Residuals in Higher-Order Rewriting*

H. J. Sander Bruggink†

May 6, 2003

Abstract. Residuals have been studied for various forms of rewriting and residual systems have been defined to capture residuals in an abstract setting. In this article we study residuals in orthogonal Pattern Rewriting Systems (PRSs). First, the rewrite relation is defined by means of a higher-order rewriting logic, and proof terms are defined that witness reductions. Then, we have the formal machinery to define a residual operator for PRSs, and we will prove that an orthogonal PRS together with the residual operator mentioned above, is a residual system. As a side-effect, all results of (abstract) residual theory are inherited by orthogonal PRSs, such as confluence, and the notion of permutation equivalence of reductions.

Contents

*This paper will appear (in a shortened form) in the proceedings of RTA 2003. Because examples have been added in this version, the numbering of the theorems and lemmas may differ from the conference version.

†Department of Philosophy, Utrecht University, email: `brugginkphil.uu.nl`, web: `http://www.phil.uu.nl/~bruggink`

1 Introduction

This paper deals with residual theory: what remains of a reduction after another reduction from the same object has been performed? Let φ and ψ be reductions. Intuitively, the residual of φ after ψ, written φ/ψ, should consist of exactly those steps of φ which were not in ψ. In the literature, residuals have been studied in various degrees of abstraction [2, 3, 4, 6, 8, 13, 14], and for various forms of reduction (e.g. reduction in the λ-calculus, first-order term rewriting, and concurrency theory). In this paper we study residuals in a subclass of Higher-order Rewriting Systems (HRSs), orthogonal Pattern Rewriting Systems (orthogonal PRSs).

Even in first-order term rewriting, calculating residuals is a non-trivial task. Performing a reduction may duplicate the redexes of other reductions, thus potentially increasing the length of their residuals. In the higher-order case, the problems caused by duplication are more severe: now, copies of the same redex may get nested. Consider the orthogonal PRS which consists of the following two rules:

$$\begin{array}{rrcl} \mu: & \lambda z.\mathsf{mu}(\lambda x.z(x)) & \to & \lambda z.z(\mathsf{mu}(\lambda x.z(x))) \\ \rho: & \lambda x.f(x) & \to & \lambda x.h(x,x) \end{array}$$

Consider the term $s = \mathsf{mu}(\lambda x.f(x))$. The rule μ can be applied to the whole term (because $(\lambda z.\mathsf{mu}(\lambda x.z(x)))(\lambda x.f(x)) =_\beta s$) and the rule ρ can be applied to the subterm $\lambda x.f(x)$, so the following steps exist from s:

$$\begin{array}{rrcl} \varphi: & \mathsf{mu}(\lambda x.f(x)) & \to & f(\mathsf{mu}(\lambda x.f(x))) \\ \psi: & \mathsf{mu}(\lambda x.f(x)) & \to & \mathsf{mu}(\lambda x.h(x,x)) \end{array}$$

The residual of ψ after φ is the reduction

$$\begin{array}{rcl} f(\mathsf{mu}(\lambda x.f(x))) & \to & h(\mathsf{mu}(\lambda x.f(x)), \mathsf{mu}(\lambda x.f(x)) \\ & \to^* & h(\mathsf{mu}(\lambda x.h(x,x)), \mathsf{mu}(\lambda x.h(x,x))) \end{array}$$

in which we see that one copy of the ρ-redex duplicates another (nested) copy of the ρ-redex.

In this paper we define a projection operator for proof terms, which are witnesses to multistep reductions. The operator projects one proof term over another and returns the residual of that proof term after the other. We define the projection operator by means of an inference system (postponing the proof that it is actually defined on orthogonal PRSs to the last part of the paper), prove that a PRS with projection operator is a residual system, and give an algorithm which calculates residuals.

2 Preliminaries

2.1 Higher-Order Rewriting

We use Higher-order Rewriting Systems (HRSs) [7]. In fact, we consider HRSs as HORSs [12] with the simply typed λ-calculus as substitution calculus. We presuppose working knowledge of the λ-calculus, but in this section we will quickly recall the important notions of HRSs.

We fix in advance a signature Σ of simply typed constants (over a set of base types $\mathcal{B}$)[1]. *Preterms* are simply typed λ-terms over Σ. We identify α-equivalent preterms. We consider $\beta\eta$-equivalence classes of preterms. Since it is well-known that β-reduction combined with restricted η-expansion ($\beta\overline{\eta}$-reduction) is both confluent (modulo α-equivalence) and strongly normalizing, we can consider $\beta\overline{\eta}$-normal forms as unique representatives of the $\beta\eta$-equivalence classes. So, we define: *terms* are preterms in $\beta\overline{\eta}$-normal form. A *context* is a term of the form $\lambda x.C_0$, such that x occurs free in C_0 exactly once.

We write stu for $(st)u$, and we use, for arbitrary (pre)terms $s, t_1, \ldots, t_n$, the following notation: $s(t_1, \ldots, t_n) = st_1 \ldots t_n$. Often, s will just be a function symbol, but the same notation is used if s is a term of the form $\lambda x_1 \ldots x_n.s_0$.

A term s is a *pattern* if all of its free variables x occur in some subterm of s of the form $x(y_1, \ldots, y_n)$, where the y_i are *distinct* bound variables.

Definition 2.1. *A* rewrite rule *is a tuple $l = \lambda x_1 \ldots x_n.l_0 \to \lambda x_1 \ldots x_n.r_0 = r$, where l (the left-hand side) and r (the right-hand side) are closed terms of the same type, and l is not η-equivalent to a variable. The rule is* left-linear *if $x_1, \ldots, x_n$ occur in l_0 exactly once.*

A Higher-order Rewrite System *(HRS) $\mathcal{H}$ is a set of rewrite rules. $\mathcal{H}$ is left-linear, if all its rules are. An HRS is a* Pattern Rewrite System *(PRS) if, for all of its rules $\lambda x_1 \ldots x_n.l_0 \to \lambda x_1 \ldots x_n.r_0$, l_0 is a pattern.*

Let $\mathcal{H}$ be an HRS. We define the rewrite relation $\to_{\mathcal{H}}$ as follows [16]: s rewrites to t, written $s \to_{\mathcal{H}} t$ (the subscript is omitted if clear from the context), if there is a context C and a rule $l \to r \in R$, such that $s \twoheadleftarrow_\beta C(l)$ and $C(r) \twoheadrightarrow_\beta t$. By $\to_{\mathcal{H}}^*$ we denote the reflexive, transitive closure of $\to_{\mathcal{H}}$.

Example 2.2. Consider the PRS which consists of the single rule

$$l = \lambda z.\mathsf{mu}(\lambda x.z(x)) \to \lambda z.z(\mathsf{mu}(\lambda x.z(x))) = r$$

and the term $s = \mathsf{mu}(\lambda x.f(x))$. Let $C = \lambda v.v(\lambda x.f(x))$. Now $C(l) \twoheadrightarrow_\beta s$. Because $C(r) \twoheadrightarrow_\beta f(\mathsf{mu}(\lambda x.f(x))) = t$, it is the case that $s \to t$. Let $C' = \lambda v.f(v(\lambda x.f(x)))$. Then $C'(l) \twoheadrightarrow_\beta t$ and $C'(r) \twoheadrightarrow_\beta f(f(\mathsf{mu}(\lambda x.f(x)))) = u$, so we know that $s \to t \to u$.

[1] All definitions must be read as having the signature as an implicit parameter.

This example shows why our definition of the higher-order rewrite relation does not require substitutions such as in first-order term rewriting: the role of the substitutions is taken over by the context, and the act of substituting is done by the β-reductions.

Sometimes, the outermost abstractions in the rules will be omitted. For instance, the rule of Example 2.2 may be written as $\mathsf{mu}(\lambda x.z(x)) \to z(\mathsf{mu}(\lambda x.z(x)))$, where it is assumed that the free variable z is in fact bound by an invisible abstraction.

The most important reason one might have to use PRSs, is the following result of Miller [10]: unification of patterns is decidable, and if two patterns are unifiable, a most general unifier can be computed. This entails that the rewriting relation induced by a PRS is decidable.

We mention the following property of higher-order rewriting. It is non-trivial due to the implicit β-reductions in su and tv. Proofs can be found in [7, 12].

Proposition 2.3. *Let s, t, u, v be terms. If $s \to^* t$ and $u \to^* v$ then $su \to^* tv$.*

2.2 Residual Theory

Residual theory was studied in, among others, [2, 3, 4, 6, 8]. In this section, we present residuals in an abstract setting, following [13, 14], which was, in turn, based on [17]. If φ and ψ are reductions from the same object, in an arbitrary form of rewriting, then what can we tell in general of what the residual of φ after ψ must look like?

The most general form of rewriting system, which, for that reason, we will use in this section, is an *abstract rewriting system* (ARS). An ARS is a structure $\mathcal{R} = \langle A, R, \mathrm{src}, \mathrm{tgt}\rangle$ where A is a set of objects, R is a set of steps, and src and tgt are functions from R to A, specifying the source and target of the steps, respectively. Two steps are called *coinitial* if they start at the same object.

An *abstract rewriting system with composition* (ARSC) is a structure $\mathcal{R} = \langle A, R, \cdot, \mathrm{src}, \mathrm{tgt}\rangle$ such that $\langle A, R, \mathrm{src}, \mathrm{tgt}\rangle$ is an ARS (the elements of R are now called *reductions*) and $\cdot$ (the composition operator) is a function from pairs $\varphi, \psi \in R$ with $\mathrm{tgt}(\varphi) = \mathrm{src}(\psi)$ to elements of R. An ARSC can be converted to an ARS in the obvious way, i.e. by leaving out the composition operator.

Definition 2.4. *A* residual system *is specified by a triple $\langle \mathcal{R}, 1, /\rangle$ where: $\mathcal{R}$ is an (abstract) rewriting system; 1 is a function from objects (of $\mathcal{R}$) to steps, such that $\mathrm{src}(1(s)) = \mathrm{tgt}(1(s)) = s$; and $/$, the projection function, is a function from pairs of coinitial steps to steps, with $\mathrm{src}(\varphi/\psi) = \mathrm{tgt}(\psi)$ and*

$\mathrm{tgt}(\varphi/\psi) = \mathrm{tgt}(\psi/\varphi)$, *such that the following identities hold:*

$$\begin{array}{rcl} 1/\varphi & = & 1 \\ \varphi/1 & = & \varphi \\ \varphi/\varphi & = & 1 \\ (\varphi/\psi)/(\chi/\psi) & = & (\varphi/\chi)/(\psi/\chi) \end{array}$$

A residual system $\langle \mathcal{R}, 1, / \rangle$ *is a* residual system with composition*, if* $\mathcal{R}$ *is an ARSC, and the following identities hold:*

$$\begin{array}{rcl} 1 \cdot 1 & = & 1 \\ \chi/(\varphi \cdot \psi) & = & (\chi/\varphi)/\psi \\ (\varphi \cdot \psi)/\chi & = & (\varphi/\chi) \cdot (\psi/(\chi/\varphi)) \end{array}$$

The result of projecting φ over ψ (i.e. φ/ψ) is called the *residual* of φ after ψ. The intuitions behind the first three identities and the requirements to sources and targets are immediately clear. Noting that if we want to project φ over ψ and then over χ, we actually have to project φ over ψ and then over χ/ψ to make sure that the steps are coinitial, the fourth identity just states that projecting φ over ψ and then over χ yields the same result as projecting φ over ψ and χ in reverse order. The two extra identities for residual systems with composition describe how the projection operator works on composed reductions. In Fig. 1 we hope to visually clearify the cube identity (on the left) and the axioms for residual systems with composition (on the right).

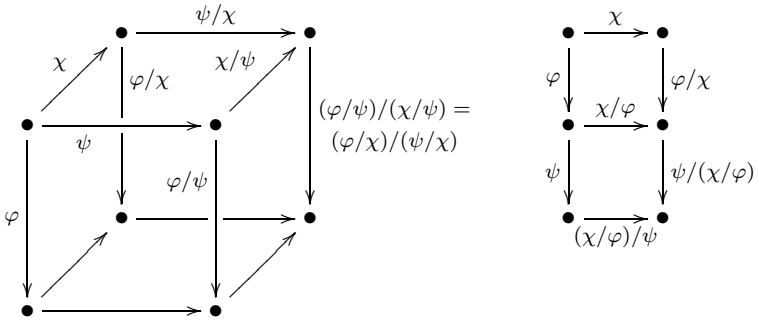

Figure 1: Axioms of residual systems

Theorem 2.5. *If* $\langle \mathcal{R}, 1, / \rangle$ *is a residual system, then* $\mathcal{R}$ *is confluent.*

Proof. Let $\langle \mathcal{R}, 1, / \rangle$ be a residual system, φ a step from a to b and ψ a step from a to c. Then ψ/φ is a step from b to some d and φ/ψ a step from c to the same object d. □

Residual theory provides an elegant formalization of the notion of equivalence of reductions: two reductions are the same if the residual of the one

$$\frac{\varphi_1 : s_1 \geq t_1 \ldots \varphi_n : s_n \geq t_n}{\rho(\varphi_1, \ldots, \varphi_n) : l(s_1, \ldots, s_n) \geq r(t_1, \ldots, t_n)}\ \mathsf{rule}$$

$$\frac{\varphi_1 : s_1 \geq t_1 \ldots \varphi_n : s_n \geq t_n}{a(\varphi_1, \ldots, \varphi_n) : a(s_1, \ldots, s_n) \geq a(t_1, \ldots, t_n)}\ \mathsf{apps}$$

$$\frac{\varphi : s \geq t}{\lambda x.\varphi : \lambda x.s \geq \lambda x.t}\ \mathsf{abs}$$

$$\frac{\varphi : s \geq u \quad \psi : u \geq t}{(\varphi \cdot \psi) : s \geq t}\ \mathsf{trans}$$

Table 1: Rewrite logic for HRSs with witnessing proof terms

after the other is an empty reduction, and vice versa. This formalization is called *permutation equivalence*. We define, for reductions φ, ψ:

$$\begin{array}{ll} \varphi \lesssim \psi & \text{if} \quad \varphi/\psi = 1 \\ \varphi \simeq \psi & \text{if} \quad \varphi \lesssim \psi \text{ and } \psi \lesssim \varphi \end{array}$$

It is not difficult to prove that $\lesssim$ is a quasi-order, and $\simeq$ is a congruence for $\cdot$ and $/$.

One of the side-effects of the main result of the paper, is that the above notion of permutation equivalence transfers directly to PRSs. Laneve & Montanari [5] give an axiomatic definition of permutation equivalence for the related format of orthogonal Combinatory Reduction Systems (CRSs), by translating CRS to first-order TRS and then using a first-order rewrite logic. We apply a higher-order rewrite logic to PRSs directly.

3 Higher-Order Rewrite Logic

In this section we give an alternative definition of the rewrite relation by means of a higher-order rewrite logic, i.e. a higher-order equational logic (see e.g. [11, 19]) without the symmetry rule (cf. [9]). The rules of the higher-order rewrite logic are presented in Table 1, together with witnessing proof terms ($\rho : l \to r$ is a rule, and a is an arbitrary function symbol or variable). Note that $l(s_1, \ldots, s_n)$ is implicitly reduced to $\beta\overline{\eta}$-normal form. The rules don't include a reflexivity rule; this rule can be easily simulated by the other rules, and is therefore left out. Note that the rule and apps rules function as axioms if $n = 0$. We write $s \geq t$ if there is a proof term φ such that $\varphi : s \geq t$.

Proposition 3.1. *$s \to^* t$ iff $s \geq t$.*

Proof. The left-to-right case of the proposition is trivial, and the right-to-left case is done by structural induction on the inference of $s \geq t$. □

In the rest of the paper, the following conventions are used: f, g range over function symbols, x, y range over variables, a, b range over function symbols *and* variables, and ρ, θ are rule symbols, where l, r are the left- and right hand side of ρ. Suppose $\varphi : s \geq t$. The terms s and t will be called the source and target of φ, respectively, and we introduce the functions $\mathrm{src}(\varphi) = s$ and $\mathrm{tgt}(\varphi) = t$. It is easily seen that $s : s \geq s$. Thus, we define the unit function 1 as $1(t) = t$. We will usually omit the argument, and just write 1 for each reduction which is the unit of some term; the exact term can be found by looking at the source or the target.

Proof terms are convenient, because they are terms, and so we have technical machinery to deal with them [1, 13, 14]. We relate proof terms to the conventional rewriting terminology in the following way: a *multistep* (or just *step* for short) is a proof term which contains no $\cdot$'s; a *proper step* is a multistep with only one rule symbol in it, and a (multistep) *reduction* is a proof term of the form $\varphi_1 \cdot \ldots \cdot \varphi_n$ (i.e. modulo associativity of $\cdot$), where the φ_i are multisteps. Note that these notions intuitively correspond with the usual non proof term based notions.

We associate to each HRS $\mathcal{H}$ the following ARSC $\dot{\mathcal{H}}$: terms are its objects, the *proof terms* of $\mathcal{H}$ are its *steps*, the src and tgt functions simply are the ones introduced above, and $\cdot$ just returns for each pair of composable proof terms φ, ψ their composition, i.e. the term $\varphi \cdot \psi$. The ARS $\hat{\mathcal{H}}$ is defined as $\dot{\mathcal{H}}$ without composition. The translation of $\mathcal{H}$ into $\hat{\mathcal{H}}$ will be done implicitly.

4 Higher-Order Term Residual Systems

From now on, we restrict our attention to PRSs. We want to define a projection function, which gives for pairs of proof terms the residual of one after the other. We do this by adding a new symbol to the language, /, and defining a relation which simplifies terms of this extended signature to terms which contain no more /'s. This relation can then be used to finally define the projection operator.

Let a pre-slash-dot term be a proof term over an extended signature which includes a polymorphic *projection operator* $/ : \alpha \to \alpha \to \alpha$ (note that every proof term is a slash-dot term as well). Slash-dot terms are pre-slash-dot terms modulo the following identities:

$$\begin{aligned} f(\varphi_1 \cdot \psi_1, \ldots, \varphi_n \cdot \psi_n) &= f(\varphi_1, \ldots, \varphi_n) \cdot f(\psi_1, \ldots, \psi_n) \\ \lambda x.(\varphi \cdot \psi) &= \lambda x.\varphi \cdot \lambda x.\psi \\ 1 \cdot \varphi &= \varphi \\ \varphi \cdot 1 &= \varphi \end{aligned}$$

The first two are called the *functorial identities*, and the last two are called the *unit identities*.

4.1 A First Attempt

Simplification of terms is usually modelled as a rewriting system. In [13, 14], the following rewriting system is presented which reduces (first-order) slash-dot terms to their corresponding proof term.

$$\begin{array}{rcl} a(\varphi_1,\ldots,\varphi_n)/a(\psi_1,\ldots,\psi_n) & \to & a(\varphi_1/\psi_1,\ldots,\varphi_n/\psi_n) \\ \rho(\varphi_1,\ldots,\varphi_n)/\rho(\psi_1,\ldots,\psi_n) & \to & r(\varphi_1/\psi_1,\ldots,\varphi_n/\psi_n) \\ \rho(\varphi_1,\ldots,\varphi_n)/l(\psi_1,\ldots,\psi_n) & \to & \rho(\varphi_1/\psi_1,\ldots,\varphi_n/\psi_n) \\ l(\varphi_1,\ldots,\varphi_n)/\rho(\psi_1,\ldots,\psi_n) & \to & r(\varphi_1/\psi_1,\ldots,\varphi_n/\psi_n) \\ (\varphi\cdot\psi)/\chi & \to & (\varphi/\chi)\cdot(\psi/(\chi/\varphi)) \\ \chi/(\varphi\cdot\psi) & \to & (\chi/\varphi)/\psi \end{array}$$

The naive method to transfer the system to the higher-order case, is to add the following rule:

$$(\lambda x.\varphi'(x)/\lambda x.\psi'(x))z \to \varphi'(z)/\psi'(z)$$

This rule pushes abstractions outwards. The variable z is just used to handle bound variables. However, the rule is not equipped to handle nesting correctly. Consider the following two-rule PRS:

$$\begin{array}{rrcl} \mu: & \lambda z.\mathsf{mu}(\lambda x.z(x)) & \to & \lambda z.z(\mathsf{mu}(\lambda x.z(x))) \\ \rho: & \lambda x.f(x) & \to & \lambda x.g(x) \end{array}$$

and the following steps:

$$\begin{array}{rcl} \mathsf{mu}(\lambda x.\rho(x)) & : & \mathsf{mu}(\lambda x.f(x)) \geq \mathsf{mu}(\lambda x.g(x)) \\ \mu(\lambda x.f(x)) & : & \mathsf{mu}(\lambda x.f(x)) \geq f(\mathsf{mu}(\lambda x.f(x))) \end{array}$$

Wee see that the proof term $\mathsf{mu}(\lambda x.\rho(x))/\mu(\lambda x.f(x))$ reduces in a number of steps to $\rho(\mathsf{mu}(\lambda x.\rho(x)/\lambda x.\rho(x)))$, and then in the final step the two copies of $\lambda x.\rho(x)$, which are not supposed to be further reduced, 'cancel each other out', resulting in the (incorrect) proof term $\rho(\mathsf{mu}(\lambda x.f(x)))$. Changing the fifth rule into

$$(\lambda x.\varphi'(x)/\lambda x.\psi'(x))z \to \varphi'(\bot z)/\psi'(\bot z)$$

where $\bot$ is a new symbol which makes sure that applications of the other rules are blocked, and adding rules to make sure that $\bot\varphi/\bot\varphi \to^* \varphi$, seems, at first sight, to solve the problem, but I have chosen another approach which I find more elegant.

4.2 Residuals of Compatible Reductions

We define the 'simplification' relation $\succcurlyeq$ between slash-dot terms and proof terms by means of the inference system Res given in Table 2 on page 10.

The R_i, $\cdot L$ and $\cdot R$ rules define how the residual operator works on terms of different forms, and the replacement rules zoom in to the interesting parts of a slash-dot term. The $\mathsf{r{+}t/}$ rule combines replacement of the / and transitivity.

We write $\vdash^{\mathcal{K}} \varphi \succcurlyeq \chi$ to denote that the inference $\mathcal{K}$ has $\varphi \succcurlyeq \chi$ as its final conclusion. The function $|\mathcal{K}|$ returns the 'depth' of an inference, i.e. if $|\mathcal{K}|$ is an inference with immediate subinferences $\mathcal{L}_1, \ldots, \mathcal{L}_n$, then $|\mathcal{K}| = \max_{0<i\leq n} |\mathcal{L}_i| + 1$. We write $\vdash^k \varphi \succcurlyeq \chi$ if an inference $\mathcal{K}$ exists such that $\vdash^{\mathcal{K}} \varphi \succcurlyeq \chi$ and $|\mathcal{K}| \leq k$. If k is omitted, $\mathcal{K}$ may be of arbitrary size, and in this case the $\vdash$ will often be omitted as well. The *principal rule* of an inference $\mathcal{K}$ is the last rule which is applied, i.e. the rule which appears at the bottom of the inference. We will assume the function $\mathsf{pr}(\mathcal{K})$ which returns the principal rule of an inference $\mathcal{K}$.

We demonstrate the inference system by two examples: the first example shows that the system yields the correct result for the example from Sect. 4.1, and the second demonstrates how the system works on proof terms with compositions.

Example 4.1. Consider the PRS from Sect. 4.1. The current framework yields the correct result:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{x/x \succcurlyeq x}\,R_1}{\rho(x)/f(x) \succcurlyeq \rho(x)}\,R_1}{\lambda x.\rho(x)/\lambda x.f(x) \succcurlyeq \lambda x.\rho(x)}\,R_5}{\mathsf{mu}(\lambda x.\rho(x))/\mu(\lambda x.f(x)) \succcurlyeq \rho(\mathsf{mu}(\lambda x.\rho(x)))}\,R_4$$

Example 4.2. Consider the following three-rule PRS (which is actually just a first-order TRS translated to the current format):

$$\begin{array}{lrcl} \rho: & \lambda x.f(x) & \to & \lambda x.g(x) \\ \theta: & \lambda x.g(x) & \to & \lambda x.h(x,x) \\ \eta: & a & \to & b \end{array}$$

In this PRS the following reductions exist: $f(a) \to g(a) \to h(a,a)$ and $f(a) \to f(b)$. The first reduction is witnessed by the proof term $\rho(a) \cdot \theta(a)$ and the second by $f(\eta)$. We calculate the both residuals of one after the other.

$$\dfrac{\dfrac{\dfrac{}{a/\eta \succcurlyeq b}\,R_4}{\rho(a)/f(\eta) \succcurlyeq \rho(b)}\,R_3 \qquad \dfrac{\dfrac{}{\eta/a \succcurlyeq b}\,R_3}{f(\eta)/\rho(a) \succcurlyeq g(\eta)}\,R_4 \qquad \dfrac{\dfrac{}{a/\eta \succcurlyeq b}\,R_4}{\theta(a)/g(\eta) \succcurlyeq \theta(b)}\,R_3}{(\rho(a) \cdot \theta(a))/f(\eta) \succcurlyeq \rho(b) \cdot \theta(b)}\,\cdot L$$

Residual rules:

$$\frac{\varphi_1/\psi_1 \succcurlyeq \chi_1 \cdots \varphi_n/\psi_n \succcurlyeq \chi_n}{a(\varphi_1,\ldots,\varphi_n)/a(\psi_1,\ldots,\psi_n) \succcurlyeq a(\chi_1,\ldots,\chi_n)}\, R_1$$

$$\frac{\varphi_1/\psi_1 \succcurlyeq \chi_1 \cdots \varphi_n/\psi_n \succcurlyeq \chi_n}{\rho(\varphi_1,\ldots,\varphi_n)/\rho(\psi_1,\ldots,\psi_n) \succcurlyeq r(\chi_1,\ldots,\chi_n)}\, R_2$$

$$\frac{\varphi_1/\psi_1 \succcurlyeq \chi_1 \cdots \varphi_n/\psi_n \succcurlyeq \chi_n}{\rho(\varphi_1,\ldots,\varphi_n)/l(\psi_1,\ldots,\psi_n) \succcurlyeq \rho(\chi_1,\ldots,\chi_n)}\, R_3$$

$$\frac{\varphi_1/\psi_1 \succcurlyeq \chi_1 \cdots \varphi_n/\psi_n \succcurlyeq \chi_n}{l(\varphi_1,\ldots,\varphi_n)/\rho(\psi_1,\ldots,\psi_n) \succcurlyeq r(\chi_1,\ldots,\chi_n)}\, R_4$$

$$\frac{\varphi/\psi \succcurlyeq \chi}{\lambda x.\varphi/\lambda x.\psi \succcurlyeq \lambda x.\chi}\, R_5$$

$$\frac{\varphi_1/\psi \succcurlyeq \varphi_1' \quad \psi/\varphi_1 \succcurlyeq \psi' \quad \varphi_2/\psi' \succcurlyeq \varphi_2'}{(\varphi_1\cdot\varphi_2)/\psi \succcurlyeq \varphi_1'\cdot\varphi_2'}\cdot L \qquad \frac{\varphi/\psi_1 \succcurlyeq \varphi' \quad \varphi'/\psi_2 \succcurlyeq \chi}{\varphi/(\psi_1\cdot\psi_2) \succcurlyeq \chi}\cdot R$$

$$\frac{\varphi \succcurlyeq \varphi' \quad \psi \succcurlyeq \psi' \quad \varphi'/\psi' \succcurlyeq \chi}{\varphi/\psi \succcurlyeq \chi}\,\mathsf{r{+}t/}$$

Replacement rules:

$$\frac{\varphi_1 \succcurlyeq \psi_1 \cdots \varphi_n \succcurlyeq \psi_n}{a(\varphi_1,\ldots,\varphi_n) \succcurlyeq a(\psi_1,\ldots,\psi_n)}\,\mathsf{repl}_a \qquad \frac{\varphi \succcurlyeq \psi}{\lambda x.\varphi \succcurlyeq \lambda x.\psi}\,\mathsf{repl}_\lambda$$

$$\frac{\varphi_1 \succcurlyeq \psi_1 \cdots \varphi_n \succcurlyeq \psi_n}{\rho(\varphi_1,\ldots,\varphi_n) \succcurlyeq \rho(\psi_1,\ldots,\psi_n)}\,\mathsf{repl}_\rho \qquad \frac{\varphi_1 \succcurlyeq \psi_1 \quad \varphi_2 \succcurlyeq \psi_2}{\varphi_1\cdot\varphi_2 \succcurlyeq \psi_1\cdot\psi_2}\,\mathsf{repl}\cdot$$

Table 2: The inference rules for Res.

$$\dfrac{\dfrac{\dfrac{}{\eta/a \succcurlyeq \eta}\,R_3}{f(\eta)/\rho(a) \succcurlyeq g(\eta)}\,R_4 \qquad \dfrac{\dfrac{}{\eta/a \succcurlyeq \eta}\,R_3}{g(\eta)/\theta(a) \succcurlyeq h(\eta,\eta)}\,R_4}{f(\eta)/(\rho(a)\cdot\theta(a)) \succcurlyeq h(\eta,\eta)}\cdot R$$

The first proof term, $\rho(b) \cdot \theta(b)$, witnesses the reduction $f(b) \to g(b) \to h(b,b)$, and the second proof term, $h(\eta,\eta)$, witnesses the reduction (in this case consisting of a single multistep) $h(a,a) \to^* h(b,b)$. So we see that we obtain the expected proof terms.

A slash-dot term φ is called *internally compatible* if there is a χ such that $\varphi \succcurlyeq \chi$. The source and target of an internally compatible slash-dot term φ with $\varphi \succcurlyeq \chi$ are defined as $\mathrm{src}(\varphi) = \mathrm{src}(\chi)$ and $\mathrm{tgt}(\varphi) = \mathrm{tgt}(\chi)$. Two slash-dot terms φ and ψ are *compatible* if φ/ψ is internally compatible. A PRS $\mathcal{H}$ is called compatible if all possible pairs of proof terms φ, ψ of $\mathcal{H}$ are compatible.

The following lemma expresses, in a sense, that proof terms are the 'final objects' of the relation $\succcurlyeq$ defined by the inference system.

Lemma 4.3. *Let φ be a proof term. Then: $\vdash^{\mathcal{K}} \varphi \succcurlyeq \psi$ if and only if $\varphi = \psi$.*

Proof. Since φ is a proof term, it does not contain /'s and we only need to consider replacement rules. The 'only if' side can then be proved by induction on $\mathcal{K}$, and since the premisses of the rules under consideration are all strictly smaller than the conclusion, the 'if' side can be proved by induction on the length φ. □

Next, we prove a few standardization properties of the proposed inference system, which will come in handy in the later proofs. Given a desired outcome, Lemma 4.4 and Lemma 4.5 are used to select the principal rule of a valid inference with the desired conclusion (if it exists).

Lemma 4.4. *Suppose $\vdash^{\mathcal{K}} \varphi/\psi \succcurlyeq \chi$.*

1. *If $\varphi = a(\varphi_1, \ldots, \varphi_n)$ and $\psi = a(\varphi_1, \ldots, \varphi_n)$, then there is an inference $\mathcal{K}'$ with $|\mathcal{K}'| \leq |\mathcal{K}|$ such that $\vdash^{\mathcal{K}'} \varphi/\psi \succcurlyeq \chi$ and $\mathsf{pr}(\mathcal{K}') = R_1$.*

2. *If $\varphi = \rho(\varphi_1, \ldots, \varphi_n)$ and $\psi = \rho(\varphi_1, \ldots, \varphi_n)$, then there is an inference $\mathcal{K}'$ with $|\mathcal{K}'| \leq |\mathcal{K}|$ such that $\vdash^{\mathcal{K}'} \varphi/\psi \succcurlyeq \chi$ and $\mathsf{pr}(\mathcal{K}') = R_2$.*

3. *If $\varphi = \rho(\varphi_1, \ldots, \varphi_n)$ and $\psi = l(\varphi_1, \ldots, \varphi_n)$, then there is an inference $\mathcal{K}'$ with $|\mathcal{K}'| \leq |\mathcal{K}|$ such that $\vdash^{\mathcal{K}'} \varphi/\psi \succcurlyeq \chi$ and $\mathsf{pr}(\mathcal{K}') = R_3$.*

4. *If $\varphi = l(\varphi_1, \ldots, \varphi_n)$ and $\psi = \rho(\varphi_1, \ldots, \varphi_n)$, then there is an inference $\mathcal{K}'$ with $|\mathcal{K}'| \leq |\mathcal{K}|$ such that $\vdash^{\mathcal{K}'} \varphi/\psi \succcurlyeq \chi$ and $\mathsf{pr}(\mathcal{K}') = R_4$.*

5. *If $\varphi = \lambda x.\varphi_0$ and $\psi = \lambda x.\psi_0$, then there is an inference $\mathcal{K}'$ with $|\mathcal{K}'| \leq |\mathcal{K}|$ such that $\vdash^{\mathcal{K}'} \varphi/\psi \succcurlyeq \chi$ and $\mathsf{pr}(\mathcal{K}') = R_5$.*

Proof. We show only (1), the other cases are analogous. We use induction on $|\mathcal{K}|$. There are exactly two rules that match the conclusion, so we distinguish the following cases:

- If $\mathsf{pr}(\mathcal{K}) = R_1$, then simply $\mathcal{K}' = \mathcal{K}$.
- If $\mathsf{pr}(\mathcal{K}) = \mathsf{r{+}t/}$, then we know, by induction hypothesis and the observation that only the repl_a rule matches conclusions of the form $a(\vec{\varphi}) \succcurlyeq \varphi'$, that there is an inference $\mathcal{L}$ as follows, where $\chi = a(\vec{\chi})$:

$$\dfrac{\dfrac{\cdots \varphi_i \succcurlyeq \varphi_i' \cdots}{a(\vec{\varphi}) \succcurlyeq a(\vec{\varphi}')}\,\mathsf{repl}_a \quad \dfrac{\cdots \psi_i \succcurlyeq \psi_i' \cdots}{a(\vec{\psi}) \succcurlyeq a(\vec{\psi}')}\,\mathsf{repl}_a \quad \dfrac{\cdots \varphi_i'/\psi_i' \succcurlyeq \chi_i \cdots}{a(\vec{\varphi}')/a(\vec{\psi}') \succcurlyeq a(\vec{\chi})}\,R_1}{a(\vec{\varphi})/a(\vec{\psi}) \succcurlyeq a(\vec{\chi})}\,\mathsf{r{+}t/}$$

  We now build the inference $\mathcal{K}'$ as follows:

$$\dfrac{\cdots \quad \dfrac{\varphi_i \succcurlyeq \varphi_i' \;\; \psi_i \succcurlyeq \psi_i' \;\; \varphi_i'/\psi_i' \succcurlyeq \chi_i}{\varphi_i/\psi_i \succcurlyeq \chi_i}\,\mathsf{r{+}t/} \quad \cdots}{a(\vec{\varphi})/a(\vec{\psi}) \succcurlyeq a(\vec{\chi})}\,R_1$$

□

Lemma 4.5. *Suppose* $\vdash^{\mathcal{K}} \varphi/\psi \succcurlyeq \chi$.

1. *If* $\varphi = \varphi_1 \cdot \varphi_2$, *then there is an inference* $\mathcal{K}'$ *with* $|\mathcal{K}'| \leq |\mathcal{K}|$ *such that* $\vdash^{\mathcal{K}'} \varphi/\psi \succcurlyeq \chi$ *and* $\mathsf{pr}(\mathcal{K}) = \cdot L$.
2. *If* $\psi = \psi_1 \cdot \psi_2$, *then there is an inference* $\mathcal{K}'$ *with* $|\mathcal{K}'| \leq |\mathcal{K}|$ *such that* $\vdash^{\mathcal{K}'} \varphi/\psi \succcurlyeq \chi$ *and* $\mathsf{pr}(\mathcal{K}) = \cdot R$.

Proof. By induction on $|\mathcal{K}|$. First we prove (i):

- If $\mathsf{pr}(\mathcal{K}) = \cdot L$, then $\mathcal{K}' = \mathcal{K}$.
- If $\mathsf{pr}(\mathcal{K}) = \mathsf{r{+}t/}$, then there must be an $\mathcal{L}$, where $|\mathcal{L}| \leq |\mathcal{K}|$, which ends as follows:

$$\dfrac{\dfrac{\varphi_1 \succcurlyeq \varphi_1' \;\; \varphi_2 \succcurlyeq \varphi_2'}{\varphi_1 \cdot \varphi_2 \succcurlyeq \varphi_1' \cdot \varphi_2'} \quad \psi \succcurlyeq \psi_0 \quad \dfrac{\varphi_1'/\psi_0 \succcurlyeq \chi_1 \;\; \psi_0/\varphi_1' \succcurlyeq \psi_0' \;\; \varphi_2'/\psi_0' \succcurlyeq \chi_2}{\varphi_1' \cdot \varphi_2'/\psi_0 \succcurlyeq \chi_1 \cdot \chi_2}}{(\varphi_1 \cdot \varphi_2)/\psi \succcurlyeq \chi_1 \cdot \chi_2}$$

  where the left subinference looks like this because only the $\mathsf{repl}\cdot$ rule matches conclusions of the form $\varphi \cdot \psi \succcurlyeq \chi$, and the right subinference is obtained by induction hypothesis.

  We can now prove:

$$\dfrac{\dfrac{\varphi_1 \succcurlyeq \varphi_1' \;\; \psi \succcurlyeq \psi_0 \;\; \varphi_1'/\psi_0 \succcurlyeq \chi_1}{\varphi_1/\psi \succcurlyeq \chi_1} \qquad \dfrac{\varphi_2 \succcurlyeq \varphi_2' \;\; \psi_0' \succcurlyeq \psi_0' \;\; \varphi_2'/\psi_0' \succcurlyeq \chi_2}{\varphi_2/\psi_0' \succcurlyeq \chi_2}}{\dfrac{\psi \succcurlyeq \psi_0 \;\; \varphi_1 \succcurlyeq \varphi_1' \;\; \psi_0/\varphi_1' \succcurlyeq \psi_0'}{\psi/\varphi_1 \succcurlyeq \psi_0'}}$$

and finally build $\mathcal{K}'$ as follows:

$$\dfrac{\varphi_1/\psi \succcurlyeq \chi_1 \;\; \psi/\varphi_1 \succcurlyeq \psi_0' \;\; \varphi_2/\psi_0' \succcurlyeq \chi_2}{(\varphi_1 \cdot \varphi_2)/\psi \succcurlyeq \chi_1 \cdot \chi_2}$$

- If $\mathsf{pr}(\mathcal{K}) = \cdot R$, then it must be the case that $\varphi/\psi = (\varphi_1 \cdot \varphi_2)/(\psi_1 \cdot \psi_2)$, and, by applying the induction hypothesis on the subinferences, the following inference $\mathcal{L}$ with $|\mathcal{L}| \leq |\mathcal{K}|$ must exist:

$$\dfrac{\dfrac{\varphi_1/\psi_1 \succcurlyeq \varphi_1' \;\; \psi_1/\varphi_1 \succcurlyeq \psi_1' \;\; \varphi_2/\psi_1' \succcurlyeq \varphi_2'}{(\varphi_1 \cdot \varphi_2)/\psi_1 \succcurlyeq \varphi_1' \cdot \varphi_2'} \qquad \dfrac{\varphi_1'/\psi_2 \succcurlyeq \varphi_1'' \;\; \psi_2/\varphi_1' \succcurlyeq \psi_2' \;\; \varphi_2'/\psi_2' \succcurlyeq \varphi_2''}{(\varphi_1' \cdot \varphi_2')/\psi_2 \succcurlyeq \varphi_1'' \cdot \varphi_2''}}{(\varphi_1 \cdot \varphi_2)/(\psi_1 \cdot \psi_2) \succcurlyeq \varphi_1'' \cdot \varphi_2''} \tag{1}$$

and we take $\mathcal{K}'$ to be the following inference:

$$\dfrac{\dfrac{\varphi_1/\psi_1 \succcurlyeq \varphi_1' \;\; \varphi_1'/\psi_2 \succcurlyeq \varphi_1''}{\varphi_1/(\psi_1 \cdot \psi_2) \succcurlyeq \varphi_1''} \qquad \dfrac{\psi_1/\varphi_1 \succcurlyeq \psi_1' \;\; \varphi_1/\psi_1 \succcurlyeq \varphi_1' \;\; \psi_2/\varphi_1' \succcurlyeq \psi_2'}{(\psi_1 \cdot \psi_2)/\varphi_1 \succcurlyeq \psi_1' \cdot \psi_2'} \qquad \dfrac{\varphi_2/\psi_1' \succcurlyeq \varphi_2' \;\; \varphi_2'/\psi_2' \succcurlyeq \varphi_2''}{\varphi_2/(\psi_1' \cdot \psi_2') \succcurlyeq \varphi_2''}}{(\varphi_1 \cdot \varphi_2)/(\psi_1 \cdot \psi_2) \succcurlyeq \varphi_1'' \cdot \varphi_2''} \tag{2}$$

Now we prove (ii).

- If $\mathsf{pr}(\mathcal{K}) = \cdot R$, then $\mathcal{K}' = \mathcal{K}$.
- If $\mathsf{pr}(\mathcal{K}) = \mathsf{r}{+}\mathsf{t}/$, then there is a $\mathcal{L}$ such that $|\mathcal{L}| \leq |\mathcal{K}|$, which ends as follows:

$$\dfrac{\varphi \succcurlyeq \varphi_0 \qquad \dfrac{\dfrac{\psi_1 \succcurlyeq \psi_{1,0} \;\; \psi_2 \succcurlyeq \psi_{2,0}}{\psi_1 \cdot \psi_2 \succcurlyeq \psi_{1,0} \cdot \psi_{2,0}} \qquad \dfrac{\varphi_0/\psi_{1,0} \succcurlyeq \varphi_0' \;\; \varphi_0'/\psi_{2,0} \succcurlyeq \chi}{\varphi_0/(\psi_{1,0} \cdot \psi_{2,0}) \succcurlyeq \chi}}{\varphi/(\psi_1 \cdot \psi_2) \succcurlyeq \chi}$$

and we take $\mathcal{K}'$ to be the following inference:

$$\dfrac{\dfrac{\varphi \succcurlyeq \varphi_0 \;\; \psi_1 \succcurlyeq \psi_{1,0} \;\; \varphi_0/\psi_{1,0} \succcurlyeq \varphi_0'}{\varphi/\psi_1 \succcurlyeq \varphi_0'} \qquad \dfrac{\varphi_0' \succcurlyeq \varphi_0' \;\; \psi_2 \succcurlyeq \psi_{2,0} \;\; \varphi_0'/\psi_{2,0} \succcurlyeq \chi}{\varphi_0'/\psi_2 \succcurlyeq \chi}}{\varphi/(\psi_1 \cdot \psi_2) \succcurlyeq \chi}$$

- If $\mathsf{pr}(\mathcal{K}) = \cdot L$, then by induction hypothesis there is a $\mathcal{L}$ such that $|\mathcal{L}| \leq |\mathcal{K}|$ which ends as in (2) above, and we take $\mathcal{K}'$ to be the inference given in (1).

□

Lemma 4.6. *If $\varphi \succcurlyeq \chi$ and $\varphi \succcurlyeq \chi'$, then $\chi = \chi'$.*

Proof. Let $\vdash^{\mathcal{K}} \varphi \succcurlyeq \chi$ and $\vdash^{\mathcal{L}} \varphi \succcurlyeq \chi'$. We prove the proposition by induction on $|\mathcal{K}| + |\mathcal{L}|$. If $\mathsf{pr}(\mathcal{K}) = \mathsf{pr}(\mathcal{L})$, then the result simply follows from the induction hypothesis. Otherwise, we consider φ/ψ and use Lemmata 4.4 and 4.5 to obtain inferences $\mathcal{K}', \mathcal{L}'$ such that $\vdash^{\mathcal{K}'} \varphi \succcurlyeq \chi$, $\vdash^{\mathcal{L}'} \varphi \succcurlyeq \chi'$ and $\mathsf{pr}(\mathcal{K}') = \mathsf{pr}(\mathcal{L}')$. Since $|\mathcal{K}'| \leq |\mathcal{K}|$, $|\mathcal{L}'| \leq |\mathcal{L}|$ we can apply the induction hypothesis on the subinferences as we would have done in the first case. □

We define the relation $\approx$ to be the reflexive, symmetric and transitive closure of $\succcurlyeq$. By Lemma 4.6 and the fact that if $\varphi \succcurlyeq \chi$ then χ is a proof term (easily proved by induction), we can take proof terms as the unique representatives of the classes of $\approx$-equivalent slash-dot terms. We can now define the projection operator // as follows: $\varphi \,//\, \psi = \chi$ if χ is the unique representative of the slash-dot term φ/ψ. Theorem 4.7 is proved in Sect. 4.3.

Theorem 4.7. *$\langle \mathcal{H}, 1, // \rangle$ is a residual system, if $\mathcal{H}$ is a compatible PRS.*

In fact, as follows from Lemma 4.9 in the next section, $\langle \mathcal{H}, 1, // \rangle$ is even a residual system with composition.

Corollary 4.8. *A compatible PRS is confluent.*

Proof. By Theorems 4.7 and 2.5. □

4.3 Proof of Theorem 4.7

In this subsection we prove Theorem 4.7, i.e. we show that a compatible PRS together with unit and projection operator is a residual system. We mention the following two auxiliary lemmas:

Lemma 4.9.

1. $(\varphi \cdot \psi)/\chi \approx \varphi/\chi \cdot \psi/(\chi/\varphi)$
2. $\chi/(\varphi \cdot \psi) \approx (\chi/\varphi)/\psi$

Proof. In order to save space, we give textual sketches of the inferences. Suppose $\varphi/\chi \succcurlyeq \varphi'$, $\chi/\varphi \succcurlyeq \chi'$, $\psi/\chi' \succcurlyeq \psi'$ and $\chi'/\psi \succcurlyeq \chi''$. (Lemmata 4.4 and 4.5 imply that the mentioned pairs must be compatible if the equations of Lemma 4.9 are internally compatible.)

(i) With one application of $\mathsf{r{+}t/}$ we prove $\psi/(\chi/\varphi) \succcurlyeq \psi'$, and then with one application of $\cdot L$ and $\mathsf{repl}\cdot$ we prove $(\varphi\cdot\psi)/\chi \succcurlyeq \varphi'\cdot\psi'$ and $\varphi/\chi\cdot\psi/(\chi/\varphi) \succcurlyeq \varphi' \cdot \psi'$, respectively, and hence $(\varphi \cdot \psi)/\chi \approx \varphi/\chi \cdot \psi/(\chi/\varphi)$.

(ii) With one application of $\cdot R$ we prove $\chi/(\varphi/\psi) \succcurlyeq \chi''$ and with one application of $\mathsf{r{+}t/}$ we prove $(\chi/\varphi)/\psi \succcurlyeq \chi''$, hence $\chi/(\varphi\cdot\psi) \approx (\chi/\varphi)/\psi$. □

Lemma 4.10. *$\approx$ is a congruence, i.e. if $\varphi \approx \psi$, then*

1. $f(\ldots, \varphi, \ldots) \approx f(\ldots, \psi, \ldots)$
2. $\rho(\ldots, \varphi, \ldots) \approx \rho(\ldots, \psi, \ldots)$
3. $\lambda x.\varphi \approx \lambda x.\psi$
4. $\varphi \cdot \chi \approx \psi \cdot \chi$ *and* $\chi \cdot \varphi \approx \chi \cdot \psi$
5. $\varphi/\chi \approx \psi/\chi$ *and* $\chi/\varphi \approx \chi/\psi$

Proof. Since $\varphi \approx \psi$, there is a ξ such that $\varphi \succcurlyeq \xi$ and $\psi \succcurlyeq \xi$.

(1-4) To prove (1), we simply apply the repl_f rule to obtain inferences of $f(\ldots, \varphi, \ldots) \succcurlyeq f(\ldots, \xi, \ldots)$ and $f(\ldots, \psi, \ldots) \succcurlyeq f(\ldots, \xi, \ldots)$. To show (2), (3) and (4) we follow the same strategy, but with repl_ρ, repl_λ and $\mathsf{repl}\cdot$, respectively.

(5) requires a little bit more work, because the 'integrated' transitivity has to be taken care of. The following two inferences do the job for the first part of (5), and the second part is similar:

$$\frac{\varphi \succcurlyeq \xi \quad \chi \succcurlyeq \chi' \quad \xi/\chi' \succcurlyeq \xi'}{\varphi/\chi \succcurlyeq \xi'} \qquad \frac{\psi \succcurlyeq \xi \quad \chi \succcurlyeq \chi' \quad \xi/\chi' \succcurlyeq \xi'}{\psi/\chi \succcurlyeq \xi'}$$

□

To prove that we are dealing with a residual system, we have to show that sources and targets match (Prop. 4.11), and that the residual axioms hold (Prop. 4.12).

Proposition 4.11. *Sources and targets match, i.e.:*

1. $\mathrm{src}(\varphi/\psi) = \mathrm{tgt}(\psi)$
2. $\mathrm{tgt}(\varphi/\psi) = \mathrm{tgt}(\psi/\varphi)$

Proof. By induction on the inferences of $\varphi/\psi \succcurlyeq \chi$ and $\psi/\varphi \succcurlyeq \xi$ we easily prove that $\mathrm{src}(\chi) = \mathrm{tgt}(\psi)$ and $\mathrm{tgt}(\chi) = \mathrm{tgt}(\xi)$. □

Proposition 4.12. *The residual axioms hold, i.e.:*

1. $1/\varphi \approx 1$
2. $\varphi/1 \approx \varphi$
3. $\varphi/\varphi \approx 1$
4. $(\varphi/\psi)/(\chi/\psi) \approx (\varphi/\chi)/(\psi/\chi)$

Proof. (1)–(3) are proved by induction on the length of φ. In addition, (2) is based on (1), and (3) on (1) and (2). We show only (2), the other two are analogous. (Below, a, b and c are used to handle the various R_1–R_4 together, e.g. in R_3, $a = \rho$, $b = l$ and $c = r$.)

$$\frac{\dfrac{}{\varphi_1/1 \succcurlyeq \varphi_1}\,\mathrm{IH} \quad \cdots \quad \dfrac{}{\varphi_n/1 \succcurlyeq \varphi_n}\,\mathrm{IH}}{a(\varphi_1,\ldots,\varphi_n)/b(1,\ldots,1) \succcurlyeq c(\varphi_1,\ldots,\varphi_n)} \qquad \frac{\dfrac{}{\varphi'/1 \succcurlyeq \varphi'}\,\mathrm{IH}}{\lambda x.\varphi'/\lambda x.1 \succcurlyeq \lambda x.\varphi'}$$

$$\frac{\dfrac{}{\varphi/1 \succcurlyeq \varphi}\,\mathrm{IH} \quad \dfrac{}{1/\varphi \succcurlyeq 1}\,(1) \quad \dfrac{}{\psi/1 \succcurlyeq \psi}\,\mathrm{IH}}{(\varphi \cdot \psi)/1 \succcurlyeq \varphi \cdot \psi}$$

In order to prove (4) we introduce the *layered size* $|\varphi|$ of a slash-dot term φ:

$$\begin{array}{rcl} |f(\varphi_1,\ldots,\varphi_n)| & = & 1 + \max_{0<i\leq n} |\varphi_i| \\ |x(\varphi_1,\ldots,\varphi_n)| & = & 1 + \max_{0<i\leq n} |\varphi_i| \\ |\rho(\varphi_1,\ldots,\varphi_n)| & = & 1 + \max_{0<i\leq n} |\varphi_i| \\ |\lambda x.\varphi| & = & |\varphi| \\ |\varphi \cdot \psi| & = & |\varphi| + 1 + |\psi| \\ |\varphi/\psi| & = & |\varphi| \end{array}$$

Now (4) is verified by induction on the sum of the layered sizes of φ, ψ and χ.

The proof follows the same pattern as the one in [13]. Suppose that either φ, ψ or χ is a composite. If φ is a composite, we have the following, where the various (underlined) steps follow from Lemma 4.10 and either the induction hypothesis or Lemma 4.9:

$$\begin{array}{ll} & \underline{((\varphi_1 \cdot \varphi_2)/\psi)}/(\chi/\psi) \\ \approx & \underline{((\varphi_1/\psi) \cdot (\varphi_2/(\psi/\varphi_1)))}/(\chi/\psi) \\ \approx & \underline{(\varphi_1/\psi)/(\chi/\psi)} \cdot ((\varphi_2/(\psi/\varphi_1))/\underline{((\chi/\psi)/(\varphi_1/\psi))}) \\ \approx_{\mathrm{IH}} & (\varphi_1/\chi)/(\psi/\chi) \cdot \underline{((\varphi_2/(\psi/\varphi_1))/((\chi/\varphi_1)/(\psi/\varphi_1)))} \\ \approx_{\mathrm{IH}} & \underline{(\varphi_1/\chi)/(\psi/\chi) \cdot ((\varphi_2/(\chi/\varphi_1))/((\psi/\varphi_1)/(\chi/\varphi_1)))} \\ \approx & \underline{((\varphi_1/\chi) \cdot (\varphi_2/(\chi/\varphi_1)))}/(\psi/\chi) \\ \approx & ((\varphi_1 \cdot \varphi_2)/\chi)/(\psi/\chi) \end{array}$$

If ψ is a composite, we do:

$$\begin{array}{ll} & \underline{(\varphi/(\psi_1 \cdot \psi_2))}/\underline{(\chi/(\psi_1 \cdot \psi_2))} \\ \approx & \underline{((\varphi/\psi_1)/\psi_2)/((\chi/\psi_1)/\psi_2)} \\ \approx_{\mathrm{IH}} & \underline{((\varphi/\psi_1)/(\chi/\psi_1))}/(\psi_2/(\chi/\psi_1)) \\ \approx_{\mathrm{IH}} & \underline{((\varphi/\chi)/(\psi_1/\chi))}/(\psi_2/(\chi/\psi_1)) \\ \approx & (\varphi/\chi)/\underline{(\psi_1/\chi \cdot \psi_2/(\chi/\psi_1))} \\ \approx & (\varphi/\chi)/((\psi_1 \cdot \psi_2)/\chi) \end{array}$$

The case that χ is a composite, is the inverse of this.

Now consider the case that none of φ, ψ, χ are composites. Suppose that $\varphi = f(\vec{\varphi})$, $\psi = f(\vec{\psi})$, and $\chi = f(\vec{\chi})$, where we use the notation $\vec{x}$ for the vector $x_1, \ldots, x_n$. By Lemma 4.4, the following inference must exist:

$$\dfrac{\dfrac{\cdots \quad \overline{\varphi_i/\psi_i \succcurlyeq \zeta_{1,i}} \quad \cdots}{f(\vec{\varphi})/f(\vec{\psi}) \succcurlyeq f(\vec{\zeta_1})} \quad \dfrac{\cdots \quad \overline{\chi_i/\psi_i \succcurlyeq \zeta_{2,i}} \quad \cdots}{f(\vec{\chi})/f(\vec{\psi}) \succcurlyeq f(\vec{\zeta_2})} \quad \dfrac{\cdots \quad \overline{\zeta_{1,i}/\zeta_{2,i} \succcurlyeq \xi_{1,i}} \quad \cdots}{f(\vec{\zeta_1})/f(\vec{\zeta_2}) \succcurlyeq f(\vec{\xi_1})}}{(f(\vec{\varphi})/f(\vec{\psi}))/(f(\vec{\chi})/f(\vec{\psi})) \succcurlyeq f(\vec{\xi_1})}$$

and similarly we obtain an inference of $(f(\vec{\varphi})/f(\vec{\chi}))/(f(\vec{\psi})/f(\vec{\chi})) \succcurlyeq f(\vec{\xi_2})$. Using the same subinferences for $\varphi_i/\psi_i \succcurlyeq \zeta_{1,i}$, $\chi_i/\psi_i \succcurlyeq \zeta_{2,i}$ and $\zeta_{1,i}/\zeta_{2,i} \succcurlyeq \xi_{1,i}$, we easily prove $(\varphi_i/\psi_i)/(\chi_i/\psi_i) \succcurlyeq \xi_{1,i}$, and similarly $(\varphi_i/\chi_i)/(\psi_i/\chi_i) \succcurlyeq \xi_{2,i}$. Since, by induction hypothesis, $(\varphi_i/\psi_i)/(\chi_i/\psi_i) \approx (\varphi_i/\chi_i)/(\psi_i/\chi_i)$, we know now that $\xi_{1,i} \approx \xi_{2,i}$, so there are $\xi_{3,i}$ such that $\xi_{1,i} \succcurlyeq \xi_{3,i} \preccurlyeq \xi_{2,i}$. Two easy inferences prove $f(\vec{\xi_1}) \succcurlyeq f(\vec{\xi_3})$ and $f(\vec{\xi_2}) \succcurlyeq f(\vec{\xi_3})$. We put everything together with transitivity and get:

$$(f(\vec{\varphi})/f(\vec{\psi}))/(f(\vec{\chi})/f(\vec{\psi})) \approx f(\vec{\xi_3}) \approx (f(\vec{\varphi})/f(\vec{\chi}))/(f(\vec{\psi})/f(\vec{\chi}))$$

The same strategy works in the other non-composite cases, e.g. if $\varphi = \rho(\vec{\varphi})$, $\psi = \rho(\vec{\psi})$, and $\chi = r(\vec{\chi})$, since the difficult nesting problems (duplicating behaviour within right-hand sides of rules) occur only on the right of the $\succcurlyeq$ symbols. □

Now, the fact that a PRS with residual operator is a residual system (Theorem 4.7) follows from Prop. 4.11 and Prop. 4.12. In fact, it follows from Lemma 4.9 that it is a residual system with composition. QED.

4.4 Computing Residuals

In Sect. 4.2 only a specification of the simplification relation was given, but Lemma 4.4 and Lemma 4.5 already hinted at the existence of an algorithm which effectively computes the representative of a slash-dot term. In this section, such an algorithm is indeed given. We also prove that it terminates for two special cases of slash-dot term, and show that if the algorithm terminates, it prints the correct answer.

Definition 4.13. *The (recursive) function* $\mathrm{sim}(\pi)$ *on proof terms* π*, is defined by the following pseudo-program:*

$\mathrm{sim}((\varphi_1/\varphi_2)/\psi)) = \mathrm{sim}(\varphi'/\psi)$
  where $\varphi' = \mathrm{sim}(\varphi_1/\varphi_2)$
$\mathrm{sim}(\varphi/(\psi_1/\psi_2)) = \mathrm{sim}(\varphi/\psi')$
  where $\psi' = \mathrm{sim}(\psi_1/\psi_2)$
$\mathrm{sim}(x(\varphi_1, \ldots, \varphi_n)/x(\psi_1, \ldots, \psi_n)) = x(\mathrm{sim}(\varphi_1/\psi_1), \ldots, \mathrm{sim}(\varphi_n/\psi_n))$

$\mathrm{sim}(f(\varphi_1,\ldots,\varphi_n)/f(\psi_1,\ldots,\psi_n)) = f(\mathrm{sim}(\varphi_1/\psi_1),\ldots,\mathrm{sim}(\varphi_n/\psi_n))$
$\mathrm{sim}(\rho(\varphi_1,\ldots,\varphi_n)/\rho(\psi_1,\ldots,\psi_n)) = r(\mathrm{sim}(\varphi_1/\psi_1),\ldots,\mathrm{sim}(\varphi_n/\psi_n))$
$\mathrm{sim}(\rho(\varphi_1,\ldots,\varphi_n)/l(\psi_1,\ldots,\psi_n)) = \rho(\mathrm{sim}(\varphi_1/\psi_1),\ldots,\mathrm{sim}(\varphi_n/\psi_n))$
$\mathrm{sim}(l(\varphi_1,\ldots,\varphi_n)/\rho(\psi_1,\ldots,\psi_n)) = r(\mathrm{sim}(\varphi_1/\psi_1),\ldots,\mathrm{sim}(\varphi_n/\psi_n))$
$\mathrm{sim}(\lambda x.\varphi/\lambda x.\psi) = \lambda x.(\varphi/\psi)$
$\mathrm{sim}((\varphi_1 \cdot \varphi_2)/\psi)) = \varphi_1' \cdot \varphi_2'$
  where $\varphi_1' = \mathrm{sim}(\varphi_1/\psi)$
    $\varphi_2' = \mathrm{sim}(\varphi_2/\psi')$
    where $\psi' = \mathrm{sim}(\psi/\varphi_1)$
$\mathrm{sim}(\varphi/(\psi_1 \cdot \psi_2)) = \mathrm{sim}(\varphi'/\psi_2)$
  where $\varphi' = \mathrm{sim}(\varphi/\psi_1)$
$\mathrm{sim}(f(\varphi_1,\ldots,\varphi_n)) = f(\mathrm{sim}(\varphi_1),\ldots,\mathrm{sim}(\varphi_n))$
$\mathrm{sim}(\rho(\varphi_1,\ldots,\varphi_n)) = \rho(\mathrm{sim}(\varphi_1),\ldots,\mathrm{sim}(\varphi_n))$
$\mathrm{sim}(\lambda x.\varphi) = \lambda x.\mathrm{sim}(\varphi)$
if *none of the above cases apply* **then**
  print *"incompatible"*

Example 4.14. Consider the PRS from Sect. 4.1 (and from Ex. 4.1). We apply the algorithm of Def. 4.13 to the term $\mathsf{mu}(\lambda x.\rho(x))/\mu(\lambda x.f(x))$. The following is one run of the algorithm:

$$\begin{array}{ll}
\mathrm{sim}(\mathsf{mu}(\lambda x.\rho(x))/\mu(\lambda x.f(x))) & = \\
(\lambda z.z(\mathsf{mu}(\lambda x.z(x))))(\mathrm{sim}(\lambda x.\rho(x)/\lambda x.f(x))) & = \\
(\lambda z.z(\mathsf{mu}(\lambda x.z(x))))(\lambda x.\mathrm{sim}(\rho(x)/f(x))) & = \\
(\lambda z.z(\mathsf{mu}(\lambda x.z(x))))(\lambda x.\rho(\mathrm{sim}(x/x))) & = \\
(\lambda z.z(\mathsf{mu}(\lambda x.z(x))))(\lambda x.\rho(x)) & \twoheadrightarrow_\beta \\
\rho(\mathsf{mu}(\lambda x.\rho(x))) &
\end{array}$$

We see that the result of the previous example is the same as the result of Ex. 4.1. So why does the algorithm work, and doesn't the rewriting system presented in Sect. 4.1? The crucial difference is that the algorithm calculates subexpressions first, and does β-reductions afterwards, while the rewriting system performs β-reductions first and only then calculates the subexpressions. And even if this was fixed in the rewriting system, it would have to be equipped with a depth-first strategy to make it correct.

Proposition 4.15.

1. *If* φ *and* ψ *are* reductions*, then* $\mathrm{sim}(\varphi/\psi)$ *terminates.*
2. *If* φ *is* internally compatible*, then* $\mathrm{sim}(\varphi)$ *terminates.*

Proof. We prove the first item first. If φ and ψ are reductions, then the computation of $\mathrm{sim}(\varphi/\psi)$ proceeds in two stages: first the compositions on the outside of the terms are dealt with, and in this stage the number of composition symbols in the proof term strictly decreases in each step; and

then, when φ and ψ are parallel steps, the length of the proof term strictly decreases in each step.

Secondly, if φ is internally compatible, then an inference $\mathcal{K}$ exists such that $\vdash^{\mathcal{K}} \varphi/ \succcurlyeq \chi$. The second item can be proved by induction on $\mathcal{K}$, using Lemma 4.4 and Lemma 4.5. □

Termination *in general* is hard to show. If a proof term φ is not internally compatible, an inference of $\varphi \succcurlyeq \chi$ is not at our disposal, so we cannot use induction on the inference. The problem is then the cases which deal with composition. In these cases the size of the terms which are passed recursively to the function, may actually be larger than the size of the term under consideration.

Conjecture 4.16. $\mathrm{sim}(\varphi)$ terminates for *all* slash-dot terms φ.

The main result of the paper does not depend on this conjecture, although, because of its not being proved, a small detour has to be followed in Sect. 5.1.

Proposition 4.17. $\mathrm{sim}(\varphi) = \chi$ *if and only if* $\varphi \succcurlyeq \chi$.

Proof. The 'only if' side is proved by recursively building an inference of $\varphi \succcurlyeq \chi$. The 'if' side is easily proved by using Lemma 4.4 and Lemma 4.5. □

5 Orthogonality

In this section we relate compatibility with the well-known notion of orthogonality. In order to define orthogonality, we need to define overlap, and this is done by associating with each proper step a set of redex positions, and then looking at the intersection of the redex positions of two coinitial proper steps.

Positions are sequences of natural numbers. If P is a set of positions, and p is a position, we write $p \star P$ for $\{pq \mid q \in P\}$. First, we need to define the set of all positions of a term. Let $\Box$ denote the empty context.

$$\begin{array}{rcl} \mathcal{P}os(\Box) & = & \emptyset \\ \mathcal{P}os(x(s_1, \ldots, s_n)) & = & \{\epsilon\} \cup \bigcup_{0<i\leq n} i \star \mathcal{P}os(s_i) \\ \mathcal{P}os(f(s_1, \ldots, s_n)) & = & \{\epsilon\} \cup \bigcup_{0<i\leq n} i \star \mathcal{P}os(s_i) \\ \mathcal{P}os(\lambda x.s) & = & \{\epsilon\} \cup 1 \star \mathcal{P}os(s) \end{array}$$

where x is a variable and f a function symbol.

Now, let φ be a proper step. We define the set of redex positions of φ, written $\mathcal{RP}os(\varphi)$, as:

$$\begin{array}{rcl} \mathcal{RP}os(x(\varphi_1, \ldots, \varphi_n)) & = & \bigcup_{0<i\leq n} i \star \mathcal{RP}os(\varphi_i) \\ \mathcal{RP}os(f(\varphi_1, \ldots, \varphi_n)) & = & \bigcup_{0<i\leq n} i \star \mathcal{RP}os(\varphi_i) \\ \mathcal{RP}os(\lambda x.\varphi_0) & = & 1 \star \mathcal{RP}os(\varphi_0) \\ \mathcal{RP}os(\rho(\varphi_1, \ldots, \varphi_n)) & = & \mathcal{P}os(l(\Box, \ldots, \Box)) \end{array}$$

Note that, since φ is a proper step, in the last equation there are no more rule symbols in the φ_i.

Two coinitial proper steps φ, ψ are *overlapping* if $\mathcal{RP}os(\varphi) \cap \mathcal{RP}os(\psi) \neq \emptyset$. A left-linear PRS is *orthogonal*, if all pairs of different, coinitial proper steps are non-overlapping.

This definition has an infinite flavour: there are infinitely many steps one has to check. Fortunately, it is well-known that an equivalent notion of orthogonality exists, based on critical pairs [7]. Since a finite PRS has only finitely many possible critical pairs, this makes the question whether a PRS is orthogonal or not decidable. We stick to the step-based definition for convenience.

5.1 Compatibility Is Orthogonality

In this subsection we will prove that compatibility and orthogonality coincide. The difficult part of the proof, but also the most important one, is to show that orthogonality implies compatibility. One way of doing so is by contraposition: we run the algorithm of Def. 4.13 and analyse in which situations it prints *incompatible*, and show that the PRS is not orthogonal in each of these cases. There is one problem: we have not succeeded in proving that the algorithm actually always terminates, so we have to follow a small detour: we transform the incompatible proof terms into incompatible reductions, and then feed those to the algorithm.

Lemma 5.1. *If $l(\varphi_1, \ldots, \varphi_n) \cdot \rho(\psi_1, \ldots, \psi_n)$ is compatible with χ, then $\rho(\varphi_1 \cdot \psi_1, \ldots, \varphi_n \cdot \psi_n)$ is compatible with χ.*

Proof. Let $\varphi = l(\varphi_1, \ldots, \varphi_n) \cdot \rho(\psi_1, \ldots, \psi_n)$ $\varphi' = \rho(\varphi_1 \cdot \psi_1, \ldots, \varphi_n \cdot \psi_n)$. Since φ and χ are compatible, it must be the case that $\vdash^{\mathcal{K}} \varphi/\chi \succcurlyeq \upsilon$ for some υ. By Lemma 4.5 there is an inference that ends as follows:

$$\frac{\chi/l(\varphi_1, \ldots, \varphi_n) \succcurlyeq \chi' \quad \chi'/\rho(\psi_1, \ldots, \psi_n) \succcurlyeq \upsilon}{\chi/(l(\varphi_1, \ldots, \varphi_n) \cdot \rho(\psi_1, \ldots, \psi_n)) \succcurlyeq \upsilon}$$

where the premisses are compatible. This is only possible if $\chi = l(\vec{\varphi})$ (and $\chi' = l(\vec{\chi'})$) or $\chi = \rho(\vec{\varphi})$ (and $\chi' = \rho(\vec{\chi'})$) and, by assuming that the rule suggested by Lemma 4.4 is applied, φ_i, χ_i and χ'_i, ψ_i are compatible. In the case the $\chi = \rho(\vec{\varphi})$, we now prove $\varphi'/\chi \succcurlyeq \upsilon'$, for some υ', by the following inference:

$$\frac{\cdots \qquad \dfrac{\chi_i/\varphi_i \succcurlyeq \chi'_i \quad \chi'_i/\psi_i \succcurlyeq \upsilon_i}{\chi_i/(\varphi_i \cdot \psi_i) \succcurlyeq \upsilon_i} \qquad \cdots}{\rho(\chi_1, \ldots, \chi_n)/\rho(\varphi_1 \cdot \psi_1, \ldots, \varphi_n \cdot \psi_n) \succcurlyeq r(\upsilon_1, \ldots, \upsilon_n)}$$

Note that, in fact, $\upsilon' = \upsilon$. The case that $\chi = l(\vec{\varphi})$ is proved by a similar inference. □

Theorem 5.2. *Let $\mathcal{H}$ be an PRS. $\mathcal{H}$ is orthogonal, if and only if $\mathcal{H}$ is compatible.*

Proof. We first prove the right-to-left implication. Assume that all coinitial reductions are compatible. This implies that all coinitial multisteps φ, ψ are compatible, i.e. an inference $\mathcal{K}$ exists such that $\vdash^{\mathcal{K}} \varphi/\psi \succcurlyeq \chi$. We easily prove, by induction on $\mathcal{K}$, that φ, ψ are non-overlapping.

To show the left-to-right implication, assume, by contraposition, that φ, ψ are coinitial, but not compatible. Consider the (meta-level) rewrite system which consists of all rules of the form

$$\rho(\varphi_1 \cdot \psi_1, \ldots, \varphi_n \cdot \psi_n) \Rightarrow l(\varphi_1, \ldots, \varphi_n) \cdot \rho(\psi_1, \ldots, \psi_2)$$

where $\rho : l \to r$ is a rule. It is not difficult to see that this rewrite system is strongly normalizing, and that its normal forms are actually reductions. So, applying this rewriting system to φ and ψ yields reductions φ', ψ', respectively. By Prop. 4.15, $\mathrm{sim}(\varphi'/\psi')$ terminates, and by (the contraposition of) Lemma 5.1, φ' and ψ' are not compatible.

Let φ_0/ψ_0 be the slash-dot term which was passed to sim in the last step before it terminated; φ_0 and ψ_0 must be multisteps. By Prop. 4.17 the algorithm prints *incompatible*. By coinitiality of φ_0 and ψ_0, it cannot cannot be the case that $\varphi_0 = f(\varphi_1, \ldots, \varphi_n)$ and $\psi_0 = g(\psi_1, \ldots, \psi_n)$, where $f \neq g$. So, the following must apply: $\varphi_0 = \rho(\varphi_1, \ldots, \varphi_n)$ and $\psi_0 \neq_{\beta\eta} l(\psi_1, \ldots, \psi_n)$. There are two possible causes of this. The first is that ψ_0 has a rule symbol within the redex pattern of l. But then overlapping, coinitial proper steps φ_0' and ψ_0' can be constructed by replacing all rule symbols, except the overlapping ones, of φ_0 and ψ_0, respectively, by their left-hand sides. The second possible cause is that one of the ψ_i occurs twice in $l(\psi_1, \ldots, \psi_n)$. However, then l cannot be left-linear. Both cases imply non-orthogonality. (The third 'cause' is that ψ_0 has a $\cdot$ inside the redex pattern of φ_0, but this cannot happen because compositions are moved outwards over function symbols and abstractions by the functorial identities, and l consists only of function symbols and abstractions.) The same argument can be applied to the symmetrical case. □

5.2 Residuals of Orthogonal PRSs

In this subsection we prove the main result of the paper, namely that an orthogonal PRS, together with the unit and projection operator, forms a residual system. The hard work has already been done; we just need to put together the results obtained so far.

Theorem 5.3. *If $\mathcal{H}$ is an orthogonal PRS, then $\langle \mathcal{H}, 1, // \rangle$ is a residual system.*

Proof. By Theorem 5.2, $\mathcal{H}$ is compatible, and therefore, by Theorem 4.7, $\langle \mathcal{H}, 1, // \rangle$ is a residual system. □

It is well-known that orthogonal PRSs are confluent, as was proved in, among others, [7, 12, 15]. Here, we obtain a new proof based on the residual theory developed in this paper. The proof emerges as a simple corollary of the main result.

Corollary 5.4. *Orthogonal PRSs are confluent.*

Proof. Let $\mathcal{H}$ be an orthogonal PRS. By Theorem 5.3, $\mathcal{H}$ is a residual system, and thus by Theorem 2.5, $\mathcal{H}$ is confluent. □

6 Concluding Remarks

In this paper, we have shown that orthogonal PRSs form a residual system. As a consequence, all results for residual systems are inherited, such as the notion of permutation equivalence and confluence. We have also given an algorithm which simplifies slash-dot terms to proof terms, and we have proven, in two special cases, that the algorithm terminates.

For the future, the following research is interesting. Firstly, it is interesting to find a proof (or a refutation) of the claim that the algorithm mentioned in the previous paragraph does *always* terminate. Not only is this interesting in its own right, it is my view that such a proof may aid us in the understanding of termination of higher-order rewriting, and provide new proof methods.

Secondly, it is interesting to see if the framework can be generalized to non-orthogonal, left-linear PRSs, or even arbitrary PRSs. For this to work, an error symbol must be added, to indicate non-compatibility. For the first-order case, the same approach was succesfully applied to left-linear TRSs in [13].

Acknowledgements

I wish to thank Vincent van Oostrom and the anonymous referees of RTA'03 for their valuable remarks on preliminary versions of this paper.

References

[1] Barnaby P. Hilken. Towards a proof theory of rewriting: The simply typed 2λ-calculus. *Theoretical Computer Science*, 170:407–444, 1996.

[2] Gérard Huet. Residual theory in λ-calculus: A formal development. *Journal of Functional Programming*, 4(3):371–394, 1994.

[3] Gérard Huet and Jean-Jacques Lévy. Computations in orthogonal rewriting systems, part I + II. In J.L. Lassez and G.D. Plotkin, editors,

Computational Logic – Essays in Honor of Alan Robinson. MIT Press, 1991.

[4] Zurab Khasidashvili and John Glauert. Relating conflict-free stable transition systems and event models via redex families. *Theoretical Computer Science*, 286(1):65–95, 2002.

[5] Cosimo Laneve and Ugo Montanari. Axiomatizing permutation equivalence. *Mathematical Structures in Computer Science*, 6(3):219–215, 1996.

[6] Jean-Jacques Lévy. *Réductions correctes et optimales dans le λ-calcul*. Thèse de doctorat d'état, Université Paris VII, 1978.

[7] Richard Mayr and Tobias Nipkow. Higher-order rewrite systems and their confluence. *Theoretical Computer Science*, 192:3–29, 1998.

[8] Paul-André Melliès. Axiomatic rewriting theory VI: Residual theory revisited. In Sophie Tison, editor, *13th International Conference on Rewriting Techniques and Applications*, pages 24–50, 2002.

[9] José Meseguer. Conditional rewriting logic as a unified model of concurrency. *Theoretical Computer Science*, 96:73–155, 1992.

[10] Dale Miller. A logic programming language with lambda-abstraction, function variables, and simple unification. *Journal of Logic and Computation*, 1(4), 1991.

[11] Tobias Nipkow and Christian Prehofer. Higher-order rewriting and equational reasoning. In W. Bibel and P. Schmitt, editors, *Automated Deduction — A Basis for Applications, Volume I: Foundations*, number 8 in Applied Logic Series, pages 399–430. Kluwer Academic Press, 1998.

[12] Vincent van Oostrom. *Confluence for Abstract and Higher-Order Rewriting*. PhD thesis, Vrije Universiteit, Amsterdam, 1994.

[13] Vincent van Oostrom and Roel de Vrijer. *Equivalence of Reductions*, chapter 8 of [18]. 2003.

[14] Vincent van Oostrom and Roel de Vrijer. Four equivalent equivalences of reductions. In *Proceedings of WRS'02 (ENTCS 70.6)*, 2003. Downloadable at: `http://www.elsevier.nl/locate/entcs/`.

[15] Femke van Raamsdonk. *Confluence and Normalisation for Higher-Order Rewriting*. PhD thesis, Vrije Universiteit, Amsterdam, 1996.

[16] Femke van Raamsdonk. Higher-order rewriting. In *10th International Conference on Rewriting Techniques and Applications*, 1999.

[17] Eugene W. Stark. Concurrent transition systems. *Theoretical Computer Science*, 64(3):221–269, 1989.

[18] Terese. *Term Rewriting Systems*. Number 55 in Camb. Tracts in Theor. Comp. Sc. Cambridge University Press, 2003.

[19] D. A. Wolfram. *The Clausal Theory of Types*. Number 21 in Camb. Tracts in Theor. Comp. Sc. Cambridge University Press, 1993.