Residuals Revisited[*]

Christina Kohl and Aart Middeldorp

Department of Computer Science, University of Innsbruck, Austria
christina.kohl@uibk.ac.at,aart.middeldorp@uibk.ac.at

Abstract

Proof terms are a useful concept for comparing computations in term rewriting. The residual operation is an important operation on proof terms, used to define projection equivalence. We present a variant of the residual system (Definition 8.7.54 of TeReSe) that is (innermost) confluent and terminating, and thus can be used to decide projection equivalence.

1 Introduction

To reason about rewrite sequences in left-linear term rewrite systems, in [3, Chapter 8] and [4] de Vrijer and van Oostrom define and compare different notions of equivalence. In this paper we are concerned with one of these notions, projection equivalence, which is defined using residuals. We present a schematic rewrite system for computing residuals that operates on proof terms. The latter are used to represent rewrite sequences. Our rewrite system is a variant of the residual system defined in [3, Definition 8.7.54 and proof of Theorem 8.7.57] and [4, Definition 6.9 and proof of Theorem 6.12]. We identify several issues with the analysis in [3, 4] and propose a solution by imposing an evaluation strategy on the residual system. We establish (innermost) confluence and termination of the adapted system, and show how these properties are used to decide projection equivalence. The decision procedure is incorporated into ProTeM, a recent tool [1] for manipulating proof terms.

2 Proof Terms

Proof terms are built from function symbols, variables, and rule symbols as well as the binary *composition* operator ; which is used in infix notation. Rule symbols represent rewrite rules and have a fixed arity which is the number of different variables in the represented rule. We use Greek letters $(\alpha, \beta, \gamma, \dots)$ as rule symbols, and uppercase letters $(A, B, C, \dots)$ for proof terms.

If α is a rule symbol then lhs_α (rhs_α) denotes the left-hand (right-hand) side of the rewrite rule represented by α. Furthermore var_α denotes the list $(x_1, \dots, x_n)$ of variables appearing in α in some fixed order. The length of this list is the arity of α. Given a rule symbol α with $\mathsf{var}_\alpha = (x_1, \dots, x_n)$ and proof terms $A_1, \dots, A_n$, we write $\langle A_1, \dots, A_n \rangle_\alpha$ for the substitution $\{x_i \mapsto A_i \mid 1 \leqslant i \leqslant n\}$. A proof term A witnesses a rewrite sequence from its source $\mathsf{src}(A)$ to its target $\mathsf{tgt}(A)$, which are computed as follows:

$$\begin{aligned}
\mathsf{src}(x) &= \mathsf{tgt}(x) = x & \mathsf{src}(f(A_1,\dots,A_n)) &= f(\mathsf{src}(A_1),\dots,\mathsf{src}(A_n))\\
\mathsf{src}(A\,;B) &= \mathsf{src}(A) & \mathsf{src}(\alpha(A_1,\dots,A_n)) &= \mathsf{lhs}_\alpha\langle \mathsf{src}(A_1),\dots,\mathsf{src}(A_n)\rangle_\alpha\\
\mathsf{tgt}(A\,;B) &= \mathsf{tgt}(B) & \mathsf{tgt}(f(A_1,\dots,A_n)) &= f(\mathsf{tgt}(A_1),\dots,\mathsf{tgt}(A_n))\\
& & \mathsf{tgt}(\alpha(A_1,\dots,A_n)) &= \mathsf{rhs}_\alpha\langle \mathsf{tgt}(A_1),\dots,\mathsf{tgt}(A_n)\rangle_\alpha
\end{aligned}$$

[*]This research is supported by FWF (Austrian Science Fund) project P27528. An extended version of this paper will appear in the proceedings of CADE-27 [2].

Here f is an n-ary function symbol. We assume $\mathsf{tgt}(A) = \mathsf{src}(B)$ whenever the composition $A\,;B$ is used in a proof term. Proof terms A and B are *co-initial* if they have the same source.

Example 1. *Consider the TRS consisting of the rules* α, β, γ *and the proof terms* A, B, C*:*

$$\begin{array}{ccc} \alpha\colon\ \mathsf{f}(\mathsf{a},x) \to \mathsf{g}(x,x) & \beta\colon\ \mathsf{a} \to \mathsf{b} & \gamma\colon\ \mathsf{b} \to \mathsf{c} \\ A = \alpha(\beta\,;\gamma) & B = (\alpha(\mathsf{a})\,;\mathsf{g}(\beta,\beta))\,;\mathsf{g}(\gamma,\gamma) & C = \mathsf{f}(\mathsf{a},\beta\,;\gamma)\,;\alpha(\mathsf{c}) \end{array}$$

We have $\mathsf{src}(A) = \mathsf{src}(B) = \mathsf{src}(C) = \mathsf{f}(\mathsf{a},\mathsf{a})$ *and* $\mathsf{tgt}(A) = \mathsf{tgt}(B) = \mathsf{tgt}(C) = \mathsf{g}(\mathsf{c},\mathsf{c})$*. The proof term* B *represents the sequence* $\mathsf{f}(\mathsf{a},\mathsf{a}) \to \mathsf{g}(\mathsf{a},\mathsf{a}) \mathrel{+\!\!\!\!\!\twoheadrightarrow} \mathsf{g}(\mathsf{b},\mathsf{b}) \mathrel{+\!\!\!\!\!\twoheadrightarrow} \mathsf{g}(\mathsf{c},\mathsf{c})$*.*

We can represent any rewrite sequence $\longrightarrow^*$ by a suitable proof term. A proof term without composition represents a multi-step, a proof term without composition and nested rule symbols represents a parallel step, and a proof term without composition and only one rule symbol represents a single step. If a proof term contains neither compositions nor rule symbols, it denotes an empty step.

3 Residuals

The residual operation computes, for co-initial proof terms A and B, which steps of A remain after performing B. The diagram on the left shows a desirable result of residuals and the diagram on the right provides the intuition behind equations (6) and (7) below:

$$\begin{tikzcd} \cdot \arrow[r, two heads, "A"] \arrow[d, two heads, "B"'] & \cdot \arrow[d, two heads, "B/A"] \\ \cdot \arrow[r, two heads, "A/B"'] & \cdot \end{tikzcd} \qquad \begin{tikzcd} \cdot \arrow[r, two heads, "A"] \arrow[d, two heads, "C"'] & \cdot \arrow[r, two heads, "B"] \arrow[d, two heads, "C/A"] & \cdot \arrow[d, two heads, "(C/A)/B"] \\ \cdot \arrow[r, two heads, "A/C"'] & \cdot \arrow[r, two heads, "B/(C/A)"'] & \cdot \end{tikzcd}$$

In [3, Definition 8.7.54] and [4, Definition 6.9] the residual $A\,/\,B$ is defined by means of the following equations:

$$\begin{aligned} x\,/\,x &= x && (1)\\ f(A_1,\ldots,A_n)\,/\,f(B_1,\ldots,B_n) &= f(A_1\,/\,B_1,\ldots,A_n\,/\,B_n) && (2)\\ \alpha(A_1,\ldots,A_n)\,/\,\alpha(B_1,\ldots,B_n) &= \mathsf{rhs}_\alpha\langle A_1\,/\,B_1,\ldots,A_n\,/\,B_n\rangle_\alpha && (3)\\ \alpha(A_1,\ldots,A_n)\,/\,\mathsf{lhs}_\alpha\langle B_1,\ldots,B_n\rangle_\alpha &= \alpha(A_1\,/\,B_1,\ldots,A_n\,/\,B_n) && (4)\\ \mathsf{lhs}_\alpha\langle A_1,\ldots,A_n\rangle_\alpha\,/\,\alpha(B_1,\ldots,B_n) &= \mathsf{rhs}_\alpha\langle A_1\,/\,B_1,\ldots,A_n\,/\,B_n\rangle_\alpha && (5)\\ C\,/\,(A\,;B) &= (C\,/\,A)\,/\,B && (6)\\ (A\,;B)\,/\,C &= (A\,/\,C)\,;(B\,/\,(C\,/\,A)) && (7)\\ A\,/\,B &= \#(\mathsf{tgt}(B))^{1} && (\text{otherwise}) \end{aligned}$$

Here A, B, C, $A_1,\ldots,A_n$, $B_1,\ldots,B_n$ are proof term *variables* that can be instantiated with arbitrary proof terms (so without $/$). The x in equation (1) denotes an *arbitrary* variable (in the underlying TRS), which cannot be instantiated.[2] For every rule α of the underlying TRS, the equation schemes (3)–(5) are suitably instantiated. For instance, for rule α of Example 1

[1]In [3, 4] the wrong definition $A\,/\,B = \#(\mathsf{tgt}(A))$ is given.
[2]In [3, Remark 8.2.21] variables are treated as constants and (1) is absent.

we obtain the equations $\alpha(A) / \alpha(B) = \mathsf{g}(A / B, A / B)$, $\alpha(A) / \mathsf{f}(\mathsf{a}, B) = \alpha(A / B)$ and $\mathsf{f}(\mathsf{a}, A) / \alpha(B) = \mathsf{g}(A / B, A / B)$. In the final defining equation, # is the rule symbol of the special *error rule* $x \to \bot$. This rule is adopted to ensure that A / B is defined for arbitrary left-linear TRSs. The defining equations are taken modulo

$$t \,;\, t \approx t \tag{8}$$

$$f(A_1, \ldots, A_n) \,;\, f(B_1, \ldots, B_n) \approx f(A_1 \,;\, B_1, \ldots, A_n \,;\, B_n) \tag{9}$$

The need for the so-called functorial identities (9) is explained in the following example (Vincent van Oostrom, personal communication).

Example 2. *Consider $A = \mathsf{f}(\mathsf{g}(\beta) \,;\, \mathsf{g}(\gamma))$ and $B = \alpha(\mathsf{a})$ in the TRS*

$$\alpha\colon\ \mathsf{f}(\mathsf{g}(x)) \to x \qquad\qquad \beta\colon\ \mathsf{a} \to \mathsf{b} \qquad\qquad \gamma\colon\ \mathsf{b} \to \mathsf{c}$$

When computing A / B without (9), the α-instance $\mathsf{f}(\mathsf{g}(A_1)) / \alpha(B_1) = A_1 / B_1$ of schema (4) does not apply to A / B since the g *in $\mathsf{f}(\mathsf{g}(A_1))$ needs to be extracted from $\mathsf{g}(\alpha) \,;\, \mathsf{g}(\gamma)$ when computing A / B. As a consequence, the (otherwise) equation kicks in, producing the proof term $\#(\mathsf{b})$ that indicates an error. With (9) in place, the result of evaluating A / B is the proof term $\beta \,;\, \gamma$, representing the desired sequence $\mathsf{a} \to \mathsf{b} \to \mathsf{c}$.*

It is not immediately clear that the defining equations on the preceding page constitute a well-defined definition of the residual operation. In [3, proof of Theorem 8.7.57] and [4, proof of Theorem 6.12] the defining equations together with (8) and (9) are oriented from left to right, resulting in a rewrite system $\mathcal{R}es$ that is claimed to be terminating and confluent. The residual of A over B is then defined as the unique normal form of A / B in $\mathcal{R}es$.

There are two problems with this approach. First of all, when is the (otherwise) rule applied? In [3] this is not specified, resulting in an imprecise rewrite semantics of $\mathcal{R}es$. Keeping in mind that A / B is supposed to be a total operation on *proof terms* (so no / in A and B), a natural solution is to adopt an innermost evaluation strategy. This ensures that C / A is evaluated before $(C / A) / B$ in the right-hand side of (6) and before $B / (C / A)$ in the right-hand side of (7). The (otherwise) condition is taken into account by imposing the additional restriction that the (otherwise) rule is applied to A / B (with A and B in normal form) only if the other rules are not applicable. The second, and more serious, problem is that $\mathcal{R}es$ is *not* confluent.

Example 3. *Consider the TRS consisting of the rules*

$$\alpha\colon\ \mathsf{f}(x, y) \to \mathsf{f}(y, x) \qquad\qquad \beta\colon\ \mathsf{a} \to \mathsf{b} \qquad\qquad \gamma\colon\ \mathsf{f}(\mathsf{a}, x) \to x$$

and the proof terms $A = f(\beta, \mathsf{a})$, $B = \alpha(\mathsf{b}, \beta)$, $C = \alpha(\mathsf{a}, \mathsf{a})$, and $D = \gamma(\mathsf{a})$. There are two ways to compute $(A \,;\, B) / (C \,;\, D)$, starting with (6) or (7):

$$\begin{aligned}
((A \,;\, B) / C) / D \;&\to\; ((A / C) \,;\, (B / (C / A))) / D \\
&\to^* (\mathsf{f}(\mathsf{a} / \mathsf{a}, \beta / \mathsf{a}) \,;\, (B / \alpha(\mathsf{a} / \beta, \mathsf{a} / \mathsf{a}))) / D \\
&\to^* (\mathsf{f}(\mathsf{a}, \beta) \,;\, (B / \alpha(\mathsf{b}, \mathsf{a}))) / D \\
&\to\; (\mathsf{f}(\mathsf{a}, \beta) \,;\, \mathsf{f}(\beta / \mathsf{a}, \mathsf{b} / \mathsf{b})) / D \\
&\to^* (\mathsf{f}(\mathsf{a}, \beta) \,;\, \mathsf{f}(\beta, \mathsf{b})) / D \\
&\to\; \mathsf{f}(\mathsf{a} \,;\, \beta, \beta \,;\, \mathsf{b}) / D \to \#(\mathsf{a})
\end{aligned}$$

$$(A / (C \,;\, D)) \,;\, (B / ((C \,;\, D) / A))$$

$$
\begin{aligned}
&\to^* ((A \,/\, C) \,/\, D) \,;\, (B \,/\, ((C \,/\, A) \,;\, (D \,/\, (A \,/\, C)))) \\
&\to^* (\mathsf{f}(\mathsf{a}, \beta) \,/\, D) \,;\, (B \,/\, (\alpha(\mathsf{b}, \mathsf{a}) \,;\, (D \,/\, \mathsf{f}(\mathsf{a}, \beta)))) \\
&\to^* \beta \,;\, (B \,/\, (\alpha(\mathsf{b}, \mathsf{a}) \,;\, \gamma(\mathsf{b}))) \\
&\to^* \beta \,;\, (\mathsf{f}(\beta, \mathsf{b}) \,/\, \gamma(\mathsf{b})) \\
&\to^* \beta \,;\, \#(\mathsf{b})
\end{aligned}
$$

The normal forms $\#(\mathsf{a})$ *and* $\beta \,;\, \#(\mathsf{b})$ *represent different failing computations:* $\mathsf{a} \to \bot$ *and* $\mathsf{a} \to \mathsf{b} \to \bot$*. The above computations are depicted in the diagrams below:*

$$\begin{tikzcd}
\mathsf{f}(\mathsf{a},\mathsf{a}) \arrow[r, "A"] \arrow[d, "C"'] & \mathsf{f}(\mathsf{b},\mathsf{a}) \arrow[r, "B"] & \mathsf{f}(\mathsf{b},\mathsf{b}) \\
\mathsf{f}(\mathsf{a},\mathsf{a}) \arrow[rr, "(A\,;B)\,/\,C", "\mathsf{f}(\mathsf{a}\,;\beta,\beta\,;\mathsf{b})"'] \arrow[d, "D"'] & & \mathsf{f}(\mathsf{b},\mathsf{b}) \\
\mathsf{a} \arrow[rr, "((A\,;B)\,/\,C)\,/\,D", "\#(\mathsf{a})"'] & & \bot
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathsf{f}(\mathsf{a},\mathsf{a}) \arrow[r, "A"] \arrow[d, "C"'] & \mathsf{f}(\mathsf{b},\mathsf{a}) \arrow[r, "B"] \arrow[dd, "\mathsf{f}(\beta,\mathsf{b})"', "(C\,;D)\,/\,A"] & \mathsf{f}(\mathsf{b},\mathsf{b}) \\
\mathsf{f}(\mathsf{a},\mathsf{a}) \arrow[d, "D"'] & & \\
\mathsf{a} \arrow[r, "A\,/\,(C\,;D)", "\beta"'] & \mathsf{b} \arrow[r, "B/((C;D)/A)", "\#(\mathsf{b})"'] & \bot
\end{tikzcd}$$

To solve this problem we propose a drastic solution. When facing a term $A \,/\, B$ with A and B in normal form, the defining equations are evaluated from top to bottom and the first equation that matches is applied. This essentially means that the ambiguity between (6) and (7) is resolved by giving preference to the former. Due to innermost evaluation, no other critical situations arise. So we arrive at the following definition, where we turned equation (8) into rule (18), which is possible due to the presence of (19).

Definition 4. The *residual TRS* for proof terms consists of the following rules:

$$
\begin{aligned}
x \,/\, x &\to x && (10)\\
f(A_1,\ldots,A_n) \,/\, f(B_1,\ldots,B_n) &\to f(A_1 \,/\, B_1,\ldots,A_n \,/\, B_n) && (11)\\
\alpha(A_1,\ldots,A_n) \,/\, \alpha(B_1,\ldots,B_n) &\to \mathsf{rhs}_\alpha\langle A_1 \,/\, B_1,\ldots,A_n \,/\, B_n\rangle_\alpha && (12)\\
\alpha(A_1,\ldots,A_n) \,/\, \mathsf{lhs}_\alpha\langle B_1,\ldots,B_n\rangle_\alpha &\to \alpha(A_1 \,/\, B_1,\ldots,A_n \,/\, B_n) && (13)\\
\mathsf{lhs}_\alpha\langle A_1,\ldots,A_n\rangle_\alpha \,/\, \alpha(B_1,\ldots,B_n) &\to \mathsf{rhs}_\alpha\langle A_1 \,/\, B_1,\ldots,A_n \,/\, B_n\rangle_\alpha && (14)\\
C \,/\, (A \,;\, B) &\to (C \,/\, A) \,/\, B && (15)\\
(A \,;\, B) \,/\, C &\to (A \,/\, C) \,;\, (B \,/\, (C \,/\, A)) && (16)\\
A \,/\, B &\to \#(\mathsf{tgt}(B)) && (17)\\
x \,;\, x &\to x && (18)\\
f(A_1,\ldots,A_n) \,;\, f(B_1,\ldots,B_n) &\to f(A_1 \,;\, B_1,\ldots,A_n \,;\, B_n) && (19)
\end{aligned}
$$

We adopt innermost evaluation with the condition that the rules (10)–(17) are evaluated from top to bottom.

The residual TRS operates on *closed* proof terms, which are proof terms without proof term variables, to ensure that $\mathsf{tgt}(B)$ in the right-hand side of (17) can be evaluated. (Variables of the underlying TRS are allowed in proof terms.)

Example 5. *Consider the TRS of Example 1. For* $D = \alpha(\beta)$ *and* $E = \alpha(\mathsf{a}) \,;\, \mathsf{g}(\beta, \beta)$ *we have*

$$D \,/\, E \;=\; \alpha(\beta) \,/\, (\alpha(\mathsf{a}) \,;\, \mathsf{g}(\beta,\beta)) \to (\alpha(\beta) \,/\, \alpha(\mathsf{a})) \,/\, \mathsf{g}(\beta,\beta) \to \mathsf{g}(\beta \,/\, \mathsf{a}, \beta \,/\, \mathsf{a}) \,/\, \mathsf{g}(\beta,\beta)$$

$$\begin{aligned}
&\to^* \mathsf{g}(\beta,\beta) \,/\, \mathsf{g}(\beta,\beta) \to \mathsf{g}(\beta \,/\, \beta, \beta \,/\, \beta) \to^* \mathsf{g}(\mathsf{b},\mathsf{b}) \\
E \,/\, D \;=\; &(\alpha(\mathsf{a}) \,;\, \mathsf{g}(\beta,\beta)) \,/\, \alpha(\beta) \to (\alpha(\mathsf{a}) \,/\, \alpha(\beta)) \,;\, (\mathsf{g}(\beta,\beta) \,/\, (\alpha(\beta) \,/\, \alpha(\mathsf{a}))) \\
&\to^* \mathsf{g}(\mathsf{a} \,/\, \beta, \mathsf{a} \,/\, \beta) \,;\, (\mathsf{g}(\beta,\beta) \,/\, \mathsf{g}(\beta \,/\, \mathsf{a}, \beta \,/\, \mathsf{a})) \to^* \mathsf{g}(\mathsf{b},\mathsf{b}) \,;\, (\mathsf{g}(\beta,\beta) \,/\, \mathsf{g}(\beta,\beta)) \\
&\to \;\mathsf{g}(\mathsf{b},\mathsf{b}) \,;\, \mathsf{g}(\beta \,/\, \beta, \beta \,/\, \beta) \to^* \mathsf{g}(\mathsf{b},\mathsf{b}) \,;\, \mathsf{g}(\mathsf{b},\mathsf{b}) \to \mathsf{g}(\mathsf{b} \,;\, \mathsf{b}, \mathsf{b} \,;\, \mathsf{b}) \to^* \mathsf{g}(\mathsf{b},\mathsf{b})
\end{aligned}$$

Lemma 6. *The residual TRS is terminating and confluent on closed proof terms.*

Proof. Confluence of the residual TRS is obvious because of the innermost evaluation strategy and the fact that there is no root overlap between its rules (due to the imposed evaluation order). Showing termination is non-trivial because of the nested occurrences of $/$ in the right-hand sides of (15) and (16). As suggested in [3, Exercise 8.7.58] one can use semantic labeling [5]. We take the well-founded algebra $\mathcal{A}$ with carrier $\mathbb{N}$ equipped with the standard order $>$ and the following weakly monotone interpretation and labeling functions:

$$\begin{gathered}
\alpha_{\mathcal{A}}(x_1,\dots,x_n) = f_{\mathcal{A}}(x_1,\dots,x_n) = \max\{x_1,\dots,x_n\} \\
;_{\mathcal{A}}(x,y) = x + y + 1 \qquad /_{\mathcal{A}}(x,y) = x \qquad \#_{\mathcal{A}}(x) = \bot_{\mathcal{A}} = 0 \\
L_{;} = L_f = L_\alpha = L_\# = L_\bot = \varnothing \qquad L_/ = \mathbb{N} \qquad \mathsf{lab}_/(x,y) = x + y
\end{gathered}$$

The algebra $\mathcal{A}$ is a quasi-model of the residual TRS. Hence termination is a consequence of termination of its labeled version. The latter follows from LPO with well-founded precedence $/_i > /_j$ for all $i > j$ and $/_0 > \,;\, > f > \alpha > \# > \bot$ for all function symbols f and rule symbols α. ◻

The residual TRS is used to define projection equivalence.

Definition 7. The *projection order* $\lesssim$ and *projection equivalence* $\simeq$ are defined on co-initial proof terms as follows: $A \lesssim B$ if $A \,/\, B \to^* \mathsf{tgt}(B)$ and $A \simeq B$ if both $A \lesssim B$ and $B \lesssim A$.

Example 8. *The proof terms A, B, and C of Example 1 are projection equivalent since the residuals $A \,/\, B$, $B \,/\, A$, $A \,/\, C$, and $C \,/\, A$ all rewrite to the same normal form* $\mathsf{g}(\mathsf{c},\mathsf{c})$.

Lemma 6 provides us with an easy decision procedure for projection equivalence: $A \simeq B$ if and only the (unique) normal forms of $A \,/\, B$ and $B \,/\, A$ with respect to the residual TRS coincide and contain neither rule symbols nor compositions. This procedure is implemented in ProTeM[3] [1], a tool for manipulating proof terms. We refer to [2] for further details.

References

[1] Christina Kohl and Aart Middeldorp. ProTeM: A proof term manipulator (system description). In *Proc. 3rd FSCD*, volume 108 of *LIPIcs*, pages 31:1–31:8, 2018. doi: 10.4230/LIPIcs.FSCD.2018.31.

[2] Christina Kohl and Aart Middeldorp. Composing proof terms. In *Proc. 27th CADE*, LNAI, 2019. Accepted for publication.

[3] Terese, editor. *Term Rewriting Systems*, volume 55 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2003.

[4] Vincent van Oostrom and Roel de Vrijer. Four equivalent equivalences of reductions. In *Proc. 2nd WRS*, volume 70(6) of *ENTCS*, pages 21–61, 2002. doi: 10.1016/S1571-0661(04)80599-1.

[5] Hans Zantema. Termination of term rewriting by semantic labelling. *Fundamenta Informaticae*, 24:89–105, 1995. doi: 10.3233/FI-1995-24124.

[3] http://informatik-protem.uibk.ac.at/