Infinitary Combinatory Reduction Systems[☆]

Jeroen Ketema[a,1], Jakob Grue Simonsen[b]

[a]*Research Institute of Electrical Communication, Tohoku University*
2-1-1 Katahira, Aoba-ku, Sendai 980-8577, Japan
[b]*Department of Computer Science, University of Copenhagen (DIKU)*
Universitetsparken 1, 2100 Copenhagen Ø, Denmark

Abstract

We define infinitary Combinatory Reduction Systems (iCRSs), thus providing the first notion of infinitary higher-order rewriting. The systems defined are sufficiently general that ordinary infinitary term rewriting and infinitary λ-calculus are special cases.

Furthermore, we generalise a number of known results from first-order infinitary rewriting and infinitary λ-calculus to iCRSs. In particular, for fully-extended, left-linear iCRSs we prove the well-known compression property, and for orthogonal iCRSs we prove that (1) if a set of redexes $\mathcal{U}$ has a complete development, then all complete developments of $\mathcal{U}$ end in the same term and that (2) any tiling diagram involving strongly convergent reductions S and T can be completed iff at least one of S/T and T/S is strongly convergent.

We also prove an ancillary result of independent interest: A set of redexes in an orthogonal iCRS has a complete development iff the set has the so-called finite jumps property.

Keywords: higher-order term rewriting, combinatory reduction systems, infinitary rewriting

---

1. Introduction

In the following pages we extend the theory of infinitary writing to a generic higher-order format. Thus, the current paper, together with its companion papers [3, 4], for the first time unifies the theory of infinitary (first-order) term rewriting and infinitary λ-calculus within one setting.

As with all forms of infinitary rewriting, it is most convenient to use a known finite formalism as the basis for the theory. Our concrete vehicle is a variant

[☆]Parts of this paper previously appeared in [1] and [2].

Email addresses: `jketema@nue.riec.tohoku.ac.jp` (Jeroen Ketema), `simonsen@diku.dk` (Jakob Grue Simonsen)

[1]This author was partially funded by the Netherlands Organisation for Scientific Research (NWO) under FOCUS/BRICKS grant number 642.000.502.

of higher-order rewriting called *Combinatory Reduction Systems (CRSs)*. This variant is used because it offers a clear separation between terms and meta-terms, which avoids some technical difficulties; of all formalisms offering such a separation, CRSs seem to be most widely used.

In the remainder of this introduction we motivate the extension from a first-order to a higher-order setting from a programming language perspective, we explain the difficulties in constructing the extension, and we outline the new techniques needed for said construction.

1.1. Motivation

Term rewriting is a useful tool in the study of declarative programming, logic, universal algebra, and automated theorem proving. In term rewriting, equations are viewed as directed replacement rules where left-hand sides are replaced by right-hand sides, but not vice versa.

From a programming perspective, term rewriting affords easy modelling of function declaration and evaluation in declarative programming languages such as Haskell, Lisp, ML, and Prolog. In these languages, functions are defined by (oriented) equations, and a function call `foo(a)` is, conceptually, evaluated by replacing `foo(a)` with the body of the function `foo`(x) where all occurrences of the formal parameter x in the body of `foo` are replaced by the actual parameter `a` [5]. This notion of replacement is essentially a rewrite step in term rewriting.

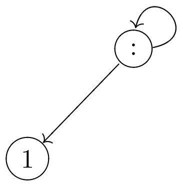
To allow for the ability to easily model many different kinds of languages, the syntax of term rewriting often has some complexities that we do not encounter in programming languages. This is especially so for the higher-order systems that are the subject of this paper: They often require us to write variable bindings explicitly, while applicative forms are more commonly found in declarative programming languages.

One particularly interesting feature of modern programming languages is the possibility to work explicitly with data structures that are *semantically* infinite — even though, in all concrete applications, program execution only examines a finite part of the data structure. For example, in Haskell, the expression $[\mathtt{0}..]$ denotes the *infinite* list of non-negative integers $[\mathtt{0}, \mathtt{1}, \mathtt{2}, \mathtt{3}, \mathtt{4}, \ldots]$. In such languages, semantically infinite lists make perfect sense due to *lazy evaluation*: No list element is actually computed until program execution specifically asks for its evaluation [6].

The theory of first-order programming with potentially infinite data structures has been developed successfully since the eighties in the form of *infinitary rewriting* [7, 8, 9, 10] and *graph rewriting* [11, 12].

In infinitary rewriting, infinite terms are defined as elements of a certain metric space, and (potentially) infinite computations must be convergent sequences in this metric space, both in order to obtain a well-defined result in the limit and to ensure that a computation can be halted after a finite number of steps in such a way that we know that a well-defined partial result has been computed (e.g. the first n elements of a list).

In graph rewriting, terms are finite graphs with possibly shared nodes and rewrite rules are pairs of such graphs; for example, the following graph represents the infinite list of ones:



The theory of both infinitary rewriting and graph rewriting is predominantly first-order, and prior higher-order research efforts have mostly been restricted to variants of λ-calculus (see below). Unfortunately, first-order constructs are insufficient for the modern programmer — his arsenal includes *higher-order* functions that take other functions as arguments. Consider for instance the function `map` which in succession applies a function to each element of a list:

$$\begin{aligned} \texttt{map}\, f\, (x{:}xs) &= (f(x)) : \texttt{map}\, f\, xs \\ \texttt{map}\, f\, [] &= [] \end{aligned}$$

While it is possible to encode functions such as `map` in first-order term rewriting, it is hard to do so without introducing a number of awkward workarounds: In first-order rewriting, the function f must be treated as a constant, whence a single rule cannot capture all possibilities for f; indeed encodings must use defunctionalisation [13, 14, 15], applicative notation [16], or some similar technique. All of these require restating the definitions using an extended syntax and possibly some bookkeeping to make certain that if something is applied to an argument, that something is actually a function.

The classical "theorist's approach" to handling higher-order functions such as `map` is to simply appeal to the machinery of λ-calculus [17]. Function evaluation in λ-calculus is expressed through its single rewrite rule

$$(\lambda x.M)N \to_\beta M\{N/x\}\,,$$

where $M\{N/x\}$ is the substitution of the parameter N for the free occurrences of variable x in the function body M. The extension of λ-calculus to infinite terms and computations [9] affords an idealised model of function evaluation, including higher-order function evaluation that can handle constructs such as `map`, but it is quite awkward to take a real-world functional program and encode it directly in λ-calculus — witness for example the encoding of natural numbers as Church numerals [17].

A much more straightforward encoding is possible by using one of the variants of *higher-order rewriting* [18, 19, 20, 21, 22, 23, 24]. For example, in the syntax of one of these variants — *Combinatory Reduction Systems (CRSs)* — the definition of `map` becomes:

$$\begin{aligned} \texttt{map}([z]F(z), \texttt{cons}(X, XS)) &\to \texttt{cons}(F(X), \texttt{map}([z]F(z), XS)) \\ \texttt{map}([z]F(z), \texttt{nil}) &\to \texttt{nil} \end{aligned}$$

that is, a de-sugared version of the declaration of `map` where variable bindings have been made explicit.

Most higher-order rewriting formats combine two notions: function-symbols-as-variables, and the ability to have bound variables [20, 25, 26, 23]. The first notion ensures that higher-order functions such as `map` may be encoded succinctly, and the second that formalisms such as λ-calculus also have succinct representations.

There are at least two ways to treat higher-order functions in the setting of lazy programming: One is the notion of higher-order graph rewriting which, alas, does not yet have a large array of generally applicable results [27]. The other is a true extension of infinitary rewriting to the higher-order setting through higher-order term rewriting.

The aim of the present paper is to provide such an extension of infinitary rewriting to the higher-order setting: We define *infinitary* Combinatory Reduction Systems, an extension of one of the oldest formats of (finitary) higher-order rewriting [18, 19, 20].

Our work allows evaluation of, say, `map` on potentially infinite lists. With appropriate shorthands, for example writing $[\mathtt{1},\mathtt{2},\ldots]$ instead of

$$\mathtt{cons}(\mathtt{succ}(\mathtt{0}),\mathtt{cons}(\mathtt{succ}(\mathtt{succ}(\mathtt{0}))),\ldots)\,,$$

we allow for rewriting of terms such as

$$\mathtt{map}([z]f(z),[\mathtt{1},\mathtt{2},\ldots])\,,$$

where f is some function (see also Figure 1). In addition, the methods developed in this paper will allow us to prove pertinent results about terms as the above. For example, we may prove (a) normalisation results showing that repeated application of the rules for `map` will yield a well-defined new infinite list, and (b) confluence results showing that any sufficiently well-behaved function substituted for f will yield identical results regardless of the way it is evaluated.

1.2. Moving beyond first-order rewriting

A number of very useful proof methods have been devised for infinitary rewriting in the first-order setting and in the restricted higher-order setting of infinitary λ-calculus. *The* natural question to ask is whether these apply when we move to the world of general higher-order terms.

The question is complicated by the existence of various formats of higher-order rewriting and their dependence on a suitable meta-calculus to handle bound variables and substitution [25]. To illustrate the problems, we show how matters go awry with λ-calculus as meta-calculus. The exact variant of (typed) λ-calculus is irrelevant for purposes of illustration; the reader just needs to know that in order to perform a rewrite step in a higher-order rewriting system, some 'bookkeeping' β-reductions or η-expansions might need to be performed before and after the rewrite step itself.

The main three problems are described below. Note that the first two problems are already encountered in infinitary λ-calculus:

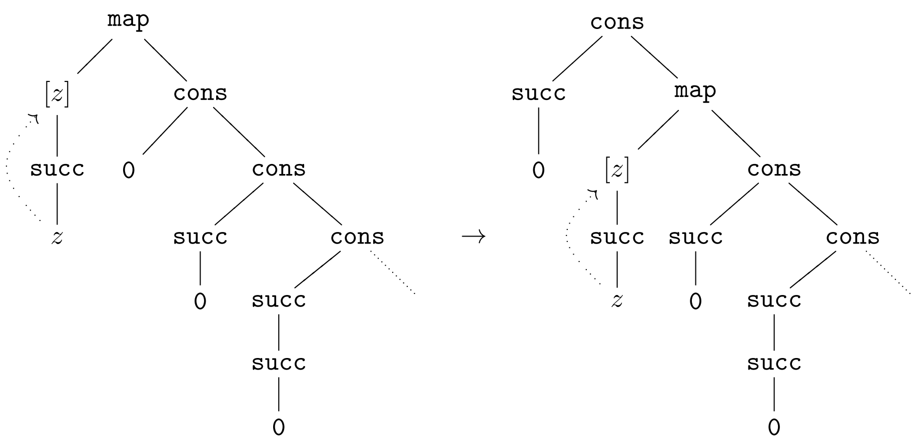


Figure 1: The first rewrite step when using `map` to apply the successor function to each element of the infinite list of natural numbers

- *Rewrite steps may nest disjoint subterms.*

  In the higher-order setting disjoint subterms may become nested in rewrite steps, a phenomenon due to higher-order systems modelling function application. Considering the rules for `map` from the previous section, we see that the function substituted for F is applied to X: rewriting *nests* X inside F. When infinite terms and infinite reductions are considered, a new phenomenon appears: It becomes possible to 'push' a redex out of a term in an infinite number of steps by use of nesting: Consider the rewrite rule $f(\lambda x.Z(x)) \to Z(f(\lambda x.Z(x))$ that nests the left-hand side of the rule in Z. We have that $f(\lambda x.g(x))$ reduces in a finite number of steps to $g(g(\ldots g(f(\lambda x.g(x)))))$ and in an infinite number of steps it reduces to $g(g(\ldots g(\ldots)))$, pushing $f(\lambda x.g(x))$ out of the term and thus erasing a redex in a non-standard way.

  Due to the above behaviour, the well-known *Strip Lemma*, often used for proving confluence, will fail to hold in many situations. Adapting the counterexample to confluence from infinitary λ-calculus [28] and employing the rule $g(Z) \to Z$ next to $f(\lambda x.Z(x)) \to Z(f(\lambda x.Z(x))$, we have that $f(\lambda x.g(x))$ not only reduces to $g(g(\ldots g(\ldots)))$ but also in one step to $f(\lambda x.x)$. These latter two terms only reduce to themselves and do not have a common reduct as required by the Strip Lemma.

  In similar vein, the crucial *compression property* might fail: There are reductions that cannot be compressed to have length at most ω, where ω is the least infinite ordinal. Failure depends on interaction with the third problem mentioned below; for details we refer the reader to Example 5.5.

- *Rewrite rules may encode non-occur checks.*

A different problem specific to the higher-order setting is that rules may encode 'non-occur checks'. Indeed, the side condition on η-reduction in the λ-calculus rule

$$\lambda x.Mx \to_\eta M \qquad \text{if } x \text{ does not occur free in } M$$

is the most well-known example of such a check, and is often internalised in the rewrite rules of higher-order systems. For example, the CRS version of the η-rule $\texttt{lam}([x]\texttt{app}(Z, x)) \to Z$ does not require a side condition, x is simply omitted as argument of Z. When infinite terms and infinite reductions are considered, it is possible to create redexes after an infinite number of steps due to such non-occur checks by 'pushing' a variable out of a term in an infinite number of steps. It is well-known that due to this behaviour the *compression property* might fail for infinitary λ-calculus with η-reduction: There are reductions involving η-steps that cannot be compressed to have length at most ω [9].

- *Rewrite steps may fail to be well-defined.*

  Consider the apparently innocent infinite term $t = Z(Z(\ldots Z(\ldots)))$ consisting of the variable Z applied to itself an infinite number of times. If we have a rewrite rule, say $f(\lambda x.Z(x)) \to t$, then we obviously want to perform steps such as $f(\lambda x.g(x)) \to g(g(\ldots g(\ldots)))$. But the term $f(\lambda x.x)$ is also a legal (finitary) term, whence the rewrite step $f(\lambda x.x) \to (\lambda x.x)((\lambda x.x)(\ldots(\lambda x.x)))$ should be allowed as well. However, the term on the right-hand side contains an infinite number of β-redexes that are part of the meta-calculus. Moreover, the term does not have an infinite normal form with respect to β-reduction: The right-hand side only β-reduces to itself. In higher-order systems of all ilks, well-behaved terms need to be in normal form with respect to the meta-calculus, whence the rewrite step above *cannot* be allowed.

  The behaviour of a single rewrite step can also mimic the phenomenon that destroys confluence of orthogonal systems in first-order infinitary rewriting and infinitary λ-calculus: Let $K^* = \lambda x.\lambda y.y$; the rewrite rule

  $$g(\lambda x.\lambda y.Z(y)) \to Z(a, Z(b, Z(a, Z(b, Z(\ldots)))))$$

  should admit the rewrite step

  $$g(K^*) \to K^*(a, K^*(b, K^*(a, K^*(b, K^*(\ldots))))) \,.$$

  Again, the right-hand side includes an infinite number of β-redexes that are part of the meta-calculus. Moreover, there are two possible *distinct* β-reducts of the right hand side that neither have normal forms, nor a common reduct:

  $$K^*(a, K^*(a, K^*(a, K^*(a, K^*(\ldots)))))$$

  and

  $$K^*(b, K^*(b, K^*(b, K^*(b, K^*(\ldots))))) \,.$$

As a sine qua non of higher-order rewriting is the ability to encode TRSs *and* λ-calculus, any attempt at studying higher-order infinitary rewriting needs to uncover sufficient criteria for the compression property to hold. Moreover, the last problem above shows that constraints need to be enforced on the allowed rewrite rules for rewriting to be well-defined. It is doubtful whether the Strip Lemma needs to be recovered, as it already fails for infinitary λ-calculus [9, 10], and we do not aim to recover it.

It turns out that the above issues can be addressed in the context of CRSs by employing two restrictions that are only mildly intrusive:

- Only allow *fully-extended* systems where bound variables *must* occur in the arguments of all meta-variables in the scope of the binding. A further benefit is that this restriction is already reasonably well-known in rewriting [29, 30].
- Disallow 'infinite towers of applications': The last of the above three problems is due to unconstrained (and infinite) application of variables in the right-hand side of rules. This is the basis for the *finite chains property* of right-hand sides of rules, as introduced below. We note that when using ordinary higher-order term rewriting systems to rewrite infinite terms, the finite chains property will be trivially satisfied as all rules have finite right-hand sides. Moreover, we note that infinite right-hand sides satisfying finite chains property are expressive enough to encode the infinite right-hand sides allowed in first-order infinitary rewriting [8].

The need to impose constraints on rewrite rules necessitates a detailed low-level description of (meta-)term and rule formation in infinitary Combinatory Reduction Systems; a significant part of the paper is devoted to this (Sections 2 through 4).

With the help of the above constraints, we can adapt the proofs of the crucial compression property from first-order infinitary rewriting and infinitary λ-calculus. Moreover, the constraints help to set up the machinery necessary for proving the basic properties of developments, which are a technical vehicle for proving confluence and normalisation. From that point on, there is no free lunch: Proving sufficient conditions for a set of redexes to have a well-defined development is significantly hampered by the presence of nestings more complicated than in λ-calculus, that is, more complicated than simply nesting a term N in a term M. These nestings may significantly change the internal sorting of the subterms of a term. Only by highly meticulous and methodical tracing of subterms does it appear possible to obtain proofs of the usual results on developments; we have adapted the elegant approach of *finite jumps* from [10] to do this, relegating the details to the appendix of this paper.

1.3. Contributions, background, and related work

The main contributions of the paper are:

1. The extension of infinitary rewriting to the higher-order setting. In the process, we exhibit a range of examples showing the main pitfalls in extending infinitary reasoning to higher-order systems.
2. A compression property for fully-extended, left-linear iCRSs, thus generalising the most basic result for all weaker notions of infinitary rewriting.
3. Theorems 6.12 and 7.2 on complete developments and tiling diagrams for orthogonal iCRSs. Theorem 6.12 shows that if a set of redexes has a complete development, then all complete developments of that set will end in the same term. Theorem 7.2 shows that both projections of two reductions across each other will be well-defined if at least one of them is. Among other things, this result is an important stepping stone towards confluence results for iCRSs.
4. A thorough exposition of a novel proof technique for proving results about developments in the infinitary setting. We extend the technique pioneered in [10] using the finite jumps property at the price of slightly more involved intermediate results.

Background and related work. The original investigation of sets of infinite terms as metric spaces was pioneered by Arnold and Nivat [31]. Rewriting of infinite first-order terms was first considered in the liberal setting of so-called *weak* or *Cauchy* convergence, by Dershowitz, Kaplan, and Plaisted [32, 33, 7]. Kennaway, Klop, Sleep, and de Vries, inspired by Farmer and Watro's paper [34], considered *strong convergence*: A more restrictive notion of infinite reduction where reductions must necessarily employ rewrite steps deeper and deeper in terms. This notion of infinite reduction has become the de facto standard in infinitary rewriting [8, 35, 36]. Recent advances include alternative approaches to defining infinite terms and their accompanying rewrite relation [37, 38], modular properties [39], and uniform normalisation [40].

Lisper has defined a separate notion of infinitary Combinatory Reduction Systems [41] and has proven a number of preliminary results for these. His notion of reduction rules is a special case of the one in the present paper: Only finite right-hand sides are allowed. Moreover, when proving a higher-order analogue to the well-known Strip Lemma, Lisper requires that no nesting of meta-variables occurs in right-hand sides of rules. This restriction materialises in several crucial places, for example when unfoldings for higher-order rules are considered, Lisper recommends switching to a *first-order* combinator system. Our treatment of confluence does not require any such restriction; we allow for infinite right-hand sides with arbitrary nestings of meta-variables (satisfying the finite chains property), where the extension to infinite right-hand sides does not complicate our proofs severely.

An infelicity in [41] is that the compression property suffers from a subtle error in the proof; indeed we have a counterexample showing that when rules are not fully-extended, compression may fail to hold, even for systems with finite right-hand sides (see Example 5.4). We show in the present paper that requiring fully-extendedness completely recovers the compression property.

1.4. Structure and reader's guide

The layout of the paper is as follows: Section 2 contains preliminary definitions. Sections 3 and 4 introduce terms and rewriting, respectively. Section 5 proves that every well-behaved rewrite sequence of transfinite length can be compressed to one of length at most ω. Section 6 contains definitions and proofs of generalisations of standard results for *developments*. Section 7 generalises a result from [10] on tiling diagrams of vital importance for showing confluence of fully-extended, orthogonal systems. Section 8 concludes.

Readers with prior knowledge of rewriting and the syntax of (ordinary, finite) Combinatory Reduction Systems can make do by noting that meta-terms are simply formed by interpreting the rules for meta-term formation top-down instead of bottom-up. A serious caveat is that 'infinite chains of immediately nested meta-variables' must be avoided — see Section 3.3.

For the reader with prior knowledge of infinitary rewriting, we introduce metrics on terms and transfinite reductions in the usual manner; compression requires a more substantial analysis than usual due to the fact that nestings can occur in reduction steps — see Section 5. Due to the problem of redexes being created after an infinite number of steps by variables being 'pushed out' of a term, we choose to require that all rules are fully-extended — see Definition 4.10. We believe fully-extendedness to be both technically simple to understand and sufficiently liberal to allow study of most interesting systems.

2. Preliminaries

Prior knowledge of CRSs [19, 20, 22] and infinitary rewriting [10] is not a requirement, but will greatly improve the reader's understanding of the text. Throughout, the notion 'infinitary Term Rewriting System' is abbreviated as iTRS and 'infinitary λ-calculus' is abbreviated as iλc.

We assume a signature Σ, each element of which has finite arity. We also assume a countably infinite set of variables and, for each finite arity, a countably infinite set of meta-variables. We denote the least infinite ordinal by ω, and arbitrary ordinals by α, β, γ, and so on. Moreover, we use $\mathbb{N}$ to denote the set of natural numbers, including zero.

To define terms and meta-terms, we first need to introduce finite meta-terms and positions. The finite meta-terms and terms below are simply the meta-terms and terms of CRSs, where the meta-terms are objects used in rule formation and the terms are the objects rewritten.

Definition 2.1. The set of *finite meta-terms* is defined inductively by the following rules, where s and $s_1, \ldots, s_n$ are again finite meta-terms:

1. each variable x is a finite meta-term,
2. if x is a variable, then $[x]s$ is a finite meta-term,
3. if Z is a meta-variable of arity n, then $Z(s_1, \ldots, s_n)$ is a finite meta-term,
4. if $f \in \Sigma$ has arity n, then $f(s_1, \ldots, s_n)$ is a finite meta-term.

A finite term is a finite meta-term without occurrences of meta-variables.

A finite meta-term of the form $[x]s$ is called an *abstraction*. Each occurrence of the variable x in s is *bound* in $[x]s$, and each subterm of s is said to occur in the *scope* of the abstraction. If s is a finite meta-term, we denote by $root(s)$ the root symbol of s. Following the definition of finite meta-terms, we define $root(x) = x$, $root([x]s) = [x]$, $root(Z(s_1, \dots, s_n)) = Z$, and $root(f(s_1, \dots, s_n)) = f$.

Example 2.2. Let `abs`, `app`, `nil`, `map`, and `cons` be function symbols with `abs` unary, `app` binary, `nil` nullary, and `map` and `cons` again binary. Then we have that $\mathtt{app}(\mathtt{abs}([x]Z(x), Z')$ and $\mathtt{map}([x]F(x), \mathtt{cons}(X, XS))$ are finite meta-terms, with Z, Z', F, X, and XS meta-variables of appropriate arity.

We shall not define rewrite rules or the rewrite relation of CRSs here, instead giving it for iCRSs in Section 4. For now, the reader may satisfy herself that rewrite rules are pairs of terms satisfying a few very natural requirements. The function *map* may be expressed as a CRS with two rules:

$$\begin{aligned} \mathtt{map}([x]F(x), \mathtt{cons}(X, XS)) &\to \mathtt{cons}(F(X), \mathtt{map}([x]F(x), XS)) \\ \mathtt{map}([x]F(x), nil) &\to \mathtt{nil} \end{aligned}$$

Likewise, the β-rule of λ-calculus may be expressed by the single rule

$$\mathtt{app}(\mathtt{abs}([x]Z(x), Z')) \to Z(Z')$$

Intuitively, positions denote the 'locations' of subterms; they are defined as follows.

Definition 2.3. The *set of positions* of a finite meta-term s, denoted $\mathcal{P}os(s)$, is the set of finite strings over $\mathbb{N}$, with ϵ the empty string, such that:

1. if $s = x$ for some variable x, then $\mathcal{P}os(s) = \{\epsilon\}$,
2. if $s = [x]t$, then $\mathcal{P}os(s) = \{\epsilon\} \cup \{0 \cdot p \mid p \in \mathcal{P}os(t)\}$,
3. if $s = Z(t_1, \dots, t_n)$, then $\mathcal{P}os(s) = \{\epsilon\} \cup \{i \cdot p \mid 1 \leq i \leq n,\, p \in \mathcal{P}os(t_i)\}$,
4. if $s = f(t_1, \dots, t_n)$, then $\mathcal{P}os(s) = \{\epsilon\} \cup \{i \cdot p \mid 1 \leq i \leq n,\, p \in \mathcal{P}os(t_i)\}$.

The *depth* of a position p, denoted $|p|$, is the number of characters in p. Given p, $q \in \mathcal{P}os(s)$, we write $p \leq q$ and say that p is a *prefix* of q, if there exists an $r \in \mathcal{P}os(s)$ such that $p \cdot r = q$. If $r \neq \epsilon$, we also write $p < q$ and say that the prefix is *strict*. Moreover, if neither $p \leq q$ nor $q \leq p$, we say that p and q are *parallel*, which we write as $p \parallel q$.

We denote by $s|_p$ the subterm of s that occurs *at* position $p \in \mathcal{P}os(s)$. Moreover, if $q \in \mathcal{P}os(s)$ and $p < q$, we say that $s|_p$ occurs *above* q. Finally, if $p > q$, then we say that $s|_p$ occurs *below* q.

3. Terms and valuations

We now proceed to define the main objects of study, namely *meta-terms* and *terms*. Furthermore, we define valuations, which are similar to substitutions as defined in the case of iTRSs and iλc and which are crucial for the definition of the rewrite relation on terms.

As it turns out, the most straightforward and liberal definition of meta-terms has rather poor properties: Applying a valuation need not necessarily yield a well-defined term. Therefore, we also introduce an important restriction on meta-terms: the *finite chains property.* This property will also prove crucial in obtaining positive results later in the paper.

3.1. Meta-terms and terms

In iTRSs and iλc, terms are defined by introducing a metric over the set of finite terms and taking the completion of that metric. That is, taking the least set of objects (with respect to set inclusion) containing the set of finite terms such that every Cauchy sequence converges [31, 8, 9] — this set will contain both the finite and infinite terms. Intuitively, with respect to such a metric, two terms s and t are close to each other if the first 'conflict' between them occurs at great depth. In iTRSs, a conflict is a position p such that $root(s|_p) \neq root(t|_p)$. In i$\lambda$c, a conflict is defined similarly, but also takes into account α-equivalence. The metric, denoted $d(s,t)$, is defined as 0 if no conflict occurs between s and t and is otherwise defined as 2^{-k}, where k denotes the minimal depth such that a conflict occurs between s and t. We take a similar approach in this paper.

To define terms and meta-terms for iCRSs, we first define the notions of a conflict and α-equivalence for *finite* meta-terms. In the definition, $s[x \to y]$ denotes the replacement in s of the occurrences of the free variable x by the variable y.

Definition 3.1. Let s and t be finite meta-terms. A *conflict* of s and t is a position $p \in \mathcal{P}os(s) \cap \mathcal{P}os(t)$ such that:

1. if $p = \epsilon$, then $root(s) \neq root(t)$ and not both s and t abstractions,
2. if $p = i \cdot q$ for $i \geq 1$, then $root(s) = root(t)$ and q a conflict of $s|_i$ and $t|_i$,
3. if $p = 0 \cdot q$, then $s = [x_1]s'$ and $t = [x_2]t'$ and q a conflict of $s'[x_1 \to y]$ and $t'[x_2 \to y]$, where y occurs neither in s' nor in t'.

The finite meta-terms s and t are *α-equivalent* if no conflict exists between them.

The metric is now defined precisely as in the case of iTRSs and iλc:

Definition 3.2. The metric d on the set of finite meta-terms is defined as:

$$d(s,t) = \begin{cases} 0 & \text{if } s \text{ and } t \text{ are } \alpha\text{-equivalent} \\ 2^{-k} & \text{otherwise} \end{cases}$$

where k is the minimal depth such that a conflict occurs between s and t.

Example 3.3. The finite meta-terms $s = [x]Z(x, f(x))$ and $s' = [y]Z(y, f(y))$ satisfy $d(s,s') = 0$. Moreover, if $t = [y]Z(y, f(z))$, then the only conflict between s and t occurs at position 021 and, hence, $d(s,t) = 2^{-3} = \frac{1}{8}$.

Precisely following the definition of terms in the case of iTRSs and iλc, we define the set of meta-terms.

Definition 3.4. The set of *meta-terms* is the metric completion of the set of finite meta-terms with respect to the metric d.

Recall that we can obtain the metric completion of a set by taking a set of equivalence classes of its Cauchy sequences such that two sequences $(s_i)_{i<\omega}$ and $(t_i)_{i<\omega}$ are in the same class iff $\lim_{i\to\omega} d(s_i, t_i) = 0$ [42]. Below, we refer to this fact several times. However, for most purposes a meta-term can simply be thought of as an infinite tree defined according to the rules of Definition 2.1. Such a tree is easily extracted from a Cauchy sequence of finite meta-terms by considering the positions and function symbols that "are constant from some point in the sequence onwards".

By definition of metric completion, the set of finite meta-terms is a subset of the set of meta-terms. Moreover, we can uniquely extend the metric d on the set of finite meta-terms to a metric on the set of meta-terms; we also denote this extended metric by d.

Example 3.5. Any finite meta-term, for instance $[x]Z(x, f(x))$, is a meta-term. Moreover, $Z'(Z'(Z'(\ldots)))$ is a meta-term, as is $Z_1([x_1]x_1, Z_2([x_2]x_2, \ldots))$.

The notions of a set of positions and a subterm of a finite meta-term carry over directly to meta-terms, we use the same notation in both cases.

The set of terms can now be defined as in the finite case [19, 20, 22], that is, by barring meta-variables from occurring. The only difference is that meta-terms instead of finite meta-terms now occur in the definition.

Definition 3.6. The set of *terms* is the set of meta-terms without occurrences of meta-variables.

Both the set of (infinite) first-order terms and the set of (infinite) λ-terms are easily shown to be included in the set of terms.

The definition of contexts carries over directly from the finite case:

Definition 3.7. A *context* is a meta-term over $\Sigma \cup \{\Box\}$ where $\Box$ is a fresh nullary function symbol. A *one-hole* context is a context in which precisely one $\Box$ occurs.

Given a context, we obtain a term by replacing the holes in the context by terms. For example, if $C[\Box]$ is a one-hole context and s is a term, we obtain a term by replacing $\Box$ by s; the new term is denoted by $C[s]$.

Replacing a hole in a context does not avoid the capture of free variables: A free variable x in s is bound by an abstraction over x in $C[\Box]$ in case $\Box$ occurs in the scope of the abstraction. This behaviour is not obtained automatically when working modulo α-equivalence: It is *always* possible find a representative from the α-equivalence class of $C[\Box]$ that does not capture the free variables in s. Therefore, we will always work with *fixed* representatives from α-equivalence classes of contexts. This convention ensures that variables will be captured properly.

Remark 3.8. Capture avoidance is disallowed for contexts as we do not want to lose variable bindings over rewrite steps in case: (i) an abstraction occurs in a context, and (ii) a variable bound by the abstraction occurs in a subterm being rewritten. Note that this means that the representative employed for the context must already be fixed *before* performing the actual rewrite step.

As motivation, consider λ-calculus: In the term $\lambda x.(\lambda y.x)z$, contracting the redex inside the context $\lambda x.\Box$ yields $\lambda x.x$, whence the substitution rules for contexts should be such that

$$(\lambda x.\Box)\{(\lambda y.x)z/\Box\} \to_\beta \lambda x.x\,.$$

If we assumed capture avoidance in effect for contexts, we would have an α-conversion in the rewrite step, whence

$$(\lambda x.\Box)\{(\lambda y.x)z/\Box\} \to_\beta \lambda w.x\,,$$

which is clearly wrong.

Henceforth, we use $f^n(s)$ for any $n \in \mathbb{N}$ and term s to denote the following inductively defined term:

$$f^n(s) = \begin{cases} s & \text{if } n = 0 \\ f(f^m(s)) & \text{if } n = m + 1 \end{cases}$$

Moreover, we use f^ω to denote the term that is the solution of the recursive equation $s = f(s)$ or, informally, $f(f(\ldots f(\ldots)))$.

As mentioned in the introduction of this section, we shall later define a restriction on meta-terms called the *finite chains property*. Intuitively, a *chain* is a sequence of contexts in a meta-term that occur 'nested right below each other'.

Definition 3.9. Let s be a meta-term. A *chain* in s is a sequence of (context, position)-pairs $(C_i[\Box], p_i)_{i<\alpha}$, with $\alpha \leq \omega$, such that for each $(C_i[\Box], p_i)$:

1. if $i + 1 < \alpha$, then $C_i[\Box]$ has *one* hole and $C_i[t_i] = s|_{p_i}$ for some term t_i, and
2. if $i + 1 = \alpha$, then $C_i[\Box]$ has *no* holes and $C_i[\Box] = s|_{p_i}$,

and such that $p_{i+1} = p_i \cdot q_i$ for all $i + 1 < \alpha$ where q_i is the position of the hole in $C_i[\Box]$.

If $\alpha < \omega$, respectively $\alpha = \omega$, then the chain is called *finite*, respectively *infinite*.

Observe that at most one $\Box$ occurs in any context $C_i[\Box]$ in a chain. In fact, $\Box$ only occurs in $C_i[\Box]$ if $i + 1 < \alpha$; if $i + 1 = \alpha$, we have $C_i[\Box] = s|_{p_i}$.

Example 3.10. Consider the term f^ω and define $(C_0[\Box], p_0) = (f^2(\Box), p)$ and $(C_1[\Box], p_1) = (f^\omega, p \cdot 11)$. Then, $(C_i[\Box], p_i)_{i<2}$ is a finite chain for any $p \in \mathcal{P}os(f^\omega)$. The sequence $(f(\Box), q_i)_{i<\omega}$ with $f^i(\Box)|_{q_i} = \Box$ is an infinite chain. Although rather pathological, $(\Box, \epsilon)_{i<\omega}$ is also an infinite chain.

3.2. Valuations

We next define valuations, the iCRS analogue of substitutions as defined in the case of iTRSs and iλc. The ingredients are the same as in the case of CRSs [20, 22], that is, we first define substitutions and substitutes and subsequently employ these in the definition of valuations. There is a subtle difference, however: The definitions are to be interpreted top-down — due to the presence of infinite terms and meta-terms — rather than bottom-up, as is also done in the case of iTRSs and iλc in relation to the finite systems these are based on.

Below, we use $\vec{x}$ and $\vec{t}$ as short-hand for, respectively, the sequences $x_1, \ldots,$ x_n and $t_1, \ldots, t_n$ with $n \geq 0$. Moreover, we assume n fixed in the next two definitions.

Definition 3.11. A *substitution* of the terms $\vec{t}$ for distinct variables $\vec{x}$ in a term s, denoted $s[\vec{x} := \vec{t}]$, is defined as:

1. $x_i[\vec{x} := \vec{t}] = t_i$,
2. $y[\vec{x} := \vec{t}] = y$, if y does not occur in $\vec{x}$,
3. $([y]s')[\vec{x} := \vec{t}] = [y](s'[\vec{x} := \vec{t}])$,
4. $f(s_1, \ldots, s_m)[\vec{x} := \vec{t}] = f(s_1[\vec{x} := \vec{t}], \ldots, s_m[\vec{x} := \vec{t}])$.

The above definition implicitly takes into account the usual variable convention [17] in the third clause to avoid the binding of free variables by the abstraction.

Assuming terms are defined by taking equivalence classes of Cauchy sequences, we obtain a representative of the class $s[\vec{x} := \vec{t}]$ in two steps. First, we select a representative $(s_i)_{i<\omega}$ of s and a sequence of representatives $(t_{1,i})_{i<\omega}$, $\ldots$, $(t_{n,i})_{i<\omega}$ for the sequence $\vec{t} = t_1, \ldots, t_n$. Second, we take $(s_i[\vec{x} := \vec{t_i}])_{i<\omega}$, that is, we substitute the sequence of ith members of the representatives chosen for $\vec{t}$ in the ith member of the representative chosen for s. Since all members are finite meta-terms it follows that substitution is well-defined: The definition can simply be read bottom-up (i.e. inductively) for each finite meta-term. It is easy to see that each resulting sequence of finite meta-terms converges and falls in the same equivalence class. Hence, applying a substitution to a term yields a well-defined term.

We now define substitutes, adopting this name from Kahrs [43].

Definition 3.12. An *n-ary substitute* is a mapping denoted $\underline{\lambda} x_1, \ldots, x_n.s$ or $\underline{\lambda}\vec{x}.s$, with s a term, such that:

$$(\underline{\lambda}\vec{x}.s)(t_1, \ldots, t_n) = s[\vec{x} := \vec{t}]\,. \tag{1}$$

Reading Equation (1) from left to right yields a rewrite rule:

$$(\underline{\lambda}\vec{x}.s)(t_1, \ldots, t_n) \to s[\vec{x} := \vec{t}]\,.$$

The rule can be seen as a *parallel β-rule.* That is, a variant of the β-rule from iλc which simultaneously substitutes multiple variables. We call the root of

$(\underline{\lambda}\vec{x}.s)$ the $\underline{\lambda}$-abstraction and the root of the left-hand side of the parallel β-rule the $\underline{\lambda}$-application.

The set of terms is easily extended with terms that contain $\underline{\lambda}$-abstractions and $\underline{\lambda}$-applications; we call the terms in this extended set $\underline{\lambda}$-terms. Extending the set creates a variant of iλc once we note that the notions of weakly and strongly convergent reductions [9, 10] carry over unchanged. The reader may note that these definitions are exactly the same as those we give for iCRSs later in the paper (see Section 4.3).

To aid the reader's understanding we now define descendants across reductions contracting parallel β-redexes. The definition is a straightforward variant of the corresponding notion in iλc; we state it explicitly to make the exact definition of descendants in iCRSs clearer.

Denote by 0 the position of the subterm on the left-hand side of a $\underline{\lambda}$-application and the position of the body of a $\underline{\lambda}$-abstraction. Moreover, denote by $1, \ldots, n$ the positions of the subterms on the right-hand side of a $\underline{\lambda}$-application. Let u be a rewrite step contracting a parallel β-redex at position p. The set of *descendants* of a position $q \in \mathcal{P}os(s)$ across, denoted q/u, is defined as $q/u = \{q\}$ in case $p \parallel q$ or $p > q$. In case $q = p \cdot 0 \cdot 0 \cdot q'$ and q is not the position of a variable bound by the $\underline{\lambda}$-abstraction of the contracted redex, we define $q/u = \{p \cdot q'\}$. Moreover, in case $q = p \cdot i \cdot q'$ with $1 \leq i \leq n$, we define $q/u = \{p \cdot q'' \mid q'' \in Q\}$ where Q is the set of positions q'' such that $p \cdot 0 \cdot q''$ is the position of a variable bound by the ith variable in the $\underline{\lambda}$-abstraction. Otherwise, we define $q/u = \emptyset$.

Employing the above definition of descendants, the notions of descendants and residuals across strongly convergent reductions carry over without change from iλc [9, 10]. Again, the definition is the same as the one for iCRSs (see Definition 4.22).

Definition 3.13. Let σ be a function that maps meta-variables to substitutes such that, for all $n \in \mathbb{N}$, if Z has arity n, then so does $\sigma(Z)$.

A *valuation* induced by σ is a relation $\bar{\sigma}$ that takes meta-terms to terms such that:

1. $\bar{\sigma}(x) = x$,
2. $\bar{\sigma}([x]s) = [x](\bar{\sigma}(s))$,
3. $\bar{\sigma}(Z(s_1, \ldots, s_m)) = \sigma(Z)(\bar{\sigma}(s_1), \ldots, \bar{\sigma}(s_m))$,
4. $\bar{\sigma}(f(s_1, \ldots, s_m)) = f(\bar{\sigma}(s_1), \ldots, \bar{\sigma}(s_m))$.

Similar to Definition 3.11, the above definition implicitly takes into account the variable convention to avoid the binding of free variables by the abstraction, this time in the second clause.

From an operational point-of-view the definition of a valuation yields a straightforward two-step way of applying it to a meta-term: In the first step each subterm of the form $Z(t_1, \ldots, t_n)$ is replaced by a subterm of the form $(\underline{\lambda}\vec{x}.s)(t_1, \ldots, t_n)$. In the second step Equation (1) is applied to each subterm of the form $(\underline{\lambda}\vec{x}.s)(t_1, \ldots, t_n)$, as introduced in the first step.

In view of the parallel β-rule introduced immediately below Definition 3.12 the second step can be seen as a complete development of the parallel β-redexes introduced in the first step:

Definition 3.14. A *development* of a set $\mathcal{U}$ of parallel β-redexes is a strongly convergent reduction such that each step contracts a residual of a redex in $\mathcal{U}$. A development $s \twoheadrightarrow t$ is called *complete* if $\mathcal{U}/(s \twoheadrightarrow t) = \emptyset$.

In the finite case [19, Remark II.1.10.1], the application of a valuation to a meta-term always yields a unique term, that is, *valuations are well-defined.* Unfortunately, this is no longer the case when infinite meta-terms are considered:

Example 3.15. Consider the meta-term

$$Z(Z(\ldots Z(\ldots)))$$

and any map that satisfies $Z \mapsto \underline{\lambda}x.x$. Clearly, this map should induce a valuation. However, applying any such valuation to $Z(Z(\ldots Z(\ldots)))$ yields:

$$(\underline{\lambda}x.x)((\underline{\lambda}x.x)(\ldots(\underline{\lambda}x.x)(\ldots)))\,.$$

This $\underline{\lambda}$-term has no complete development, as no matter how many parallel β-redexes are contracted, it reduces only to itself and not to a term.

In the above example, $\bar{\sigma}(Z(Z(\ldots Z(\ldots))))$ is not well-defined due to the fact that *no* map can have the properties of a valuation induced by σ *and* be defined on $Z(Z(\ldots Z(\ldots)))$.

Well-definedness of valuations does not depend on one unique meta-variable being present in the meta-term. The same behaviour can be witnessed in case different meta-variables of different arities are present as long as we can define a valuation that assigns $\underline{\lambda}\vec{x}.y$ to each meta-variable Z in the meta-term with y in $\vec{x}$ such that y corresponds to an argument of Z which is a meta-variable.

We are faced with even more intricate problems: Applying a valuation to a meta-term may yield distinct $\underline{\lambda}$-terms as of reduction of $\underline{\lambda}$-terms is not necessarily confluent.

Example 3.16. Consider a signature with nullary functions symbols a and b. Moreover, consider the meta-term

$$Z(a, Z(b, Z(a, Z(b, Z(\ldots)))))\,.$$

Applying the valuation that assigns to Z the substitute $\underline{\lambda}xy.y$ yields the $\underline{\lambda}$-term:

$$(\underline{\lambda}xy.y)(a, (\underline{\lambda}xy.y)(b, (\underline{\lambda}xy.y)(a, (\underline{\lambda}xy.y)(b, (\underline{\lambda}xy.y)(\ldots)))))\,,$$

which is depicted in Figure 2(a). The term reduces by means of two different developments to the $\underline{\lambda}$-terms:

$$(\underline{\lambda}xy.y)(a, (\underline{\lambda}xy.y)(a, (\underline{\lambda}xy.y)(a, (\underline{\lambda}xy.y)(a, (\underline{\lambda}xy.y)(\ldots)))))\,,$$

as depicted in Figure 2(b), and:

$$(\underline{\lambda}xy.y)(b, (\underline{\lambda}xy.y)(b, (\underline{\lambda}xy.y)(b, (\underline{\lambda}xy.y)(b, (\underline{\lambda}xy.y)(\ldots)))))\,,$$

as depicted in Figure 2(c). These last two $\underline{\lambda}$-terms have no common reduct with respect to parallel β-reduction; they reduce only to themselves. Note that a similar problem occurs in iλc where confluence fails unless certain restrictions are enforced [9].

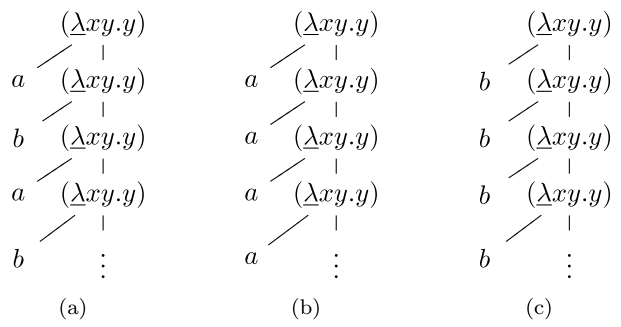


Figure 2: The $\underline{\lambda}$-terms from Example 3.16

The situation is unsatisfactory: We would like valuations to be defined on as many meta-terms as possible. By the above examples, this is not possible in general. In the following subsection we identify a class of meta-terms that avoids these problems, yet is sufficiently expressive.

Remark 3.17. The problematic examples in the present section are all analogues of similar examples in first-order infinitary rewriting. If we introduce for each $\underline{\lambda}\vec{x}.x_i$ with x_i in $\vec{x}$ a function symbol $\overline{(\underline{\lambda}\vec{x}.x_i)}$ of arity equal to the number of variables in $\vec{x}$, then replacing the parallel β-rule by the set of all *first-order* rules of the form

$$\overline{(\underline{\lambda}\vec{x}.x_i)}(z_1, \ldots, z_n) \to z_i$$

does not change the behaviour of the examples in this section. Moreover, all examples then involve infinite chains of first-order redexes (called collapsing towers in [8]). Ruling out these chains in first-order rewriting is a sufficient condition for complete developments to exist [8]; this also holds in our case, as we show in the next section (without introducing first-order rules). Note that this result does not carry over to iλc, as there one may have that β-redexes may occur in subterms with an abstraction at the root (something not allowed in the case of the parallel β-redex, which requires the s in $\underline{\lambda}\vec{x}.s$ to be a term, not just a $\underline{\lambda}$-term).

3.3. Finite chains property

The examples exhibiting problems with valuations all share a common feature: They involve $\underline{\lambda}$-terms with infinite chains of parallel β-redexes. Thus, they involve in particular meta-terms with infinite chains of meta-variables.

Definition 3.18. Let s be a meta-term. A *chain of meta-variables* in s is a chain $(C_i[\Box], p_i)_{i<\alpha}$ in s, with $\alpha \leq \omega$, such that for each $i < \alpha$ it holds that $C_i[\Box] = Z(t_1, \ldots, t_n)$ with $t_j = \Box$ for exactly one $1 \leq j \leq n$.

A meta-term s is said to satisfy the *finite chains property* if no infinite chain of meta-variables occurs in s.

Example 3.19. An example of a class of meta-terms satisfying the finite chains property is the class of finite meta-terms. The class of meta-terms with infinitely nested chains of finite chains of meta-variables 'guarded' by abstractions or function symbols also satisfies the finite chains property. The following meta-term is an example of a meta-term in the latter class:

$$[x_1]Z_1([x_2]Z_2(\ldots [x_n]Z_n(\ldots)))$$

As a special case we have that any meta-term in which all meta-variables occur as $Z(s_1, \ldots, s_n)$ with no meta-variables occurring at the roots of $s_1, \ldots, s_m$ satisfies the finite chains property.

Examples of meta-terms that do *not* satisfy the finite chains property are $Z(Z(\ldots Z(\ldots)))$ and $Z_1(Z_2(\ldots Z_n(\ldots)))$.

The following holds for meta-terms satisfying the finite chains property and is used in Lemma 5.1 to show that compressed reductions are well-behaved.

Proposition 3.20. *Let s be a meta-term satisfying the finite chains property and let γ be a map that assigns to each $p \in \mathcal{P}os(s)$ the number of prefix positions of p at which* no *meta-variable occurs. For any $n \in \mathbb{N}$, the number of positions p with $\gamma(p) = n$ is finite.*

Proof. Consider s as a finitely-branching tree. Remove from this tree all positions p for which $\gamma(p) > n$. Observe that any position $p' < p$ will satisfy $\gamma(p') \leq \gamma(p)$. Hence, if $\gamma(p) \leq n$, no prefix of p will be removed. Thus, the graph resulting from this removal is again a tree; call this tree $\mathcal{T}$.

Assume that $\mathcal{T}$ contains an infinite path such that for every position p along the path we have $\gamma(p) \leq n$. As non-meta-variables can only occur at n positions along the path, there exists a position q such that only meta-variables occur at positions of which q is a prefix, contradicting the finite chains property. Hence, no infinite path occurs in $\mathcal{T}$. As $\mathcal{T}$ is finitely branching, König's Lemma yields that $\mathcal{T}$ is finite, implying that the number of positions p for which $\gamma(p) \leq n$, and *a fortiori* $\gamma(p) = n$ is also finite. $\square$

We next show that all valuations are total on the set of meta-terms satisfying the finite chains property.

Proposition 3.21. *Let s be a meta-term satisfying the finite chains property and let $\bar{\sigma}$ be a valuation. There is a unique term that is the result of applying $\bar{\sigma}$ to s.*

Proof. Employing the two-step operational view when applying $\bar{\sigma}$ — as described in the previous section — it is immediate by the definition of valuations that the first step of applying $\bar{\sigma}$ to s has a unique result. Denote this result by s_σ and denote the set of all parallel β-redexes in s_σ by $\mathcal{U}$. The result now follows if we can show that $\mathcal{U}$ has a complete development ending in a term and, moreover, that each development of $\mathcal{U}$ ends in the same term.

To start, observe that to repeatedly rewrite the root of s_σ by means of the parallel β-redex requires the root to be of the form

$$(\underline{\lambda}\vec{x}.x_i)(t_1, \ldots, t_n)\,,$$

where $1 \leq i \leq n$ and t_i is again a redex of this form. This is only possible if there exists in s_σ an infinite chain of such redexes starting at the root. However, by definition of valuations this means that an infinite chain of meta-variables occurs in s, which is impossible as s satisfies the finite chains property. Thus, the root can only be rewritten finitely often in a development. Applying the same reasoning to the roots of the subterms, we obtain that all possible reductions are strongly convergent and that there exists a complete development reducing the redexes in $\mathcal{U}$ in an outside-in fashion. As all parallel β-redexes occur in $\mathcal{U}$ and as no $\underline{\lambda}$-applications and $\underline{\lambda}$-abstractions occur in s, the result of the complete development, which we denote by $\bar{\sigma}(s)$, is necessarily a term.

To show that each complete development ends in the same term, observe that we can consider each parallel β-redex $(\underline{\lambda}x_1, \ldots, x_n.s)(t_1, \ldots, t_n)$ to be a sequence of β-redexes:

$$(\lambda x_1.(\ldots((\lambda x_n.s)t_n)\ldots))t_1 .$$

This means that each complete development in our variant of iλc corresponds to a complete development in iλc extended with some function symbols. As each complete development in iλc ends in the same term [9, 10], a result independent of any function symbols that are added, the same holds for the redexes in $\mathcal{U}$. Hence, $\bar{\sigma}(s)$ is unique. □

3.4. Discussion

We have chosen to define infinite terms in what we believe is the most common way in infinitary rewriting. However, other possibilities exist, as do other ways to obtain the set of meta-terms satisfying the finite chains property, and yet other ways to obtain well-defined valuations. For the benefit of the expert reader, we briefly review some of these in this subsection; the reader only interested in the development of iCRSs may safely skip this material.

Infinite terms.. Defining (infinite) (meta-)terms by means of metric completion is one of three currently known ways of introducing these terms. In a first-order setting (infinite) terms can alternatively be defined by means of partial functions (from the set of all possible positions to elements of the chosen signature) [44, 45] or by means of the domain-theoretical construct of ideal completion (first endowing the set of finite terms with a partial order) [46]. It turns out that each of these three approaches is isomorphic to the others as shown in [47] by co-algebraic means.

Extending each of the three approaches to a higher-order setting requires us to take account of α-equivalence, just as we have done for the metric completion approach (following the definition of infinite λ-terms from [9]). In the case of the approach using ideal completion, a definition has been given in [47], albeit for the slightly different class of higher-order terms used to define HRSs [26, 21]. The partial function approach has not yet been extended to higher-order terms. Hence, the theory of infinite higher-order terms can be called sketchy at best and we deem it neither proper nor fruitful to rectify this situation in the current paper.

A further complication is the fact that currently no co-algebraic notion exists that can rightly be said to capture the concept of (infinite) higher-order terms with α-equivalence. Indeed, the known (categorical) algebraic notions used to describe finite higher-order terms do not properly dualise to include all infinite terms defined above [47]. For example, dualising the notion of binding algebras from [48] will only allow for (infinite) terms with a finite number of free variables. Hence our use of the informal concepts of bottom-up and top-down instead of inductive and co-inductive, respectively.

Meta-terms satisfying the finite chains property.. The set of meta-terms satisfying the finite chains property can alternatively be defined by slightly altering the employed depth measure and metric.

Given a term s and a position $p \in \mathcal{P}os(s)$, define the depth measure D:

$$\begin{aligned} D(s, \epsilon) &= 0 \\ D(Z(t_1, \ldots, t_n), i \cdot p') &= D(t_i, p') \\ D([x]t, 0 \cdot p') &= 1 + D(t, p') \\ D(f(t_1, \ldots, t_n), i \cdot p') &= 1 + D(t_i, p') \end{aligned}$$

The difference with the usual depth measure $|p|$ is that we are *not* counting meta-variables towards the depth.

Next, define the metric d_a:

$$d_a(s,t) = \begin{cases} 0 & \text{if } s \text{ and } t \text{ are } \alpha\text{-equivalent} \\ 2^{-k} & \text{otherwise} \end{cases}$$

where k is the minimal depth — with respect to the depth measure D — such that a conflict occurs between s and t.

The meta-terms without infinite chains of meta-variables are now defined by taking the metric completion of the set of finite meta-terms with respect to d_a. That precisely the meta-terms without infinite chains of meta-variables are obtained is an immediate consequence of the meta-variables not counting towards the depth.

The above construction for the set of meta-terms satisfying the finite chains property is inspired by similar constructions for iλc defining subsets of the set of infinite λ-terms by slightly altering the notion of the depth measure used in the employed metric [9]. For example, the set containing no λ-terms with infinite chains of λ-abstractions (i.e. not containing subterms of the form $\lambda x_1.\lambda x_2 \ldots \lambda x_n \ldots$) can be defined in this way.

One reason we did not choose to define meta-terms using the changed metric is that the concept of meta-variables does not occur in all formats for higher-order rewriting. For example, in HRSs [21, 26] no distinction is made between term formation in rules ('meta-terms') and term formation in the terms to be rewritten. Thus, when extending our results to other formats, we would have to either work with two different metrics, one for rules and one for general terms, or with one, very restrictive metric that would disallow perfectly harmless terms as these could otherwise occur as right-hand sides of rules, with possibly detrimental results.

Remark 3.22. We conjecture that, using the above metric, it is possible to show that valuations are total on the set of meta-terms satisfying the finite chains property by using Kahrs' uniform continuity approach [38]. That is, by showing that valuations define a uniformly continuous map from the set of finite meta-terms to the set of finite terms which uniquely extends to a map on the metric completions of its domain and codomain. The above metric is required, as valuations are not uniformly continuous given the metric from Definition 3.2 — due to the problems outlined at the end of Section 3.2, which led to the definition of the finite chains property.

Kahrs' approach thus requires entangling the definition of the metric with the finite chains property. We find this approach ill-suited for illustrative purposes and hence prefer not to use it. Moreover, our current proof provides an interesting application of the theory of developments from iλc.

Well-defined valuations.. As shown in [9], confluence can be recovered in iλc in the following way: Define a term s as being *root-active* if every reduct of s has a β-redex at the root; introduce a fresh nullary function symbol $\perp$ and assume that every root-active term can be rewritten to $\perp$. This result can be used to recover confluence for our system with the parallel β-rule: simply use the encoding of $\underline{\lambda}$-terms as λ-terms from the proof of Proposition 3.21 and apply the confluence theorem from [9]. Well-definedness of valuations — without the finite chains property — then follows along the lines of Proposition 3.21, observing that if an infinite number of parallel β-redexes is contracted at a certain position, then the subterm at that position must be root-active.

Remark 3.23. We conjecture that, rewriting root-active $\underline{\lambda}$-terms to $\perp$, we can show that (a) the compression property holds and that (b) if a set of redexes has a complete development, then all complete developments of the set end in the same term. However, Lemma 6.15, which is crucial for the confluence proof in [2], will fail. To see this, consider the following two rewrite rules, the first of which is *not* allowed starting from Section 4 onwards:

$$\begin{aligned} f([x]Z(x)) &\to Z^\omega \\ g(Z) &\to Z \end{aligned}$$

Considering the term $f([x]g(x))$, we have that $f([x]g(x)) \to g^\omega$ is a complete development of the redex at the root of the term (see Section 6). However, the set consisting of both redexes in $f([x]g(x))$ either reduces to $\perp$ — by first contracting the $g(Z) \to Z$-redex and next the redex at the root — or to g^ω, which only reduces to itself.

4. Rewrite rules and reductions

Having defined terms, we move on to define rewrite rules and reductions.

4.1. Rewrite rules

We give a number of definitions that are direct extensions of the corresponding definitions from CRS theory.

Definition 4.1. A finite meta-term is a *pattern* if each of its meta-variables has only distinct bound variables as its arguments. Moreover, a meta-term is *closed* if all its variables occur bound.

Example 4.2. The meta-terms $f([x]Z(x), Z')$ and $f([x]g(Z(x)), y)$ are patterns. The meta-term $g(Z(Z'))$ is not a pattern as the meta-variable Z' occurs as an argument of the meta-variable Z. The pattern $f([x]g(Z(x)), y)$ is not closed due to the free occurrence of the variable y.

We next define rewrite rules and iCRSs. As in the case of iTRSs, the definitions are identical to the definitions given in the finite case, with exception of the restrictions on the right-hand sides of the rewrite rules [7, 8]. In the case of iTRSs the finiteness restriction is lifted from the right-hand sides. Here, this is also done, but at the same time the finite chains property is put into place.

Definition 4.3. A *rewrite rule* is a pair (l, r), denoted $l \to r$, where l is a finite meta-term and r is a meta-term, such that:

1. l is a pattern with a function symbol at the root,
2. all meta-variables that occur in r also occur in l,
3. l and r are closed, and
4. r satisfies the finite chains property.

The meta-terms l and r are called, respectively, the *left-hand side* and the *right-hand side* of the rewrite rule.

An *infinitary Combinatory Reduction System (iCRS)* is a pair $\mathcal{C} = (\Sigma, R)$ with Σ a signature and R a set of rewrite rules.

Example 4.4. We have that $f([x]Z(x), Z') \to Z(Z')$ is a rewrite rule. Moreover, $g(Z(Z')) \to Z'$ is not a rule as $g(Z(Z'))$ is not a pattern and $f([x]Z(x), Z') \to X$ is not a rule as X does not occur on the left-hand side.

Both left-hand and right-hand sides of rewrite rules satisfy the finite chains property. In the case of left-hand sides this follows by their finiteness, and in the case of right-hand sides this is by the definition.

It follows easily that iTRSs and iλc are iCRSs if we interpret their rewrite rules as rules in the above sense. By definition of iTRSs and iλc only finite chains of meta-variables occur in the right-hand sides of the rewrite rules.

In the remainder of the paper there are a few references to the assumption that right-hand sides satisfy the finite chains property; the references can be found in the proofs of Lemmas 5.1, 6.7, and 6.15 and Theorem 7.2. In each of these cases there is no advantage in using finite right-hand sides instead of infinite right-hand sides satisfying the finite chains property; we would still need to reason that meta-variables only occur in finite chains to be able to complete the proofs.

We now define rewrite steps.

Definition 4.5. A *rewrite step* is a pair of terms (s, t), denoted $s \to t$, adorned with a one-hole context $C[\Box]$, a rewrite rule $l \to r$, and a valuation $\bar{\sigma}$ such that $s = C[\bar{\sigma}(l)]$ and $t = C[\bar{\sigma}(r)]$. The term $\bar{\sigma}(l)$ is called an *$l \to r$-redex*, or simply a *redex*. The redex *occurs* at position p and depth $|p|$ in s, where p is the position of the hole in $C[\Box]$.

A position q of s is said to occur in the *redex pattern* of the redex at position p if $q \geq p$ and if there does not exist a position q' with $q \geq p \cdot q'$ such that q' is the position of a meta-variable in l.

Example 4.6. The term $f([x]h(x), a)$ rewrites to $h(a)$ by contracting the redex of the rule $f([x]Z(x), Z') \to Z(Z')$ occurring at position ϵ, that is, at the root.

As both left-hand and right-hand sides of rewrite rules satisfy the finite chains property, it follows by Proposition 3.21 that rewrite steps are well-defined.

We now mention some standard restrictions on rewrite rules that we shall need later in the paper:

Definition 4.7. A rewrite rule is *left-linear*, if each meta-variable occurs at most once in its left-hand side. Moreover, an iCRS is *left-linear* if all its rewrite rules are.

Definition 4.8. Let s and t be finite meta-terms that have no meta-variables in common. The meta-term s *overlaps* t if there exists a non-meta-variable position $p \in \mathcal{P}os(s)$ and a valuation $\bar{\sigma}$ such that $\bar{\sigma}(s|_p) = \bar{\sigma}(t)$.

Two rewrite rules *overlap* if their left-hand sides overlap and if the overlap does not occur at the root when two copies of the same rule are considered. An iCRS is *orthogonal* if all its rewrite rules are left-linear and no two (possibly the same) rewrite rules overlap.

Example 4.9. The rule $g([x]Z(x)) \to b$ overlaps $f(g([x]Z(x))) \to h([x]Z(x))$ at position 1. The rules $f(g([x]Z(x))) \to h([x]Z(x))$ and $f(h([x]Z(x))) \to Z(a)$ do not overlap.

In case the rewrite rules $l_1 \to r_1$ and $l_2 \to r_2$ overlap at position p, it follows that p cannot be the position of a bound variable in l_1. If it were, we would obtain for some valuation $\bar{\sigma}$ and variable x that $\bar{\sigma}(l_1|_p) = x = \bar{\sigma}(l_2)$, which would imply that l_2 does not have a function symbol at the root, as required by the definition of rewrite rules.

Moreover, it is easily seen that if two left-linear rules overlap in an infinite term, there is also a finite term in which they overlap. As left-hand sides are *finite* meta-terms, we may thus appeal to standard ways of deeming CRSs orthogonal by inspection of their rules.

Definition 4.10. A pattern is *fully-extended* [29, 30], if, for each of its meta-variables Z and each abstraction $[x]s$ having an occurrence of Z in its scope, x is an argument of that occurrence of Z. Moreover, a rewrite rule is *fully-extended* if its left-hand side is, and an iCRS is *fully-extended* if all its rewrite rules are.

Example 4.11. The pattern $f(g([x]Z(x)))$ is fully-extended, as is the rewrite rule $f(g([x]Z(x))) \to h([x]Z(x))$. The pattern $g([x]f(Z(x), Z'))$, with Z' occurring in the scope of the abstraction $[x]$, is not fully-extended as x does not occur as an argument of Z'. As a result, the rule $g([x]f(Z(x), Z')) \to Z(Z')$ is not fully-extended and no $g([x]f(Z(x), Z')) \to Z(Z')$-redex occurs at the root of $g([x]f(x, x))$. On the other hand, such a redex does occur at the root of $g([x]f(x, a))$; contracting this redex yields a.

4.2. Transfinite reductions

We can now define transfinite reductions. The definition is identical to those of iTRSs and iλc [8, 9].

Definition 4.12. A *transfinite reduction* with domain $\alpha > 0$ is a sequence of terms $(s_\beta)_{\beta<\alpha}$ adorned with a rewrite step $s_\beta \to s_{\beta+1}$ for each $\beta + 1 < \alpha$. In case $\alpha = \alpha' + 1$, the reduction is *closed* and of length α'. In case α is a limit ordinal, the reduction is called *open* and of length α. The reduction is *weakly continuous* or *Cauchy continuous* if, for every limit ordinal $\gamma < \alpha$, the distance between s_β and s_γ tends to 0 as β approaches γ from below. The reduction is *weakly convergent* or *Cauchy convergent* if it is weakly continuous and closed.

An open transfinite reduction is lacking a well-defined final term, while a closed reduction does have such a term.

Example 4.13. Consider the rewrite rule $f([x]Z(x)) \to Z(f([x]Z(x)))$ and observe that $f([x]x) \to f([x]x)$. Define $s_\beta = f([x]x)$ for all $\beta < \omega{\cdot}2$. The reduction $(s_\beta)_{\beta<\omega\cdot 2}$, where in each step we contract the redex at the root, is open and weakly continuous. Adding the term $f([x]x)$ to the end of the reduction yields a weakly convergent reduction. Both reductions are of length $\omega \cdot 2$.

As in [8, 9, 10], we prefer to reason about strongly convergent reductions. This ensures that descendants are always well-defined and that we can restrict our attention to reductions of length at most ω by the so-called *compression property*, as shown in Section 5.

Definition 4.14. Let $(s_\beta)_{\beta<\alpha}$ be a transfinite reduction. For each rewrite step $s_\beta \to s_{\beta+1}$, let d_β denote the depth of the contracted redex. The reduction is *strongly continuous* if it is weakly continuous and if, for every limit ordinal $\gamma < \alpha$, the depth d_β tends to infinity as β approaches γ from below. The reduction is *strongly convergent* if strongly continuous and closed.

Example 4.15. The reductions from Example 4.13 are neither strongly continuous nor strongly convergent, as all contracted redexes occur at the root, that is, at depth 0. On the other hand, given again the rule $f([x]Z(x)) \to Z(f([x]Z(x)))$, we have that the depth of the contracted redexes increases along the following reduction:

$$f([x]g(x)) \to g(f([x]g(x)) \to \cdots \to g^n(f([x]g(x))) \to g^{n+1}(f([x]g(x))) \to \cdots$$

The reduction is open and strongly continuous. Extending the reduction with the term g^ω yields a strongly convergent reduction. Both reductions are of length ω.

Example 4.16. Consider the rules for map from Example 2.2. Let g be a unary function symbol and let s and t be the terms satisfying the recursive equations $s = \mathtt{cons}(\mathtt{nil}, s)$ and $t = \mathtt{cons}(g(\mathtt{nil}), t)$, that is, s and t represent, respectively, the infinite list of nils and $g(\mathtt{nil})$s. We have that

$$\begin{aligned}\mathtt{map}([x]g(x), s) &\to \mathtt{cons}(g(\mathtt{nil}), \mathtt{map}([x]g(x), s)) \\ &\to \mathtt{cons}(g(\mathtt{nil}), \mathtt{cons}(g(\mathtt{nil}), \ldots)) \\ &\to \cdots t\end{aligned}$$

is a strongly convergent reduction of length ω.

Notation 4.17. By $s \twoheadrightarrow^{\alpha} t$, respectively $s \twoheadrightarrow^{\leq\alpha} t$, we denote a *strongly convergent* reduction of ordinal length α, respectively of ordinal length at most α. By $s \twoheadrightarrow t$ we denote a *strongly convergent* reduction of arbitrary ordinal length and by $s \to^{*} t$ we denote a reduction of finite length. Reductions are usually ranged over by capital letters such as D, S, and T. The concatenation of reductions S and T is denoted by $S; T$.

Note that the concatenation of any finite number of strongly convergent reductions is a strongly convergent reduction. With respect to strongly convergent reductions we also have the following:

Lemma 4.18. *If $s \twoheadrightarrow t$, then the number of steps contracting redexes at depths less than $d \in \mathbb{N}$ is finite for any d.*

Proof. This is exactly the proof of [8, Lemma 3.5]. □

Corollary 4.19. *Every strongly convergent reduction has countable length.*

4.3. Descendants and residuals

The definition of a descendant across a rewrite step $\bar{\sigma}(l) \to \bar{\sigma}(r)$ follows the definition of valuations and substitutions, and is thus defined in two steps. The first step defines descendants in $\bar{\sigma}(r)$ where only the valuation is applied and not Equation (1). The second step defines descendants across the application of Equation (1).

The second step has already been described in Section 3.2 and we refer the reader to that section for further details. Do note that the positions of variables bound by contracted parallel β-redexes do not have any descendants. As a consequence, positions of variables bound by redexes being reduced in iCRSs will not have descendants either. This behaviour is analogous to that of the descendants defined in [19].

In addition to the above, it follows from the definition given below that positions occurring in the redex pattern of a contracted redex do not have any descendants either. Defining these positions and the positions of variables bound by contracted redexes to be without descendants has as advantage that each position in the resulting term will descent from *at most* one position in the original term. This simplifies the theory we need to develop to prove confluence properties of orthogonal iCRSs. Needless to say, the definition given below is easily adapted to deal differently with bound variables and positions occurring in redex patterns.

We continue to define the first step in the definition of descendants. Recall that we denote by 0 both the position of the subterm on the left-hand side of a $\underline{\lambda}$-application and the position of the body of a $\underline{\lambda}$-abstraction and that we denote by $1, \ldots, n$ the positions of the subterms on the right-hand side of a $\underline{\lambda}$-application. This means that $(\underline{\lambda}\vec{x}.s)(t_1, \ldots, t_n)|_0 = (\underline{\lambda}\vec{x}.s)$, $\underline{\lambda}\vec{x}.s|_0 = s$, and $Z(t_1, \ldots, t_n)|_i = (\underline{\lambda}\vec{x}.s)(t_1, \ldots, t_n)|_i = t_i$ for $1 \leq i \leq n$. We denote by $\bar{\sigma}(l) \to r_\sigma$ the rewrite step $\bar{\sigma}(l) \to \bar{\sigma}(r)$ where the valuation is applied to r but not Equation (1).

Definition 4.20. Let $l \to r$ be a rewrite rule, $\bar{\sigma}$ a valuation, and $p \in \mathcal{P}os(\bar{\sigma}(l))$. Suppose $u : \bar{\sigma}(l) \to r_\sigma$. The set $p/_1 u$ is defined as follows:

- if a position $q \in \mathcal{P}os(l)$ exists such that $p = q \cdot q'$ and $root(l|_q) = Z$, then define $p/_1 u = \{p' \cdot 0 \cdot 0 \cdot q' \mid p' \in P\}$ with $P = \{p' \mid root(r|_{p'}) = Z\}$,
- if no such position exists, then define $p/_1 u = \emptyset$.

Note that $\mathcal{P}os(r) \subseteq \mathcal{P}os(r_\sigma)$ by the notation of positions in subterms of the form $(\underline{\lambda}\vec{x}.s)(t_1, \ldots, t_n)$. From this it follows that $P \subseteq \mathcal{P}os(r_\sigma)$.

We can now give the full definition of a descendant across a rewrite step.

Definition 4.21. Let $u : C[\bar{\sigma}(l)] \to C[\bar{\sigma}(r)]$ be a rewrite step, such that p is the position of the hole in $C[\square]$, and let $q \in \mathcal{P}os(C[\bar{\sigma}(l)])$. The set of *descendants* of q *across* u, denoted q/u, is defined as $q/u = \{q\}$ in case $p \parallel q$ or $p > q$. In case $q = p \cdot q'$, the set is defined as $q/u = \{p \cdot q'' \mid q'' \in Q\}$, where Q is the set of descendants of $q'/_1 u'$ with $u' : \bar{\sigma}(l) \to r_\sigma$ across a complete development of the parallel β-redexes in r_σ.

Descendants across a reduction are defined as for iTRSs and iλc.

Definition 4.22. Let $s_0 \twoheadrightarrow^\alpha s_\alpha$ and let $P \subseteq \mathcal{P}os(s_0)$. The set of *descendants* of P across $s_0 \twoheadrightarrow^\alpha s_\alpha$, denoted $P/(s_0 \twoheadrightarrow^\alpha s_\alpha)$, is defined as follows:

- if $\alpha = 0$, then $P/(s_0 \twoheadrightarrow^\alpha s_\alpha) = P$,
- if $\alpha = 1$, then $P/(s_0 \to s_1) = \bigcup_{p \in P} p/(s_0 \to s_1)$,
- if $\alpha = \beta + 1$, then $P/(s_0 \twoheadrightarrow^{\beta+1} s_{\beta+1}) = (P/(s_0 \twoheadrightarrow^\beta s_\beta))/(s_\beta \to s_{\beta+1})$,
- if α is a limit ordinal, then $p \in P/(s_0 \twoheadrightarrow^\alpha s_\alpha)$ iff $p \in P/(s_0 \twoheadrightarrow^\beta s_\beta)$ for all large enough $\beta < \alpha$.

In the case of orthogonal iCRSs, if there exists a redex at a position p employing a rewrite rule $l \to r$ that is not contracted in a rewrite step and if p has descendants across the step, then there exists a redex at each descendant of p that also employs the rule $l \to r$. Hence, for orthogonal systems there exists a well-defined notion of *residual* by strongly convergent reductions. We overload the notation $\cdot/\cdot$ to denote both the descendant and the residual relation.

Notation 4.23. Let $s \twoheadrightarrow t$. Assume $P \subseteq \mathcal{P}os(s)$ and $\mathcal{U}$ a set of redexes in s. We denote the descendants of P across $s \twoheadrightarrow t$ by $P/(s \twoheadrightarrow t)$ and the residuals of $\mathcal{U}$ across $s \twoheadrightarrow t$ by $\mathcal{U}/(s \twoheadrightarrow t)$. Moreover, if $P = \{p\}$ and $\mathcal{U} = \{u\}$, then we also write $p/(s \twoheadrightarrow t)$ and $u/(s \twoheadrightarrow t)$. Finally, if $s \twoheadrightarrow t$ consists of a single step contracting a redex u, then we sometimes write $\mathcal{U}/u$.

Example 4.24. Consider the strongly convergent reduction from Example 4.15:

$$f([x]g(x)) \to g(f([x]g(x)) \to \cdots \to g^n(f([x]g(x))) \to \cdots g^\omega$$

and call it S. The position of the subterm $g(x)$ in the term $f([x]g(x))$ is 10. We have:

$$\begin{aligned} 10/(&f([x]g(x)) \to g(f([x]g(x))) \\ &= \{10\}/(f([x]g(x)) \to g(f([x]g(x))) \\ &= \{\epsilon, 110\} \end{aligned}$$

and for the positions ϵ and 0 in the redex pattern of $f([x]g(x))$:

$$\{\epsilon, 0\}/(f([x]g(x)) \to g(f([x]g(x))) = \emptyset\,.$$

Moreover, for S we have:

$$10/S = \{\epsilon, 1, 11, 111, \ldots\}\,.$$

The following two lemmas provide some insight in the interplay between residuals and strongly convergent reductions; they are the respective analogues of Lemmas 12.5.12 and 12.5.4 in [10].

Lemma 4.25. *Let P be a set of positions in a term s and let $s \twoheadrightarrow t$. If every step in $s \twoheadrightarrow t$ occurs at depth strictly greater than d, then P and $P/(s \twoheadrightarrow t)$ have exactly the same members at depth $\leq d$.*

Proof. As each step $s_i \to s_{i+1}$ occurs at depth $> d$, we have $p/(s_i \to s_{i+1}) = \{p\}$ for every $p \in P$ at depth $\leq d$, which by definition of descendants entails that $p/(s \twoheadrightarrow t) = \{p\}$. □

Lemma 4.26. *For every fully-extended, left-linear iCRS, if $s \twoheadrightarrow t$ is a reduction of limit ordinal length, then for every redex u in t there exists a term s' in $s \twoheadrightarrow t$ such that u is the unique residual of a redex in s'.*

Proof. Suppose that u is a redex in t that occurs at position p. By definition of rewrite rules, it follows that the left-hand side of the rewrite rule employed in u is finite. Hence, there exists a depth d such that all positions in the redex pattern of u have depth strictly less than d. By strong convergence we may write $s \twoheadrightarrow t$ as $s \twoheadrightarrow s' \twoheadrightarrow t$ such that all steps in $s' \twoheadrightarrow t$ occur below depth d. By left-linearity and fully-extendedness it follows that a redex v occurs at position p in s' with u the unique residual of v. □

In Lemma 4.28 below, we need the notion of a reduct of a subterm (in addition to descendants and residuals).

Definition 4.27. Let $s_0 \twoheadrightarrow^{\alpha} s_\alpha$. Moreover, let $p_0 \in \mathcal{P}os(s_0)$ and $p_\alpha \in \mathcal{P}os(s_\alpha)$. The subterm $s_\alpha|_{p_\alpha}$ is called a *reduct* of $s_0|_{p_0}$ if for every $\beta \leq \alpha$ there exists a position q_β in s_β with $q_\alpha = p_\alpha$ such that:

- if $\beta = 0$, then $q_0 = p_0$,
- if $\beta = \beta' + 1$, then $q_\beta = q_{\beta'}$ unless $s_{\beta'} \to s_{\beta'+1}$ contracts a redex strictly above $q_{\beta'}$ in which case $q_\beta \in q_{\beta'}/(s_{\beta'} \to s_{\beta'+1})$, and
- if β is a limit ordinal, then $q_\beta = q_\gamma$ for all large enough $\gamma < \beta$.

A position $q \geq p_\alpha$ in s_α is said to occur in a *reduct* $s_\alpha|_{p_\alpha}$ of $s_0|_{p_0}$ if, for all positions $p_\alpha < p' \leq q$ in s_α, the subterm $s_\alpha|_{p'}$ is a reduct of a subterm strictly below p_0 in s_0.

The above notion generalises the usual notion of a reduct. The usual notion is obtained by taking the root position for every q_β. There is a slight difference between reducts and descendants: Contracting a redex at a position p yields a reduct at position p, while the position p does not have a descendant.

The lemma below precludes nestings of residuals from occurring in reductions unless the conditions mentioned in the lemma are met. This is due to the fact that nestings of subterms can only be created by substitution of bound variables. A full proof of the lemma can be found in [4] on page 29.

Lemma 4.28. *Let $s_0 \twoheadrightarrow^{\alpha} s_\alpha$ and suppose u_α and v_α in s_α are residuals of redexes in s_0. Denote for all $\gamma \leq \alpha$ by u_γ and v_γ, respectively, the unique redexes at positions p_γ and q_γ in s_γ of which u_α and v_α are residuals. Assume for all $\gamma < \alpha$ that if the step $s_\gamma \to s_{\gamma+1}$ contracts a redex at prefix position of q_γ then the redex is a residual of a redex in s_0. Then, given that a variable bound by an abstraction in the redex pattern of u_α occurs in v_α, it follows that (a) $p_0 < q_0$ and (b) q_α occurs in the reduct $s_\alpha|_{p_\alpha}$ of $s_0|_{p_0}$.*

Proof (Sketch). Suppose that a variable bound by an abstraction in the redex pattern of u_α occurs in v_α. We reason by transfinite induction on α, the length of the reduction $s_0 \twoheadrightarrow^{\alpha} s_\alpha$. In case $\alpha = 0$, the result is immediate since bound variables can only occur below the abstraction by which they are bound.

In case α is a successor ordinal, suppose either that (a) $p_0 \not< q_0$, or that (b) $q_\alpha \geq p_\alpha$, but q_α does not occur in the reduct $s_\alpha|_{p_\alpha}$ of $s_0|_{p_0}$. The proof is then a straightforward, but tedious analysis on these two cases.

In case α is a limit ordinal, the result is immediate by strong convergence and the induction hypothesis, since residuals occur at finite depth. □

5. Compression

Compression is a feature of (strongly convergent) infinitary rewriting that, in essence, allows us to shorten reductions of arbitrary lengths to much more manageable reductions of length at most ω.

$$\begin{tikzcd}
\bar{\sigma}_0(l) \arrow[r, twoheadrightarrow, "\leq\omega"] \arrow[d] & \bar{\sigma}_\omega(l) \arrow[d] \\
\bar{\sigma}_0(r) \arrow[r, twoheadrightarrow, "\leq\omega"] & \bar{\sigma}_\omega(r)
\end{tikzcd}$$

Figure 3: Lemma 5.1

In this section, we prove the compression property for fully-extended, left-linear iCRSs. Fully-extendedness and left-linearity ensure that no redex is created by either making two subterms equal in an infinite number of steps or by erasing some variable in an infinite number of steps. We will also show that these two assumptions cannot be omitted.

We note that the proof of the compression property in earlier, more restrictive settings, for example iλc [9, 10] are essentially independent of the concrete notion of rewriting, except for the details of the crucial observation that, for left-linear systems, one can 'pull back' a redex across a reduction that does not affect the positions in the redex pattern. We prove this property specifically for iCRSs in the following lemma:

Lemma 5.1. *For every fully-extended, left-linear iCRS, if $\bar{\sigma}_0(l) \twoheadrightarrow^{\leq\omega} \bar{\sigma}_\omega(l) \to \bar{\sigma}_\omega(r)$ such that l is the redex pattern of the redex contracted in $\bar{\sigma}_\omega(l) \to \bar{\sigma}_\omega(r)$ and such that no step along $\bar{\sigma}_0(l) \twoheadrightarrow \bar{\sigma}_\omega(l)$ occurs at a position in the pattern of l, then $\bar{\sigma}_0(l) \to \bar{\sigma}_0(r) \twoheadrightarrow^{\leq\omega} \bar{\sigma}_\omega(r)$ (see Figure 3).*

Proof. Let $\bar{\sigma}_0(l) \twoheadrightarrow^{\leq\omega} \bar{\sigma}_\omega(l) \to \bar{\sigma}_\omega(r)$. By left-linearity and fully-extendedness we have that $\bar{\sigma}_0(l) \to \bar{\sigma}_0(r)$. Hence, $\bar{\sigma}_0(r) \twoheadrightarrow^{\leq\omega} \bar{\sigma}_\omega(r)$ is left to prove.

Since the left-hand side of each rewrite rule is a pattern and since no redex contracted along $\bar{\sigma}_0(l) \twoheadrightarrow^{\leq\omega} \bar{\sigma}_\omega(l)$ occurs at a position in the redex pattern of l, it follows that $\bar{\sigma}_0(l) \twoheadrightarrow \bar{\sigma}_\omega(l)$ consists of a finite number of interleaved, strongly convergent reductions of length at most ω: one reduction for each meta-variable Z that occurs in l, reducing $\sigma_0(Z)(\vec{x})$ to $\sigma_\omega(Z)(\vec{x})$. By Lemma 4.18 we may write:

$$\sigma_0(Z)(\vec{x}) \to^* \sigma_1(Z)(\vec{x}) \to^* \cdots \to^* \sigma_d(Z)(\vec{x}) \to^* \sigma_{d+1}(Z)(\vec{x}) \to^* \cdots \sigma_\omega(Z)(\vec{x}) \,,$$

where for each $d \geq 0$ we have that all steps in $\sigma_d(Z)(\vec{x}) \twoheadrightarrow \sigma_\omega(Z)(\vec{x})$ occur at depth d or below. Hence, $\sigma_d(Z)(\vec{x}) \twoheadrightarrow \sigma_\omega(Z)(\vec{x})$ is possibly empty. Moreover, by left-linearity, we may replace the variables $\vec{x}$ by arbitrary terms $\vec{t}$ to obtain a reduction $\sigma_d(Z)(\vec{t}) \to^* \sigma_{d+1}(Z)(\vec{t})$. No nesting of the terms in $\vec{t}$ can occur, as the free variables in $\vec{t}$ are also free in $\sigma_d(Z)(\vec{t})$.

We now show for all $d \geq 0$ that there exists a reduction $s_d \to^* s_{d+1}$ with all rewrite steps occurring at depth d or below and such that $d(s_d, \bar{\sigma}_\omega(r)) \leq 2^{-d}$ and $s_0 = \bar{\sigma}_0(r)$. To do so, consider the map γ which assigns to each $p \in \mathcal{P}os(r)$ the number of prefix positions of p at which *no* meta-variable occurs and define for each d a sequence of meta-terms $r_{d,0}, r_{d,1}, r_{d,2}, \ldots$ with $r_{d,0} = r$ and $r_{d,i+1} =$

$\bar{\sigma}'_{d,i}(r'_{d,i})$ where $r'_{d,i}$ is obtained from $r_{d,i}$ by labelling each meta-variable with its position in $r_{d,i}$ and such that $\bar{\sigma}'_{d,i}$ is the valuation induced by the map $\sigma'_{d,i}$ defined as:

$$\sigma'_{d,i}(Z^p) = \begin{cases} \sigma_{d-\gamma(p)}(Z) & \text{if } \gamma(p) < d \text{ and } |p| < i \\ \sigma_0(Z) & \text{if } \gamma(p) \geq d \text{ and } |p| < i \\ Z & \text{if } |p| \geq i \end{cases}$$

where during application of $\bar{\sigma}'_{d,i}$ we temporarily consider the meta-variables introduced in the last clause to be function symbols of appropriate arity. By Lemma 3.20 and since no further nestings can be created, as remarked above, all remaining meta-variables will occur at progressively greater depths along the defined sequence of meta-terms. Hence, the sequence converges to a term, we define this term to be s_d.

For each Z^p with $\sigma_{d-\gamma(p)}$ applied, consider the reduction $\sigma_{d-\gamma(p)}(Z)(\vec{x}) \to^* \sigma_{d+1-\gamma(p)}(Z)(\vec{x})$. By definition of s_d of s_{d+1}, it follows that $s_d \to^* s_{d+1}$. Since all steps in $\sigma_{d-\gamma(p)}(Z)(\vec{x}) \to^* \sigma_{d+1-\gamma(p)}(Z)(\vec{x})$ for Z at position p occur at depth $d - \gamma(p)$ or below and since there are $\gamma(p)$ prefix positions of p at which no meta-variable occurs, all rewrite steps in $s_d \to^* s_{d+1}$ occur at depth d or below. Moreover, since all rewrite steps in $\sigma_{d-\gamma(p)}(Z)(\vec{x}) \twoheadrightarrow \sigma_\omega(Z)(\vec{x})$ also occur at depth $d - \gamma(p)$ or below, we also have $d(s_d, \bar{\sigma}_\omega(r)) \leq 2^{-d}$.

By construction of $s_d \to^* s_{d+1}$ it follows that

$$s_0 \to^* s_1 \to^* \cdots \to^* s_d \to^* s_{d+1} \to^* \cdots \bar{\sigma}_\omega(r)$$

is a strongly convergent reduction of length at most ω. Moreover, as $\bar{\sigma}_0(r) = s_0$, we have $\bar{\sigma}_0(r) \twoheadrightarrow^{\leq\omega} \bar{\sigma}_\omega(r)$. Hence, the result follows. $\square$

The above result does not hold in case we allow steps along $\bar{\sigma}_0(l) \twoheadrightarrow \bar{\sigma}_\omega(l)$ to occur at positions in the pattern of l. To see this, consider the following two rules:

$$f(a) \to f(b)$$
$$f(Z) \to g(Z)$$

and define $l = f(Z)$ and $r = g(Z)$ with $\sigma_0(Z) = a$ and $\sigma_\omega(Z) = b$. We obtain $\bar{\sigma}_0(l) = f(a) \to f(b) = \bar{\sigma}_\omega(l)$ and $\bar{\sigma}_\omega(l) = f(b) \to g(b) = \bar{\sigma}_\omega(r)$ by, respectively, the first and second rule, where the first step occurs at the position of the function symbol f in the pattern $f(Z)$. Although we also have $\bar{\sigma}_0(l) = f(a) \to g(a) = \bar{\sigma}_0(r)$ by the second rule, $g(a)$ cannot be reduced to $g(b)$, as required by the lemma.

The main result of this section is now as follows:

Theorem 5.2 (Compression). *For every fully-extended, left-linear iCRS, if $s \twoheadrightarrow^\alpha t$, then $s \twoheadrightarrow^{\leq\omega} t$.*

Proof. Let $s \twoheadrightarrow^\alpha t$ and proceed by ordinal induction on α. By the proof of Theorem 12.7.1 in [10] it suffices to show that the theorem holds for $\alpha = \omega + 1$:

The cases where α is 0, a limit ordinal, or a successor ordinal greater than $\omega+1$ do not depend on the definition of rewriting.

For the case $\alpha = \omega + 1$, it suffices to note that the details of the proof in [10, Theorem 12.7.1] are independent of the notion of rewriting as long as the properties of Lemmas 4.18 and 5.1 hold. □

We next show that the assumptions of left-linearity and fully-extendedness cannot be omitted from the previous theorem. In addition, we show that omitting the finite chains property from the definition of rewrite rules can also make compression fail.

Example 5.3 (Failure of compression without left-linearity). In case left-linearity is omitted, failure of compression follows if we interpret the counterexample to compression for non-left-linear iTRSs [8] in the context of iCRSs. Suppose we have at our disposal the following three rewrite rules:

$$\begin{aligned} f(Z, Z) &\to c \\ a &\to g(a) \\ b &\to g(b) \end{aligned}$$

Obviously, the first of the above rules is not left-linear. Now consider the following reduction of length $\omega + 1$:

$$f(a, b) \to^* f(g(a), g(b)) \to^* f(g^2(a), g^2(b)) \to^* \cdots f(g^\omega, g^\omega) \to c$$

The reduction cannot the compressed to a reduction of length at most ω because ω steps are required to reduce both $g(a)$ and $g(b)$ to g^ω and because the two arguments of f differ as long as $g(a)$ and $g(b)$ have not been reduced to g^ω.

Example 5.4 (Failure of compression without fully-extendedness). Consider the following two rewrite rules:

$$\begin{aligned} f([x]Z) &\to Z \\ g(Z) &\to h(g(Z)) \end{aligned}$$

The first of the above two rewrite rules is not fully-extended, as the meta-variable Z on the left-hand side occurs in the scope of the abstraction $[x]$, while x is not an argument of Z. Now consider the following reduction:

$$f([x]g(x)) \to f([x]h(g(x))) \to \cdots f([x]h^\omega) \to h^\omega$$

The reduction cannot be compressed to a reduction of length at most ω, because ω steps are required to reduce $g(x)$ to h^ω and because the variable x occurs bound as long as $g(x)$ has not been reduced to h^ω.

As an alternative to the above, failure of compression in the case of non-fully-extendedness also follows by interpreting $\lambda\beta\eta$-calculus in the context of iCRSs: The η-rule is not fully-extended. Failure of compression to reductions of at most length ω is demonstrated in [9]. However, as can be deducted from

Lemma 5 in [49], a slightly different compression property does hold in the case of $\lambda\beta\eta$-calculus: Each reduction can be compressed to a reduction of length at most $\omega + \omega$. This result is not mentioned explicitly in [49], but a similar result occurs in Theorem 9 of that paper: It is proved that compression to at most $\omega + \omega$ holds for $\lambda\beta\eta$-calculus extended with a fresh constant $\perp$ and a "rule schema" $t \to \perp$ that applies to terms t not reducible to head normal form.

Lastly, we show that the finite chains property, which underlies much of treatment of iCRSs (see Sections 1.2 and 3.3), is also needed for compression:

Example 5.5 (Failure of compression without the finite chains property). Assume we have at our disposal the following two rewrite rules:

$$\begin{aligned} f([x]Z(x), [y]Z'(y)) &\to Z'(Z^\omega) \\ g(Z) &\to h(g(Z)) \end{aligned}$$

Obviously, the right-hand side of the first rule does not satisfy the finite chains property. Now consider the following reduction:

$$f([x]x, [y]g(y)) \to f([x]x, [y]h(g(y))) \to \cdots f([x]x, [y]h^\omega) \to h^\omega$$

Compression fails, as the first rule cannot be applied to $f([x]x, [y]g(y))$, or for that matter to any $f([x]x, [y]h^n(g(y)))$ with $n \in \mathbb{N}$, because we have:

$$f([x]x, [y]h^n(g(y))) = \bar{\sigma}(f([x]Z(x), [y]Z'(y))) \, ,$$

with $\sigma(Z) = \underline{\lambda} x.x$ and $\sigma(Z') = \underline{\lambda} y.h^n(g(y))$, and:

$$\bar{\sigma}(Z'(Z^\omega)) = (\underline{\lambda} y.h^n(g(y)))((\underline{\lambda} x.x)((\underline{\lambda} x.x)(\ldots((\underline{\lambda} x.x)(\ldots))))) \, ,$$

which obviously has no complete development of its parallel β-redexes.

6. Developments

In this section we prove that each complete development of the same set of redexes in an orthogonal iCRS ends in the same term.

Assuming in the remainder of this section that every iCRS is orthogonal and that s is a term and $\mathcal{U}$ a set of redexes in s, we first define developments:

Definition 6.1. A *development* of $\mathcal{U}$ is a strongly convergent reduction such that each step contracts a residual of a redex in $\mathcal{U}$. A development $s \twoheadrightarrow t$ is called *complete* if $\mathcal{U}/(s \twoheadrightarrow t) = \emptyset$. Moreover, a development is called *finite* if $s \twoheadrightarrow t$ is finite.

A complete development of a set of redexes does not necessarily exist in the infinite case. Consider for example the rule $f(Z) \to Z$ and the term f^ω. The set of all redexes in f^ω does not have a complete development: After any (partial) development a residual of a redex in f^ω always remains at the root of the resulting term. Hence, any complete development will have an infinite number of root-steps and thus is not strongly convergent, contrary to requirements.

Remark 6.2. Although the above the definition and the results below concern orthogonal iCRSs, they can be interpreted in the more liberal context of orthogonal sets of redexes (where no restrictions are placed upon the iCRSs but where it is assumed that there is no overlap between the patterns of the redexes that occur in $\mathcal{U}$). No modification of either the above definition or the proofs below is necessary. Even though this is the case, we opt to work in the context of orthogonal iCRSs, as this seems most common throughout the literature on rewrite systems.

6.1. Paths and finite jumps

To prove that each complete development of the same set of redexes ends in the same term, we extend the technique of the Finite Jumps Developments Theorem [10] to orthogonal iCRSs.

The crucial notion is that of a *path*, first used in (finitary) λ-calculus for characterising redex families [50, 51]. Intuitively, a path is a sequence of nodes that "jumps back and forth" between (positions in) a term and the right-hand sides of rules of redexes that occur in that term. Consider a set $\mathcal{U}$ of redexes in a term s. A path 'traces' through s starting at the root and proceeding to increasingly greater depths. If a redex in $\mathcal{U}$, or a variable bound by a redex in $\mathcal{U}$, is encountered on a path, a 'jump' is made to the right-hand side of the employed rewrite rule. The path continues there until a meta-variable is met, at which point a jump back to the original term is made (see Figure 4).

As paths 'jump' when redexes and bound variables are met, each path intuitively traces possible nestings of elements of $\mathcal{U}$ that would occur when contracting a redex from $\mathcal{U}$. To reason about the notoriously difficult problem of nestings in higher-order systems, one may then consider the set of all paths induced by $\mathcal{U}$ and the changes to the paths by contraction of redexes. Note that even though a path can be infinite, it will turn out that a set of redexes has well-defined complete developments iff every path has a *finite* number of jumps.

Our definition of path is necessarily more complicated than the one for iλc in [10]: There, the only rule to consider is β-reduction, the right-hand side of which consists of a simple substitution. In the presence of the more complex rules of iCRSs, we need to ensure that each possible right-hand side is treated correctly when defining a path. Apart from purely syntactical changes, we extend the notion of paths from [10] in one crucial way: We also consider *projections* of paths. Projections afford a way of extracting the syntactic symbols from paths to obtain the final (and unique) term of a complete development of $\mathcal{U}$.

In the following, we denote by p_u the position of the redex u in s. Moreover, we say that a variable x is *bound by a redex* u if x is bound by an abstraction $[x]$ which occurs in the left-hand side of the rewrite rule employed in u.

Definition 6.3. A *path* of s with respect to $\mathcal{U}$ is a sequence of alternating nodes and edges. Each node is labelled either (s, p) with $p \in \mathcal{P}os(s)$ or (r, p, p_u) with r the right-hand side of a rewrite rule, $p \in \mathcal{P}os(r)$, and $u \in \mathcal{U}$. Each edge is directed and either unlabelled or labelled with an element of $\mathbb{N}$.

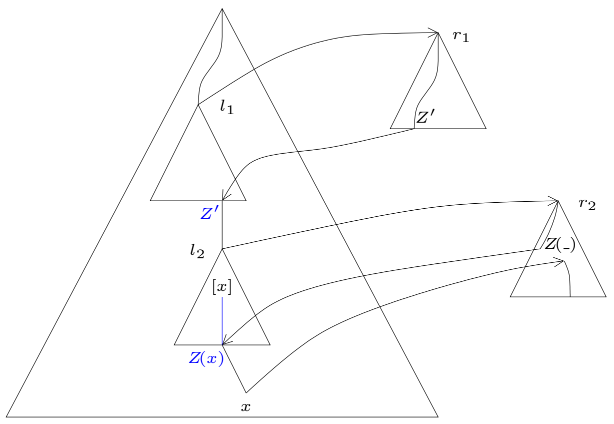


Figure 4: A path tracing through a term: when a redex or a bound variable is met in a trace, a ‘jump’ is made to the right-hand side of the employed rewrite rule and the trace continues there until a meta-variable is encountered

Every path starts with a node labelled (s, ϵ). If a node n of a path is labelled (s, p) and has an outgoing edge to a node n', then:

1. if $s|_p$ is neither a redex in $\mathcal{U}$ nor a variable bound by a redex in $\mathcal{U}$, then for some $i \in \mathcal{P}os(s|_p) \cap \mathbb{N}$ the node n' is labelled $(s, p \cdot i)$ and the edge from n to n' is labelled i,
2. if $s|_p$ is a redex $u \in \mathcal{U}$ with $l \to r$ the employed rewrite rule, then the node n' is labelled (r, ϵ, p_u) and the edge from n to n' is unlabelled,
3. if $s|_p$ is a variable x bound by a redex $u \in \mathcal{U}$ with $l \to r$ the employed rewrite rule, then the node n' is labelled $(r, p' \cdot i, p_u)$ and the edge from n to n' is unlabelled, such that (r, p', p_u) was the last node before n with p_u, $root(r|_{p'}) = Z$, and $l|_{q \cdot i} = x$ with q the unique position of Z in l.

If a node n of a path is labelled (r, p, p_u) and has an outgoing edge to a node n', then:

1. if $root(r|_p)$ is not a meta-variable, then for some $i \in \mathcal{P}os(r|_p) \cap \mathbb{N}$ the node n' is labelled $(r, p \cdot i, p_u)$ and the edge from n to n' is labelled i,
2. if $root(r|_p)$ is a meta-variable Z, then the node n' is labelled $(s, p_u \cdot q)$ and the edge from n to n' is unlabelled, such that $l \to r$ is the rewrite rule employed in u and such that q is the unique position of Z in l.

A path ends in case we encounter a nullary function symbol or a variable not bound by a redex of s in $\mathcal{U}$ (this is automatically the case for any variable that occurs on the right-hand side of a rewrite rule). This is immediate by the fact that $\mathcal{P}os(t) \cap \mathbb{N}$ is empty in case t is either a variable or nullary function symbol.

We say that a path is *maximal* if it is not a proper prefix of another path. We write a path Π as a (possibly infinite) sequence of alternating nodes and edges $\Pi = n_1 e_1 n_2 \cdots$.

Definition 6.4. Let $\Pi = n_1 e_1 n_2 \cdots$ be a path of s with respect to $\mathcal{U}$. The *path projection* $\phi(\Pi)$ of Π is a sequence of alternating nodes and edges $\phi(\Pi) = \phi(n_1)\phi(e_1)\phi(n_2)\cdots$. Each node $\phi(n)$ is either unlabelled or labelled with a function symbol or variable such that:

1. if n is labelled (s, p), then $\phi(n)$ is unlabelled if $s|_p$ is a redex in $\mathcal{U}$ or a variable bound by such a redex and it is labelled $root(s|_p)$ otherwise, and
2. if n is labelled (r, p, q), then $\phi(n)$ is unlabelled if $root(r|_p)$ is a meta-variable and it is labelled $root(r|_p)$ otherwise.

Each edge $\phi(e)$ is either labelled with an element of $\mathbb{N}$ or labelled ϵ such that if e is labelled i, then $\phi(e)$ has the same label, and if e is unlabelled, then $\phi(e)$ is labelled ϵ.

Note that the nodes of path projections are either unlabelled or labelled with *function symbols*, *variables*, or *abstractions*; this is contrary to paths whose nodes are labelled with *pairs* and *triples*.

Example 6.5. Consider the orthogonal iCRS that only has the following rewrite rule, also denoted $l \to r$:

$$f([x]Z(x), Z') \to Z(g(Z(Z'))) .$$

Given the terms $s = f([x]g(x), a)$ and $t = g(g(g(a)))$ and the set $\mathcal{U}$ containing the only redex in s, we have that $s \to t$ is a complete development of $\mathcal{U}$.

The term s has one maximal path with respect to $\mathcal{U}$:

$$\begin{aligned}(s,\epsilon) \to (r,\epsilon,\epsilon) \to (s,10) \xrightarrow{1} (s,101) \to (r,1,\epsilon) \xrightarrow{1} (r,11,\epsilon) \\ \to (s,10) \xrightarrow{1} (s,101) \to (r,111,\epsilon) \to (s,2)\end{aligned}$$

Moreover, the term t has one maximal path with respect to $\mathcal{U}/(s \to t) = \emptyset$:

$$(t,\epsilon) \xrightarrow{1} (t,1) \xrightarrow{1} (t,11) \xrightarrow{1} (t,111) .$$

The path projections of the maximal paths are, respectively,

$$\cdot \xrightarrow{\epsilon} \cdot \xrightarrow{\epsilon} g \xrightarrow{1} \cdot \xrightarrow{\epsilon} g \xrightarrow{1} \cdot \xrightarrow{\epsilon} g \xrightarrow{1} \cdot \xrightarrow{\epsilon} \cdot \xrightarrow{\epsilon} a$$

and

$$g \xrightarrow{1} g \xrightarrow{1} g \xrightarrow{1} a .$$

Let $\mathcal{P}(s,\mathcal{U})$ denote the set of path projections of *maximal paths* of s with respect to $\mathcal{U}$. The following two results can be witnessed in the above example, their proofs are simple, but tedious and lengthy, hence occur in Appendix Appendix A.2.

Proposition 6.6. *The map ϕ defines a bijection between the set of paths and the set of path projections, respectively between maximal paths and the path projections in $\mathcal{P}(s,\mathcal{U})$.*

Lemma 6.7. *Let $u \in \mathcal{U}$ and let $s \to t$ be the rewrite step contracting u. There exists a bijection between $\mathcal{P}(s,\mathcal{U})$ and $\mathcal{P}(t,\mathcal{U}/u)$. Given a path projection $\phi(\Pi) \in \mathcal{P}(s,\mathcal{U})$, its image under the bijection is obtained by deleting finite sequences of unlabelled nodes and ϵ-labelled edges from $\phi(\Pi)$.*

We next define a property of $\mathcal{U}$, based on $\mathcal{P}(s,\mathcal{U})$: The *finite jumps property*. We also introduce some terminology to relate a term to $\mathcal{P}(s,\mathcal{U})$.

Definition 6.8. The set $\mathcal{U}$ has the *finite jumps property* if no path projection occurring in $\mathcal{P}(s,\mathcal{U})$ contains an infinite sequence of unlabelled nodes and ϵ-labelled edges. Moreover, a term t *matches* $\mathcal{P}(s,\mathcal{U})$ if, for all $\phi(\Pi) \in \mathcal{P}(s,\mathcal{U})$ and all prefixes of $\phi(\Pi)$ ending in a node $\phi(n)$ labelled f, it holds that $root(t|_p) = f$, where p is the concatenation of the edge labels in the prefix (starting at the first node of $\phi(\Pi)$ and ending in $\phi(n)$).

We now prove an ancillary result concerning finite jumps.

Proposition 6.9. *If $\mathcal{U}$ has the finite jumps property, then there exists a unique term, denoted $\mathcal{T}(s,\mathcal{U})$, that matches $\mathcal{P}(s,\mathcal{U})$.*

Proof (Sketch). Let $\mathcal{P}_p(s,\mathcal{U})$ denote the set of all prefixes of path projections in $\mathcal{P}(s,\mathcal{U})$ such that the concatenation of the edge labels for each prefix is p and such that each prefix ends in a labelled node. The proof proceeds by induction on p in a fashion identical to the proof of Proposition 12.5.8 in [10]. □

We can now prove the Finite Jumps Developments Theorem:

Theorem 6.10 (Finite Jumps Developments Theorem)**.** *If $\mathcal{U}$ has the finite jumps property, then:*

1. *every complete development of $\mathcal{U}$ ends in $\mathcal{T}(s,\mathcal{U})$,*
2. *for any $p \in \mathcal{P}os(s)$, the set of descendants of p by a complete development of $\mathcal{U}$ is independent of the complete development,*
3. *for any redex u of s, the set of residuals of u by a complete development of $\mathcal{U}$ is independent of the complete development, and*
4. *$\mathcal{U}$ has a complete development.*

Proof (Sketch). (1) Assuming there exists a complete development, the proof is identical to the proof of Proposition 12.5.9 in [10], except that Lemma 6.7 is employed instead of tracing. The proof proceeds by ordinal induction showing that for every s_α in the complete development with residuals $\mathcal{U}_\alpha = \mathcal{U}/(s \twoheadrightarrow s_\alpha)$ of $\mathcal{U}$, we have that $\mathcal{P}(s_\alpha,\mathcal{U}_\alpha)$ can be obtained from $\mathcal{P}(s,\mathcal{U})$ by deleting finite sequences of unlabelled nodes and ϵ-labelled edges from the elements of $\mathcal{P}(s,\mathcal{U})$. The proof is then concluded by employing Proposition 6.9.

(2) In analogy to [19, Section II.2], the proof proceeds by labelling subterms where the rewrite rules take into account the labels that occur such that each reduction in the labelled version corresponds to a reduction in the original iCRS by removal of all labels and vice versa. If the subterms of some term have a certain label k, then the descendants of these subterms across some reduction

are precisely the subterms labelled k in the final term. Finally, the first clause of the current theorem is applied, taking into account the labelling.

(3) By the second clause of the current proof and orthogonality of the assumed iCRS.

(4) As in the proof of Proposition 12.5.9 in [10] it suffices to observe that contracting redexes in an outside-in fashion only affects parts of maximal paths following the node corresponding to the root of the contracted redex (this follows by inspection of the proof of Lemma 6.7). As only finite sequences of nodes and edges occur and since terms are finitely branching, the contracted redexes occur at increasingly greater depths along the constructed reduction, yielding a strongly convergent reduction that is a complete development. □

6.2. Developments

With the Finite Jumps Developments Theorem in hand, we can now precisely characterise the sets of redexes having complete developments. This characterisation seems to be new.

Recall that we are working with an orthogonal iCRS and that $\mathcal{U}$ is a set of redexes in a term s.

Lemma 6.11. *The set $\mathcal{U}$ has a complete development iff $\mathcal{U}$ has the finite jumps property.*

Proof. To prove that the finite jumps property holds if $\mathcal{U}$ has a complete development, suppose $\mathcal{U}$ does not have the finite jumps property. That is, there exists a path projection $\phi(\Pi)$ of s with respect to $\mathcal{U}$ that ends in an infinite sequence of unlabelled nodes and ϵ-labelled edges.

We show by ordinal induction for every s_α in the complete development, with residuals $\mathcal{U}_\alpha = \mathcal{U}/(s \twoheadrightarrow s_\alpha)$ of $\mathcal{U}$, that there exists a path projection $\phi(\Pi_\alpha)$ of s_α with respect to $\mathcal{U}_\alpha$ that has an infinite sequence of unlabelled nodes and ϵ-labelled edges. For $s_0 = s$, this is immediate.

For $s_{\alpha+1}$, we have by the induction hypothesis that there exists path projection $\phi(\Pi_\alpha)$ that has an infinite sequence of unlabelled nodes and ϵ-labelled edges. By Lemma 6.7, the path projection $\phi(\Pi_{\alpha+1})$ is obtained by deleting finite sequences of unlabelled nodes and ϵ-labelled edges from $\phi(\Pi_\alpha)$. Hence, $\phi(\Pi_{\alpha+1})$ must also have an infinite sequence of unlabelled nodes and ϵ-labelled edges.

For s_α with α a limit ordinal, we have by strong convergence that $\phi(\Pi_\alpha)$ can be obtained from $\phi(\Pi)$ by deleting all unlabelled nodes and ϵ-labelled edges deleted in the previous steps. Also by strong convergence, the deleted nodes and edges occur at an increasingly greater distance from the starting node of the path projections considered along the complete development. Hence, an increasing part of the infinite sequence of unlabelled nodes and ϵ-labelled edges is stable along the complete development. This implies that $\phi(\Pi_\alpha)$ also has such an infinite sequence.

Thus, a path projection that has an infinite sequence of unlabelled nodes and ϵ-labelled edges is present after each complete development of $\mathcal{U}$. By definition of paths and path projections this implies that a residual of a redex in $\mathcal{U}$ occurs in the final term of the complete development. However, this contradicts the fact that no residuals of redexes in $\mathcal{U}$ occur in the final term of a complete development. Hence, $\mathcal{U}$ must have the finite jumps property.

That $\mathcal{U}$ has a complete development if it has the finite jumps property is an immediate consequence of Theorem 6.10(4). □

The result we were aiming at now follows easily.

Theorem 6.12. *If $\mathcal{U}$ has a complete development, then all complete developments of $\mathcal{U}$ end in the same term.*

Proof. By Lemma 6.11, if $\mathcal{U}$ has a complete development, then it has the finite jumps property. But then, each complete development of $\mathcal{U}$ ends in the same term by Theorem 6.10(1). □

6.3. Properties of developments

We next prove a number of properties of complete developments that will be of use in [3, 4]. Again, recall that we assume to be working in an orthogonal iCRS and that $\mathcal{U}$ is a set of redexes in a term s.

Notation 6.13. If there exists a complete development of $\mathcal{U}$ resulting in a term t, then we write $s \Rightarrow t$, where the arrow is adorned with $\mathcal{U}$ if needed.

Lemma 6.14. *If $\mathcal{U}$ has a complete development and if $s \twoheadrightarrow t$ is a (not necessarily complete) development of $\mathcal{U}$, then $\mathcal{U}/(s \twoheadrightarrow t)$ has a complete development.*

Proof. As $\mathcal{U}$ has the finite jumps property by Lemma 6.11, it follows by inspection of the proof of Theorem 6.10(1) that $\mathcal{U}/(s \twoheadrightarrow t)$ also has the finite jumps property. Hence, the result now follows by applying Lemma 6.11 to $\mathcal{U}/(s \twoheadrightarrow t)$. □

Lemma 6.15. *If $\mathcal{U}$ has a complete development and if u is a redex in s, then $\mathcal{U} \cup \{u\}$ has a complete development.*

Proof. Perform a complete development of $\mathcal{U}$, resulting in a term t. By definition of valuations and substitutions, nestings of u can only be created by the redexes above u in the initial term. As there are only finitely many such redexes and as the right-hand side of each rewrite rules satisfies the finite chains property, only finite chains of residuals of u occur in t (though infinite nestings are still possible). Repeatedly contract a residual of u that is at minimal depth. As only finite chains of residuals of u occur and as residuals of u cannot nest other residuals of u, it follows by terms being finitely branching that the minimal depth at which redexes are contracted increases after a finite number of steps. Hence, the reduction defined in this way is strongly convergent and contracts all residuals of u, resulting in a complete development of $\mathcal{U} \cup \{u\}$. □

Remark that the above lemma fails in case we do not require right-hand sides of rewrite rules to satisfy the finite chains property:

Example 6.16 (Failure of Lemma 6.15 without the finite chains property). Consider the following two rewrite rules, the first of which does not satisfy the finite chains property:

$$\begin{aligned} f([x]Z(x)) &\to Z^\omega \\ g(Z) &\to Z \end{aligned}$$

Now consider the term $f([x]g(x))$. Obviously, the singleton set consisting of the $g(Z) \to Z$-redex has a complete development:

$$f([x]g(x)) \to f([x]x)\,.$$

However, the set consisting of both redexes in $f([x]g(x))$ does *not* have a complete development: If we first reduce the $g(Z) \to Z$-redex, we can no longer contract the remaining $f([x]Z(x)) \to Z^\omega$-redex. If the $f([x]Z(x)) \to Z^\omega$-redex is contracted first, any development of the set of residuals of the $g(Z) \to Z$-redex from the original term must leave a residual at the root of the final term of the development, hence this development is not complete.

Proposition 6.17. *Let $\mathcal{U}$ have a complete development $s \Rightarrow t$ and let v be a redex in s. The following diagram commutes:*

$$\begin{tikzcd} s \arrow[r, "v"] \arrow[d, Rightarrow, "\mathcal{U}"] & t' \arrow[d, Rightarrow, "\mathcal{U}/(s\to^{v} t')"] \\ t \arrow[r, Rightarrow, "v/(s\Rightarrow^{\mathcal{U}} t)"'] & s' \end{tikzcd}$$

Proof. Immediate by Lemmas 6.14 and 6.15, Theorem 6.12 and the fact that $(\mathcal{U} \cup \{v\})/(s \to^v t') = \mathcal{U}/(s \to^v t')$ and $(\mathcal{U} \cup \{v\})/(s \Rightarrow t) = v/(s \Rightarrow t)$. □

Lemma 6.18. *If $\mathcal{U}$ is finite, then $\mathcal{U}$ has a finite complete development.*

Proof. By induction on the number of redexes in $\mathcal{U}$. If $\mathcal{U}$ is empty, we are done. Otherwise, by the finiteness of $\mathcal{U}$, there exists a redex $v \in \mathcal{U}$ such that no redexes from $\mathcal{U}$ occur in its arguments. Contract v. As no redexes occur in the arguments of v, the set $\mathcal{U}/v$ contains one redex less than $\mathcal{U}$. The induction hypothesis now furnishes the result. □

The following proposition establishes a limited form of commutativity.

Proposition 6.19. *Let $\mathcal{U}$ and $\mathcal{V}$ be sets of redexes in s such that $\mathcal{U}$ has a complete development $s \Rightarrow t$ and $\mathcal{V}$ is finite. The following diagram commutes:*

$$\begin{tikzcd} s \arrow[r, Rightarrow, "\mathcal{V}"] \arrow[d, Rightarrow, "\mathcal{U}"] & t' \arrow[d, Rightarrow, "\mathcal{U}/(s\Rightarrow^{\mathcal{V}} t')"] \\ t \arrow[r, Rightarrow, "\mathcal{V}/(s\Rightarrow^{\mathcal{U}} t)"'] & s' \end{tikzcd}$$

Proof. By Lemma 6.18, we have that $\mathcal{V}$ has a finite complete development. Denote this development by

$$s = s_0 \to s_1 \to \cdots \to s_n = t' ,$$

where $s_i \to s_{i+1}$ is assumed to contract a redex v_{i+1}. By Proposition 6.17 we can erect the following diagram, where S_i denotes $s_0 \to^* s_i$ and T_{i+1} denotes $s_i \twoheadrightarrow t_i$ and where $(\mathcal{U}/S_i)/(s_i \to s_{i+1}) = \mathcal{U}/S_{i+1}$ by definition of residuals:

$$\begin{tikzcd}
s_0 \arrow[r, "v_1"] \arrow[d, Rightarrow, "\mathcal{U}"] & s_1 \arrow[r, "v_2"] \arrow[d, Rightarrow, "\mathcal{U}/S_1"] & \cdot \arrow[r, dotted, no head] \arrow[d, Rightarrow, "\mathcal{U}/S_2"] & \cdot \arrow[r, "v_n"] & s_n \arrow[d, Rightarrow, "\mathcal{U}/S_n"] \\
t_0 \arrow[r, Rightarrow, "v_1/T_1"'] & t_1 \arrow[r, Rightarrow, "v_2/T_2"'] & \cdot \arrow[r, dotted, no head] & \cdot \arrow[r, Rightarrow, "v_n/T_n"'] & t_n
\end{tikzcd}$$

The reduction $v_1/T_1; v_2/T_2; \ldots; v_n/T_n$ is a complete development of $\mathcal{V}/T_1$: By definition only residuals from redexes in $\mathcal{V}$ are contracted. Moreover, if not all residuals were contracted, then neither does S_n contract all residuals of $\mathcal{V}$, which is impossible by definition. Hence, by defining $t = t_0$ and $s' = t_n$, the result follows. $\square$

7. Tiling diagrams

We next introduce the notion of a *tiling diagram* from [10]. Moreover, we characterise the completion of tiling diagrams for orthogonal iCRSs. This characterisation extends Theorem 12.6.5 in [10] by considering not only reductions of limit ordinal length but also reductions of successor ordinal length.

Definition 7.1. A *tiling diagram* of two strongly convergent reductions $S : s_{0,0} \to^{\alpha} s_{\alpha,0}$ and $T : s_{0,0} \to^{\beta} s_{0,\beta}$ is a rectangular arrangement of strongly convergent reductions as depicted in Figure 5 such that (1) each reduction $S_{\gamma,\delta} : s_{\gamma,\delta} \twoheadrightarrow s_{\gamma+1,\delta}$ is a complete development of a set of redexes in $s_{\gamma,\delta}$, and similarly for $T_{\gamma,\delta} : s_{\gamma,\delta} \twoheadrightarrow s_{\gamma,\delta+1}$, (2) the leftmost vertical reduction is S and the topmost horizontal reduction is T, and (3) for each γ and δ the set of redexes developed in $S_{\gamma,\delta}$ is the set of residuals of the redex contracted in $s_{\gamma,0} \to s_{\gamma+1,0}$ across the (strongly convergent) reduction $T_{\gamma,[0,\delta]} : s_{\gamma,0} \to s_{\gamma,1} \to \cdots s_{\gamma,\delta}$ (symmetrically for $T_{\gamma,\delta}$).

For $S_{[0,\alpha],\beta}$ we usually write S/T and we call this reduction the *projection* of S across T (similarly for $T_{\alpha,[0,\beta]}$ and T/S). Moreover, if T consists of a single step contracting a redex u, we also write S/u (symmetrically T/u).

For orthogonal iCRSs, the following theorem now characterises the completion of tiling diagrams. As mentioned already above, the theorem extends Theorem 12.6.5 in [10] to reductions of arbitrary length. The proof focuses on the successor ordinal cases, that is, the new aspect in the characterisation; the proof for the limit ordinal case can be copied almost verbatim from [10], as explained in the proof.

$$\begin{tikzcd}
 & & & T & & & \\
 & s_{0,0} \arrow[r] \arrow[d] & s_{0,1} \arrow[r, dotted, no head] \arrow[d, two heads] & s_{0,\delta} \arrow[r] \arrow[d, two heads] & s_{0,\delta+1} \arrow[r, dotted, no head] \arrow[d, two heads] & s_{0,\beta} \arrow[d, two heads] & \\
 & s_{1,0} \arrow[r, two heads] \arrow[d, dotted, no head] & s_{1,1} \arrow[r, dotted, no head] \arrow[d, dotted, no head] & s_{1,\delta} \arrow[r, two heads] \arrow[d, dotted, no head] & s_{1,\delta+1} \arrow[r, dotted, no head] \arrow[d, dotted, no head] & s_{1,\beta} \arrow[d, dotted, no head] & \\
S & s_{\gamma,0} \arrow[r, two heads] \arrow[d] & s_{\gamma,1} \arrow[r, dotted, no head] \arrow[d, two heads] & s_{\gamma,\delta} \arrow[r, two heads, "T_{\gamma,\delta}"] \arrow[d, two heads, "S_{\gamma,\delta}"'] & s_{\gamma,\delta+1} \arrow[r, dotted, no head] \arrow[d, two heads] & s_{\gamma,\beta} \arrow[d, two heads] & S/T \\
 & s_{\gamma+1,0} \arrow[r, two heads] \arrow[d, dotted, no head] & s_{\gamma+1,1} \arrow[r, dotted, no head] \arrow[d, dotted, no head] & s_{\gamma+1,\delta} \arrow[r, two heads] \arrow[d, dotted, no head] & s_{\gamma+1,\delta+1} \arrow[r, dotted, no head] \arrow[d, dotted, no head] & s_{\gamma+1,\beta} \arrow[d, dotted, no head] & \\
 & s_{\alpha,0} \arrow[r, two heads] & s_{\alpha,1} \arrow[r, dotted, no head] & s_{\alpha,\delta} \arrow[r, two heads] & s_{\alpha,\delta+1} \arrow[r, dotted, no head] & s_{\alpha,\beta} & \\
 & & & T/S & & &
\end{tikzcd}$$

Figure 5: A tiling diagram

Theorem 7.2. *Let S and T be strongly convergent reductions starting from the same term. Suppose that a tiling diagram for S and T exists except that it is unknown whether S/T and T/S are strongly convergent and end in the same term. The following are equivalent:*

1. *The tiling diagram of S and T can be completed: S/T and T/S are strongly convergent and end in the same term.*
2. *S/T is strongly convergent.*
3. *T/S is strongly convergent.*

Proof. The first statement trivially implies the second and third. Hence, we only need to prove that the first holds under assumption of either the second or the third statement. Without loss of generality there are three cases to consider depending on the lengths, α and β, of S and T.

- In case $\alpha = \alpha' + 1$ and $\beta = \beta' + 1$, write $\mathcal{U} = u/T_{\alpha',[0,\beta']}$ and $\mathcal{V} = v/S_{[0,\alpha'],\beta'}$, where u is the redex contracted in $s_{\alpha',0} \to s_{\alpha,0}$ and v the redex contracted in $s_{0,\beta'} \to s_{0,\beta}$.

  Assume that S/T is strongly convergent. By definition of tiling diagrams, $S_{\alpha',\beta'}$ is a (not necessarily complete) development of $\mathcal{U} \cup \mathcal{V}$ and $T_{\alpha',\beta'}; S_{\alpha',\beta}$ is a complete development of $\mathcal{U} \cup \mathcal{V}$. By Lemma 6.14 and Theorem 6.12, it now follows that there exists a complete development of $(\mathcal{U} \cup \mathcal{V})/S_{\alpha',\beta'}$ that starts in $s_{\alpha',\beta}$ and ends in the same term as $T_{\alpha',\beta'}; S_{\alpha',\beta}$. As $(\mathcal{U} \cup \mathcal{V})/S_{\alpha',\beta'} = \mathcal{V}/S_{\alpha',\beta'}$ by definition of $S_{\alpha',\beta'}$, the complete development can be chosen to be a complete development of residuals of v in $s_{\alpha',\beta}$. Hence, it completes the tiling diagram as required.

The case where T/S is strongly convergent is symmetrical, as α and β are both successor ordinals.

- In case α is a limit ordinal and $\beta = \beta' + 1$, write v for the redex contracted in $s_{0,\beta'} \to s_{0,\beta}$. That S/T and T/S end in the same term follows immediately in case S/T and T/S are both strongly convergent, as this implies that for larger $\gamma < \alpha$ more residuals of v up to greater depths must occur at the same positions in $s_{\gamma,\beta'}$ and $s_{\alpha,\beta'}$. Hence, we only need to prove strong convergence.

  Assume that S/T is strongly convergent and T/S is not. By assumption there exists a position p of minimal depth d such that an infinite number of residuals of v is contracted in T/S. As only finitely many residuals of v occur in $s_{\alpha,\beta'}$ at depth less than d, it follows by Lemma 4.28 and the fact that right-hand sides of rewrite rules satisfy the finite chains property that an infinite collapsing chain of residuals of v exists in $s_{\alpha,\beta'}$. By strong convergence of $S_{[0,\alpha],\beta'}$, there exists a $\gamma < \alpha$ such that all steps in $S_{[\gamma,\alpha],\beta'}$ occur below depth d. By definition of γ, each $s_{\kappa,\beta}$ with $\gamma < \kappa < \alpha$ has a finite collapsing chain of residuals of v at position p. No infinite chain can occur at position p in $s_{\kappa,\beta'}$, since each $T_{\kappa,\beta'}$ is strongly convergent. The finite chain of residuals of v becomes arbitrary large along $S_{[\gamma,\alpha],\beta'}$, otherwise no infinite chain exists in $s_{\alpha,\beta'}$. However, this implies that for every point along the strongly convergent reduction S/T a redex is contracted at position p somewhere later along the reduction, contradiction. Hence, T/S is strongly convergent.

  Assume now that T/S is strongly convergent and S/T is not. By assumption there exists a position p of minimal depth d such that an infinite number of reductions occurs at p in S/T. Moreover, by strong convergence of $S_{[0,\alpha],\beta'}$ and the minimality of d, there exists a $\gamma < \alpha$ such that all redexes contracted in $S_{[\gamma,\alpha],\beta'}$ and $S_{[\gamma,\alpha],\beta}$ occur at depth d or below.

  In $s_{\gamma,\beta'}$ only a finite number of residuals of v occur at depth less than d. Hence, the subterms of $s_{\gamma,\beta}$ at depth d or below consist of finite chains of parallel subterms that occur at depth d in $s_{\gamma,\beta'}$, with all residuals of v contracted. By definition of γ and since the chains consist of parallel subterms, it follows that all redexes contracted in $S_{[\gamma,\alpha],\beta}$ occur within the chains and that no further nestings can be created among the chains. In fact, since the subterms are parallel, further nestings cannot even be created within the chains. But then, there exists a point along $S_{[\gamma,\alpha],\beta}$ such that precisely one of the subterms of $s_{\gamma,\beta'}$ is responsible for the infinite number of reductions at p. Since S is strongly convergent, this implies that the subterm has at its root a collapsing chain of residuals of v that becomes arbitrary large along $S_{[\gamma,\alpha],\beta'}$. Thus, there exists an infinite collapsing chain in the limit $s_{\alpha,\beta'}$. Because the chain cannot be erased by contracting other residuals of v, otherwise the infinite reduction in at p in

$S_{[\gamma,\alpha],\beta}$ does not exist, it follows that T/S cannot be strongly convergent as a complete development of residuals of v in $s_{\alpha,\beta'}$, contradiction. Hence, S/T is strongly convergent.

- If α and β are limit ordinals, then the result follows by Theorem 12.6.5 in [10]. The proof is independent of the details of rewriting, except for its use of one lemma — Lemma 12.5.12 in [10] — which also holds in the case of iCRSs and is Lemma 4.25 above. □

Although the above proof uses the finite chains property, it is unclear to us whether this property is essential. Its use can possibly be removed, albeit likely at the expense of a much more involved proof.

8. Conclusion and outlook

We have defined infinitary Combinatory Reduction Systems (iCRSs), the first notion of infinitary higher-order rewriting. The main technical results of this paper have been a compression property showing, as appropriate for all infinitary rewriting systems, that strongly convergent reductions in fully-extended, left-linear systems may be compressed to have length at most ω, as well as generalisations of well-known results concerning developments and tiling diagrams in orthogonal systems. We furthermore have unearthed a number of complications, in particular the need for finite chains and fully-extendedness that are not present in first-order infinitary rewriting and infinitary λ-calculus.

We stress to the reader that these are complications *generic* to any reasonable notion of infinitary higher-order rewriting. In particular, the need for the finite chains property will certainly surface as all notions of higher-order rewriting employ a notion of substitution in the very definition of a rewrite step, and the presence of *in*finite chains in right-hand sides of rewrite rules is inimical to well-defined substitutions of higher-order terms. We note that in many applications of infinitary rewriting, rules with finite right-hand sides are likely to be sufficient, and the difficulties with infinite chains are thus unlikely to surface in such settings.

In two companion papers [3, 4], we establish results for confluence and normalisation of orthogonal, fully-extended iCRSs; the material generalises most of the current results on infinitary rewriting in the literature.

8.1. Future work and open questions

The most pertinent and most important open question is whether the techniques developed in this paper and its companion papers can be used to extend infinitary rewriting to other formats of higher-order rewriting. We conjecture that the proof methods we have developed for iCRSs will find use in proving basic properties for such notions of rewriting, where the discussion and conjectures in Section 3.4 should be taken into account.

With regard to the current paper at least one interesting open question remains. Unlike what has been done for the first-order case [8], we have not attempted to show that compressed reductions are Lévy equivalent to the original ones. We conjecture that the proof of Theorem 5.2 actually constructs such an equivalent reduction.

We observe that as rewriting is traditionally also used for computing with equational logic, our work also allows for modelling of formulas in *infinitary logic* with quantifiers and bound variables [52] in the same fashion as is usually done in ordinary rewriting with ordinary logic; we do not yet know whether this has any implications for the possible use of infinitary logic in any practical matters.

We have also observed that the compression property fails for non-fully-extended systems: There are reductions that cannot be compressed to have length at most ω. Possibly, there exists a 'weak' compression property that allows reductions in non-fully-extended systems to be compressed to some ordinal length $\alpha > \omega$ (where α must necessarily be countable as all strongly convergent reductions have at most countable length). We conjecture that such a weak compression property *cannot* exist.

Acknowledgement

The authors extend their thanks to the anonymous referees for their diligent work and many comments that have led to substantial improvements in the readability of the paper.

References

[1] J. Ketema, J. G. Simonsen, Infinitary combinatory reduction systems, in: Proceedings of the 16th International Conference on Rewriting Techniques and Applications (RTA 2005), Vol. 3467 of Lecture Notes in Computer Science, Springer-Verlag, 2005, pp. 438–452.

[2] J. Ketema, J. G. Simonsen, On confluence of infinitary combinatory reduction systems, in: Proceedings of the 12th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR 2005), Vol. 3835 of Lecture Notes in Artificial Intelligence, Springer-Verlag, 2005, pp. 199–214.

[3] J. Ketema, J. G. Simonsen, Infinitary combinatory reduction systems: Normalising reduction strategies, Logical Methods in Computer Science 6 (1:7) (2010) 1–35.

[4] J. Ketema, J. G. Simonsen, Infinitary combinatory reduction systems: Confluence, Logical Methods in Computer Science 5 (4:3) (2009) 1–29.

[5] A. B. Webber, Modern Programming Languages: A Practical Introduction, Franklin, Beedle and Associates, 2003.

[6] P. Henderson, J. H. Morris Jr., A lazy evaluator, in: Proceedings of the 3rd ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages (POPL '76), The ACM Press, 1976, pp. 95–103.

[7] N. Dershowitz, S. Kaplan, D. A. Plaisted, Rewrite, rewrite, rewrite, rewrite, rewrite, ..., Theoretical Computer Science 83 (1) (1991) 71–96.

[8] R. Kennaway, J. W. Klop, R. Sleep, F.-J. de Vries, Transfinite reductions in orthogonal term rewriting systems, Information and Computation 119 (1) (1995) 18–38.

[9] J. R. Kennaway, J. W. Klop, M. R. Sleep, F.-J. de Vries, Infinitary lambda calculus, Theoretical Computer Science 175 (1) (1997) 93–125.

[10] R. Kennaway, F.-J. de Vries, Infinitary rewriting, in: Terese [16], Chapter 12, pp. 668–711.

[11] M. J. Plasmeijer, M. C. J. D. van Eekelen, Functional Programming and Parallel Graph Rewriting, Addison-Wesley, 1993.

[12] E. Barendsen, Term graph rewriting, in: Terese [16], Chapter 13, pp. 712–743.

[13] J. Reynolds, Definitional interpreters for higher-order programming languages, in: Proceedings of the ACM Annual Conference, Vol. 2, The ACM Press, 1972, pp. 717–740.

[14] O. Danvy, L. R. Nielsen, Defunctionalization at work, in: Proceedings of the 3rd ACM SIGPLAN International Conference on Principles and Practices of Declarative Programming (PPDP '01), The ACM Press, 2001, pp. 162–174.

[15] N. Andersen, N. D. Jones, Flow analysis of lazy higher-order functional programs, Theoretical Computer Science 375 (2007) 120–136.

[16] Terese (Ed.), Term Rewriting Systems, Vol. 55 of Cambridge Tracts in Theoretical Computer Science, Cambridge University Press, 2003.

[17] H. P. Barendregt, The Lambda Calculus: Its Syntax and Semantics, revised Edition, Elsevier Science, 1985.

[18] P. Aczel, A general Church-Rosser theorem, Tech. rep., University of Manchester (Jul. 1978).

[19] J. W. Klop, Combinatory reduction systems, Ph.D. thesis, Rijksuniversiteit Utrecht (1980).

[20] J. W. Klop, V. van Oostrom, F. van Raamsdonk, Combinatory reduction systems: introduction and survey, Theoretical Computer Science 121 (1 & 2) (1993) 279–308.

[21] T. Nipkow, C. Prehofer, Higher-order rewriting and equational reasoning, in: W. Bibel, P. Schmitt (Eds.), Automated Deduction – A Basis for Applications, Vol. 1, Kluwer, 1998, pp. 399–430.

[22] F. van Raamsdonk, Higher-order rewriting, in: Terese [16], Chapter 11, pp. 588–667.

[23] J. Glauert, D. Kesner, Z. Khasidashvili, Expression reduction systems and extensions: An overview, in: Processes, Terms and Cycles: Steps on the Road to Infinity, Vol. 3838 of Lecture Notes in Computer Science, Springer-Verlag, 2005, pp. 496–553.

[24] T. Yamada, Confluence and termination of simply typed term rewriting systems, in: Proceedings of the 12th International Conference on Rewriting Techniques and Applications (RTA 2001), Vol. 2051 of Lecture Notes in Computer Science, Springer-Verlag, 2001, pp. 338–352.

[25] V. van Oostrom, Confluence for abstract and higher-order rewriting, Ph.D. thesis, Vrije Universiteit Amsterdam (1994).

[26] R. Mayr, T. Nipkow, Higher-order rewrite systems and their confluence, Theoretical Computer Science 192 (1998) 3–29.

[27] C. Bertolissi, P. Baldan, H. Cirstea, C. Kirchner, A rewriting calculus for higher-order term graphs, Electronic Notes in Theoretical Computer Science 127 (2005) 21–41.

[28] A. Berarducci, Infinite λ-calculus and non-sensible models, in: Logic and Algebra, Vol. 180 of Lecture Notes in Pure and Applied Mathematics, Marcel Dekker, 1996, pp. 339–378.

[29] M. Hanus, C. Prehofer, Higher-order narrowing with definitional trees, in: Proceedings of the 7th International Conference on Rewriting Techniques and Applications (RTA '96), Vol. 1103 of Lecture Notes in Computer Science, Springer-Verlag, 1996, pp. 138–152.

[30] V. van Oostrom, Higher-order families, in: Proceedings of the 7th International Conference on Rewriting Techniques and Applications (RTA '96), Vol. 1103 of Lecture Notes in Computer Science, Springer-Verlag, 1996, pp. 392–407.

[31] A. Arnold, M. Nivat, The metric space of infinite trees. Algebraic and topological properties, Fundamenta Informaticae 3 (4) (1980) 445–476.

[32] N. Dershowitz, S. Kaplan, Rewrite, rewrite, rewrite, rewrite, rewrite, in: Proceedings of the 16th Annual ACM Symposium on Principles of Programming Languages (POPL '89), 1989, pp. 250–259.

[33] N. Dershowitz, S. Kaplan, D. A. Plaisted, Infinite normal forms, in: Proceedings of the 16th International Colloquium on Automata, Languages and Programming (ICALP '89), Vol. 372 of Lecture Notes in Computer Science, Springer-Verlag, 1989, pp. 249–262.

[34] W. M. Farmer, R. J. Watro, Redex capturing in term graph rewriting, in: Proceedings of the 4th Conference on Rewriting Techniques and Applications (RTA '91), Vol. 488 of Lecture Notes in Computer Science, 1991, pp. 13–24.

[35] R. Kennaway, V. van Oostrom, F.-J. de Vries, Meaningless terms in rewriting, The Journal of Functional and Logic Programming 1.

[36] M. Dezani-Ciancaglini, P. Severi, F.-J. de Vries, Infinitary lambda calculus and discrimination of Berarducci trees, Theoretical Computer Science 298 (2) (2003) 275–302.

[37] S. Blom, An approximation based approach to infinitary lambda calculi, in: Proceedings of the 15th International Conference on Rewriting Techniques and Applications (RTA 2004), Vol. 3091 of Lecture Notes in Computer Science, Springer-Verlag, 2004, pp. 221–232.

[38] S. Kahrs, Infinitary rewriting: Meta-theory and convergence, Acta Informatica 44 (2) (2007) 91–121.

[39] J. G. Simonsen, On modularity in infinitary rewriting, Information and Computation 204 (6) (2006) 957–988.

[40] J. W. Klop, R. de Vrijer, Infinitary normalization, in: S. N. Artëmov, H. Barringer, A. S. d'Avila Garcez, L. C. Lamb, J. Woods (Eds.), We Will Show Them: Essays in Honour of Dov Gabbay, Vol. 2, College Publications, 2005, pp. 169–192.

[41] B. Lisper, Infinite unfolding and transformations of nondeterministic programs, Fundamenta Informaticae 66 (4) (2005) 415–439.

[42] J. L. Kelley, General Topology, Vol. 27 of Graduate Texts in Mathematics, Springer-Verlag, 1975.

[43] S. Kahrs, Compilation of combinatory reduction systems, in: Proceedings of the 1st International Workshop on Higher-Order Algebra, Logic, and Term Rewriting (HOA '93), Vol. 816 of Lecture Notes in Computer Science, Springer-Verlag, 1993, pp. 169–188.

[44] J. A. Goguen, J. W. Thatcher, E. G. Wagner, J. B. Wright, Initial algebra semantics and continuous algebras, Journal of the ACM 24 (1) (1977) 68–95.

[45] B. Courcelle, Fundamental properties of infinite trees, Theoretical Computer Science 25 (2) (1983) 95–169.

[46] G. Berry, J.-J. Lévy, Minimal and optimal computations of recursive programs, Journal of the ACM 26 (1979) 148–175.

[47] J. Ketema, Böhm-like trees for rewriting, Ph.D. thesis, Vrije Universiteit, Amsterdam (2006).

[48] M. Fiore, G. Plotkin, D. Turi, Abstract syntax and variable binding (extended abstract), in: Proceedings of the 14th Symposium on Logic in Computer Science (LICS '99), Computer Society Press of the IEEE, 1999, pp. 193–202.

[49] P. Severi, F.-J. de Vries, An extensional Böhm model, in: Proceedings of the 13th International Conference on Rewriting Techniques and Applications (RTA 2002), Vol. 2378 of Lecture Notes in Computer Science, Springer-Verlag, 2002, pp. 159–173.

[50] A. Asperti, V. Danos, C. Laneve, L. Regnier, Paths in the λ-calculus, in: Proceedings of the Ninth International IEEE Symposium on Logic in Computer Science (LICS '94), IEEE Computer Society Press, 1994, pp. 426–436.

[51] A. Asperti, C. Laneve, Paths, computations and labels in the λ-calculus, Theoretical Computer Science 142 (2) (1995) 277–297.

[52] M. Makkai, Admissible sets and infinitary logic, in: J. Barwise (Ed.), Handbook of Mathematical Logic, North-Holland, 1977, pp. 233–282.

Appendix A. Full proofs omitted from the main text

Appendix A.1. Proof of Theorem 5.2

Proof of Theorem 5.2. Let $s \twoheadrightarrow^{\alpha} t$ and proceed by ordinal induction on α. By the proof of [10, Theorem 12.7.1] it suffices to show that the theorem holds for $\alpha = \omega + 1$: The cases where α is 0, a limit ordinal, or a successor ordinal greater than $\omega + 1$ do not depend on the definition of rewriting.

Suppose $\alpha = \omega + 1$ and write

$$s = s_0 \to s_1 \to \cdots s_\omega \to s_{\omega+1} = t\,.$$

The redex contracted in $s_\omega \to s_{\omega+1}$, call it u, occurs at a position p at depth d_u in s_ω. By definition of rewrite rules, the rule employed in u, say $l \to r$, has a finite left-hand side. Hence, there exists a $d_l > d_u$ such that all positions in the redex pattern of u have depth strictly less than d_l.

By Lemma 4.18, we may write $s \twoheadrightarrow t$ as:

$$s_0 \to^* s_n \twoheadrightarrow s_\omega \to s_{\omega+1}$$

where all rewrite steps in $s_n \twoheadrightarrow s_\omega$ occur at depth d_l or below. Moreover, by left-linearity and fully-extendedness it follows that a redex v occurs at position p in s_n with u the unique residual of v. Contracting v in s_n yields a term t'.

Observe for some $m \in \mathbb{N}$ that there exists a context $C[\Box, \ldots, \Box]$ with $m + 1$ holes, which all occur at depth d_u, such that we may write $s_n \twoheadrightarrow s_\omega \to s_{\omega+1}$ as:

$$C[\bar{\sigma}(l), s'_1, \ldots, s'_m] \twoheadrightarrow C[\bar{\sigma}'(l), s''_1, \ldots, s''_m] \to C[\bar{\sigma}'(r), s''_1, \ldots, s''_m]\,.$$

Existence follows as each rewrite step in $s_n \twoheadrightarrow s_\omega$ occurs at depth $d_l > d_u$ or below and as all positions in the redex pattern of redex v occur at or at depth d_u or below.

By definition of $C[\Box, \ldots, \Box]$ we have that $t' = C[\bar{\sigma}(r), s'_1, \ldots, s'_m]$, where t' is the result of contracting v in s_n. Moreover, the reduction $s_n \twoheadrightarrow s_{\omega+1}$ interleaves the reductions $\bar{\sigma}(l) \twoheadrightarrow^{\leq \omega} \bar{\sigma}'(l) \to \bar{\sigma}'(r)$ and $s'_i \twoheadrightarrow^{\leq \omega} s''_i$, with $1 \leq i \leq m$, where for the first of these reductions, there exists a reduction $\bar{\sigma}(l) \to \bar{\sigma}(r) \twoheadrightarrow^{\leq \omega} \bar{\sigma}'(r)$ by Lemma 5.1.

By Lemma 4.18, we may write $\bar{\sigma}(r) \twoheadrightarrow^{\leq \omega} \bar{\sigma}'(r)$ as:

$$\bar{\sigma}(r) = \bar{\sigma}_0(r) \to^* \bar{\sigma}_1(r) \to^* \cdots \to^* \bar{\sigma}_d(r) \to^* \bar{\sigma}_{d+1}(r) \to^* \cdots \bar{\sigma}'(r)$$

and each $s'_i \twoheadrightarrow^{\leq \omega} s''_i$ as:

$$s'_i = s_{i,0} \to^* s_{i,1} \to^* \cdots \to^* s_{i,d} \to^* s_{i,d+1} \to^* \cdots s''_i\,,$$

where for each $d \geq 0$ we have that steps in $\bar{\sigma}_d(r) \twoheadrightarrow \bar{\sigma}'(r)$ and $s_{i,d} \twoheadrightarrow s''_i$ occur at depth d or below. Hence, $\bar{\sigma}_d(r) \twoheadrightarrow \bar{\sigma}'(r)$ and $s_{i,d} \twoheadrightarrow s''_i$ may be empty from some d onwards.

We now show for all $d \geq 0$ that there exists a reduction $t_d \to^* t_{d+1}$ with all rewrite steps occurring at depth $d_u + d$ or below and such that $d(t_d, t) \leq 2^{-(d_u+d)}$. To do so, define the following for each $d \geq 0$:

$$t_d = C[\bar{\sigma}_d(r), s_{1,d}, \ldots, s_{m,d}]$$

and consider $\bar{\sigma}_d(r) \to^* \bar{\sigma}_{d+1}(r)$ and $s_{i,d} \to^* s_{i,d+1}$ for all $1 \leq i \leq m$. Obviously, we have:

$$t_d = C[\bar{\sigma}_d(r), s_{1,d}, \ldots, s_{m,d}] \to^* C[\bar{\sigma}_{d+1}(r), s_{1,d+1}, \ldots, s_{m,d+1}] = t_{d+1}\,.$$

Since all steps in $\bar{\sigma}_d(r) \to^* \bar{\sigma}_{d+1}(r)$ and $s_{i,d} \to^* s_{i,d+1}$ occur at depth d or below and since the holes in the context $C[\Box, \ldots, \Box]$ occur at depth d_u, all rewrite steps in $t_d \to t_{d+1}$ occur at depth $d_u + d$ or below. Moreover, since all rewrite steps in $\bar{\sigma}_d(r) \twoheadrightarrow \bar{\sigma}'(r)$ and $s_{i,d} \twoheadrightarrow s''_i$ also occur at depth d or below, we also have that $d(t_d, t) \leq 2^{-(d_u+d)}$.

By construction of the reductions $t_d \to^* t_{d+1}$ it follows that

$$t_0 \to^* t_1 \to^* \cdots \to^* t_d \to^* t_{d+1} \to^* \cdots t$$

is a strongly convergent reduction of length at most ω. Since $t' = t_0$, we have that $t' \twoheadrightarrow^{\leq\omega} t$. Hence, as $s \to^* t'$, it follows that $s \twoheadrightarrow^{\leq\omega} t$, as required. $\Box$

Appendix A.2. Proof of Proposition 6.6 and Lemma 6.7

As in Section 6, we assume in this appendix that we are working in an orthogonal iCRS and that $\mathcal{U}$ is a set of redexes in a term s. We first prove Proposition 6.6.

Proof of Proposition 6.6. As each path projection derives from a path, we have by definition that ϕ is *surjective.* Similar for the path projections in $\mathcal{P}(s,\mathcal{U})$ and the maximal paths, as each path projection in $\mathcal{P}(s,\mathcal{U})$ derives from a maximal path.

To prove that ϕ is *injective*, suppose there exist (maximal) paths Π and Π' such that $\phi(\Pi) = \phi(\Pi')$. By definition of ϕ both paths and the path projection consist of the same number of nodes and edges. Let Π^* be the longest shared prefix of Π and Π'. The prefix Π^* is non-empty, as each path of s starts with (s, ϵ). There are now two cases to consider depending on Π^* ending in either an edge or a node.

In case Π^* ends in an edge, the next node is uniquely determined by the definition of paths. Hence, as Π and Π' have the same number of nodes and edges we can extend Π^* with that unique node, contradiction.

In case Π^* ends in a node, both paths extend Π^*, otherwise $\Pi = \Pi'$ or the paths differ in the number of nodes or edges. In case the extension is with an unlabelled edge in case of one of the paths, the other path must also extend Π^* with an unlabelled edge. This follows by the definition of paths. In case the extension is with an edge labelled i, the other path must also extend Π^* with an edge labelled i. This follows by definition of paths and by $\phi(\Pi) = \phi(\Pi')$.

Hence, in case Π^* ends in a node a contradiction also follows. We can conclude that ϕ is an injection both between paths and path projections and between maximal paths and the path projections in $\mathcal{P}(s, \mathcal{U})$. $\square$

To prove Lemma 6.7, we define a map θ_u taking maximal paths Π of s with respect to $\mathcal{U}$ to maximal paths of t with respect to $\mathcal{U}/u$, where $u \in \mathcal{U}$ and $s \to t$ by contracting u. The definition of $\theta_u(\Pi)$ employs a partial map ψ_Π that has three arguments: a node of Π, a finite string over $\mathbb{N}$, and a partial map from $\mathcal{U} - \{u\}$ to finite strings over $\mathbb{N}$.

We first define ψ_Π. In the definition, given a partial map ρ, we denote by $\rho[x \mapsto y]$ the partial map ρ' defined as:

$$\rho'(z) = \begin{cases} y & \text{if } z = x \\ \rho(z) & \text{otherwise} \end{cases}$$

Definition Appendix A.1. Let Π be a maximal path of s with respect to $\mathcal{U}$, let $u \in \mathcal{U}$, and let $s \to t$ by contracting u. Define $\psi_\Pi(n, q_t, \rho)$ as:

1. If n is labelled (s, p) with the subterm at p neither a redex in $\mathcal{U}$ nor a variable bound by a redex in $\mathcal{U}$, then
   (a) if n has no outgoing edge, define $\psi_\Pi(n, q_t, \rho) = (t, q_t)$,
   (b) if n has an edge labelled i to n', define $\psi_\Pi(n, q_t, \rho) = (t, q_t) \xrightarrow{i} \psi_\Pi(n', q_t \cdot i, \rho)$.
2. If n is labelled (s, p_v) with $v \in \mathcal{U} - \{u\}$ and if n has an unlabelled edge to n', define $\psi_\Pi(n, q_t, \rho) = (t, q_t) \to \psi_\Pi(n', q_t, \rho[v \mapsto q_t])$,
3. If n is labelled (s, p_u) and if n has an unlabelled edge to n', define $\psi_\Pi(n, q_t, \rho) = \psi_\Pi(n', q_t, \rho)$.
4. If n is labelled (s, p) with $s|_p$ a variable bound by $v \in \mathcal{U} - \{u\}$ and if n has an unlabelled edge to n', define $\psi_\Pi(n, q_t, \rho) = (t, q_t) \to \psi_\Pi(n', \rho(v), \rho)$.
5. If n is labelled (s, p) with $s|_p$ a variable bound by u and if n has an unlabelled edge to n', define $\psi_\Pi(n, q_t, \rho) = \psi_\Pi(n', q_t, \rho)$.
6. If n is labelled (r, p, p_v) with $r|_p$ not a meta-variable and $v \in \mathcal{U} - \{u\}$, then
   (a) if n has no outgoing edge, define $\psi_\Pi(n, q_t, \rho) = (r, p, q_t)$,
   (b) if n has an edge labelled i to n', define $\psi_\Pi(n, q_t, \rho) = (r, p, q_t) \xrightarrow{i} \psi_\Pi(n', q_t, \rho)$.
7. If n is labelled (r, p, p_u) with $r|_p$ not a meta-variable, then
   (a) if n has no outgoing edge, define $\psi_\Pi(n, q_t, \rho) = (t, q_t)$,
   (b) if n has an edge labelled i to n', define $\psi_\Pi(n, q_t, \rho) = (t, q_t) \xrightarrow{i} \psi_\Pi(n', q_t \cdot i, \rho)$.
8. If n is labelled (r, p, p_v) with $r|_p$ a meta-variable and $v \in \mathcal{U} - \{u\}$ and if n has an unlabelled edge to n', which is labelled $(s, p_v \cdot q)$, define $\psi_\Pi(n, q_t, \rho) = (r, p, q_t) \to \psi_\Pi(n', q_t \cdot q, \rho)$.
9. If n is labelled (r, p, p_u) with $r|_p$ a meta-variable and if n has an unlabelled edge to n', which is labelled $(s, p_u \cdot q)$, define $\psi_\Pi(n, q_t, \rho) = \psi_\Pi(n', q_t, \rho)$.

Let $\perp$ be the completely undefined map. We define the following.

Definition Appendix A.2. Let $u \in \mathcal{U}$ and let Π be a maximal path of s with respect to $\mathcal{U}$. The map θ_u is defined as:

$$\theta_u(\Pi) = \psi_\Pi((s, \epsilon), \epsilon, \perp) .$$

Note that $\theta_u(\Pi)$ is calculated by iterating ψ_Π. After a finite number of iterations, a finite prefix of $\theta_u(\Pi)$ is obtained.

In the next proposition we show that θ_u is well-defined: $\theta_u(\Pi)$ is a maximal path of t with respect to $\mathcal{U}/u$.

Remark Appendix A.3. In the proposition we make use of the fact that the definition of a rewrite step $\bar{\sigma}(l) \to \bar{\sigma}(r)$ follows the definition of valuations and substitutions and, hence, is defined in two steps. The first step defines descendants in $\bar{\sigma}(r)$ where only the valuation is applied and not Equation (1), we denote the result of this step by r_σ. The second step defines descendants across application of Equation (1). Reading the equation from left to right as a *parallel β-rule*, the second step is essentially a complete development of parallel β-redexes.

Proposition Appendix A.4. *Let Π be a maximal path of s with respect to $\mathcal{U}$ and let $u \in \mathcal{U}$. For each finite number of iterations of ψ_Π in the calculation of $\theta_u(\Pi)$ the following holds:*

- *Either no nodes and edges have been generated, or the generated nodes and edges, with exception of the edge generated last, form a path, and the edge generated last is valid when extending the path to a longer one.*
- *For all defined $\rho(v)$ in the third argument of ψ_Π it holds that $\rho(v) \in \mathcal{P}os(t)$ and that a residual of v occurs at $\rho(v)$ in t.*

Distinguishing by the particular clause of Definition Appendix A.1 employed in the last iteration, the following also holds:

(1) (a) nothing; (b) q_t is descendant of p and the next node generated is $(t, q_t \cdot i)$, which together with the previously generated nodes and edges forms a path;

(2) a residual of v occurs at q_t in t and the next node generated is (r, ϵ, q_t), which together with the previously generated nodes and edges forms a path;

(3) $q_t = p_u$ and the next node generated is (t, q_t), which together with the previously generated nodes and edges forms a path;

(4) if n' is labelled $(r, p' \cdot i, p_v)$, then the next node generated is $(r, p' \cdot i, \rho(v))$, which together with the previously generated nodes and edges forms a path;

(5) there are two subcases:

- *if in the previous iteration clause (3) is employed followed by a number of iterations employing in turn clauses (5) and (9), then $q_t = p_u$;*

- *if in the previous iteration clause (1) is employed or if clause (7) is employed followed by a number of iterations employing in turn clauses (5) and (9), then $q_t = q'_t \cdot i$ where q'_t is the q_t from either clause (1) or (7);*

in both cases the next node generated is (t, q_t), which together with the previously generated nodes and edges forms a path.

(6) *(a) nothing; (b) a residual of v occurs at q_t and the next node generated is $(r, p \cdot i, q_t)$, which together with the previously generated nodes and edges forms a path;*

(7) *(a) nothing; (b) $q_t = p_u \cdot q$ where q is a descendant of p across a complete development of the parallel β-redexes in r_σ and the next node generated is $(t, q_t \cdot i)$, which together with the previously generated nodes and edges forms a path;*

(8) *q_t is a descendant of p_v and the next node generated is $(t, q_t \cdot q)$, which together with the previously generated nodes and edges forms a path;*

(9) *there are two subcases:*

- *if the next iteration does* not *employ clause (5), then q_t is a descendant of $p_u \cdot q$;*
- *otherwise, q_t is as in clause (5);*

in both cases the next node generated is (t, q_t), which together with the previously generated nodes and edges forms a path.

In addition, if Π is a maximal path of s with respect to $\mathcal{U}$, then $\theta_u(t)$ is a maximal path of t with respect to $\mathcal{U}/u$.

Proof. Let Π be a maximal path of s with respect to $\mathcal{U}$, let $u \in \mathcal{U}$, and let $s \to t$ by contracting u. We prove the lemma by induction on the number of iterations of ψ_Π.

Below, when we say that something is a path of t, we implicitly assume that it is a path of t with respect to $\mathcal{U}/u$.

Base case. By definition of ψ_Π only clauses (1), (2), and (3) can be employed in the first iteration. The other clauses either require a bound variable at the root of s, which is impossible, or they require the label of the node in the first argument to be a triple, which is not the case. We deal with each of the possible clauses in turn:

(1) In this case a node labelled (t, ϵ) is generated and possibly an edge labelled i. Obviously, (t, ϵ) is a path. The partial map $\perp$ is unaffected by this clause, thus satisfying the necessary requirements.

    As the current iteration implies that no redex from $\mathcal{U}$ occurs at the root of s, we have that $q_t = \epsilon \in \mathcal{P}os(t)$ is a descendant of $p = \epsilon$ and $root(t|_\epsilon) = root(s|_\epsilon)$. Hence, in case of clause (a), the path is maximal like Π. In case of clause (b), the edge labelled i is allowed and the next node generated node must be (t, i). This node forms a path together with (t, ϵ) and the edge labelled i. By construction of paths, the finite prefix of Π considered thus far is not maximal and there is nothing more to show.

(2) In this case a node labelled (t, ϵ) and an unlabelled edge are generated. Obviously, (t, ϵ) is a path. As orthogonality is assumed, a redex v', which is a residual of v, occurs at $\epsilon \in \mathcal{P}os(t)$. Hence, $\perp[v \mapsto \epsilon]$ satisfies the necessary requirements.

As $v \in \mathcal{U} - \{u\}$, it holds that $v' \in \mathcal{U}/u$. Hence, the unlabelled edge is allowed, and the next node generated must be (r, ϵ, ϵ), where r is the right-hand side of the rewrite rule employed in v'. This node forms a path together with (t, ϵ) and the unlabelled edge. By construction of paths, the finite prefix of Π considered thus far is not maximal and there is nothing more to show.

(3) Obviously, $\perp$ is unaffected by this clause, thus satisfying the necessary requirements. Moreover, $q_t = \epsilon \in \mathcal{P}os(t)$ is equal to $p = \epsilon$, and by definition of ψ_Π the next generated node must be (t, ϵ), which is a path. By construction of paths, the finite prefix of Π considered thus far is not maximal and there is nothing more to show.

Induction step. Assume we have proven the lemma up to some arbitrary number of iterations. We next prove that it also holds in case of one more iteration. We deal with each of the possible clauses in turn:

(1) In this case the only possible clauses employed in the previous iteration are (1), (8), and (9). All other clauses force the label of the node in the first argument of ψ_Π to be a triple, which is not the case.

A node labelled (t, q_t) is generated and possibly an edge labelled i. By the clauses possible in the previous iteration, (t, q_t) forms a path together with the previously generated nodes and edges. The partial map ρ is unaffected by this clause, thus satisfying the necessary requirements.

Also by the clauses possible in the previous iteration, $q_t \in \mathcal{P}os(t)$ is a descendant of p and $root(t|_{q_t}) = root(s|_p)$. Hence, in case of clause (a), the path is maximal like Π. In case of clause (b), the edge labelled i is allowed and the next node generated must be $(t, q_t \cdot i)$. This node forms a path together with previously generated nodes and edges. By construction of paths, the finite prefix of Π considered thus far is not maximal and there is nothing more to show.

(2) As before, the only possible clauses employed in the previous iteration are (1), (8), and (9). All other clauses force the label of the node in the first argument of ψ_Π to be a triple, which is not the case.

A node labelled (t, q_t) and an unlabelled edge are generated. By the clauses possible in the previous iteration, (t, q_t) forms a path together with the previously generated nodes and edges. Moreover, as orthogonality is assumed, a redex v', which is a residual of v, occurs at $q_t \in \mathcal{P}os(t)$. Hence, as ρ satisfies the necessary requirements, $\rho[v \mapsto q_t]$ does so too.

As $v \in \mathcal{U} - \{u\}$, it holds that $v' \in \mathcal{U}/u$. Hence, the unlabelled edge is allowed, and the next node generated is (r, ϵ, q_t), where r is the right-hand side of the rewrite rule employed in v'. This node forms a path together with previously generated nodes and edges. By the construction of paths, the finite prefix of Π considered thus far is not maximal and there is nothing more to show.

(3) In this case the only possible clauses employed in the previous iteration are (1) and (8). All other clauses, except (9), force the label of the node in the first argument of ψ_Π to be a triple, which is not the case. Clause (9) is impossible as it requires the redex u to occur above itself in s.

Obviously, ρ is unaffected by this clause, thus satisfying the necessary requirements. By the clauses possible in the previous iteration and the definition of descendants, $q_t = p_u$. Also by the clauses possible in the previous iteration, the next node generated is (t, q_t) and the node forms a path together with previously generated nodes and edges. By construction of paths, the finite prefix of Π considered thus far is not maximal and there is nothing more to show.

(4) As before, the only possible clauses employed in the previous iteration are (1), (8), and (9). All other clauses force the label of the node in the first argument of ψ_Π to be a triple, which is not the case.

A node labelled (t, q_t) and an unlabelled edge are generated. By the clauses possible in the previous iteration, (t, q_t) forms a path together with the previously generated nodes and edges. The partial map ρ is unaffected by this clause, thus satisfying the necessary requirements.

Also by the clauses possible in the previous iteration, q_t is a descendant of p and $root(t|_{q_t})$ is a variable bound by a residual v' of v in t, where by construction $\rho(v)$ is the position of v'. Hence, the unlabelled edge is allowed. That a node labelled $(r, p', \rho(v))$, where r is the right-hand side of the rewrite rule employed in v', has been generated as the last node labelled with $\rho(v)$ follows by definition of ψ_Π. Hence, the next node generated is $(r, p \cdot i, \rho(v))$. This node forms a path together with previously generated nodes and edges. By construction of paths, the finite prefix of Π considered thus far is not maximal and there is nothing more to show.

(5) As before, the only possible clauses employed in the previous iteration are (1), (8), and (9). All other clauses force the label of the node in the first argument of ψ_Π to be a triple, which is not the case.

Obviously, ρ is unaffected by this clause, thus satisfying the necessary requirements. By the clauses in the previous iteration, the requirements of the two subcases are satisfied and the next node generated is (t, q_t).

By construction of paths, the finite prefix of Π considered thus far is not maximal and there is nothing more to show.

(6) In this case the only possible clauses employed in the previous iteration are (2), (4), and (6). Clauses (1), (8), and (9) force the label of the node in the first argument of ψ_Π to be a tuple, which is not the case. Clauses (3) and (5) force v to be equal to u, which is not allowed.

A node labelled (r, p, q_t) is generated and possibly an edge labelled i. By the clauses possible in the previous iteration, (r, p, q_t) forms a path together with the previously generated nodes and edges. The partial map ρ is unaffected by this clause, thus satisfying the necessary requirements.

As r is left unchanged, it holds in case of clause (a), that the path is maximal like Π. In case of clause (b), the edge labelled i is allowed and the next node generated is $(r, p \cdot i, q_t)$. This node forms a path together with previously generated nodes and edges. By construction of paths, the finite prefix of Π considered thus far is not maximal and there is nothing more to show.

(7) In this case the only possible clauses employed in the previous iteration are (3), (5), and (7). Clauses (1), (8), and (9) force the label of the node in the first argument of ψ_Π to be a tuple, which is not the case. Clauses (2) and (4) force v to be unequal to u.

A node labelled (t, q_t) is generated and possibly an edge labelled i. By the clauses possible in the previous iteration (t, q_t) forms a path together with the previously generated nodes and edges. The partial map ρ is unaffected by this clause, thus satisfying the necessary requirements.

Also by the clauses possible in the previous iteration, $q_t = p_u \cdot q$ and $\mathit{root}(t|_{q_t}) = \mathit{root}(r_\sigma|_p)$, where q is a descendant of p across complete development of the parallel β-redexes in r_σ. Hence, in case of clause (a), the path is maximal like Π. In case of clause (b), the edge labelled i is allowed. Moreover, the next node generated is $(t, q_t \cdot i)$. This node forms a path together with previously generated nodes and edges. By construction of paths, the finite prefix of Π considered thus far is not maximal and there is nothing more to show.

(8) In this case the only possible clauses employed in the previous iteration are (2), (4), and (6). Clauses (1), (8), and (9) force the label of the node in the first argument of ψ_Π to be a tuple, which is not the case. Clauses (3) and (5) force v to be equal to u, which is not allowed.

In this case a node labelled (r, p, q_t) and an unlabelled edge are generated. By the clauses possible in the previous iteration, (r, p, q_t) forms a path together with the previously generated nodes and edges. The partial map ρ is unaffected by this clause, thus satisfying the necessary requirements.

As r is left unchanged, the unlabelled edge is allowed. Moreover, as a residual of v occurs at q_t, the next node generated is $(t, q_t \cdot q)$. This node forms a path together with previously generated nodes and edges. By construction of paths, the finite prefix of Π considered thus far is not maximal and there is nothing more to show.

(9) In this case the only possible clauses employed in the previous iteration are (3), (5), and (7). Clauses (1), (8), and (9) force the label of the node in the first argument of ψ_Π to be a tuple, which is not the case. Clauses (2) and (4) force v to be unequal to u.

Obviously, ρ is unaffected by this clause, thus satisfying the necessary requirements. By the clauses in the previous iteration, the requirement that q_t is a descendant of $p_u \cdot q$ is satisfied in the first subcase, and the requirements of clause (5) are satisfied in the second subcase. Moreover, the next node generated is (t, q_t). By construction of paths, the finite prefix of Π considered thus far is not maximal and there is nothing more to show.

There can be no infinite cycle of iterations employing clauses (5) and (9) in the construction of $\theta_u(\Pi)$, because that implies the existence of an infinite chain of meta-variables in r. Hence, $\theta_u(\Pi)$ is a well-defined path of t with respect to $\mathcal{U}/u$. In case Π is finite, the induction shows that $\theta_u(\Pi)$ is maximal. In case Π is infinite, so is $\theta_u(\Pi)$, and we conclude that $\theta_u(\Pi)$ is a maximal path. $\square$

The next lemma relates the maximal paths of s with respect to $\mathcal{U}$ to the maximal paths of t with respect to $\mathcal{U}/u$. In the proof of the lemma we leave out the labels of the explicitly denoted edges.

Proposition Appendix A.5. *The map θ_u is a bijection.*

Proof. Let $u \in \mathcal{U}$ and $s \to t$ by contracting u. By Proposition Appendix A.4, we have that θ_u maps maximal paths of s with respect to $\mathcal{U}$ to maximal paths of t with respect to $\mathcal{U}/u$.

To prove that θ_u is *surjective*, let Π_t be a maximal path of t with respect to $\mathcal{U}/u$. We are done if $\Pi_t = \theta_u(\Pi_s)$ for some maximal path Π_s of s with respect to $\mathcal{U}$. Otherwise, Π_t has a finite non-empty prefix Π'_t in common with $\theta_u(\Pi_s)$ for some maximal path Π_s of s with respect to $\mathcal{U}$. The prefix is non-empty as any path of t begins with (t, ϵ). Let Π'_t be the longest finite prefix of Π_t such that $\theta_u(\Pi_s) = \Pi'_t \to \cdots$ for some maximal path Π_s. We have $\Pi_s = \Pi'_s \to \cdots$ for some finite path Π'_s. By definition of θ_u we can extend the prefix Π'_t with a new node precisely when we can extend Π'_s. Hence, Π'_t cannot be the *longest* finite prefix with $\theta_u(\Pi_s) = \Pi'_t \to \cdots$ for some maximal path Π_s in s, since we can extend Π'_s to form a new maximal path with more nodes, contradiction. Hence, $\Pi_t = \theta_u(\Pi_s)$ for some maximal path Π_s.

To prove that θ_u is *injective*, suppose there exist two maximal paths Π and Π' of s with respect to $\mathcal{U}$ such that $\theta_u(\Pi) = \theta_u(\Pi')$. Let Π^* be the longest prefix shared between Π and Π'. The prefix Π^* is non-empty, as any path of s begins with (s, ϵ). There are now two cases to consider depending on Π^* ending in either an edge or a node.

In case Π^* ends in an edge, the next node is uniquely determined by the definition of paths. Hence, as Π and Π' are maximal, we can extend Π^* with that unique node, contradiction.

In case Π^* ends in a node, at least one of Π and Π' extends Π^*, otherwise $\Pi = \Pi'$. In case the extension is with an unlabelled edge, the other path must also extend Π^* with an unlabelled edge. This follows by the definition of paths and by Π and Π' being maximal. Otherwise, in case the extension is with an edge labelled i, the other path must also extend Π^* with an edge labelled i. This follows by definition of paths and as $\theta_u(\Pi) = \theta_u(\Pi')$. Hence, in case Π^* ends in a node a contradiction also follows and we can conclude that θ_u is injective. □

We finally prove Lemma 6.7.

Proof of Lemma 6.7. By Proposition Appendix A.5, the map θ_u is a bijection between the maximal paths of s with respect to $\mathcal{U}$ and the maximal paths of t with respect to $\mathcal{U}/u$. By Proposition 6.6, a bijection exists between the set of paths and the set of path projections mapping unlabelled edges to ϵ-labelled edges and labelled edges to edges with the same label. Hence, θ_u induces a bijection θ'_u between $\mathcal{P}(s,\mathcal{U})$ and $\mathcal{P}(t,\mathcal{U}/u)$. By examining the construction of θ_u, we see that it only deletes unlabelled edges and nodes corresponding to meta-variables of u and variables bound by u. Moreover, it is evident that if an infinite sequence of nodes and unlabelled edges were deleted, the right-hand side of the rule of u would contain an infinite chain of meta-variables, contradicting the definition of meta-terms. Hence, $\phi(\theta'_u(\Pi))$ can be obtained from $\phi(\Pi)$ by deleting only finite sequences of unlabelled nodes and ϵ-labelled edges, as required. □

Appendix A.3. Proof of Proposition 6.9 and Theorem 6.10

Proof of Proposition 6.9. Let $\mathcal{P}_p(s,\mathcal{U})$ denote the set of all prefixes of path projections in $\mathcal{P}(s,\mathcal{U})$ such that the concatenation of the edge labels for each prefix is p and such that each prefix ends in a labelled node. The proof proceeds by induction on p.

Consider $\mathcal{P}_\epsilon(s,\mathcal{U})$. By the finite jumps property $\mathcal{P}_\epsilon(s,\mathcal{U})$ is non-empty and by the definition of paths, $\mathcal{P}_\epsilon(s,\mathcal{U})$ has at most one element. Hence, $\mathcal{P}_\epsilon(s,\mathcal{U})$ is a singleton set. By definition of paths, the unique prefix in $\mathcal{P}_\epsilon(s,\mathcal{U})$ has precisely one labelled node. Suppose the label is f. It follows that t only matches $\mathcal{P}(s,\mathcal{U})$ if $root(t|_\epsilon) = f$.

Now suppose $\mathcal{P}_p(s,\mathcal{U})$ is a singleton set such that the final node node of the unique prefix in the set is labelled f, where f is either a variable, a function symbol of arity n, or an abstraction. In the last two cases, consider $\mathcal{P}_{p\cdot i}(s,\mathcal{U})$ for either $1 \leq i \leq n$ or $i = 0$. By the finite jumps property, the definition of paths, and the fact that $\mathcal{P}_p(s,\mathcal{U})$ is a singleton set, we have that $\mathcal{P}_{p\cdot i}(s,\mathcal{U})$ is a singleton set. Suppose that final node of the unique prefix in $\mathcal{P}_{p\cdot i}(s,\mathcal{U})$ is labelled g. It follows that t only matches $\mathcal{P}(s,\mathcal{U})$ if $root(t|_{p\cdot i}) = g$.

Since all sets $\mathcal{P}_p(s,\mathcal{U})$ are singleton sets there exist terms that match $\mathcal{P}(s,\mathcal{U})$. Moreover, if t is such a term, then we have for all $p \in \mathcal{P}os(t)$ that $\mathcal{P}_p(s,\mathcal{U})$ exists and is a singleton set (look at all prefixes of p), and if the final labelled node of the unique prefix in such a set has label f, then $root(t|_p) = f$. Hence, the term t is unique. $\square$

Proof of Theorem 6.10. (1) Suppose there exists a complete development. We show by ordinal induction that for every s_α in the complete development with residuals $\mathcal{U}_\alpha = \mathcal{U}/(s \twoheadrightarrow s_\alpha)$ of $\mathcal{U}$, we have that $\mathcal{P}(s_\alpha,\mathcal{U}_\alpha)$ can be obtained from $\mathcal{P}(s,\mathcal{U})$ by deleting finite sequences of unlabelled nodes and ϵ-labelled edges from the elements of $\mathcal{P}(s,\mathcal{U})$. Obviously, for $s_0 = s$, this is immediate.

For $s_{\alpha+1}$, it follows by the induction hypothesis that $\mathcal{P}(s_\alpha,\mathcal{U}_\alpha)$ can be obtained from $\mathcal{P}(s,\mathcal{U})$ by deleting finite sequences of unlabelled nodes and ϵ-labelled edges from the elements of $\mathcal{P}(s,\mathcal{U})$. Moreover, by Lemma 6.7, we have that $\mathcal{P}(s_{\alpha+1},\mathcal{U}_{\alpha+1})$ can be obtained from $\mathcal{P}(s_\alpha,\mathcal{U}_\alpha)$ by deleting finite sequences of unlabelled nodes and ϵ-labelled edges. Hence, $\mathcal{P}(s_{\alpha+1},\mathcal{U}_{\alpha+1})$ can also be obtained from $\mathcal{P}(s,\mathcal{U})$ by deleting finite sequences of unlabelled nodes and ϵ-labelled edges.

For s_α with α a limit ordinal, we have by strong convergence that $\mathcal{P}(s_\alpha,\mathcal{U}_\alpha)$ can be obtained from $\mathcal{P}(s,\mathcal{U})$ by deleting all unlabelled nodes and ϵ-labelled edges deleted in the previous steps. As $\mathcal{P}(s,\mathcal{U})$ has the finite jumps property, $\mathcal{P}(s_\alpha,\mathcal{U}_\alpha)$ can only be obtained by deleting finite sequences of unlabelled nodes and ϵ-labelled edges.

Hence, each $\mathcal{P}(s_\alpha,\mathcal{U}_\alpha)$ has the finite jumps property, as $\mathcal{P}(s,\mathcal{U})$ has the finite jumps property and as each $\mathcal{P}(s_\alpha,\mathcal{U}_\alpha)$ can be obtained by deleting only finite sequences of unlabelled nodes and ϵ-labelled edges.

By Proposition 6.9 we have for each $\mathcal{P}(s_\alpha,\mathcal{U}_\alpha)$ that there is a unique term $\mathcal{T}(s_\alpha,\mathcal{U}_\alpha)$ that matches it. By inspection of the proof of the proposition it easily follows that the unlabelled nodes and ϵ-labelled edges are irrelevant for the construction of $\mathcal{T}(s_\alpha,\mathcal{U}_\alpha)$. Hence, $\mathcal{T}(s_\alpha,\mathcal{U}_\alpha) = \mathcal{T}(s,\mathcal{U})$ for all α. Moreover, since the chosen complete development was arbitrary, it follows that the final term of each complete development is $\mathcal{T}(s,\mathcal{U})$.

(2) In analogy to [19, Section II.2], let $\mathcal{K}$ be a set of labels including a special *empty* label ε. Define for all function symbols f and variables x and for all labels $k \in \mathcal{K}$ the *labelled alternatives* f^k and x^k, where f and f^k have the same arity. A *labelling* of a (meta-)term replaces each function symbol and variable (including the variables that occur in abstractions) by a labelled alternative, assuming that the labels of variables are ignored where *bindings* and *valuations* are concerned.

The labelled version of the assumed orthogonal iCRS includes for every rewrite rule $l \to r$ and every possible labelling l' of l a rewrite rule $l' \to r'$, where r' is the labelling of r that labels all function symbols and variables with ε. The labelled version of the iCRS is easily shown to be orthogonal (see [19, Proposition II.2.6]).

Each reduction in the labelled version corresponds to a reduction in the original iCRS by removal of all labels. Moreover, given a reduction in the original iCRS and a labelling of the initial term, there exists a unique reduction in the labelled version such that removal of the labels yields the reduction we started out with.

Given a term in which some subterms are labelled k, it is easily shown that the descendants of these subterms across some reduction are precisely the subterms labelled k in the final term. Moreover, these descendants are exactly the descendants obtained in the corresponding unlabelled reduction. The result now follows by the first clause of the current proof when applied to the labelled version of the assumed iCRS, when trivially extended to the slightly more liberal notion of valuations.

(3) By the second clause of the current proof and orthogonality of the assumed iCRS.

(4) Consider a maximal path Π with a node (s, p) such that p is the position of a redex in $\mathcal{U}$. By definition of paths and path projections we have for the first node $n = (s, p)$ in Π with p the position of a redex that the concatenation of the edge label of the prefix of $\phi(\Pi)$ that ends in $\phi(n)$ is p. Moreover, by inspection of the proof of Lemma 6.7 it follows that contracting the redex at position p deletes a sequence of unlabelled nodes and ϵ-labelled edges from $\phi(\Pi)$ directly following the node $\phi(n)$.

Repeatedly contract a residual of a redex in $\mathcal{U}$ that is at minimal depth. Since only finite sequences of nodes and edges occur and since terms are finitely branching, it follows by the above observations regarding paths that the contracted redexes occur at increasingly greater depths along the constructed reduction. Hence, the reduction is strongly convergent and since redexes at minimal depth are contracted the reduction must also be a complete development. □