Skew and ω-Skew Confluence and Infinite Normal Forms

Zena M. Ariola[⋆1] and Stefan Blom[2]

[1] University of Oregon
[2] University of Innsbruck

Abstract. The notion of skew confluence was introduced to characterize non-confluent term rewriting systems that had unique infinite normal forms or Böhm like trees. This notion however is not expressive enough to deal with all possible sources of non-confluence in the context of infinite terms or terms extended with letrec. To that end, we present a new notion called ω-skew confluence. We show that ω-skew confluence constitutes a sufficient and necessary condition for uniqueness of infinite normal forms over infinite terms or terms with letrec.

1 Introduction

It is well-known that uniqueness of normal forms follows from confluence and termination and that given termination, confluence and uniqueness of normal forms are equivalent [1, 2]. There are however interesting systems that lack the notion of normal form. For example, according to the rewriting rule given below:

$$\mathsf{F}\ x \to \mathsf{Cons}(x, \mathsf{F}\ x)$$

the term $\mathsf{F}\ 1$ does not have a normal form. It has however a well-defined meaning consisting of the infinite list of ones. This implies that for non-terminating rewrite systems an alternative viable semantics could consist of the notion of *infinite normal form* [3, 4]. This notion is a generalization of the Böhm Tree in the lambda calculus [5].

The notion of infinite normal form is related to the notion of information content, also called instant semantics [6] or direct approximation [7, 8]. Intuitively, the information content of a term is that part of the infinite normal form which is already present in the term. The information content of a term can itself be finite or infinite. In our example, where we build towards an infinite list of ones, we never have an infinite list present in the term, but only finite prefixes. Hence, we speak of finite information content. If we consider the information content of infinite terms then it is obvious that the information content itself can be infinite. Since graphs can be used to represent infinite terms, it could happen that a finite object has infinite information content, like in the graph in Fig. 1.

⋆ Supported by National Science Foundation grant number CCR-0204389

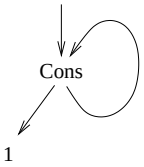


Fig. 1. Finite representation of an infinite list of ones.

An important property of infinite normal forms is *uniqueness*. This property is similar to uniqueness of normal forms and states that every two convertible terms have the same infinite normal form. Confluence implies uniqueness of infinite normal forms. However, confluence is not necessary for guaranteeing uniqueness.

Skew confluence[3], as introduced in [9, 10], characterizes rewrite systems that have unique trees with respect to a notion of direct approximant or notion of finite information content. The idea behind skew confluence is that if there exists a computation that develops a certain set of information, then any other computation can be extended to develop a superset of that information. The theory of skew confluence works well for term rewriting. It can also be applied smoothly to some forms of term graph rewriting. However, the application to other forms of term graph rewriting and to infinitary rewriting is not as smooth as one would hope for. The reason is that the information content can be infinite. To illustrate the problem of infinite terms with finite and infinite information content, we consider term M below:

$$M \equiv (\lambda x.f\,x\,(f\,x\,(f\,x\,(\cdots))))(I\,I)$$

On one side, term M rewrites to a normal form in two steps:

$$(\lambda x.f\,x\,(f\,x\,(f\,x\,(\cdots))))(I\,I) \underset{\beta}{\longrightarrow} (\lambda x.f\,x\,(f\,x\,(f\,x\,(\cdots))))I \underset{\beta}{\longrightarrow} f\,I\,(f\,I\,(f\,I\,(\cdots)))$$

This normal form has infinite information content. Term M also rewrites to a term M_1 which does not even have a finite normalizing sequence:

$$M \equiv (\lambda x.f\,x\,(f\,x\,(f\,x\,(\cdots))))(I\,I) \underset{\beta}{\longrightarrow} M_1 \equiv f\,(I\,I)\,(f\,(I\,I)\,(f\,(I\,I)\,(\cdots)))$$

Moreover, in each reduct of M_1 only finitely many of the $(I\,I)$ redexes have been reduced, which means that each reduct has finite information content. In Section 3, we will give a similar example for term graphs.

We introduce a new variant of skew confluence, called *ω-skew confluence*, which is more suitable to the case of infinite information content. The idea behind ω-skew confluence is that if there exists a computation that develops a certain set of information, then any other computation can be extended to develop a set of information that contains a given element from the first set, but not necessarily all of its elements.

[3] The name was suggested to us by Jan Willem Klop.

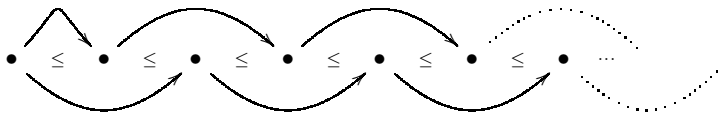


Fig. 2. A skew confluent ARS which is not confluent

Proving (ω-)skew confluence of a non-confluent rewrite system can be rather tedious and non-confluence often complicates matters. In the case of cyclic calculi the source of non-confluence is often a subset of the rewrite rules which deals with substitution and/or unwinding. The idea behind the lifting theory in [11] is to partition the problem into a problem of the substitution rules and a problem for the other rules. The intention is that only the part dealing with substitution is non-confluent. Thus, one deals with non-confluence in a simplified setting and in the more complicated setting, one can use confluence in the proofs. In this paper we present an abstract version of this lifting theory. We start with a well-behaved abstract rewrite system (ARS for short) and an infinite normal form with finite information content. We assume the ARS is equipped with a partial order. Next, we consider an extension of the rewrite system where we use the ideal completion of the original set of objects as the semantics of the objects of the extension. We define a construction which lifts the notion of information content from the original ARS to the extension and provides conditions under which this lifted notion of information content yields unique infinite normal forms.

The paper is organized as follows: We start in Section 2 with an intuitive description of the three distinct notions of confluence. In Section 3, we present a counterexample to confluence which arises from unwinding a graph in different ways. We discuss how confluence modulo bisimilarity provides a solution. We also explain the need of skew and ω-skew confluence to cope with the loss of confluence when the substitution rules are extended with other rewrite rules. In Section 4, we formalize skew and ω-skew confluence and the notion of infinite normal form. In Section 5, we use these notions to show the consistency of the call-by-name and call-by-need cyclic calculi. In Section 6, we present an abstract version of the lifting theory. We conclude in Section 7.

2 Confluence, Skew Confluence and ω-Skew Confluence

We give an intuitive description of these properties through a series of examples. We define the set of objects A as the set consisting of the bottom element $\bot$ and two copies of the set of natural numbers:

$$A = \{\bot\} \cup \mathbb{N} \cup \{\underline{n} \mid n \in \mathbb{N}\} \ .$$

We define the reduction relation $\underset{1}{\rightarrow}$ on A as follows:

$$\bot \underset{1}{\rightarrow} 0, \ \ \bot \underset{1}{\rightarrow} \underline{0}, \ n \underset{1}{\rightarrow} n+1, \ \underline{n} \underset{1}{\rightarrow} \underline{n+1}, \ 2n \underset{1}{\rightarrow} \underline{2n}, \ \underline{2n+1} \underset{1}{\rightarrow} 2n+1 \ .$$

We then have that $(A, \underset{1}{\rightarrow})$ is confluent.

We define next the reduction relation $\underset{2}{\rightarrow}$ on A as follows:

$$\bot \underset{2}{\rightarrow} 0, \quad \bot \underset{2}{\rightarrow} \underline{0}, \; n \underset{2}{\rightarrow} n+1, \; \underline{n} \underset{2}{\rightarrow} \underline{n+1} \; .$$

We have that $(A, \underset{2}{\rightarrow})$ is not confluent. To that end, we define a quasi order $\preceq$ (*i.e.,* $\preceq$ is reflexive and transitive) and an equivalence $\sim$ on A (*i.e.,* $\sim$ is reflexive, symmetric and transitive) as follows:

$$a \preceq a', \text{ if } |a| \leq |a'| \text{ and } a \sim a', \text{ if } a \preceq a' \wedge a' \preceq a \; ,$$

where $|.| : A \rightarrow \mathbb{N}$ is defined as follows:

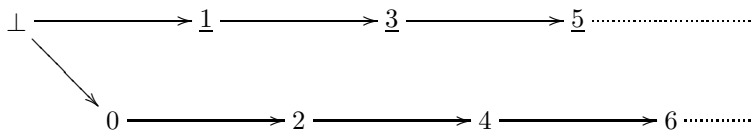
$$| \bot | = 0, |n| = n+1, |\underline{n}| = n+1 \; .$$

We then obtain that $(A, \underset{2}{\rightarrow})$ is confluent modulo $\sim$. According to confluence, divergent reductions can converge to the same term. Whereas, according to confluence modulo, divergent reductions can reach equivalent terms.

We define the reduction relation $\underset{3}{\rightarrow}$ on A as follows:

$$\bot \underset{3}{\rightarrow} 0, \quad \bot \underset{3}{\rightarrow} \underline{1}, \; n \underset{3}{\rightarrow} n+2, \; \underline{n} \underset{3}{\rightarrow} \underline{n+2} \; .$$

We have that $(A, \underset{3}{\rightarrow})$ is neither confluent nor confluent modulo $\sim$. The situation is depicted below (in Fig. 2 we also give a schematic presentation):



To cope with this situation we need skew confluence which states that diverging reductions can always be extended in the same direction.

To explain ω-skew confluence, consider the ARS $(\mathbb{N} \cup \{\infty\}, \rightarrow)$, where

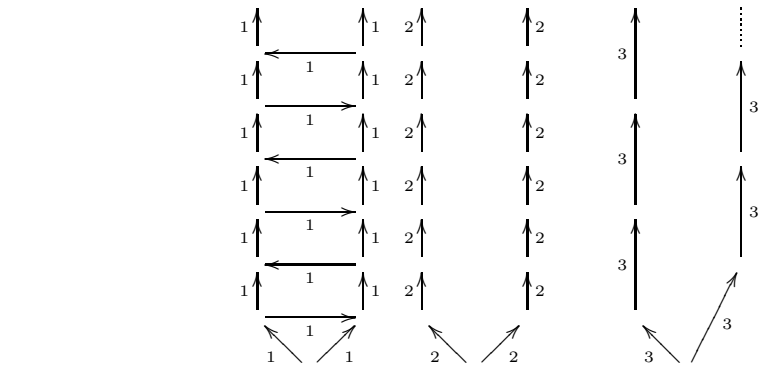
$$i \rightarrow i+1, i \in \mathbb{N} \text{ and } 0 \rightarrow \infty$$

and let $|\alpha| = \{i \in \mathbb{N} \mid i \leq \alpha\}$ We also change the definition of the quasi-ordering as follows:

$$a \preceq a', \text{ if } |a| \subseteq |a'|$$

For all $\alpha \in \mathbb{N} \cup \{\infty\}$ we have that the infinite normal form of α is $\mathbb{N}$. However, the ARS is not skew confluent: $0 \rightarrow\!\!\!\!\rightarrow \infty$ and $0 \rightarrow\!\!\!\!\rightarrow 60$, but there does not exist an α, such that $60 \rightarrow\!\!\!\!\rightarrow \alpha$ and $\infty \preceq \alpha$.

See Fig. 3 and 4 for an illustration of the reduction graphs of $\bot$ for each of the four reduction relations.



Confluent	yes	no	no
Confluent modulo $\sim$	yes	yes	no
Skew confluent	yes	yes	yes

Fig. 3. Some reduction graphs and their confluence properties.

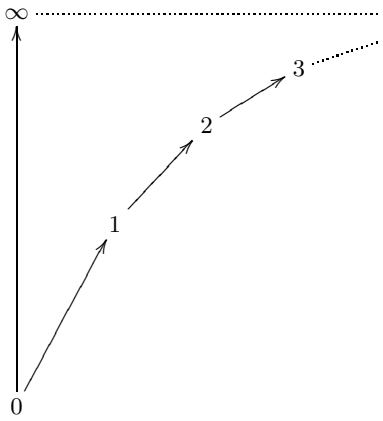


Fig. 4. An illustration of the need for ω-skew confluence.

3 Lack of Confluence in Term Graphs

The need for a less restrictive notion of confluence arises in practice if one wants to provide a more accurate foundation of programming languages. To reason about either execution or optimizations one has to deal with the notion of sharing and cyclic structures [12–14]. As pointed out by Wadsworth [15], these concerns can be accommodated by considering term graph rewriting as opposed to term (or tree) rewriting.

As pointed out in [16, 17], term graphs can be nicely represented as terms with letrec's. The cyclic structure depicted in Fig. 1 would be represented as

$$\langle x \mid x = \mathsf{Cons}(1, x) \rangle$$

The advantage of this representation is that one can apply existing term rewrite rules directly to the cyclic term[4]. However, the old rewrite rules are not enough: we must also use rules that modify the letrec structure. Typical examples of such rules are the rules for external, internal and cyclic substitution:

$$\begin{aligned} \langle C[x] \mid x = M, E \rangle &\xrightarrow{\text{es}} \langle C[M] \mid x = M, E \rangle \\ \langle M \mid x = C[y], y = N, E \rangle &\xrightarrow{\text{is}} \langle M \mid x = C[N], y = N, E \rangle \\ \langle M \mid x = C[x], E \rangle &\xrightarrow{\text{cs}} \langle M \mid x = C[C[x]], E \rangle \end{aligned}$$

where $C[x]$ stands for a one-hole context filled with variable x. The problem with these simple rules is that confluence is lost. The classical example is:

$$\begin{array}{ccccc} & & M & & \\ & & \equiv & & \\ \langle x \mid x = \mathsf{F}(x) \rangle & \xrightarrow{\text{es}} & \langle \mathsf{F}(x) \mid x = \mathsf{F}(x) \rangle & \xrightarrow{\text{cs}} & \langle \mathsf{F}(x) \mid x = \mathsf{F}(\mathsf{F}(x)) \rangle \\ \downarrow{\scriptstyle \text{cs}} & & & & \equiv \\ & & & & M_o \\ \langle x \mid x = \mathsf{F}(\mathsf{F}(x)) \rangle & & & & \\ \equiv & & & & \\ M_e & & & & \end{array}$$

The cyclic terms M_o and M_e do not have a common reduct because any reduct of M_o will contain an odd number of F symbols and any reduct of M_e an even number.

The lack of confluence of the substitution rules themselves is not a major point of concern. Substitution rules transform terms to bisimilar terms. That is, into terms which represent the same infinite tree. From this it follows that although the reduction is not confluent, it is confluent modulo bisimulation.

However, when we combine the substitution rules with a rewrite system then matters can become worse. For example, consider the TRS

$$\begin{aligned} \mathsf{F}(\mathsf{F}(X)) &\to \mathsf{F}(\mathsf{G}(X)) \\ \mathsf{G}(\mathsf{G}(X)) &\to \mathsf{G}(\mathsf{F}(X)) \end{aligned}$$

[4] Because of the capability of the letrec to represent graphs with cycles, we refer to terms with letrec's as cyclic terms.

This TRS is confluent and terminating. However, when we apply these rewrite rules to M_o and M_e we get:

$$M_e \to \langle x \mid x = \mathsf{F}(\mathsf{G}(x))\rangle \equiv M'_e$$
$$M_o \to \langle \mathsf{F}(x) \mid x = \mathsf{F}(\mathsf{G}(x))\rangle \equiv M'_o$$

As before, a count of the symbols leads to the conclusion that these M'_e and M'_o do not have a common reduct. But unlike before, we also lose confluence modulo bisimulation. All reducts of M'_e will be of the form

$$\langle (\mathsf{FG})^n(x) \mid x = (\mathsf{FG})^m(x)\rangle$$

where $(\mathsf{FG})^0(x) = x$ and $(\mathsf{FG})^{n+1}(x) = \mathsf{F}(\mathsf{G}((\mathsf{FG})^n(x)))$. All reducts of M'_o will be of the form

$$\langle (\mathsf{FG})^n(\mathsf{G}((\mathsf{FG})^k(x))) \mid x = (\mathsf{FG})^m(x)\rangle \text{ or } \langle (\mathsf{FG})^n(\mathsf{F}((\mathsf{FG})^k(x))) \mid x = (\mathsf{FG})^m(x)\rangle$$

Let us consider the symbols we find after following paths of even length starting from the root. In the case of a reduct of M'_e, we find only F symbols, which means that we find infinitely many F symbols. In the case of a reduct of M'_o, it is difficult to say what symbol we find in the prefix. But once we get to the loop, we find only G symbols. Hence, we find at most finitely many F symbols. If two graphs are bisimilar then the symbols reachable by paths of a certain length are identical. Hence, it is impossible that any reduct of M'_e is bisimilar to a reduct of M'_o.

We know that our rewrite system is not confluent modulo bisimulation. However, neither can we produce random results: any reduct of the original M will be of the form

$$\langle \mathsf{F}(?(?(?(\cdots(x)\cdots)))) \mid x = (\mathsf{FG})^m(x)\rangle$$

where ? stands for F or G. Any term of this form will reduce to

$$\langle \mathsf{F}(\mathsf{G}(\mathsf{F}(\mathsf{G}(\cdots(x)\cdots)))) \mid x = (\mathsf{FG})^m(x)\rangle$$

From this we can derive that given any two reducts of M, we can reduce the second reduct until its alternating F/G prefix is at least as big as the alternating F/G prefix of the first reduct. This means that when we look at prefixes, we have skew confluence. If we look at unwindings then the F/G prefix of the unwinding of M'_e is an infinite alternating F/G sequence, but the unwindings of the reducts of M'_o only start with a finite alternating F/G sequence. So when we look at unwindings, we do not have skew confluence. However, we do have ω-skew confluence because by reducing M'_o we can match any finite prefix of the unwinding of M'_e.

4 Skew and ω-Skew Confluence and Infinite Normal Forms

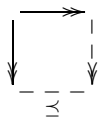
In this section we introduce skew confluence, ω-skew confluence and infinite normal forms with (in)finite information content.

To define skew confluence we need a way of telling if an object is better than another object. We formalize this by considering an ARS and a quasi order.

Definition 1 (skew confluence). *Given an ARS $\mathcal{A} \equiv (A, \rightarrow)$ and a quasi order $(A, \preceq)$. The ARS $\mathcal{A}$ is skew confluent up to $\preceq$ if*

$$\forall a_1, a_2, a_3 \in A : a_1 \rightarrow\!\!\!\!\!\rightarrow a_2 \wedge a_1 \rightarrow\!\!\!\!\!\rightarrow a_3 \Rightarrow \exists a_4 : a_2 \preceq a_4 \wedge a_3 \rightarrow\!\!\!\!\!\rightarrow a_4 \ .$$

The commutative diagram for skew confluence is



Confluence implies skew confluence. More precisely, if the reduction relation is increasing in a quasi order then confluence implies skew confluence up to that quasi order:

Proposition 1. *Given an ARS $\mathcal{A} \equiv (A, \rightarrow)$ and a quasi order $(A, \preceq)$. If $\rightarrow \subset \preceq$ and $\mathcal{A}$ is confluent then $\mathcal{A}$ is skew confluent up to $\preceq$.*

We will now formally define infinite normal forms with finite information content. But first we introduce the following definition.

Definition 2. *Given a partial order $(D, \leq)$,*

- *we define the downward closure of a set $C \subset D$, denoted by $\downarrow C$, as*

$$\downarrow C = \{b \in D \mid b \leq a \in C\} \ ;$$

- *a subset I of D is an ideal iff (i) I is non-empty, (ii) I is directed: $\forall a, b \in I, \exists c \in I, a \leq c$ and $b \leq c$, (iii) I is downward closed: $\forall c \in I$, if $\exists d \in D, d \leq c$ then $d \in I$.*

Definition 3. *Given an ARS $(A, \rightarrow)$ and a partial order $(D, \leq)$.*

- *A notion of finite information content is a function $\omega : A \rightarrow D$, such that $a_1 \rightarrow a_2 \Rightarrow \omega(a_1) \leq \omega(a_2)$.*
- *The infinite normal form generated by ω is defined as*

$$\mathrm{Inf}_\omega(a) = \downarrow\{\omega(a') \mid a \rightarrow\!\!\!\!\!\rightarrow a'\}$$

- *A notion of information content yields unique infinite normal forms if*

$$a \rightarrow a' \Rightarrow \mathrm{Inf}_\omega(a) = \mathrm{Inf}_\omega(a')$$

- *The information order $\leq_\omega$ is defined by*

$$a_1 \leq_\omega a_2 \text{ if } \omega(a_1) \leq \omega(a_2) \ .$$

Next, we prove that uniqueness of infinite normal forms with finite information content is equivalent to skew confluence up to.

Theorem 1. *A notion of information content ω yields unique infinite normal forms iff the ARS is skew confluent up to $\leq_\omega$.*

Proof. "$\Rightarrow$" Assuming uniqueness of infinite normal form. If $a_1 \twoheadrightarrow a_2$ and $a_1 \twoheadrightarrow a_3$ then $\mathrm{Inf}_\omega(a_1) = \mathrm{Inf}_\omega(a_3)$ by induction on the length of the reduction from a_1 to a_3. Because $a_1 \twoheadrightarrow a_2$, we know that $\omega(a_2) \in \mathrm{Inf}_\omega(a_1)$. So we also know that $\omega(a_2) \in \mathrm{Inf}_\omega(a_3)$. That means that there exists an a_4, such that $a_3 \twoheadrightarrow a_4$ and $\omega(a_2) \leq \omega(a_4)$.

"$\Leftarrow$" Assuming skew confluence, we have to show $\mathrm{Inf}_\omega(a_1) = \mathrm{Inf}_\omega(a_2)$.

"$\supset$" Trivial.

"$\subset$" Given $d \in \mathrm{Inf}_\omega(a_1)$, there exist and a_3, such that $a_1 \twoheadrightarrow a_3$ and $d \leq \omega(a_3)$. By skew confluence there exists an a_4, such that $a_2 \twoheadrightarrow a_4$ and $a_3 \leq_\omega a_4$. By definition, we have $\omega(a_4) \in \mathrm{Inf}_\omega(a_2)$ and since $d \leq \omega(a_4)$ this implies $d \in \mathrm{Inf}_\omega(a_2)$.

To define infinite normal forms with infinite information content, we use the notion of ideal completion of a partial order. If D is a partial order then $\mathcal{I}(D)$ denotes the ideal completion of D. Using this notation, we define infinite normal form with infinite information content as follows:

Definition 4. *Given an ARS $(A, \to)$ and a partial order $(D, \leq)$.*

- *A notion of infinite information content is a function $\omega : A \to \mathcal{I}(D)$, such that $a_1 \to a_2 \Rightarrow \omega(a_1) \subseteq \omega(a_2)$.*
- *The infinite normal form generated by ω is defined as*

$$\mathrm{Inf}_\omega(a) = \cup\{\omega(a') \mid a \twoheadrightarrow a'\}$$

- *The information order $\leq_\omega$ is defined by*

$$a_1 \leq_\omega a_2 \text{ if } \omega(a_1) \subset \omega(a_2) \ .$$

The examples in Sections 2 and 3 show that skew confluence up to the information order is not a necessary condition for uniqueness of infinite normal forms with infinite information content. The notion that is necessary and sufficient is the notion of ω-skew confluence, which we introduce next.

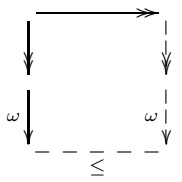
Definition 5 (ω-skew confluence). *Given an ARS $\mathcal{A} \equiv (A, \to)$, a partial order $(D, \leq)$ and a function $\omega : A \to \mathcal{I}(D)$. The ARS $\mathcal{A}$ is ω-skew confluent with respect to ω if*

$$\forall a_1, a_2, a_3, d_1, a_1 \twoheadrightarrow a_2 \wedge a_1 \twoheadrightarrow a_3 \wedge d_1 \in \omega(a_2) : \exists a_4, d_2 : a_3 \twoheadrightarrow a_4 \wedge d_2 \in \omega(a_4) \wedge d_1 \leq d_2 \ .$$

For use in diagrams we define $\xrightarrow[\omega]{}$ by

$$a \xrightarrow{\omega} d, \text{ if } d \in \omega(a) \ .$$

Note that the ω in ω-skew confluence is *not* the function ω used in the definition, but the first infinite ordinal. The commutative diagram for ω-skew confluence is



Theorem 2. *A notion of infinite information content ω yields unique infinite normal forms iff the ARS is ω-skew confluent with respect to ω.*

Proof. Observe that

$$\mathrm{Inf}_\omega(a) = \{d \mid a \twoheadrightarrow a' \underset{\omega}{\longrightarrow} d\} \ .$$

"$\Rightarrow$" Assuming uniqueness of infinite normal form. If $a_1 \twoheadrightarrow a_2 \underset{\omega}{\longrightarrow} d_1$ and $a_1 \twoheadrightarrow a_3$ then $\mathrm{Inf}_\omega(a_1) = \mathrm{Inf}_\omega(a_3)$ by induction on the length of the reduction from a_1 to a_3. We know that $d_1 \in \mathrm{Inf}_\omega(a_1)$. So we also know that $d_1 \in \mathrm{Inf}_\omega(a_3)$. That means that there exists an a_4, such that $a_3 \twoheadrightarrow a_4$ and $a_4 \underset{\omega}{\longrightarrow} d_1$.

"$\Leftarrow$" Assuming skew confluence, we have to show $\mathrm{Inf}_\omega(a_1) = \mathrm{Inf}_\omega(a_2)$.

"$\supset$" Trivial.

"$\subset$" Given $d \in \mathrm{Inf}_\omega(a_1)$, there exist and a_3, such that $a_1 \twoheadrightarrow a_3$ and $a_3 \underset{\omega}{\longrightarrow} d$. By skew confluence there exists an a_4 and d', such that $a_2 \twoheadrightarrow a_4$ and $a_4 \underset{\omega}{\longrightarrow} d'$ and $d \leq d'$. This implies that $a_4 \underset{\omega}{\longrightarrow} d$ and hence $d \in \mathrm{Inf}_\omega(a_2)$.

Proposition 2. *Given an ARS $\mathcal{A} \equiv (A, \to)$, a partial order $(D, \leq)$ and a function $\omega : A \to \mathcal{I}(D)$. If $\mathcal{A}$ is skew confluent up to $\leq_\omega$ then $\mathcal{A}$ is ω-skew confluent with respect to ω.*

Observe that we have ω-skew confluence iff

$$\forall a_1, a_2, a_3 \forall d_1 \exists a_4 \exists d_2 : a_1 \twoheadrightarrow a_2 \wedge a_1 \twoheadrightarrow a_3 \wedge d_1 \in \omega(a_2), a_3 \twoheadrightarrow a_4 \wedge d_2 \in \omega(a_4) \wedge d_1 \leq d_2$$

and skew confluence iff

$$\forall a_1, a_2, a_3 \exists a_4 \forall d_1 \exists d_2 : a_1 \twoheadrightarrow a_2 \wedge a_1 \twoheadrightarrow a_3 \wedge d_1 \in \omega(a_2), a_3 \twoheadrightarrow a_4 \wedge d_2 \in \omega(a_4) \wedge d_1 \leq d_2 \ .$$

5 Cyclic Lambda Calculi

We now apply the notions of skew and ω-skew confluence to extensions of the call-by-name lambda-calculus and of the call-by-need lambda-calculus [18, 19] with cyclic structures, as defined below.

Definition 6. *The set of cyclic lambda terms $\Lambda\circ$ is defined as follows:*

$$\begin{array}{ll} \textit{Terms} & M ::= x \ \mid \ \lambda x.M \ \mid \ M\ N \mid \ \langle M \mid D \rangle \\ \textit{Declarations} & D ::= x_1 = M_1, \ldots, x_n = M_n \end{array}$$

where the variables $x_i, 1 \leq i \leq n$, are distinct from each other and the order of the equations does not matter.

$$
\begin{array}{lll}
(\lambda x.M)\ N & \xrightarrow[\beta\circ]{} \langle M \mid x = N\rangle & \\
\langle C[x] \mid x = M, D\rangle & \xrightarrow[\text{es}]{} \langle C[M] \mid x = M, D\rangle & \\
\langle M \mid x = C[y], y = N, D\rangle & \xrightarrow[\text{is}]{} \langle M \mid x = C[N], y = N, D\rangle & \\
\langle\langle M \mid D_1\rangle \mid D_2\rangle & \xrightarrow[\text{em}]{} \langle M \mid D_1, D_2\rangle & \\
\langle M \mid x = \langle N \mid D_1\rangle, D_2\rangle & \xrightarrow[\text{im}]{} \langle M \mid x = N, D_1, D_2\rangle & \\
\langle M \mid D\rangle\ N & \xrightarrow[\text{lift}]{} \langle M\ N \mid D\rangle & \\
M\ \langle N \mid D\rangle & \xrightarrow[\text{lift}]{} \langle M\ N \mid D\rangle & \\
\lambda x.\langle M \mid D_1, D_2\rangle & \xrightarrow[\text{lift}]{} \langle \lambda x.\langle M \mid D_1\rangle \mid D_2\rangle & (x = x, D_1) \perp D_2 \neq \{\} \\
\langle M \mid D_1, D_2\rangle & \xrightarrow[\text{gc}]{} \langle M \mid D_1\rangle & D_2 \neq \{\}, D_2 \perp \langle M \mid D_1\rangle \\
\langle M \mid\rangle & \xrightarrow[\text{gc}]{} M & \\
M^\sigma & \xrightarrow[\text{cp}]{} M & \sigma : \mathcal{V} \to \mathcal{V}
\end{array}
$$

Fig. 5. The cyclic call-by-name lambda calculus

5.1 The Cyclic Call-by-name Calculus

As an application of skew-confluence we consider the cyclic call-by-name calculus, given in Fig. 5. Note that M^σ doesn't stand for substitution but for defined variable collapsing, as defined next.

Definition 7. *Given a function σ from recursion variables V to $\mathcal{V}$, M^σ is defined as follows:*

$$
\begin{array}{lll}
x^\sigma & = \sigma(x) & \\
(M\ N)^\sigma & = M^\sigma\ N^\sigma & \\
(\lambda x.M)^\sigma & = \lambda x.M^\sigma & \text{if } \sigma(x) = x \wedge \sigma^{-1}(x) = \{x\} \\
\langle M \mid x_1 = M_1, \cdots, x_n = M_n\rangle^\sigma & = \langle M^\sigma \mid x_1^\sigma = M_1^\sigma, \cdots, x_n^\sigma = M_n^\sigma\rangle & \\
 & \qquad \text{if } \sigma(x_i) = \sigma(x_j) \Rightarrow M_i^\sigma = M_j^\sigma &
\end{array}
$$

For example if M is $\lambda z.\langle x \mid x = z\,y, y = z\,x\rangle$ and $\sigma(x) = \sigma(y) = u$ then

$$M^\sigma \equiv \lambda z.\langle u \mid u = z\,u\rangle$$

The symbol $\perp$ is used to denote independence of definitions: $D_2\ \perp \langle M \mid D_1\rangle$ means that the set of variables that occur as the left-hand side of an equation in D_2 does not intersect with the set of free variables of M and of D_1. This guarantees that the free variables of M and of D_1 do not refer to the variables defined in D_2.

In Section 3, we already pointed out that the substitution rules lead to non-confluence. However, $\lambda\circ$ is skew-confluent with respect to the following notion of finite information content.

Definition 8. *Given $M, N \in \Lambda\circ$, the finite information content of M is given by the function $\omega_{\lambda\circ}$. Given M, $\omega_{\lambda\circ}(M)$ returns the normal form of M with respect*

$$
\begin{array}{lll}
(\lambda x.M)N & \xrightarrow[\beta\circ]{} & \langle M \mid x = N\rangle \\
\langle C[x] \mid x = V, D\rangle & \xrightarrow[\mathrm{es_V}]{} & \langle C[V] \mid x = V, D\rangle \\
\langle M \mid x = C[x_1], x_1 = V, D\rangle & \xrightarrow[\mathrm{is_V}]{} & \langle M \mid x = C[V], x_1 = V, D\rangle \\
\langle M \mid D\rangle N & \xrightarrow[\mathrm{lift}]{} & \langle MN \mid D\rangle \\
M\langle N \mid D\rangle & \xrightarrow[\mathrm{lift}]{} & \langle MN \mid D\rangle \\
\lambda x.\langle M \mid D, VD\rangle & \xrightarrow[\mathrm{lift}]{} & \langle \lambda x.\langle M \mid D\rangle \mid VD\rangle \qquad (x = x, D) \perp VD \neq \{\} \\
\langle\langle M \mid D\rangle \mid D'\rangle & \xrightarrow[\mathrm{em}]{} & \langle M \mid D, D'\rangle \\
\langle M \mid x = \langle N \mid D\rangle, D_1\rangle & \xrightarrow[\mathrm{im}]{} & \langle M \mid x = N, D, D_1\rangle \\
\langle M \mid D, D'\rangle & \xrightarrow[\mathrm{gc}]{} & \langle M \mid D\rangle \qquad \{\} \neq D', D' \perp \langle M \mid D\rangle \\
\langle M \mid \rangle & \xrightarrow[\mathrm{gc}\Box]{} & M \\
M & \xrightarrow[\mathrm{cp_V}]{} & N \qquad \exists\sigma : \mathcal{V} \to \mathcal{V}, N^\sigma \equiv M \text{ and} \\
 & & \qquad \forall x \not\equiv x', \sigma(x) \equiv \sigma(x') : \\
 & & \qquad \sigma(x) \text{ bound to a value in } M \\
C_{\mathrm{safe}}[M\ N] & \xrightarrow[\mathrm{name}]{} & C_{\mathrm{safe}}[\langle x \mid x = M\ N\rangle] \qquad x \text{ a new variable}
\end{array}
$$

where

$$
\begin{array}{rcl}
C_{\mathrm{safe}} & ::= & C' \mid C[\lambda x.C'] \mid C[C'\ M] \mid C[M\ C'] \ . \\
C' & ::= & \Box \mid \langle C' \mid D\rangle \\
V & ::= & x \mid \lambda x.M \\
VD & ::= & x_1 = V_1, \cdots, x_n = V_n
\end{array}
$$

Fig. 6. The cyclic call-by-need lambda calculus

to the following rules:

$$
\begin{array}{rcll}
(\lambda x.M)N & \xrightarrow[\omega_{\lambda\circ}]{} & \Omega & \beta\omega \\
\langle C[x] \mid x = M, D\rangle & \xrightarrow[\omega_{\lambda\circ}]{} & \langle C[\Omega] \mid x = M, D\rangle & es\omega \\
\Omega M & \xrightarrow[\omega_{\lambda\circ}]{} & \Omega & @\omega \\
\langle M \mid D\rangle & \xrightarrow[\omega_{\lambda\circ}]{} & M \qquad D \perp M & \mathrm{gc}\omega
\end{array}
$$

Examples: $\omega_{\lambda\circ}(\langle \lambda x.yz \mid y = I\rangle) = \lambda x.\Omega$, $\omega_{\lambda\circ}(\langle x \mid x = x\rangle) = \Omega$, $\omega_{\lambda\circ}(\langle xy \mid y = I\rangle x) = (x\Omega)x$, and $\omega_{\lambda\circ}(\langle xx \mid x = I\rangle) = \Omega$. Note that even though $\langle xy \mid y = I\rangle x$ is a lift redex, its information content is not Ω.

Theorem 3. *The ARS $\lambda\circ$ is skew confluent up to $\leq_{\omega_{\lambda\circ}}$.*

A direct proof can be found in [10]. In the next section, we will develop a theory which allows us to prove uniqueness of infinite normal forms from a list of other properties.

5.2 A Cyclic Call-by-need Calculus

The cyclic call-by-need calculus $\lambda\circ_{\mathrm{need}}$ is given in Fig. 6.

Definition 9. *Given $M, N \in \Lambda\circ$, the infinite information content of M is given by the following function $\omega_{\lambda^\circ_{\text{need}}}$:*

$$\omega_{\lambda^\circ_{\text{need}}}(M) = \downarrow \{\omega_{\lambda\circ}(N) \mid M \xrightarrow[\text{es}]{}\!\!\!\!\!\rightarrow N\}$$

Due to the fact that the information content is infinite, we do not have skew confluence up to the information order. Consider the following two reductions:

$$\begin{aligned} M \equiv \langle x \mid x = \lambda z.z\ y, y = \lambda z'.z'\ (x\ z')\rangle &\xrightarrow[\lambda\circ_{\text{need}}]{}\!\!\!\!\!\rightarrow \langle \lambda z.z\ y \mid y = \lambda z'.z'\ ((\lambda z.z\ y)\ z')\rangle \\ &\xrightarrow[\lambda\circ_{\text{need}}]{} \langle \lambda z.z\ y \mid y = \lambda z'.z'\ (z'\ y)\rangle \equiv M_1 \end{aligned}$$

and

$$M \equiv \langle x \mid x = \lambda z.z\ y, y = \lambda z'.z'\ (x\ z')\rangle \twoheadrightarrow \langle x \mid x = \lambda z.z\ (\lambda z'.z'\ (x\ z'))\rangle \equiv M_2\ .$$

We have that $\omega_{\lambda^\circ_{\text{need}}}(M_1) = \text{Inf}_{\text{need}}(M_1)$, because the only redexes in M_1 and any of its reducts are value substitutions, which are performed as part of the computation of the information content. However, there cannot exist M_3 such that $M_2 \xrightarrow[\lambda\circ_{\text{need}}]{}\!\!\!\!\!\rightarrow M_3$ and $\omega_{\lambda^\circ_{\text{need}}}(M_1) \subseteq \omega_{\lambda^\circ_{\text{need}}}(M_3)$ because $\omega_{\lambda^\circ_{\text{need}}}(M_1)$ is infinite whereas the information content of any reduct of M_2 is finite. The reason is that in the unwinding of M we have an infinite number of β-redexes. When we rewrite M into M_1 we do all of those redexes at once and when we rewrite M into M_2 we destroy the opportunity to do them in one step. The consistency of $\lambda^\circ_{\text{need}}$ is guaranteed by the following theorem.

Theorem 4. *The ARS $\lambda^\circ_{\text{need}}$ is ω-skew confluent with respect to $\leq_{\omega_{\lambda^\circ_{\text{need}}}}$.*

A direct proof can be found in [20]. In the next section, we will develop a theory which allows us to prove consistency from a list of other properties.

6 Lifting Infinite Normal Forms

The infinite normal forms of both the cyclic call-by-name and call-by-need lambda calculi are closely related to unwinding. The information content for the cyclic call-by-name calculus can be seen as a two steps process. First, one computes the normal form with respect to the $es\omega$ and $gc\omega$ rules. Second, one applies the notion of information content associated to the lambda-calculus [8]. The call-by-need information content of a term is the information content of the unwinding of the term. In this section, we study how to derive this notion of information content in an abstract setting.

We first introduce in Section 6.1 the notion of a finite basis and its properties. The finite basis will be modeling the lambda calculus. In Section 6.2 we consider extensions consisting of infinite objects over the basis and objects whose semantics are infinite objects. In Section 6.3, we consider infinite normal forms of extensions.

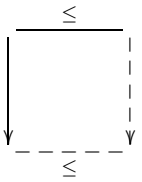
6.1 Finite Basis

We start from an ARS equipped with a partial order on its objects and a notion of finite information content. Using the partial order we can build the ideal completion [21], which will function as the set of possibly infinite objects. To be able to lift the notion of information content to infinite objects, we must require that the notion of information content and the rewrite relation are also monotonic with respect to the partial order. We formalize this starting point with the notion of a *finite basis*, for which we need one auxiliary definition: monotonicity of a rewrite relation with respect to an order.

Definition 10. *Given an ARS $(A, \to)$ and a partial order $(A, \leq)$, we say that $\to$ is monotonic with respect to $\leq$ if*

$$a \to a' \wedge a \leq a'' \Rightarrow \exists a''' : a' \leq a''' \wedge a'' \to a'''$$

The diagram of monotonicity is



Definition 11 (finite basis). *A tuple $(A, \to, \leq_A, \omega, D, \leq_D)$ is a finite basis if*

- *$(A, \to)$ is an ARS;*
- *$(A, \leq_A)$ and $(D, \leq_D)$ are partial orders;*
- *$\omega : A \to D$ is a finite notion of information content on $(A, \to)$, which yields unique infinite normal forms and is monotone with respect to $\leq_A$;*
- *$\to$ is monotone with respect to $\leq_A$.*

Example 1. Let Λ stand for the set of lambda-calculus terms and ω_{LL} stand for the function which given a term M returns the normal form of M with respect to the following ω_{LL}-rules [7]:

$$\begin{array}{ll} (\lambda x.M)\, N & \xrightarrow[\omega_{\mathrm{LL}}]{} \Omega \\ \Omega\, M & \xrightarrow[\omega_{\mathrm{LL}}]{} \Omega \end{array}$$

Then one has that $(\Lambda, \xrightarrow[\beta]{}, \leq_\Omega, \omega_{\mathrm{LL}}, \Lambda, \leq_\Omega)$ is a finite basis.

Next, we consider rewrite systems, referred to as extensions, whose objects have infinite objects as semantics. Moreover, we want these rewrite systems to mimic the behavior of their finite counterparts. For the cyclic lambda calculi mimicking the finite lambda calculus meant that the rewrite relation induced by the cyclic calculi was contained in the infinitary lambda calculus and that finite reductions in an approximation could be lifted to reduction in the extension. The equivalent of this involves lifting the reduction in a finite basis to a reduction on its ideal completion.

6.2 Extensions

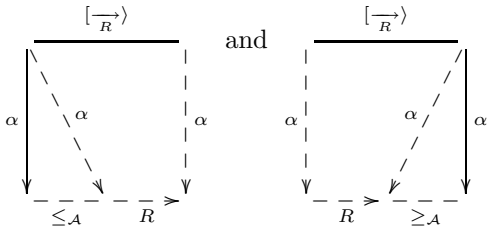
The set of infinite terms can be seen as the ideal completion of the set of finite terms under the prefix order $\leq_\Omega$. Therefore, we treat the ideal completion of a set of objects as infinite objects. We then define a rewrite relation on ideals as follows: we say that an ideal rewrites to another if every sufficiently large element of the first ideal rewrites to an element of the second and every sufficiently large element of the second ideal can be obtained by rewriting an element of the first. This is in a way similar to how Corradini defined complete developments of an infinite set of redexes in an infinite term [22].

Definition 12. *Given a finite basis* $\mathcal{A} = (A, \underset{\mathcal{A}}{\longrightarrow}, \leq_A, \omega, D, \leq_D)$*. The operator* $[\cdot\rangle : (A \times A) \to \mathcal{I}(A) \times \mathcal{I}(A)$ *is defined by* $I_1[\underset{R}{\longrightarrow}\rangle I_2$ *if*

$$\forall a \in I_1 : \exists a' \in I_1, a'' \in I_2 : a \leq a' \underset{R}{\longrightarrow} a''$$

$$\forall a'' \in I_2 : \exists a' \in I_2, a \in I_1 : a \underset{R}{\longrightarrow} a' \geq a''$$

If for $I \in \mathcal{I}(A)$, we denote $a \in I$ as $I \underset{\alpha}{\longrightarrow} a$ then we can phrase this definition with the following 2 diagrams:



Example 2. Consider the infinitary lambda calculus term

$$(\lambda x.f\ x\ (f\ x\ (f\ x\ (\cdots))))(I\ I)$$

from the introduction. The reduction of $I\ I$ to I can be matched:

$$(\lambda x.f\ x\ (f\ x\ (f\ x\ (\cdots))))(I\ I)\ [\underset{\beta}{\longrightarrow}\rangle(\lambda x.f\ x\ (f\ x\ (f\ x\ (\cdots))))I$$

because

$$\begin{gathered}(\lambda x.\Omega)\ (I\ I) \underset{\beta}{\longrightarrow} (\lambda x.\Omega)\ I \\ (\lambda x.f\ x\ (\Omega))\ (I\ I) \underset{\beta}{\longrightarrow} (\lambda x.f\ x\ (\Omega))\ I \\ (\lambda x.f\ x\ (f\ x\ (\Omega)))\ (I\ I) \underset{\beta}{\longrightarrow} (\lambda x.f\ x\ (f\ x\ (\Omega)))\ I \\ \vdots\ \vdots\ \vdots\end{gathered}$$

It is obvious that for any single step we can do this. So we also have

$$(\lambda x.f\ x\ (f\ x\ (f\ x\ (\cdots))))I\ [\underset{\beta}{\longrightarrow}\rangle\ f\ I\ (f\ I\ (f\ I\ (\cdots)))$$

and

$$(\lambda x.f\,x\,(f\,x\,(f\,x\,(\cdots))))(I\,I)\;[\underset{\beta}{\longrightarrow}\rangle\;f\,(I\,I)\,(f\,(I\,I)\,(f\,(I\,I)\,(\cdots)))$$

Interestingly, we can also do infinite developments. An infinite development, of course, means performing a finite development on the prefix, so we have to use $[\underset{\beta}{\longrightarrow\!\!\!\!\!\rightarrow}\rangle$ rather than $[\underset{\beta}{\longrightarrow}\rangle$. The fact is that

$$f\,(I\,I)\,(f\,(I\,I)\,(f\,(I\,I)\,(\cdots)))\;[\underset{\beta}{\longrightarrow\!\!\!\!\!\rightarrow}\rangle\;f\,I\,(f\,I\,(f\,I\,(\cdots)))$$

follows from

$$\begin{gathered}
f\,(I\,I)\,\Omega \underset{\beta}{\longrightarrow\!\!\!\!\!\rightarrow} f\,I\,\Omega \\
f\,(I\,I)\,(f\,(I\,I)\,\Omega) \underset{\beta}{\longrightarrow\!\!\!\!\!\rightarrow} f\,I\,(f\,I\,(\Omega)) \\
f\,(I\,I)\,(f\,(I\,I)\,(f\,(I\,I)\,(\Omega))) \underset{\beta}{\longrightarrow\!\!\!\!\!\rightarrow} f\,I\,(f\,I\,(f\,I\,(\Omega))) \\
\vdots\;\;\vdots\;\;\vdots
\end{gathered}$$

Now that we can lift any relation from an order to its ideal completion, it is logical to also extend information content from the order to the ideal completion. Naturally the information contained in an ideal is infinite, so we define the information content of an ideal as the downward closure of the set of information contents of its elements:

Definition 13. *Given a finite basis* $\mathcal{A} = (A, \underset{\mathcal{A}}{\longrightarrow}, \leq_A, \omega, D, \leq_D)$ *and* $I \in \mathcal{I}(A)$. *Let*

$$\omega_\infty(I) = {\downarrow}\{\omega(a) \mid a \in I\}$$

This is well-defined because of the monotonicity of ω with respect to $\leq_{\mathcal{A}}$. Next, we consider the abstract version of an extension which contains objects whose semantics are infinite objects over the basis:

Definition 14. *A tuple* $\mathcal{A}^X \equiv (A^X, \underset{\mathcal{A}^X}{\longrightarrow}, [\![\cdot]\!])$ *is an extension of a finite basis* $\mathcal{A} \equiv (A, \underset{\mathcal{A}}{\longrightarrow}, \leq_A, \omega, D, \leq_D)$, *if* $(A^X, \underset{\mathcal{A}^X}{\longrightarrow})$ *is an ARS and* $[\![\cdot]\!] : A^X \to \mathcal{I}(A)$.

The cyclic lambda calculi are extensions of the lambda calculus in this sense because the semantics of a cyclic lambda term is its unwinding, which is an infinite term. For an extension to make sense, we require it to be sound and complete with respect to the basis [23, 24]. In other words, the extension cannot do more than the basis (soundness) and the extension can simulate everything the basis can do (completeness). To define soundness we use the $[\cdot\rangle$ operator. To define completeness we use a simple lifting property:

Definition 15. *Given a finite basis* $\mathcal{A} \equiv (A, \underset{\mathcal{A}}{\longrightarrow}, \leq_A, \omega, D, \leq_D)$, *its extension* $\mathcal{A}^X \equiv (A^X, \underset{\mathcal{A}^X}{\longrightarrow})$ *and a function* $[\![\cdot]\!] : A^X \to \mathcal{I}(A)$. *Then,*

- $\mathcal{A}^X$ *is infinitarily sound with respect to* $[\![\cdot]\!]$ *and* $\mathcal{A}$ *if*

$$s \underset{\mathcal{A}^X}{\longrightarrow} t \Rightarrow [\![s]\!]\;[\underset{\mathcal{A}}{\longrightarrow\!\!\!\!\!\rightarrow}\rangle\;[\![t]\!]$$

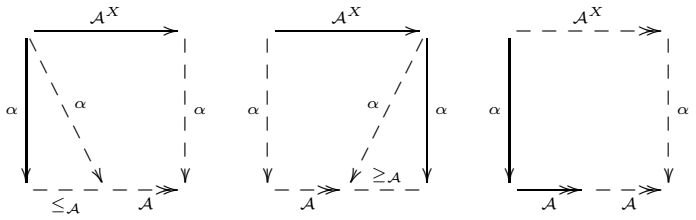


Fig. 7. Soundness and completeness of an extension as commutative diagrams

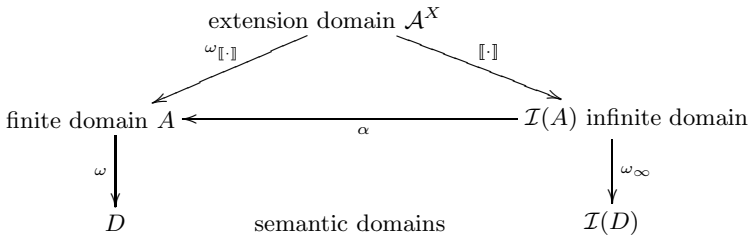


Fig. 8. Overview of the given domains and their connections.

– $\mathcal{A}^X$ *is infinitarily complete respect to* $[\![\cdot]\!]$ *and* $\mathcal{A}$ *if*

$$\forall a, s : a \in [\![s]\!] \wedge a \xrightarrow[\mathcal{A}]{}\!\!\!\!\rightarrow a' \Rightarrow \exists t, a'' : s \xrightarrow[\mathcal{A}^X]{}\!\!\!\!\rightarrow t \wedge a' \xrightarrow[\mathcal{A}]{}\!\!\!\!\rightarrow a'' \in [\![t]\!]$$

In order to be able to draw diagrams, we denote $a \in [\![s]\!]$ by $s \xrightarrow[\alpha]{} a$ as well. Using this notation, the diagrams for soundness and completeness can be drawn as given in Fig. 7.

At this point we have all of the domains that we want to consider: finite and infinite semantic domains, finite and infinite basis domains and an extension domain. Notions of information content from the finite base to the semantic domain are given as well. This is enough to define an infinite notion of information content on the extension. To define a finite notion of information content, we also need a function for the extension to finite objects. We will call this function $\omega_{[\![\cdot]\!]}$. It will be explained later. In Fig. 8 we have given an overview of the domains and the connection between them.

6.3 Infinite Normal Forms

We start with infinite information content because we only need the semantics of the extended objects. The idea is that the information content of an extended object is just the information content of the unwinding.

Theorem 5. *Given a finite basis* $\mathcal{A} \equiv (A, \xrightarrow[\mathcal{A}]{}, \leq_A, \omega, D, \leq_D)$ *and an extension* $\mathcal{A}^X \equiv (A^X, \xrightarrow[\mathcal{A}^X]{}, [\![\cdot]\!])$. *If*

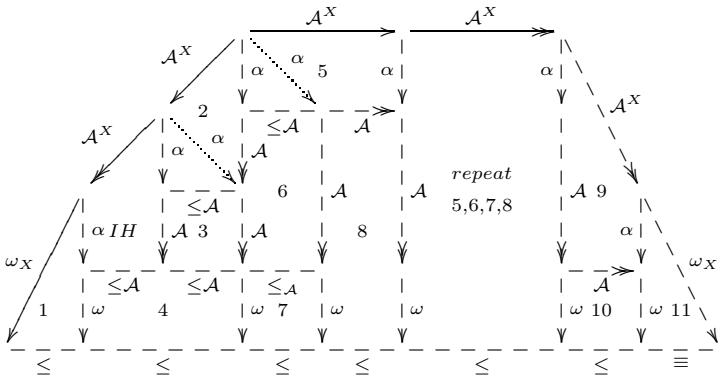
1. $\mathcal{A}^X$ *is infinitarily sound with respect to* $[\![\cdot]\!]$ *and* $\mathcal{A}$*;*
2. $\mathcal{A}^X$ *is infinitarily complete with respect to* $[\![\cdot]\!]$ *and* $\mathcal{A}$*;*

then $\omega_X : A^X \to \mathcal{I}(D)$ *defined by*

$$\omega_X(s) = \omega_\infty([\![s]\!])$$

is a notion of infinite information content which yields unique infinite normal forms.

Proof. To show that ω_X is a notion of infinite information content we need to show that if $s \xrightarrow[\mathcal{A}^X]{} s'$ then $\omega_X(s) \subseteq \omega_X(s')$. If $d \in \omega_X(s)$ then there exists an $a \in A$ such that $a \in [\![s]\!]$ and $d \leq \omega(a)$. By soundness of the extension there exist a' and a'' such that $a' \in [\![s]\!]$, $a \leq_{\mathcal{A}} a'$, $a'' \in [\![s']\!]$ and $a' \xrightarrow[\mathcal{A}]{} a''$. By monotonicity of ω w.r.t. $\leq_{\mathcal{A}}$, we have that $\omega(a) \leq \omega(a')$. Because ω is a notion of information content, we have $\omega(a') \leq \omega(a'')$. Hence $d \in \omega_X(s')$. To show that ω_X yields unique infinite normal forms, we need to show that $\xrightarrow[\mathcal{A}^X]{}$ is ω-skew confluent up to $\leq_{\omega_X}$. We do this by constructing a commutative diagram. The numbers indicate the order in which the construction can take place and below the diagram we indicate the reasons for the pieces of the diagram.



1, 11 Definition of ω_X.
2, 5 Because $\mathcal{A}^X$ is infinitarily sound with respect to $[\![\cdot]\!]$ and $\mathcal{A}$
4,7,10 Because ω is monotonic w.r.t. $\leq_{\mathcal{A}}$.
3,6 Because $\xrightarrow[\mathcal{A}]{}$ and $\leq_{\mathcal{A}}$ commute.
8 Because $\xrightarrow[\mathcal{A}]{}$ is skew confluent up to $\leq_\omega$.
9 Because $\mathcal{A}^X$ is infinitarily complete with respect to $[\![\cdot]\!]$ and $\mathcal{A}$.
10 Because ω is monotonic w.r.t. $\xrightarrow[\mathcal{A}]{}$.

Next, we consider the more complicated matter of defining a notion of finite information content on an extension. This situation arises when the extension

contains the computation of the semantics as a proper subset. This is the case in the call-by-name cyclic lambda calculus. The unwinding of a cyclic lambda term is an infinite normal form itself. One can define the rewrite system $[\![\cdot]\!]$ as external substitution

$$\langle C[x] \mid x = M, D\rangle \xrightarrow[[\![\cdot]\!]]{} \langle C[M] \mid x = M, D\rangle$$

and $\omega_{[\![\cdot]\!]}$ as the normal form with respect to

$$\langle M \mid x_1 = M_1, \cdots, x_n = M_n\rangle \xrightarrow[\omega_{[\![\cdot]\!]}]{} M[x_1 := \Omega, \cdots, x_n := \Omega]$$

Then the information content of a cyclic term M becomes:

$$\omega_{\lambda\circ}(M) = \omega_{\mathrm{LL}}(\omega_{[\![\cdot]\!]}(M))$$

We copy the idea for our abstract setting. That is, we assume that $[\![\cdot]\!]$ is given as an infinite normal form with finite information content. Based on this assumption, we define the information content of an extended object as the base information content of the unwiding information content of the extended object. Unfortunately, it is not guaranteed that this composition is actually a notion of information content. For example, consider the TRS

$$\begin{aligned}\mathsf{F}(\mathsf{F}(X)) &\to \mathsf{F}(\mathsf{G}(X))\\ \mathsf{G}(\mathsf{G}(X)) &\to \mathsf{G}(\mathsf{F}(X))\end{aligned}$$

Let $\omega(M)$ denote the largest possible alternating prefix of M. For example, $\omega(\mathsf{F}(\mathsf{F}(X))) = \mathsf{F}(\Omega)$. The function ω is a proper notion of information content, but if we consider the cyclic extension

$$\begin{aligned}\mathsf{F}(\mathsf{F}(X)) &\to \langle x \mid x = \mathsf{F}(\mathsf{G}(X))\rangle\\ \mathsf{G}(\mathsf{G}(X)) &\to \langle x \mid x = \mathsf{G}(\mathsf{F}(X))\rangle\\ \langle x \mid x = M, D\rangle &\to \langle M \mid x = M, D\rangle\\ \langle \mathsf{F}(x) \mid x = M, D\rangle &\to \mathsf{F}(\langle x \mid x = M, D\rangle)\\ \langle \mathsf{G}(x) \mid x = M, D\rangle &\to \mathsf{G}(\langle x \mid x = M, D\rangle)\end{aligned}$$

then $\omega \circ \omega_{[\![\cdot]\!]}$ is not a notion of information content:

$$(\omega \circ \omega_{[\![\cdot]\!]})(\mathsf{F}(\mathsf{F}(X))) = \omega(\mathsf{F}(\mathsf{F}(X))) = \mathsf{F}(\Omega)$$

and

$$(\omega \circ \omega_{[\![\cdot]\!]})(\langle x \mid x = \mathsf{F}(\mathsf{G}(X))\rangle) = \omega(\Omega) = \Omega$$

So $\omega\circ\omega_{[\![\cdot]\!]}$ is not monotonic with respect to the reduction relation of the extension. The problem is that the extended rewrite rules destroy a part of the unwinding, which was already part of the information content. The solution therefore is to require that every rewrite step in the extension preserve enough unwinding to compute at least the information content of the left-hand side:

$$\forall s, t \in A^X : s \xrightarrow[\mathcal{A}^X]{} t \Rightarrow \exists a \in A : a \leq_{\mathcal{A}} \omega_{[\![\cdot]\!]}(s) \wedge a \leq_{\mathcal{A}} \omega_{[\![\cdot]\!]}(t) \wedge \omega(a) = \omega(\omega_{[\![\cdot]\!]}(s))$$

So the rewrite step from s to t preserves a part of the unwinding a and a is enough to compute the information content of s.

For the call-by-name calculus this property holds, because if $M \xrightarrow[\lambda\circ]{} N$ then it is either not a $\beta\circ$ step and $\omega_{[\![\cdot]\!]}(M) \leq_\Omega \omega_{[\![\cdot]\!]}(N)$ and we can take $a = \omega_{[\![\cdot]\!]}(M)$ or it is a $\beta\circ$ step $C[(\lambda x.P)\,Q] \xrightarrow[\beta\circ]{} C[\langle P \mid x = Q\rangle]$. In this case we can take $a = \omega_{[\![\cdot]\!]}(C[\Omega])$.

Theorem 6. *Given a finite basis $\mathcal{A} \equiv (A, \xrightarrow[\mathcal{A}]{}, \leq_A, \omega, D, \leq_D)$, an extension $\mathcal{A}^X \equiv (A^X, \xrightarrow[\mathcal{A}^X]{}, [\![\cdot]\!])$ and a subset $\xrightarrow[[\![\cdot]\!]]{}$ of $\xrightarrow[\mathcal{A}^X]{}$ plus a notion of information content $\omega_{[\![\cdot]\!]}$ that computes $[\![\cdot]\!]$. If*

1. *$\mathcal{A}^X$ is infinitarily sound with respect to $[\![\cdot]\!]$ and $\mathcal{A}$;*
2. *$\mathcal{A}^X$ is infinitarily complete with respect to $[\![\cdot]\!]$ and $\mathcal{A}$;*
3. *$\forall s,t \in A^X : s \xrightarrow[\mathcal{A}^X]{} t \Rightarrow \exists a \in A : a \leq_\mathcal{A} \omega_{[\![\cdot]\!]}(s) \wedge a \leq_\mathcal{A} \omega_{[\![\cdot]\!]}(t) \wedge \omega(a) = \omega(\omega_{[\![\cdot]\!]}(s))$*

then

$$\omega_X(M) = \omega(\omega_{[\![\cdot]\!]}(M))$$

is a notion of information content which yields unique infinite normal forms.

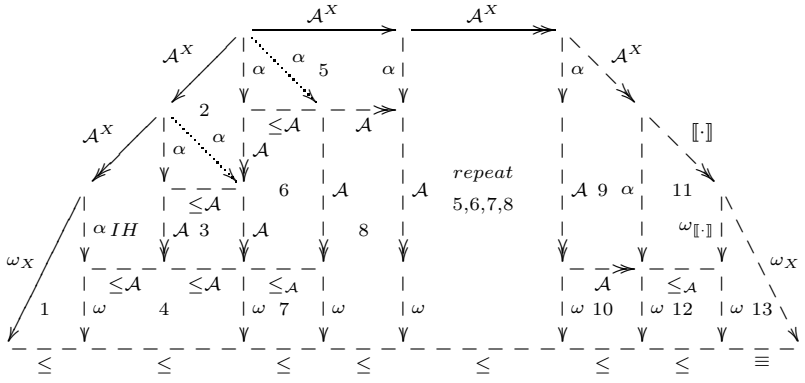
Proof. To show that ω_X is a notion of finite information content we need to show that if $M \xrightarrow[\mathcal{A}^X]{} M'$ then $\omega_X(M) \leq \omega_X(M')$. By condition 3, we can find $a \in A$ such that

$$a \leq_\mathcal{A} \omega_{[\![\cdot]\!]}(M) \wedge a \leq_\mathcal{A} \omega_{[\![\cdot]\!]}(M') \wedge \omega(a) = \omega(\omega_{[\![\cdot]\!]}(M))$$

Because ω is monotone with respect to $\leq_\mathcal{A}$, we have

$$\omega_X(M) = \omega(\omega_{[\![\cdot]\!]}(M)) = \omega(a) \leq \omega(\omega_{[\![\cdot]\!]}(M')) = \omega_X(M')$$

To show that ω_X yields unique infinite normal forms, we need to show that $\xrightarrow[\mathcal{A}^X]{}$ is skew confluent up to $\leq_{\omega_X}$. We do this by constructing a commutative diagram. The numbers indicate the order in which the construction can take place and below the diagram we indicate the reasons for the pieces of the diagram.



1, 13	Definition of ω_X.
2, 5	Because $\mathcal{A}^X$ is infinitarily sound with respect to $[\![\cdot]\!]$ and $\mathcal{A}$
4,7,10,12	Because ω is monotonic w.r.t. $\leq_{\mathcal{A}}$.
3,6	Because $\xrightarrow[\mathcal{A}]{}$ and $\leq_{\mathcal{A}}$ commute.
8	Because $\xrightarrow[\mathcal{A}]{}$ is skew confluent up to $\leq_\omega$.
9	Because $\mathcal{A}^X$ is infinitarily complete with respect to $[\![\cdot]\!]$ and $\mathcal{A}$.
10	Because ω is monotonic w.r.t. $\xrightarrow[\mathcal{A}]{}$.
11	Because $\xrightarrow[[\![\cdot]\!]]{}$ and $\omega_{[\![\cdot]\!]}$ compute $[\![\cdot]\!]$.

7 Conclusions

The applicability of rewriting to model software systems demands the introduction of new properties which give an assurance of the correctness of the formalization. We have introduced two new notions: skew-confluence and ω-skew confluence. Whereas confluence allows one to reason about normal forms, the two new properties allow one to reason about infinite normal forms. Infinite objects arise in many different scenarios. For example, they appear in memory or in a source program. Therefore, properties that guarantee their uniqueness are important.

We have developed an abstract framework to be able to construct a new system while deriving properties from the old one. In particular, we have defined the notion of a finite basis, consisting of an ARS and a notion of information content with suitable properties. We have defined how to extend the ARS to its ideal completion. We have defined an extension as an ARS, whose objects have the ideal completion of the basis as their semantics and we have defined when such an extension is sound and complete. We have shown that a finite basis and a sound and complete extension give rise to a notion of infinite information content on the extension. Assuming that we have an extension which contains the computation of the semantics as a proper subset, we also gave a notion of finite information content.

We think that using all of this machinery is easier than giving a direct proof of uniqueness of infinite normal forms in the extension. With a direct proof, we have to deal with effects of non-confluence for most of the proof. When we use the extension approach, we have a lot more statements to prove, but most of them can be proven in a context where the rewrite system is confluent. For example, in the case of the lambda calculus we can use the fact that the lambda calculus is confluent while proving that the lambda calculus is a basis. Proving that the cyclic lambda calculus is infinitarily sound for the $\beta\circ$ rule is made easy by the fact that the rewrite system consisting of external substitution, external lift and $\beta\circ$ is confluent. Proving soundness for the other rules is somewhat harder, because this involves a property of non-confluent rules. However, it should be noted that what we really have to do is to prove that those rules preserve the unwinding. Hence, the result can be reused for other calculi. Finally, to prove completeness it is sufficient to prove a confluent subset complete.

References

1. Klop, J.W.: Term rewriting systems. In Abramsky, S., Gabbay, D., Maibaum, T., eds.: Handbook of Logic in Computer Science. Volume II., Oxford University Press (1992) 1–116
2. Terese: Term Rewriting Systems. Cambdrige University Press (2003)
3. Kennaway, J.R., Klop, J.W., Sleep, M.R., de Vries, F.J.: Infinitary lambda calculus. In: Proc. Rewriting Techniques and Applications, Kaiserslautern. (1995)
4. Kennaway, J.R., Klop, J.W., Sleep, M.R., de Vries, F.J.: Transfinite reductions in orthogonal term rewriting systems. Information and Computation **119** (1995)
5. Barendregt, H.P.: The Lambda Calculus: Its Syntax and Semantics. Revised edn. Volume 103 of Studies in Logic and the Foundations of Mathematics. Elsevier (1984)
6. Welch, P.: Continuous Semantics and Inside-out Reductions. In: λ-Calculus and Computer Science Theory, Italy (Springer-Verlag LNCS 37). (1975)
7. Lévy, J.J.: Réductions Correctes et Optimales dans le Lambda-Calcul. PhD thesis, Universite Paris VII (1978)
8. Wadsworth, C.: Semantics And Pragmatics Of The Lambda-Calculus. PhD thesis, University of Oxford (1971)
9. Ariola, Z.M., Blom, S.: Cyclic lambda calculi. In Abadi, M., Ito, T., eds.: Theoretical Aspects of Computer Software. Volume 1281 of Lecture Notes in Computer Science., Springer Verlag (1997) 77–106
10. Ariola, Z.M., Blom, S.: Skew confluence and the lambda calculus with letrec. Annals of Pure and Applied Logic **117** (2002) 95–168
11. Blom, S.: Lifting Infinite Normal Form Definitions from Term Rewriting to Term Graph Rewriting. In: TERMGRAPH 2002 - International Workshop on Term Graph Rewriting. (2002)
12. Barendregt, H., Brus, T., van Eekelen, M., Glauert, J., Kennaway, J., van Leer, M., Plasmeijer, M., Sleep, M.R.: Towards an intermediate language based on graph rewriting. In: Proc. Conference on Parallel Architecture and Languages Europe (PARLE '87), Eindhoven, The Netherlands, Springer-Verlag LNCS 259. (1987)
13. Sleep, M.R., Plasmeijer, M.J., van Eekelen, M.C.D.J., eds.: Term Graph Rewriting: Theory and Practice. John Wiley & Sons (1993)
14. Ariola, Z.M., Arvind: Properties of a first-order functional language with sharing. Theoretical Computer Science **146** (1995) 69–108
15. Wadsworth, C.: The Relation between Computational and Denotational Properties for Scott's D_∞-Models of the Lambda-Calculus. Theoretical Computer Science **5** (1976)
16. Ariola, Z.M., Klop, J.W.: Equational term graph rewriting. Fundamentae Informaticae **26** (1996) 207–240 Extended version: CWI Report CS-R9552.
17. Ariola, Z.M., Klop, J.W.: Lambda calculus with explicit recursion. Information and computation **139** (1997) 154–233
18. Ariola, Z.M., Felleisen, M., Maraist, J., Odersky, M., Wadler, P.: The call-by-need lambda calculus. In: Proc. ACM Conference on Principles of Programming Languages. (1995) 233–246
19. Ariola, Z.M., Felleisen, M.: The call-by-need lambda calculus. Journal of Functional Programming **7** (1997)
20. Ariola, Z.M., Blom, S.: Lambda calculi plus letrec. Technical Report IR-434, Department of Mathematics and Computer Science, Vrije Universiteit Amsterdam (1997)

21. Hennessy, M. In: Algebraic Theory of Processes. MIT Press (1988)
22. Corradini, A.: Term rewriting in CT_Σ. In Gaudel, M.C., Jouannaud, J.P., eds.: Proc. Colloquium on Trees in Algebra and Programming (CAAP '93), Springer-Verlag LNCS 668. (1993) 468–484
23. Kennaway, J.R., Klop, J.W., Sleep, M.R., de Vries, F.J.: The adequacy of term graph rewriting for simulating term rewriting. In Sleep, M.R., Plasmeijer, M.J., van Eekelen, M.C.D.J., eds.: Term Graph Rewriting: Theory and Practice, John Wiley & Sons (1993) 157–168
24. Ariola, Z.M.: Relating graph and term rewriting via Böhm models. Applicable Algebra in Engineering, Communication and Computing **7** (1996)