{fun} consists of a set of variables, for each variable x a set of function symbols $\{X_k \mid k \geq 0\}$ and for each $F \in \{S, I, K\}$ a set of function symbols $\{F_k \mid k \geq 0\}$. The indices represent the arities. The rules of CL^{fun} are:

$$\begin{array}{l}
S_{n+3}(\alpha_m(x_1,\ldots,x_m), \beta_k(y_1,\ldots,y_k), z, u_1,\ldots,u_n) \;\to \\
\qquad\qquad \alpha_{n+m+2}(x_1,\ldots,x_m, z, \beta_{k+1}(y_1,\ldots,y_k,z), u_1,\ldots,u_n) \\
K_{n+2}(\alpha_m(x_1,\ldots,x_m), \beta_k(y_1,\ldots,y_k), u_1,\ldots,u_n) \;\to\; \alpha_{m+n}(x_1,\ldots,x_m,u_1,\ldots,u_n) \\
I_{n+1}(\alpha_m(x_1,\ldots,x_m), u_1,\ldots,u_n) \;\to\; \alpha_{m+n}(x_1,\ldots,x_m,u_1,\ldots,u_n)
\end{array}$$

where α, $\beta \in \{S, K, I, X, Y, \ldots\}$ and $n, m, k \geq 0$.

This is the right concept: CL is embedded in the currying of CL^{fun}, as we will show now. We need some definitions. One *caveat,* in order to make the following definition work we take as equivalent definition of Ter(CL) the following BNF definition:

$$b ::= S \mid K \mid I \mid x$$

$$t ::= (b) \mid (bt_1) \mid (bt_1t_2) \ldots$$

This means that as standard notation we put brackets around maximal subterms of the form $bt_1 \cdots t_n$: e.g. $(Sx(xyz)(Kx)z)$.

DEFINITION 6.4

- By simultaneous induction we define ι_{min}, ι_{max} :Ter(CL) $\rightarrow$Ter(CL^{fun}).
  - $\iota_{min}(x) = x$, for x a variable
  - $\iota_{max}(x) = X$, for x a variable
  - $\iota_{max}(F) = \iota_{min}(F) = F$, for $F \in \{S, K, I\}$,
  - $\iota_{min}((ft_1 \cdots t_n)) = \iota_{max}(f)_n(\iota_{min}(t_1) \ldots \iota_{min}(t_n))$
  - $\iota_{max}((ft_1 \cdots t_n)) = \iota_{max}(f)_n(\iota_{max}(t_1) \ldots \iota_{max}(t_n))$
- By induction we define U :Ter(CL^{fun}) $\rightarrow CL$:
  - $U(x) = x$, for x a variable,
  - $U(F_n(t_1, \ldots, t_n)) = (u(F)U(t_1) \cdots U(t_n))$, for $F \in \{S, K, I, X, \ldots\}$

  where

  - $u(X_n) = x$, for X_n the n-ary "variable-function symbol" corresponding to x
  - $u(F_n) = F$, for $F \in \{S, K, I\}$

Let us give some examples:

$$\iota_{min}((xy(Sxy)) = X_2(y, S_2(x, y))$$

$$\iota_{max}((xy(Sxy)) = X_2(Y_0, S_2(X_0, Y_0))$$

$$U(X_2(Y_0, S_2(x, y))) = U(X_2(Y_0, S_2(X_0, Y_0))) = (xy(Sxy))$$

It will be clear that these translations preserve patterns of redexes, i.e., in particular they preserve reduction steps.

LEMMA 6.5

1. $t \rightarrow s$ *in* $CL \Leftrightarrow \iota_{min}(t) \rightarrow \iota_{min}(s)$ *in* $CL^{fun} \Leftrightarrow \iota_{max}(t) \rightarrow \iota_{max}(s)$ *in* CL^{fun} *for all* $t, s \in$*Ter*(CL),

2. $U(t) \rightarrow U(s) \Leftrightarrow t \rightarrow s$ *for all* $t, s \in$*Ter*(CL^{fun}) □

It now follows immediately that:

LEMMA 6.6 *$\iota_{max} : CL \longrightarrow (CL^{fun})^{cur}$ is an embedding.* □

Since CL^{fun} is not orthogonal we have to work a little to prove the CR property. However, via the above translations we can derive it from the CR property of CL itself.

LEMMA 6.7 *CL^{fun} is CR.*

PROOF. Consider a fork $t_1 \twoheadleftarrow^* t \twoheadrightarrow^* t_2$ in CL^{fun}. Applying U we get a fork $U(t_1) \leftarrow^* U(t) \rightarrow^* U(t_2)$ in CL. By confluence of CL there exists a join $U(t_1) \rightarrow^* s \leftarrow^* U(t_2)$ in CL. Applying ι_{min} we obtain the join $\iota_{min}(U(t_1)) \rightarrow^* \iota_{min}(s) \leftarrow^* \iota_{min}(U(t_2))$ in CL^{fun}. But $t = (\iota_{min}(U(t)))^\sigma$ for some substitution σ which replaces some variables $x, y, \ldots$ by $X_0, Y_0, \ldots$. Hence $t_1 = (\iota_{min}(U(t_1)))^\sigma \leftarrow^* (\iota_{min}(s))^\sigma \rightarrow^* (\iota_{min}(U(t_2)))^\sigma = t_2$. □

LEMMA 6.8 *$CL^{fun} \oplus M$ is CR.*

PROOF. CL^{fun} and M are CR and disjoint. Hence we can apply Toyama's modularity theorem 5.5 for CR. □

LEMMA 6.9 *$(CL \oplus M)^{cur}$ is not UN$^{\rightarrow}$.*

PROOF. $CL + M^{cur}$ is an embedding of $(CL^{fun})^{cur} + M^{cur} = (CL^{fun} \oplus M)^{cur}$ via ι_{max}. Since $CL + M^{cur}$ is not UN$^{\rightarrow}$ via Lemma 6.2, it follows that $(CL^{fun} \oplus M)^{cur}$ is not UN$^{\rightarrow}$.

□

These last two lemmas imply that $CL^{fun} \oplus M$ is the sought for counterexample.

7 Acknowledgements

We would like to acknowledge the friendly help and criticism of Vincent van Oostrom, Aart Middeldorp, Femke van Raamsdonk, Yoshihito Toyama and Hans Zantema. We used Paul Taylors commutative diagram package.

References

[Bar84] H.P. Barendregt. *The Lambda Calculus, its Syntax and Semantics.* North-Holland, 2nd edition, 1984.

[BT88] V. Breazu-Tannen. Combining algebra and higher-order types. In *Third Annual Symposium on Logic in Computer Science*, pages 82–90, Edinburgh, 1988. IEEE.

[CF58] H.B. Curry and R. Feys. *Combinatory Logic*, volume I. North-Holland, 1958.

[DJ89] N. Dershowitz and J.-P. Jouannaud. Rewrite systems. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, chapter 15. North-Holland, 1989.

[DM79] N. Dershowitz and Z. Manna. Proving termination with multiset orderings. *Comm. AM*, 22(8):465–476, 1979.

[FH88] A.J. Field and P.G. Harrison. *Functional programming.* Addison-Wessley, 1988.

[Gal92] J.H. 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, 1992.

[Hue80] G. Huet. Confluent reductions: Abstract properties and applications to term rewriting systems. *JACM*, 27(4):797–821, 1980.

[KKSdV93] J.R. Kennaway, J.W. Klop, M.R. Sleep, and F.J. de Vries. Comparing curried and uncurried rewriting. In H.P. Barendregt, M. Bezem, and J.W. Klop, editors, *Dirk van Dalen Festschrift*, volume V of *Quaestiones Infinitae*. Department of Philosophy, Utrecht University, 1993.

[Klo80] J.W. Klop. *Combinatory Reduction Systems.* Mathematical Centre Tracts Nr. 127. CWI, Amsterdam, 1980. PhD Thesis.

[Klo92] J.W. Klop. Term rewriting systems. In S. Abramsky, D. Gabbay, and T. Maibaum, editors, *Handbook of Logic in Computer Science, Volume II.* Oxford University Press, 1992.

[Mid90] A. Middeldorp. *Modular Properties of Term Rewriting Systems.* PhD thesis, Vrije Universiteit, Amsterdam, November 1990.

[Toy87] Y. Toyama. On the Church-Rosser property for the direct sum of term rewriting systems. *JACM*, 34(1):128–143, 1987.