On Equal μ-Terms

Jörg Endrullis, Clemens Grabmayer[1], Jan Willem Klop, Vincent van Oostrom

Vrije Universiteit Amsterdam, Department of Computer Science, De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands

Universiteit Utrecht, ZENO Research Institute, Department of Philosophy Heidelberglaan 8, 3584 CS Utrecht, The Netherlands

Abstract

We consider the rewrite system R_μ with $\mu x.M \to_\mu M[x{:=}\mu x.M]$ as its single rewrite rule, where the signature consists of the variable binding operator commonly designated by μ, first order symbols, in this paper restricted to a binary function symbol F, and possibly some constant symbols. This kernel system denoting recursively defined objects occurs in several contexts, e.g. it is the framework of recursive types, with F as the function type constructor. For general signatures this rewriting system is widely used to represent and manipulate infinite regular trees.

The main concern of this paper is the convertibility relation for μ-terms as given by the μ-rule, in particular its decidability. This relation is sometimes called weak μ-equality, in contrast with strong μ-equality, which is given by equality of the possibly infinite tree unwinding of μ-terms. While strong equality has received much attention, the opposite is the case for weak μ-equality.

We present three alternative proofs for decidability of weak μ-equality. The first two proofs build upon an ingenious proof method of Cardone and Coppo. Prior to that, we prepare the way by an analysis of α-conversion. We then give a decidability proof in an 'α-free' way, essentially treating μ-terms as first-order terms, and next a proof in higher-order style, employing α-equivalence classes and viewing R_μ as a higher-order rewriting system.

The third decidability proof is again in the α-free way, exploiting the regular nature of the set of μ-reducts, enabling an appeal to the theory of tree automata.

We conclude with additional results concerning decidability of reachability, and upward joinability of μ-reduction, and of convertibility by α-free μ-reduction.

[1]This author was funded by the NWO-project *Realising Optimal Sharing* (ROS).

Email addresses: `joerg@few.vu.nl` (Jörg Endrullis), `clemens@phil.uu.nl` (Clemens Grabmayer), `jwk@cs.vu.nl` (Jan Willem Klop), `Vincent.vanOostrom@phil.uu.nl` (Vincent van Oostrom)

1. Introduction

Let us consider the infinite wave pattern as follows:

$$\cap\cup\cap\cup\cap\cup\cap\cup\cap\cup\cap\cup\cap\cup\cap\cup\cap\cup\cap\cup\cap\cdots$$

This pattern is actually suggested by the very origin of the Greek letter μ descending from the Egyptian hieroglyph 𓈖 and the Phoenician symbol 𐤌, meaning water. The question arises whether this pattern has a finite representation. Indeed it has, namely:

$$\mu x.\cap\cup x$$

with the μ-rule

$$\mu x.s \to s[\![x{:=}\mu x.s]\!]$$

Then we have:

$$\mu x.\cap\cup x \to \cap\cup\mu x.\cap\cup x \to \cap\cup\cap\cup\mu x.\cap\cup x \to \cap\cup\cap\cup\cap\cup\mu x.\cap\cup x \to \ldots$$

However, there are other representations of the same pattern, for example:

$$\mu x'.\cap\cup x'$$
$$\cap\mu y.\cup\cap y$$
$$\cap\cup\mu z.\cap\cup z$$
$$\mu w.\cap\cup\cap\cup w$$

Now the question arises when two such representations are the same (finitely)? Thus we arrive at the main endeavour of this paper.

This paper studies a particular orthogonal higher-order rewrite system that we will call R_μ, containing terms built from constants c, d, …, variables x, y, z, …, a binary function symbol F, and a higher order symbol μ allowing to construct terms such as $\mu x.\mathsf{F}(x,\mathsf{c})$. Everything in this paper generalizes to a more general first-order signature Σ, but for our present purposes, with recursive types as main application, the specific signature as mentioned will be assumed. There is a single rewrite rule, the μ-rule, which reads $\mu x.M \to M[x{:=}\mu x.M]$ (in meta-notation using the schematic variable M and meta-substitution). So we have the rewrite sequence:

$$\begin{aligned}\mu x.\mathsf{F}(x,x) &\to \mathsf{F}(\mu x.\mathsf{F}(x,x), \mu x.\mathsf{F}(x,x))\\ &\to \mathsf{F}(\mathsf{F}(\mu x.\mathsf{F}(x,x), \mu x.\mathsf{F}(x,x)), \mu x.\mathsf{F}(x,x)) \to \ldots\end{aligned}$$

where we could continue rewriting with as limit result the infinite binary branching tree of F's. However, in most of the paper we will only consider finite reduction sequences, and for such reduction sequences we have the confluence property, as a consequence of the general confluence theorem for orthogonal higher-order rewrite systems. For studies involving the use of μ-terms in infinitary term rewriting we refer to Inverardi and Zilli [**?**], and Corradini and Gadducci [**?**].

The rewrite system R_μ embodies the recursion principle in a most concentrated form, replacing the fixed point combinators Y as employed in λ-calculus. In fact, we can consider R_μ to be a subcalculus of λ-calculus, via the translation replacing μ by $Y \circ \lambda$ for some fixed point combinator Y; the μ-rule then becomes a derived rule. As we will remark below, this translation preserves unsolvability; meaningless terms are carried over to unsolvables in the λ-calculus.

Just as we can view λ-calculus from the finitary or the infinitary perspective, the latter leading to semantical notions such as Böhm trees, its subsystem R_μ can also be viewed finitary or infinitary. The infinitary perspective of R_μ is in fact rather well-known, as it pertains to tree unwinding semantics of e.g. recursive types [**?**]. The ensuing equality is called 'strong equality' in [**?**], holding when μ-terms M, N have the same possibly infinite tree unwinding. There are many proof systems for strong equality, and many algorithms for deciding strong equality.

However, the finitary aspects of R_μ are much less known. For instance, the basic notion of (finitary) convertibility using the μ-rule, was first studied in depth by Cardone and Coppo [**?**]. They provided a beautiful and ingenious proof method to show decidability, using standard reductions and a special purpose proof system, but as it turned out recently, the proof in [**?**] seems to be not entirely conclusive. Our proof below of decidability follows their proof strategy, but adds a crucial ingredient in the form of annotated or virtual μ-binders.

In order to explain the structure of our paper in more detail, we will elaborate now how we have approached the problem of α-conversion, and whether we perform μ-reduction modulo α, or not. It turns out that the main theme in many proofs of properties of μ-reduction resides in cycle detection, also named loop checking. In order to guarantee termination of loop checking, finiteness of the subterm closure $\mathsf{SC}(M)$ is exploited, where $\mathsf{SC}(M)$ is obtained from a μ-term M by alternating root reduction (unfolding) and subterm taking. Finiteness of the subterm closure guarantees the appearance of cycles, that is, the reoccurrence of terms that have been generated earlier. Our decidability proofs of weak μ-equality hinge upon such a concept of loop checking.

However, taking subterms of a term $\mu x.M$ is problematic, because a binder μx is severed, whereafter the previously bound x's are 'loosely dangling'. Indeed, the corresponding higher-order 'rule' $\mu x.M(x) \to M(x)$ violates the restriction on rules that the left-hand side and right-hand side must be be closed (the right-hand side is not), as found in higher-order rewriting. When we work with α-equivalence classes, it is not clear at all how to make sense of this situation, and this is the heart of the problem in the theory of μ-terms. To deal with this problem, various methods have been proposed and used, all depending on the precise way how α-conversion and α-equivalence classes are handled. Here we can distinguish three ways to define α-conversion inductively:

(I) The method used by Schroer [**?**]. This method consists in adopting new constants for the administration and management of bound variables, much like the 'de Bruijn indices'. In our paper this method will be employed in Section 8, where we treat decidability of weak μ-equality in the higher-order way, based on α-equivalence classes.

(II) The inductive definition employed by Kahrs [**?**]. In deciding α-equivalence $M \equiv_\alpha N$, we successively peel off binders from both sides M, N, keeping on the way a record, an ordered stack, of equations between the peeled off variables. Together with the other decomposition rule (F-decomposition), we then obtain a finite tree of subgoals, and the terminal equations of this tree can be simply decided by looking up in the stack, whether we have a success or a failure. These judgments at the leaves of the tree, percolating upwards, then determine the judgment of the original α-equivalence question. Interestingly, we will recognize aspects of this simple decision procedure for α-equivalence later on in the more complicated setting of deciding weak μ-equivalence (in particular the tree search, and the bookkeeping of discarded binders).

$$\dfrac{\varepsilon \vdash \mu xy.\mathsf{F}(x,y) = \mu yx.\mathsf{F}(y,x)}{\dfrac{x = y \vdash \mu y.\mathsf{F}(x,y) = \mu x.\mathsf{F}(y,x)}{\dfrac{x = y,\ y = x \vdash \mathsf{F}(x,y) = \mathsf{F}(y,x)}{\dfrac{x = y,\ y = x \vdash x = y}{x \neq y \qquad y \neq x \quad \dfrac{x = y \vdash x = y}{}} \qquad \dfrac{x = y,\ y = x \vdash y = x}{}}}}$$

Figure 1: A proof of α-equivalence in the style of Kahrs [**?**].

(III) The definition of α-equivalence as $\equiv_\alpha := (\leftarrow_\alpha \cup \rightarrow_\alpha)^*$, the convertibility relation generated by single-step α-renaming $\rightarrow_\alpha$.

In Hendriks and van Oostrom [**?**] these three approaches are proved to be equivalent. This does not mean that their use is interchangeable without more; depending on the problem context one or more of methods (I)–(III) may be preferable and the right one to use.

So there are various ways to deal with α-equivalence. But the basic strategic choice is to commit oneself either to treat matters α-free, and clearly separating α-renaming and μ-reduction; or to work all the way with α-equivalence classes and a corresponding notion of μ-reduction. The advantage of the first commitment is that if we have succeeded in eliminating the concerns to deal with α-conversion—and below we will show how in detail this can be done—then we are for all purposes actually in a first-order setting[3]. This first commitment we have adopted in our first decidability proof for weak μ-equality, extending the proof method of Cardone and Coppo, see Section 7. It is also adopted in our third decidability proof, which uses tree automata techniques, made possible by our 'first-order rendering' of μ-terms, see Section 9.

[3]The situation is similar as in the case of weak β-reduction where one does not reduce under a λ-binding: there is no need for α-conversion, and hence one obtains the simplicity of a first-order setting.

Our second decidability proof, in Section 8, is under the regime of the second commitment. It still follows the Cardone–Coppo proof strategy, but now in a pure higher-order setting, so that μ-terms are α-equivalence classes. The advantage is that then we do not have to deal with reduction modulo α; this is seamlessly integrated in the notion of reduction in higher-order rewriting. Here a certain disadvantage is that the higher-order setting is less concrete, and also less well-known.

Summing up, we advocate the adagium that α-freeness entails a first-order setting. In our paper we have endeavored to put light on the decidability question for weak μ-equality from both paradigm perspectives, α-free with the ensuing first-order setting, and higher-order, with the ensuing built-in α-equivalence. This is the rationale for the different proofs that we have developed.

An interesting meta-observation about the inclusion of the core system R_μ in the full λ-calculus, is that while in the λ-calculus 'every interesting property is undecidable', as validated for many instances by Scott's theorem, in the core system R_μ by contrast 'every interesting property is decidable'. To substantiate this informal slogan, we have included proofs of decidability of reachability $\twoheadrightarrow_{\mu/\alpha}$, and also of unsolvability, and of upward-joinability $\uparrow_{\mu/\alpha}$.

In this paper we will sometimes refer to the rewrite system R_μ as 'μ-calculus', in analogy with the λ-calculus, to facilitate our way of speaking. As a caveat we point out that we do not intend to evoke associations with well-known important and more expressive μ-calculi such as the modal μ-calculus or Parigot's $\lambda\mu$-calculus (see also [**?**]). It will be clear that our paper concentrates on syntactic aspects of R_μ, rather than semantic studies of fixed point rules such as by Fiore and Plotkin [**?**] on FPC.

2. Preliminaries

We start with setting up the notations. We also introduce the notion of meaningless μ-terms, and the Böhm Tree, or infinite tree unwinding, of a μ-term.

Definition 1 (**μ-terms**). *Ter*(μ) is the set of μ-terms over the first-order signature $\Sigma = \{\mathsf{c}, \mathsf{d}, \ldots\} \cup \{\mathsf{F}\}$ with constants c, d, ..., and binary function symbol F, more precisely:

(i) $x, y, z, \ldots \in \mathit{Ter}(\mu)$ (*variables*),

(ii) $\mathsf{c}, \mathsf{d}, \ldots \in \mathit{Ter}(\mu)$ (*constants*), and

(iii) $M, N \in \mathit{Ter}(\mu) \implies \mathsf{F}(M, N) \in \mathit{Ter}(\mu)$, and

(iv) $M \in \mathit{Ter}(\mu)$ and x a variable $\implies \mu x.M \in \mathit{Ter}(\mu)$.

Note that the main application of μ-terms is for recursive types, where the type constructor is usually written as $\to$, as in [**?**]. To avoid overloading with our reduction step notation we use the binary F.

By the *size* $\mathrm{size}(M)$ of a μ-term M we mean the number of symbols and bindings in M, or more precisely, the number of generation steps of M according to Definition 1. The definition of free and bound variables is analogous as in the

λ-calculus. By $\mathsf{FV}(M)$ (by $\mathsf{BV}(M)$) we denote the set of variables that occur free (occur bound) in M, that is, we define inductively:

$$\begin{array}{rlrl}
\mathsf{FV}(x) = \{x\} & \text{variables } x & \mathsf{BV}(x) = \emptyset & \text{variables } x \\
\mathsf{FV}(\mathsf{c}) = \{\varepsilon\} & \text{constants } \mathsf{c} & \mathsf{BV}(\mathsf{c}) = \emptyset & \text{constants } \mathsf{c} \\
\mathsf{FV}(\mu x.M) = \mathsf{FV}(M) \setminus \{x\} & & \mathsf{BV}(\mu x.M) = \mathsf{BV}(M) \cup \{x\} & \\
\mathsf{FV}(\mathsf{F}(M,N)) = \mathsf{FV}(M) \cup \mathsf{FV}(N) & & \mathsf{BV}(\mathsf{F}(M,N)) = \mathsf{BV}(M) \cup \mathsf{BV}(N) &
\end{array}$$

We write $\mathcal{P}os(M) \subseteq \mathbb{N}^*$ for the set of positions of a μ-term M, defined by:

$$\begin{aligned}
\mathcal{P}os(x) = \mathcal{P}os(\mathsf{c}) &= \{\varepsilon\} \qquad \text{for variables } x \text{ and constants } \mathsf{c} \\
\mathcal{P}os(\mu x.M) &= \{\varepsilon\} \cup \{1p \mid p \in \mathcal{P}os(M)\} \\
\mathcal{P}os(\mathsf{F}(M,N)) &= \{\varepsilon\} \cup \{1p \mid p \in \mathcal{P}os(M)\} \cup \{2p \mid p \in \mathcal{P}os(N)\}
\end{aligned}$$

For $p \in \mathcal{P}os(M)$, we write $M|_p$ for the subterm of M at position p:

$$M|_\varepsilon = M \qquad \mu x.M|_{1p} = M|_p \qquad \mathsf{F}(M,N)|_{1p} = M|_p \qquad \mathsf{F}(M,N)|_{2p} = N|_p$$

We write $M \to^p N$ if there is a step $M \to N$ such that $p \in \mathcal{P}os(M)$ is the position of the contracted redex. We write $p < q$ if the sequence of numbers representing position p is a prefix of that of q. So p is *above* q.

We use $=$ to denote syntactic equality of μ-terms, and $\equiv_\alpha$ for equality modulo α-conversion. We write $M[\![x{:=}N]\!]$ for *α-converting* substitution, which we denote using double brackets in order to distinguish it from *α-free* (potentially variable-capturing) substitution that we denote by $M[x{:=}N]$. Formally, for α-converting substitution $M[\![y{:=}N]\!]$ we employ Curry's definition (see e.g. [**?** , Definition C.1]) with μ for abstraction and F for application:

$$\begin{array}{rcl}
x[\![x{:=}N]\!] & := & N \\
y[\![x{:=}N]\!] & := & y \qquad (\text{if } y \neq x) \\
\mathsf{F}(M_1, M_2)[\![x{:=}N]\!] & := & \mathsf{F}(M_1[\![x{:=}N]\!], M_2[\![x{:=}N]\!]) \\
(\mu x.M_1)[\![x{:=}N]\!] & := & \mu x.M_1 \\
(\mu y.M_1)[\![x{:=}N]\!] & := & \mu z.(M_1[\![y{:=}z]\!])[\![x{:=}N]\!] \qquad (\text{if } y \neq x) \\
& & \quad \text{where } z = y \text{ if } x \notin \mathsf{FV}(M_1) \text{ or } y \notin \mathsf{FV}(N), \\
& & \quad \text{else } z \text{ is the first variable in the ordered} \\
& & \quad \text{sequence of variables that does neither} \\
& & \quad \text{occur in } M_1 \text{ nor in } N
\end{array}$$

Furthermore, α-free substitution $M[x{:=}N]$ is formally defined by analogous clauses but with the difference that the last one is replaced by the following simpler clause:

$$(\mu y.M_1)[x{:=}N] \quad := \quad \mu y.M_1[x{:=}N] \qquad (\text{if } y \neq x)$$

As stated by the proposition below, α-free substitutions that do not lead to the capture of variables coincide with α-converting substitutions. Variable-capture does not occur in a substitution $M[x{:=}N]$ if and only if N *is free for* x in M: there is no free occurrence of x in M below a binding μy such that $y \in \mathsf{FV}(N)$.

Proposition 2. *If N is free for x in M, then $M[\![x{:=}N]\!] = M[x{:=}N]$.*

Definition 3 (**μ-rewrite relation**)**.** The *μ-rule* (or *unfolding rule*) is:

$$\mu x.M \to M[x{:=}\mu x.M] \qquad (\text{if } \mu x.M \text{ is free for } x \text{ in } M)$$

This rule induces the *α-free μ-rewrite relation* $\to_\mu$ on μ-terms (note that, as usual, the rewrite rule may be applied within contexts). By extending $\to_\mu$-steps on both sides by α-conversion steps, $\to_{\mu/\alpha} := \equiv_\alpha \cdot \to_\mu \cdot \equiv_\alpha$, the *$\mu$-rewrite relation (modulo α-conversion)* on μ-terms is obtained.

Definition 4 (**Weak μ-equality**)**.** The relation *weak μ-equality* on μ-terms is defined as $=_{\mu/\alpha} := (\leftarrow_{\mu/\alpha} \cup \to_{\mu/\alpha})^* \cup \equiv_\alpha$.[4] By $=_\mu$ we denote the convertibility relation $(\leftarrow_\mu \cup \to_\mu)^*$ with respect to the α-free μ-rewrite relation $\to_\mu$.

Meaningless μ-terms. To illustrate some of the preliminary notations and notions we consider 'meaningless' μ-terms that are analogous to meaningless terms in λ-calculus (see e.g. [**? ?**]) of a certain kind, namely the 'root-active' ones: such λ-terms enable infinite rewrite sequences with infinitely many $\to_\beta$-reduction steps at the root position. These λ-terms are 'meaningless' in the sense that no information can be obtained from them because their root does not stabilize eventually.

This leads us to the following definition.

Definition 5 (**root normal forms, and root-active μ-terms**)**.** A *root reduction step* is a step in which the redex contracted is situated at the root. (So a root step does not occur in a non-trivial context.[5]) A *root-$\to_{\mu/\alpha}$-step* is a $(\equiv_\alpha \cdot \to_\mu \cdot \equiv_\alpha)$-step in which the $\to_\mu$-step is a root reduction step.

A *root normal form* of a μ-term is a normal form with respect to root-$\to_{\mu/\alpha}$-steps. A μ-term M is called *root-active* if it admits an infinite $\to_{\mu/\alpha}$-rewrite sequence that contains infinitely many root-$\to_{\mu/\alpha}$-steps.

Proposition 6. *(i) μ-terms in root normal form are either variables, or constants, or of the form $\mathsf{F}(A, B)$ for some μ-terms A, B.*

(ii) Let M be a μ-term. The following four statements are equivalent:

(a) M is root-active.

(b) M admits an infinite $\to_{\mu/\alpha}$-reduction consisting only of root-$\to_\mu$-steps.

(c) M does not reduce by $\to_{\mu/\alpha}$-steps (hence neither by root-$\to_{\mu/\alpha}$-steps) to a root normal form.

(d) $M \equiv_\alpha \mu x_n \dots x_1.x_k$ for some $n \geq k \geq 1$ and variables $x_1, \dots, x_n$.

[4] Note that $=_{\mu/\alpha}$ coincides with the convertibility relation $(\leftarrow_{\mu/\alpha} \cup \to_{\mu/\alpha})^*$ with respect to $\to_{\mu/\alpha}$: this is because every α-renaming step can be mimicked by a $\leftarrow_{\mu/\alpha}$-step that introduces a vacuous μ-binding at the root of the term, followed by a $\to_{\mu/\alpha}$-step that removes the vacuous μ-binding again and carries out the α-renaming step. Furthermore, $=_{\mu/\alpha}$ also coincides with $(\leftarrow_\mu \cup \equiv_\alpha \cup \to_\mu)^*$, the convertibility relation with respect to $\to_\mu$ modulo α-equivalence.

[5] A context is a μ-term with one occurrence of a hole $\Box$.

Let us denote, for $n \geq k \geq 1$, a root-active μ-term of the form $\mu x_n \dots x_1.x_k$ (with some variables $x_1, \dots, x_n$) by $\langle n, k \rangle$.

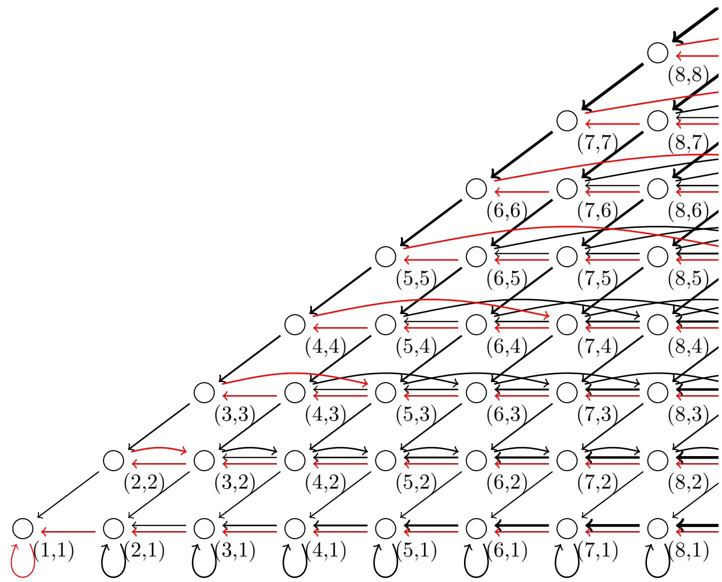


Figure 2: Reduction graph for cyclic μ-terms.

Remark 7. The following $\to_{\mu/\alpha}$-steps (provided to us by Felice Cardone, personal communication) comprise a complete picture of the $\to_{\mu/\alpha}$-steps that are possible between root-active μ-terms:

$$
\begin{array}{lll}
\langle 1, 1 \rangle \to_{\mu/\alpha} \langle 1, 1 \rangle & & \\
\langle n, k \rangle \to_{\mu/\alpha} \langle n-1, k \rangle & & \text{if } n > k \\
\langle n, k \rangle \to_{\mu/\alpha} \langle n-1, k-1 \rangle & & \text{if } k > 1 \\
\langle n, k \rangle \to_{\mu/\alpha} \langle n+k-1, k \rangle & &
\end{array}
$$

The ARS (abstract reduction system) constituted by these steps is displayed in Figure 2. The root reduction steps are colored red in the picture. It is easy to show that all root-active μ-terms are $\to_{\mu/\alpha}$-convertible to the simplest root-active μ-term $\langle 1, 1 \rangle = \mu x.x$. So, in contrast with the λ-calculus, the property 'root active' is decidable.

Hence Scott's theorem for the λ-calculus fails for the μ-calculus: Scott's theorem states that if a non-trivial property P of λ-terms is closed under β-convertibility, then P and $\neg P$ are undecidable properties. Here 'non-trivial' means: P and $\neg P$ are non-empty.

The following exercise is worth noting: For its formulation, let Φ be the translation from μ-terms to λ-terms that replaces a μ by $Y \circ \lambda$ (or simply $Y\lambda$

assuming the terms are written as words using the usual bracket convention), where Y is an arbitrary fixed point combinator, leaving the other symbols in the μ-terms unchanged.

Exercise 8. Show that: For all μ-terms M, M is root-active if and only if $\Phi(M)$ is root-active. Note that root-active λ-terms correspond to 'unsolvable' terms in λ-calculus, where root-steps are defined as steps at depth 0 with respect to the depth measure (001) (see [**?** , 12.10]).

Does also the following hold: for all μ-terms M, M is root-active if and only if $\Phi(M)$ is root-active with respect to the depth measure (111)?

Definition 9 (**Infinite tree unfolding/Böhm Trees of μ-terms**)**.** This definition is entirely analogous to the coinductive definition of Böhm Trees (BT) in the λ-calculus and for term rewriting systems (see further [**?** , sec. 2, (vii), and fn. 3] and [**?**]), except that we replace 'head reduction' by 'root reduction'. So, for $M \in \mathit{Ter}(\mu)$:

$$\begin{aligned}
\mathsf{BT}_\mu(M) &= \bot && \text{if } M \text{ has no root normal form}\\
\mathsf{BT}_\mu(\mu x.M) &= \mathsf{BT}_\mu(M[\![x{:=}\mu x.M]\!]) && \text{if } \mu x.M \text{ has a root normal form}\\
\mathsf{BT}_\mu(x) &= x && \text{for a variable } x\\
\mathsf{BT}_\mu(\mathsf{c}) &= \mathsf{c} && \text{for a constant } \mathsf{c}\\
\mathsf{BT}_\mu(\mathsf{F}(M,N)) &= \mathsf{F}(\mathsf{BT}_\mu(M), \mathsf{BT}_\mu(N))
\end{aligned}$$

Note that the first clause applies for root-active (meaningless) μ-terms of the form (n,k) discussed above. In the last clause we have borrowed term notation to denote a possibly infinite term tree, in a self-explaining way. In this case we peel off the root normal context $\mathsf{F}(\Box,\Box)$ and coinductively append the BT_μ's of M and N. Later on we will define the notion of the set of generators of a BT_μ; the present M, N in the last clause are among these generators. But we give this definition only after introducing the so-called subterm closure.

Definition 10 (**Strong μ-equality**)**.** *Strong μ-equality* $=_{\mathsf{BT}_\mu}$ is the equivalence relation on μ-terms that is induced by equality of the Böhm trees (or tree unfoldings): for μ-terms M, $M =_{\mathsf{BT}_\mu} N$ holds if $\mathsf{BT}_\mu(M) = \mathsf{BT}_\mu(N)$.

We remark that weak μ-equality implies strong μ-equality. The converse does not hold, as the following examples illustrate. Both representations

$$\mu x.{\cap}{\cup}x \qquad\qquad \text{and} \qquad\qquad \mu x.{\cap}{\cup}{\cap}{\cup}x$$

represent the same infinite wave pattern ${\cap}{\cup}{\cap}{\cup}{\cap}{\cup}{\cap}{\cup}{\cap}{\cup}{\cap}\cdots$. Nevertheless, they are not equal with respect to weak μ-equality, that is, they are not convertible by a finite sequence of folding, unfolding and α-renaming steps.

The following two μ-terms can be viewed as encodings into μ-terms over the signature we consider of the two μ-terms from the example above:

$$M = \mu x.\mathsf{F}(\mathsf{c}, x) \qquad\qquad \text{and} \qquad\qquad N = \mu x.\mathsf{F}(\mathsf{c}, \mathsf{F}(\mathsf{c}, x))$$

The μ-terms M and N are equal with respect to strong μ-equality: both unfold in infinitely steps to the Böhm tree $\mathsf{F}(\mathsf{c}, \mathsf{F}(\mathsf{c}, \mathsf{F}(\mathsf{c}, \ldots)))$. The terms are, however, not equal with respect to weak μ-equality.

3. Avoiding α-conversion in μ-calculus

We start with showing that α-conversion can be avoided along μ-reduction from a μ-term M by choosing the variables of binders in M to be distinct and distinct from the free variables. We will call such μ-terms *simple*.

Since the phenomenon of α-conversion plays a rôle in any higher-order calculus, we set the stage by showing that α-conversion cannot be avoided in the prototypical higher-order calculus: the λ-calculus with β-reduction [**?**].

Example 11 (**α-conversion in λ-calculus**)**.** We consider the following reduction in which α-conversion cannot be avoided, despite all λs binding distinct variables initially:

$$(\lambda x.xx)\lambda y.\lambda z.yz \to^1 (\lambda y.\lambda z.yz)\lambda y.\lambda z.yz \to^2 \lambda z.(\lambda y.\lambda z.yz)z \to^3 \lambda z.\lambda z'.zz'$$

In the end the bound variable z *must* be renamed to z' in order to avoid capturing the free z by λz. Without such *self-capture* α-conversion can be avoided. A particular example where α-conversion can be avoided are developments.

We show that, unlike what is the case in the λ-calculus, in the μ-calculus α-conversion can *always* be avoided. Intuitively, μ-reductions share the property with β-developments that no new redexes can be created along a reduction. The latter property was shown in [**?**] to entail the absence of self-capturing so that α-conversion can be avoided along β-developments.[6] Here we adapt, and instantiate, that proof to μ-reduction.

Definition 12 (**Self-capture**)**.** For a given term M, a position p of a binder μx *binds* (*captures*) a position $p1q$ of a variable y, if the occurrence of y at position q in $M|_{p1}$ is free (in $M|_{p1}$) and $x = y$ ($x \neq y$).

A *chain* is a series of connected links, where a *link* is a binding or a converse capturing. More precisely, a chain is a sequence of positions $p_1, \ldots, p_n$ such that for every $1 \leq i < n$ we have that p_i binds p_{i+1}, or p_i is captured by p_{i+1}. We frequently identify the positions p_i with the symbol occurrences at position p_i.

A chain is *self-capturing* if it starts with x and ends with μx, for some x. A μ-term is *self-capture-free* (*capture-avoiding*) if there is no self-capturing chain (*scc*).

Example 13. The μ-term $\mu x.\mathsf{F}(y, \mu y.x)$ contains the chain $11, \varepsilon, 121, 12$ connecting y to μy via μx and x, which is self-capturing. Indeed, α-conversion is needed for contracting the outermost μ-redex yielding $\mathsf{F}(y, \mu y'.\mu x.\mathsf{F}(y, \mu y.x))$.

[6] In the terminology of [**?**] absence of self-capturing is expressed as *holding* (being connected by a chain in the sense of Definition 12) being *parting* (never relating two residuals of the same redex), and established for all combinatory reduction systems [**?**] (second-order term rewriting systems).

In a chain binding and converse capturing alternate. Chains are finite since if p binds o which is captured by q,[7] then $p < q < o$. Links are not self-capturing and self-capture-freeness is closed under taking subterms.

The following proposition and its proof are illustrated by Figure 4.

Proposition 14. *Let $\mu x.M$ contain no scc. Suppose that a free occurrence of x in M is chained to a binder μy. Then y does not occur free in M.*

Proof. Otherwise there is a chain y,μx,x,$\ldots$,μy in $\mu x.M$, contradicting self-capture-freeness. $\square$

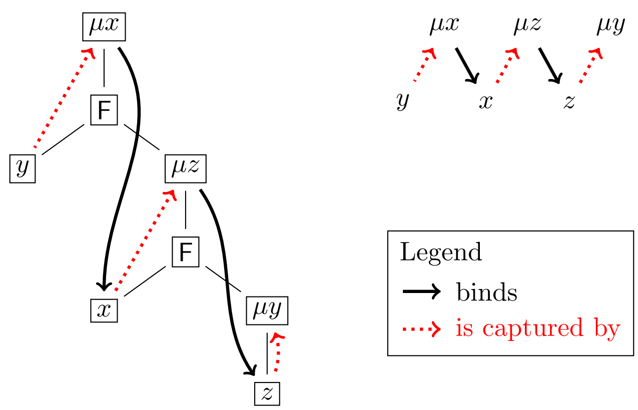


Figure 3: A self-capturing chain of length 5 for the term $\mu x.\mathsf{F}(y, \mu z.\mathsf{F}(x, \mu y.z))$.

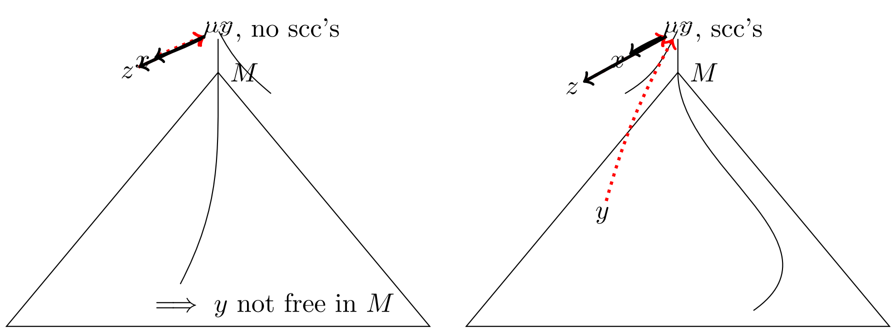


Figure 4: Assumptions of Proposition 14 (left); Proof of the proposition (right).

[7] In the terminology of [?], on which [?] was based, p *grips* q. That is, we decompose gripping using the more elementary notion of link, as a binding link followed by a converse capturing link, allowing also to deal uniformly with free variables.

Example 15. Consider the following example in Figure 3:

$$M = \mu x.\mathsf{F}(y, N)$$
$$N = \mu z.\mathsf{F}(x, P)$$
$$P = \mu y.z$$

The chain displayed in the right part of that figure shows the alternation of links and converse capturings. Note however, that the binders μx, μz, μy at the peaks of this zigzag figure are actually not horizontal, but 'sliding down' along a branch of the term, as one sees in the left part of the figure, and also in Figure 4. Note that it holds: $M \to_\mu^\epsilon \mathsf{F}(y, \mu z.\mathsf{F}(M, P))$. The last term contains two μz's, the displayed one, and the one in M. Contracting the displayed μ-redex without α-renaming causes y to be captured.

By the following lemma, redexes for which the assumptions of Proposition 14 hold can be contracted by means of α-free substitution.

Lemma 16 (**α-free**). *Suppose that if there is a chain from a free occurrence of x in M to a binder μy, then y is not free in N. Then $M[\![x{:=}N]\!] = M[x{:=}N]$.*

Proof. By induction and cases on the formation of the term substituted in, the only interesting case being when the term is an abstraction $\mu y.M$ with $x \neq y$. Then either x is not free in M or else by the assumption, y is not free in N, hence

$$\begin{aligned}(\mu y.M)[\![x{:=}N]\!] &= \mu y.M[\![y{:=}y]\!][\![x{:=}N]\!] \\ &= \mu y.M[\![x{:=}N]\!] \\ &= \mu y.M[x{:=}N] \\ &= (\mu y.M)[x{:=}N]\end{aligned}$$

where the induction hypotheses is applied in the third equality. □

Having shown that self-capture-freeness entails that α-conversion can be avoided, we next show that it is preserved by μ-reduction.

Lemma 17 (**Self-capture-free preservation**). *Self-capture-freeness is preserved by μ-reduction.*

Proof. The proof is by tracing back positions, paths, links, and chains. The (dynamic) *trace* relation $[\![\phi\rangle\!\rangle$, simply denoted by $\triangleright$ if the step ϕ is clear from the context, cf. [**?** , Section 8.6.1], between the positions of the source and target of a reduction step ϕ contracting a μ-redex $\mu z.M$ at position o:

$$C[\mu z.M]_o \to^o C[M[\![z{:=}\mu z.M]\!]]_o$$

is defined by the following clauses:

$$\begin{array}{rccll} \text{(context)} & p & \triangleright & p & \text{if } o \nleq p \\ \text{(body)} & o1p & \triangleright & op & \text{if } o1p \text{ not bound by } o \\ \text{(copy)} & op & \triangleright & oqp & \text{if } o1q \text{ bound by } o \end{array}$$

In the remainder of the proof we use primed variables, e.g. p', to range over positions in the target of ϕ, indicating the unique positions they trace back to, their *origins*, by unpriming, e.g. p, and employ that the symbols at these positions in the respective terms are the same. We claim that for every link connecting p', q' there is chain connecting p, q in the same direction. Since the claim entails that chains trace back to chains, the result follows. To prove the claim assume w.l.o.g. that p' is the position of a binder μx and q' the position of a variable y, and distinguish cases on the relative positions of $p' < q'$.

If both p', q' are in the same component (context, body, or copy) then the path between them is in the same component as well, and the origin of such a path is a path between their origins p,q.

If p' is in the context and q' in the body then the origin of the path between them is the same path with 1 inserted at o, and we conclude using that $y \neq z$ by the condition on the body case.

If p' is in the context and q' in a copy at o' then q' must be free in $o'1$, and the origin of the path is the same path with the path from o up to o' removed, from which we conclude.

If p' is in the body and q' in a copy at o' then q' must be free in $o'1$, and we conclude using that there is a chain connecting p to q via first the binder corresponding to the copy, i.e. o' with 1 inserted at o, and next o. □

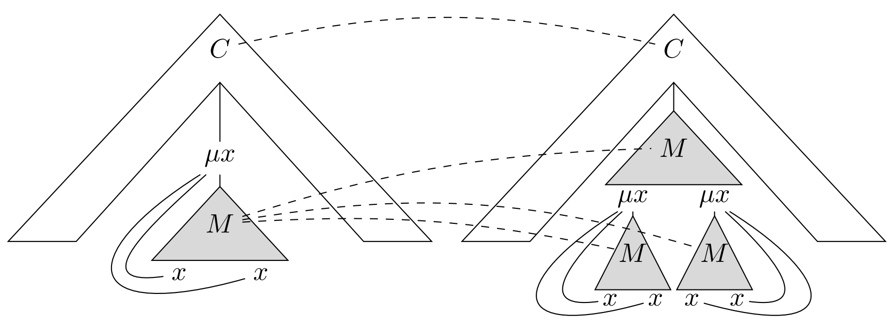


Figure 5: Tracing symbols along a step.

Theorem 18 (α-free μ). *For any μ-term M, there is an α-equivalent term M' such that any μ-reduction from M can be lifted to a μ-reduction from M' contracting redexes using α-free substitution only.*

Proof. Let M' be obtained by renaming the binders on each chain in M so as to be distinct, and distinct from the free variables, e.g. by choosing all of them to be distinct. Then M' is self-capture-free and we conclude by the previous lemmata. □

The theorem justifies treating μ-terms as if they were first-order terms, as we will do with the exception of Section 8.

We proceed with showing a substitution lemma for α-free substitution.

Remark 19. The (μ-calculus variant of the) standard Substitution Lemma [**?** , Lemma 2.1.16] will not do for our purposes as it implicitly makes use of α-conversion to guarantee a stronger invariant, the so-called variable convention, than can be guaranteed in this paper. Typically, a single μ-step both copies and nests binders, violating the variable convention.

Lemma 20 (**α-free substitution**)**.** *$M[x{:=}N][y{:=}L] = M[y{:=}L][x{:=}N[y{:=}L]]$, under the condition of Lemma 16, $x \neq y$, and x not free in L.*

Proof. By induction and cases on the structure of M.

(z) If $x = z$, then both sides of the equality yield $N[y{:=}L]$, using $x \neq y$.

If $y = z$, then both sides of the equality yield L using $x \neq y$, and x not free in L to establish $L[x{:=}\dots] = L$.

Otherwise, both sides of the equality yield z.

($\mu z.M_1$) If $x = z$, then both sides of the equality yield $\mu x.M_1[y{:=}L]$, using $x \neq y$.

If $y = z$, then either x is not free in M_1 and then both sides of the equality yield M, or else by the assumption, y is not free in N and both sides yield $\mu y.M_1[x{:=}N]$.

Otherwise, $\mu z.M_1[x{:=}N][y{:=}L] = \mu z.M_1[y{:=}L][x{:=}N[y{:=}L]]$ and we conclude by the induction hypothesis for M_1.

($\mathsf{F}(M_1, M_2)$) Then the statement follows again from the induction hypothesis. ☐

4. The subterm closure and its finiteness

In this section we exhibit a key notion in the theory of μ-terms and various decidability proofs, namely the notion of subterm closure, as well as the corresponding key lemma stating that this subterm closure is finite. Both the notion and its property were originally conceived by Brandt and Henglein [**?**].

Note: All terms in this section are assumed to be capture-avoiding, which implies that $\to_\mu$-rewrite sequences never lead to μ-terms in which μ-redexes could only be contracted after suitable renamings of bound variables.

Consider the $\mu\pi$-calculus obtained by adjoining the following root-reduction rule schemata π:

$$\begin{aligned} \mathsf{F}(M_1, M_2) &\to M_i \quad \text{for } i \in \{1,2\} && (\mathsf{F}\text{-}\textit{decomposition}) \\ \mu x.M &\to M && (\mu\text{-}\textit{decomposition}) \end{aligned}$$

to the μ-calculus restricted to root-reduction:

$$\mu x.M \to M[x{:=}\mu x.M] \qquad (\textit{root } \mu\text{-}\textit{reduction})$$

These reductions are not allowed in a context, and they pertain to μ-terms taken literally, not modulo α. Since a π-step projects onto a subterm, it may only shorten chains. Hence the above results carry over immediately, i.e. α-conversion is not needed in the $\mu\pi$-calculus either.

In this section, $\overset{\varepsilon}{\to}_{\mu\pi}$ will be used to denote $\mu\pi$-root-reduction, i.e. $\mu\pi$-reduction at position ε, the root.

Warning: The present reduction $\overset{\varepsilon}{\twoheadrightarrow}_{\mu\pi}$ should not be confused with $\twoheadrightarrow_{\mu/\alpha}$, a sequence of general μ-reduction steps, even though both relations overlap; $\overset{\varepsilon}{\twoheadrightarrow}_{\mu\pi}$ may involve some root-μ-steps, just as $\twoheadrightarrow_{\mu/\alpha}$.

Definition 21. The *subterm closure* $\mathsf{SC}(M)$ of a μ-term M is defined as the set of reducts of M with respect to $\overset{\varepsilon}{\to}_{\mu\pi}$:

$$\mathsf{SC}(M) = \{N \mid M \overset{\varepsilon}{\twoheadrightarrow}_{\mu\pi} N\}\ .$$

Example 22. We consider the cyclic μ-term $M \equiv \mu xyz.y$, i.e. the unsolvable (3,2) in Figure 1. The subterm closure $\mathsf{SC}(M)$ contains 5 elements; see Figure 6. Note that $\mu xyz.y \equiv_\alpha \mu zyz.y$, that is, we take μ-terms literally without α-conversion.

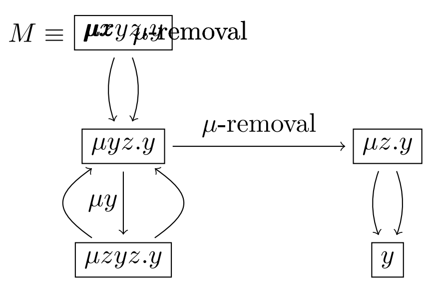


Figure 6: The subterm closure of $\mu xyz.y$.

Example 23. Let $M \equiv \mu x.\mathsf{F}(x, \mu y.\mathsf{F}(x, y))$. Then $\mathsf{SC}(M)$ has 11 elements, see Figure 7. The reduction relation $\overset{\varepsilon}{\to}_{\mu\pi}$ is not SN (terminating). Nevertheless, the set of generated terms is always finite. This finiteness will be crucial in the subsequent sections.

Before giving the detailed proof of finiteness of the subterm closure, let us give some quick intuition of the finiteness. Let $M \equiv \mu x.\mathsf{F}(x, \mu y.\mathsf{F}(x, y))$ as in Example 23, and consider the following $\mu\pi$-reduction starting from M:

$$\begin{aligned}
& \mu x.\mathsf{F}(x, \mu y.\mathsf{F}(x, y)) \\
\to_\mu\ & \mathsf{F}(\underline{\mu x.\mathsf{F}(x, \mu y.\mathsf{F}(x, y))}, \mu y.\mathsf{F}(\underline{\mu x.\mathsf{F}(x, \mu y.\mathsf{F}(x, y))}, y)) \\
\to_\pi\ & \mu y.\mathsf{F}(\underline{\mu x.\mathsf{F}(x, \mu y.\mathsf{F}(x, y))}, y) \\
\to_\mu\ & \mathsf{F}(\underline{\mu x.\mathsf{F}(x, \mu y.\mathsf{F}(x, y))}, \underline{\mu y.\mathsf{F}(\mu x.\mathsf{F}(x, \mu y.\mathsf{F}(x, y)), y)}) \\
\to_\pi\ & \underline{\mu y.\mathsf{F}(\mu x.\mathsf{F}(x, \mu y.\mathsf{F}(x, y)), y)}
\end{aligned}$$

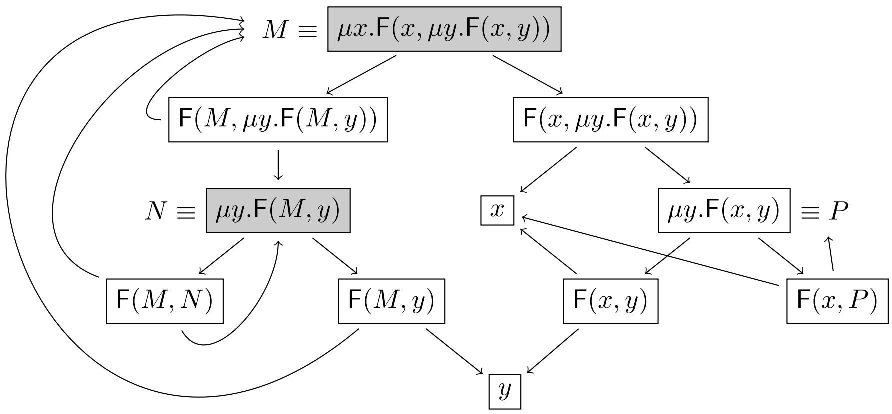


Figure 7: The subterm closure of $\mu x.\mathsf{F}(x, \mu y.\mathsf{F}(x, y))$. (For the shaded boxes see the explanation below Proposition 37 in Section 6.)

The underlined subterms[8] indicate terms that are substitutions created by a previous μ-step at the root. That is, the underlined terms have been encountered before in the reduction. Now observe that the size of the non-underlined part is decreasing with every $\mu\pi$-step. It decreases until size 0, that is, the whole term is underlined. To construct the subterm closure we only need to consider repetition-free $\mu\pi$-reductions. The above argument of the decreasing size of the non-underlined context can be formalized to a termination proof for repetition-free $\mu\pi$-reductions. Then since $\mu\pi$-reductions are finitely branching, we conclude finiteness of the subterm closure by Königs Lemma.

Remark 24. We can also prove the finiteness of the subterm closure $\mathsf{SC}(M)$ by the following appeal to RPO, recursive path order. This can be seen as a more refined version of underlining. See Figure 8 where we have shown $\mathsf{SC}(M)$ as in Figure 7, now enhanced with labels yielding the RPO finiteness proof. The notation M^0 means that the head symbol of M (where M occurred 'earlier' in the figure), has label 0, but the other labels are unchanged. Likewise for N^0 and P^0.

(i) Label each x, y, $\ldots$, F, μ in M with a natural number $n \in \mathbb{N}$, such that higher occurrences have a higher label. A labeling with that property is called 'decreasing'.

(ii) Second, we label the $\mu\pi$-rules. For F-decomposition and μ-decomposition we take the obvious definition, where the subterm resulting after the step keeps the labels it had before the step.

[8] Basically, this underlining argument was also suggested in personal correspondence by Cardone and Coppo.

(iii) However, for the root-μ-reduction rule we have the crux of the definition:

$$\mu x^{n+1}.M \to M[x{:=}\mu x^0.M] \tag{1}$$

Note that a 0-labeled μ-redex cannot 'fire', be contracted. So the label of the contractum of a μ-redex is nullified. But, beware, not the labels of μ's inside the contractum, they remain 'what they were'. So a decreasing labeling will not stay decreasing in general. However, that does not matter.

(iv) Show that labeled $\mu\pi$-reduction is SN. So there are only finitely many labeled reducts. Clearly, after erasure of the labels they are all in $\mathsf{SC}(M)$.

(v) Vice versa, show that an unlabeled $\mu\pi$-reduction *without repetition* can be 'lifted' to a labeled one. Here the intuition is that μx^0-redexes are 'old', and have used their 'one-shot firing power'.

(vi) Conclude that $\mathsf{SC}(M)$ is finite, using that RPO is terminating. For RPO see e.g. IPO, iterative lexicographic path orders [**?**].

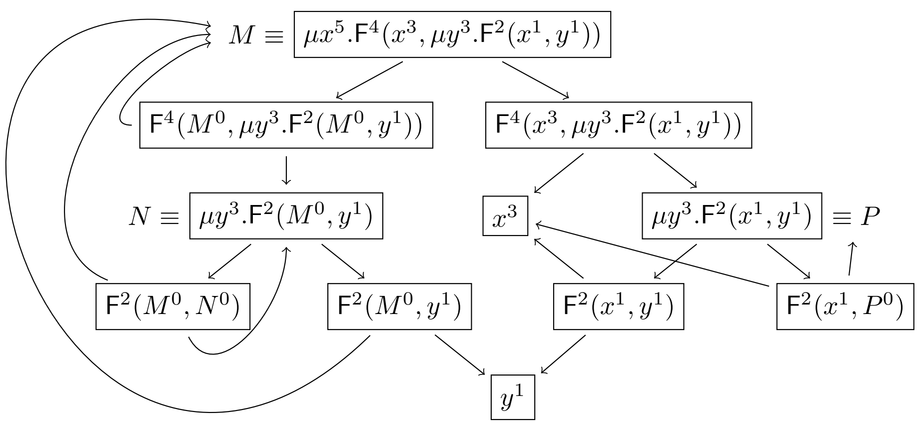


Figure 8: Finiteness of the subterm closure using a recursive path order.

Substitutions do or do not contribute to a subterm closure reduction. That the substitution of a μ-reduction step can be split off, is a consequence of the following technical factorization lemma.

Lemma 25. *Under the conditions of Lemma 16, if $M[x{:=}N] \overset{\varepsilon}{\twoheadrightarrow}_{\mu\pi} K$ then either*

(i) it is a $[x{:=}N]$-instance of a reduction $M \overset{\varepsilon}{\twoheadrightarrow}_{\mu\pi} L$, where the condition of Lemma 16 holds for L; or

(ii) $M \overset{\varepsilon}{\twoheadrightarrow}_{\mu\pi} K$; or

(iii) it has a prefix which is a $[x{:=}N]$*-instance of a reduction* $M \overset{\varepsilon}{\twoheadrightarrow}_{\mu\pi} x$*.*

Proof. The proof is by induction on the length of the reduction. If the length is 0, then case (i) holds by setting $L = M$. Otherwise, $M[x{:=}N] \overset{\varepsilon}{\twoheadrightarrow}_{\mu\pi} K' \to K$ for some term K' and we have by the induction hypothesis for this reduction either

(i) it is a $[x{:=}N]$-instance of a reduction $M \overset{\varepsilon}{\twoheadrightarrow}_{\mu\pi} L'$, where the condition of Lemma 16 holds for L'. We distinguish cases on L':

- (y) Then, since variables are normal forms, we must have $x = y$ and $N = K'$, and we are in case (iii).
- ($\mu y.L_1$) If $x = y$ then $K' = L'$ and we are in case (ii).
  If $x \neq y$ then $K' = \mu y.L_1[x{:=}N]$. If $K' \to L_1[x{:=}N] = K$ then taking $L = L_1$ brings us in case (i) using $x \neq y$ to show that the condition of Lemma 16 still holds. If $K' \to L_1[x{:=}N][y{:=}K'] = K$ then by the α-free Substitution Lemma $K = L_1[y{:=}L'][x{:=}N]$ and setting $L = L_1[y{:=}L']$ yields case (i) again, using that chains and free occurrences of x in L' trace back to L along the μ-step, as established above, so the condition of Lemma 16 still holds.
- ($\mathsf{F}(L_1, L_2)$) Then $K' = \mathsf{F}(L_1[x{:=}N], L_2[x{:=}N]) \to L_i[x{:=}N] = K$ and we are in case (i) setting $L = L_i$;

(ii) $M \overset{\varepsilon}{\twoheadrightarrow}_{\mu\pi} K'$, hence $M \overset{\varepsilon}{\twoheadrightarrow}_{\mu\pi} K' \to K$; or

(iii) it has a prefix which is a $[x{:=}N]$-instance of a reduction $M \overset{\varepsilon}{\twoheadrightarrow}_{\mu\pi} x$. Then the same holds for the reduction extended by the step $K' \to K$. □

Remark 26. The above lemma is unsatisfactory in that the second case is only there to compensate for the standard but problematic concept of M being considered a subterm of $\mu x.M$. It is problematic since it allows to free bound variables, here by means of the π-rule, breaking α-conversion. For a proper higher-order notion of subterm as introduced in Section 8, this case can be eliminated, see Lemma 55. That holds also for every higher-order recursive program scheme [**?**].

Theorem 27. $\mathsf{SC}(M)$ *is finite.*

Proof. The proof is by induction and cases on the structure of M.

- (x) $\mathsf{SC}(x) = \{x\}$.
- ($\mu x.M$) $\mathsf{SC}(\mu x.M) \subseteq \{\mu x.M\} \cup \mathsf{SC}(M) \cup \mathsf{SC}(M)[x{:=}\mu x.M]$. This is obvious in case a witness $\mu x.M \overset{\varepsilon}{\twoheadrightarrow}_{\mu\pi} N$ of minimal length to $N \in \mathsf{SC}(\mu x.M)$ is empty or starts with $\mu x.M \to M$. Otherwise, the witness starts with $\mu x.M \to M[x{:=}\mu x.M]$ and we conclude by applying the previous lemma to its suffix, noting that case (iii) cannot occur as it would contradict minimality. Hence we conclude by the induction hypothesis for M.

($\mathsf{F}(M_1, M_2)$) $\mathsf{SC}(\mathsf{F}(M_1, M_2)) = \{\mathsf{F}(M_1, M_2)\} \cup \mathsf{SC}(M_1) \cup \mathsf{SC}(M_2)$ and we conclude by the induction hypothesis for the M_i. $\square$

Remark 28. It is easy to load the induction in the proof to show that the cardinality of $\mathsf{SC}(M)$ is in fact bounded by $2^h - 1$ with h the height of M.

Remark 29. In the computation induced by the proof only residuals of μ-redexes in the initial term (no 'copies') are contracted. That is, μ-developments suffice to compute the subterm closure.

Let us comment on this interesting analogy between the classical Finite Developments Theorem in λ-calculus on the one hand, and the finiteness of the subterm closure SC on the other hand. We can formalize this statement easily by employing again an underlining argument, in a way that is somewhat dual to the one above, as follows. Underline in the initial term M all μ's. An underlined μ that is copied (by the contraction of another underlined μ-redex) looses its underlining. Now during the generation of $\mathsf{SC}(M)$, that is, during the $\pi\mu$-reduction, only underlined μ-redexes may be contracted. So the initially present underlined μ's have only a 'one shot firing power'; and a copied μ-redex cannot fire anymore. This situation coincides with the classical notion of developments of β-redexes in the λ-calculus, where only underlined β-redexes may be contracted.

5. The subterm closure and standard reductions

The subterm closure will be of fundamental importance in the coming decidability proofs. Also, the notion of standard reduction figures prominently in these proofs. Remarkably, both notions are intimately related. We will expose this relationship in the present section.

Definition 30. Let $\mathcal{R} : M_0 \to^{p_0}_{\mu/\alpha} M_1 \to^{p_1}_{\mu/\alpha} \dots$ be a reduction. Then $\mathcal{R}$ is *outside-in* if

$$\forall m > n.\ \text{not } p_m < p_n$$

Thus, $\mathcal{R}$ is outside-in if later redex positions are not above earlier ones. In other words, a contraction of a redex freezes every redex higher-up, and they remain frozen. Outside-in reductions according to the definition above coincide with *standard reductions* for R_μ viewed as a HRS [**?**].

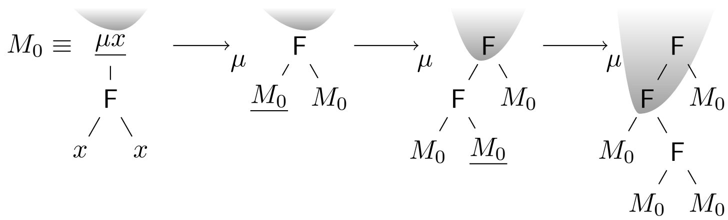
In an infinite μ-reduction $\mathcal{R} : M_0 \to M_1 \to \dots$ there can be infinitely many different redexes contracted. Remarkably, in an infinite *standard* reduction this is not the case. A strengthening of this observation is as follows:

Proposition 31. *Let* $\mathcal{R} : M_0 \to^{p_0}_{\mu/\alpha} M_1 \to^{p_1}_{\mu/\alpha} \dots$ *be a standard reduction. Then*

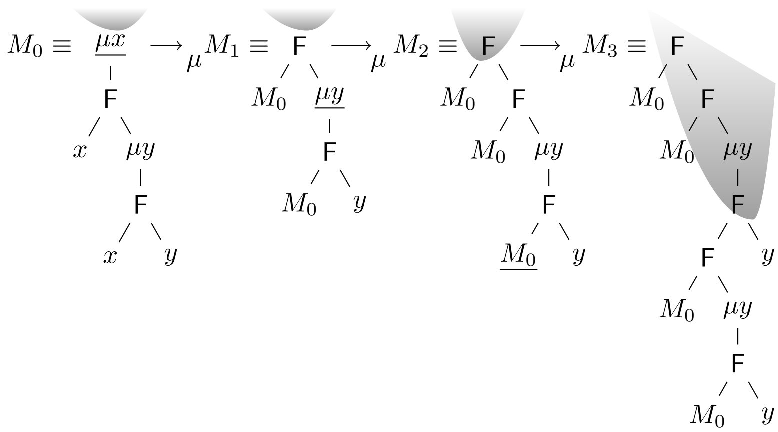
$$\forall i.\ \forall p \in \mathcal{P}os(M_i).\ (\neg\exists j < i.\ p < p_j) \implies M_i|_p \in \mathsf{SC}(M_0)$$

This proves in particular the observation above, that a standard reduction $\mathcal{R}$ contracts only finitely many redexes, even if $\mathcal{R}$ is infinite. Before giving the easy proof, let us first see what this proposition amounts to, by considering two examples.

Example 32. A standard reduction starting from $\mu x.\mathsf{F}(x,x)$



Example 33. We reconsider the term from Figure 7, the computation of the subterm closure of $\mu x.\mathsf{F}(x, \mu y.\mathsf{F}(x,y))$:



Both examples display standard reductions. That means that an increasingly large prefix 'crystallizes out', becomes frozen, by the requirement that everything above a redex contraction will be immutably fixed; no activity is allowed in that part. In the first example the frozen part only contains F's, but the second example freezes a μ-redex. Let us call the separation between frozen prefix and the lower part determined by it, the *snow line*. Now Proposition 31 states that every subterm having its root below the snow line, is an element of $\mathsf{SC}(M_0)$.

Proof of Proposition 31. For M_0 the proposition is trivially true, as the frozen prefix then still is empty, and $\mathsf{SC}(M_0)$ is closed under subterms. If we assume as induction hypothesis (IH) that the proposition holds for M_i, it is proved for M_{i+1} as follows. The redex $\mathcal{R}_i = M_i|_{p_i}$ at p_i, contracted in $M_i \to_{\mu/\alpha} M_{i+1}$, is situated below the snow line of M_i, because the reduction is standard. Thus by (IH) $\mathcal{R}_i \in \mathsf{SC}(M_0)$. Then the claim follows since the contractum of $\mathcal{R}_i$ is in $\mathsf{SC}(M_0)$ by root μ-reduction, and $\mathsf{SC}(M_0)$ is closed under subterms. $\square$

Remark 34. We said above 'increasingly large prefix'; this is so if the original term M_0 does not contain an unsolvable (circular) subterm.

Corollary 35. *Let* $\mathcal{R} : M_0 \to^{p_0}_{\mu/\alpha} M_1 \to^{p_1}_{\mu/\alpha} \dots$ *be a standard reduction.*

(i) All redexes $M_i|_{p_i}$ contracted in $\mathcal{R}$ are in $\mathsf{SC}(M_0)$.

(ii) $\mathcal{R}$ employs only finitely many different redexes (even if $\mathcal{R}$ is infinite). $\square$

If we define standard reduction in an inductive fashion, then the correspondence with the subterm closure becomes obvious:

Definition 36. The *(α-free) inductive standard relation* $\twoheadrightarrow_{\mathsf{std}}$ on the set of capture-avoiding μ-terms is defined inductively:

(i) $x \twoheadrightarrow_{\mathsf{std}} x$ for every variable or constant x,

(ii) $\mu x.M \twoheadrightarrow_{\mathsf{std}} N$ whenever $M[x{:=}\mu x.M] \twoheadrightarrow_{\mathsf{std}} N$,

(iii) $\mu x.M \twoheadrightarrow_{\mathsf{std}} \mu x.M'$ whenever $M \twoheadrightarrow_{\mathsf{std}} M'$, and

(iv) $\mathsf{F}(M, N) \twoheadrightarrow_{\mathsf{std}} \mathsf{F}(M', N')$ whenever $M \twoheadrightarrow_{\mathsf{std}} M'$ and $N \twoheadrightarrow_{\mathsf{std}} N'$.

Observe that (ii) corresponds to μ-steps at the root, (iii) to μ-removal, and (iv) to F-decomposition in the definition of the subterm closure. From the perspective of standard reduction, (iii) and (iv) decrease the height of the snow line, that is, enlarge the frozen prefix of the term.

It is easy to see that for capture-avoiding μ-terms the inductive standard relation $\twoheadrightarrow_{\mathsf{std}}$ coincides with the reduction relation generated by α-free standard reduction, which in turn is extensionally equivalent with the ordinary α-free reduction relation $\twoheadrightarrow_{\mu}$. These observations justify the use of the notion $\twoheadrightarrow_{\mathsf{std}}$.

It is instructive to compare this definition with the various proof systems in this paper by presenting it as the proof system depicted in Figure 9. Here and later we present these systems flipped upside–down with premises at the bottom and conclusions on top (cf. the motivation at the start of Section 7).

$$\frac{\mu x.M \twoheadrightarrow_{\mathsf{std}} N}{M[x{:=}\mu x.M] \twoheadrightarrow_{\mathsf{std}} N}$$

$$\frac{\mu x.M \twoheadrightarrow_{\mathsf{std}} \mu x.N}{M \twoheadrightarrow_{\mathsf{std}} N}$$

$$\frac{\mathsf{F}(M_1, M_2) \twoheadrightarrow_{\mathsf{std}} \mathsf{F}(N_1, N_2)}{M_1 \twoheadrightarrow_{\mathsf{std}} N_1 \qquad M_2 \twoheadrightarrow_{\mathsf{std}} N_2}$$

$$\frac{x \twoheadrightarrow_{\mathsf{std}} x}{}$$

Figure 9: Proof system for $\twoheadrightarrow_{\mathsf{std}}$.

6. Strong μ-equality

Although we are not primarily concerned with the infinite tree unfolding BT_μ of μ-terms and the ensuing strong μ-equality, we will briefly discuss regularity of BT_μ and decidability of strong μ-equality, in order to indicate that the methods and notions used in this paper are convenient for treating them.

Proposition 37. *For all μ-terms M, $\mathsf{BT}_\mu(M)$ is a regular tree.*

Proof. This is an immediate consequence of the finiteness of $\mathsf{SC}(M)$. Indeed, the generators of $\mathsf{BT}_\mu(M)$ can be defined as those elements in $\mathsf{SC}(M)$ that are reachable from M by a sequence of root μ-steps, followed by an F-decomposition step. The generators are the 'determinants' for BT_μ. Since they are finite in number due to Theorem 27, $\mathsf{BT}_\mu(M)$ is a regular tree. □

In Example 23, Figure 7, the generators of $\mathsf{BT}_\mu(M)$ are thus identified as $\{M, N\}$, indicated by a gray shading of their boxes in Figure 7.

Locally in this section we use the following notion: Given a goal equation $A = B$ between of capture-avoiding μ-terms, the *deductive closure* of the set $\{A = B\}$ is the set of all equations between μ-terms that can be produced starting on the set $\{A = B\}$ by repeated application of the following two generation rules from equations to equations: F-decomposition (see Section 4) simultaneously on either side of an equation, and root-μ-reduction (root-unfolding, see Section 4) simultaneously on either side of an equation.

Theorem 38. *Strong μ-equality $=_{\mathsf{BT}_\mu}$ is decidable.*

Proof Sketch. It suffices to show decidability of strong μ-equality between capture-avoiding μ-terms A and B. If these terms are not capture-avoiding, then we step over to capture-avoiding α-variants, using that BT_μ, and hence also $=_{\mathsf{BT}_\mu}$, is invariant under α-equivalence.

Given a goal equation $A = B$ between of capture-avoiding μ-terms, we compute the deductive closure of the set $\{A = B\}$. We find that this closure is finite, and hence can be constructed effectively: by induction on the generation of the deductive closure it can be proved that all equations $P = Q$ in it are contained in $\mathsf{SC}(A) \times \mathsf{SC}(B)$, which is a finite set, as a consequence of Theorem 27.

In the deductive closure of $\{A = B\}$ we only have to inspect whether it contains an 'inconsistency' or not (an inconsistency is either an equation between a variable and a constant, or an equation between different variables or constants, or an equation between a variable or a constant and a μ-term starting with F). In the latter case we conclude: $A =_{\mathsf{BT}_\mu} B$; and in the former: $A \neq_{\mathsf{BT}_\mu} B$. □

Example 39. Consider $A = \mu x.\mathsf{F}(x, x)$ and $B = \mu xy.\mathsf{F}(x, y)$. Abbreviate $C = \mu y.\mathsf{F}(B, y)$. Then the deductive closure is $\{A = B, A = C\}$. No inconsistency has appeared, and hence: $A =_{\mathsf{BT}_\mu} B$.

Remark 40. The deductive closure is intimately connected with the notion of bisimulation; in fact the equations are pairs in what can be called a bisimulation between the μ-terms A and B. Instead of the deductive closure sketched in the proof above, one can use the presentation of a downward growing search tree, branching at an F-decomposition, and equipped with loop checks at repeated occurrences of equations. This loop checked tree is in fact the mirror image of a proof of $A = B$ in the proof system of Brandt–Henglein [**?**], as elaborated in Grabmayer [**? ?**].

7. A proof system for μ-convertibility

In this section we will present our first proof of decidability of μ-convertibility. It is based on a method, including a proof system, devised by Cardone and Coppo, as we will discuss.

Usually, proofs in a deductive system are rendered with the axioms on top, working downwards to the goal equation. We prefer to flip over these proof figures, as in tableau proof systems, with the goal equation as the root on top, then have unary–binary branching towards the bottom layer, the axioms, which hence are 'terminals' of the proof tree; see Figure 10. In this way it is easier to connect derivations in the system with the (for rewrite sequences quite intuitive) downwards-direction of standard reductions that are formalized by derivations.

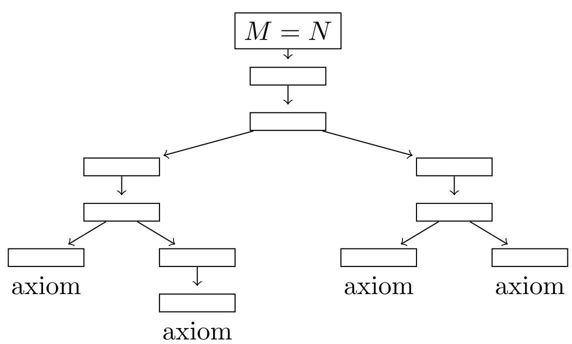


Figure 10: Proof rendered top-down.

7.1. The proof system of Cardone and Coppo

We first recall the original proof system of Cardone and Coppo, see Figure 11. Again, we present the system flipped upside–down with premises at the bottom and conclusions on top.

Note that for the proof system to work, terms have to be considered modulo α-equivalence. For example, consider the equation

$$\mu x.\mu z.\mathsf{F}(x, z) =_{\mu/\alpha} \mu y.\mathsf{F}(y, \mu z.\mathsf{F}(y, z))$$

Then to prove the equation, we would like to employ μ-removal followed by a μ-step in the left-hand side, and finally an axiom. However, μx-removal requires

$$\frac{\mu x.M(x) = N}{M(\mu x.M(x)) = N} \quad \textit{μ-step in lhs}$$

$$\frac{M = \mu y.N(y)}{M = N(\mu y.N(y))} \quad \textit{μ-step in rhs}$$

$$\frac{\mu x.M = \mu x.N}{M = N} \quad \textit{μ-removal}$$

$$\frac{\mathsf{F}(M_1, M_2) = \mathsf{F}(N_1, N_2)}{M_1 = N_1 \qquad M_2 = N_2} \quad \mathsf{F}\textit{-decomposition}$$

$$M = M \quad \textit{axiom}$$

Figure 11: Original proof system of Cardone and Coppo.

the binders on the left-hand side and on the right-hand side of the equation to be equal. For this reason we need to consider all terms in this system modulo α-conversion.

Since the set of variables is infinite, α-equivalence classes are a priori infinite too, and we need additional arguments to make the search space finite. A possibility to make the system from Figure 11 finite would be to α-convert the terms in the starting equation such that they are capture-avoiding (see Section 3), and restrict α-conversion throughout the derivation to terms over the set of (the union of the) initially used binders. However, the latter requires a proof that completeness of the system is not lost.

For this reason, we strive for a proof system where the μ-terms are taken literally, without α-conversion. To this end we extend the system of Cardone and Coppo with annotations, similar in spirit to the context in Kahrs's proof system for deciding α-equivalence [**?**].

7.2. A proof system with annotations

An important notation is the use of annotated μ-bindings, written as (μx), (μy), or as vectors (μxyz) or $(\mu\vec{x})$. So if $M \in \mathit{Ter}(\mu)$, then $(\mu x)M$ is an annotated μ-term. We only employ the annotation vectors $(\mu\vec{x})$ at the *root* of a term.

Definition 41. An *annotated μ-term* is an expression of the form $()M$, or $(\mu x_1 \dots x_n)M$ where $M \in \mathit{Ter}(\mu)$, and $x_1, \dots, x_n$ are variables. (Here $()M$ and $(\mu x_1 \dots x_n)M$ are the annotations of a μ-term M by an empty μ-binding prefix, and by the μ-binding prefix $\mu x_1 \dots x_n$, respectively.) By $\mathit{AnnTer}(\mu)$ we denote the set of all annotated μ-terms.

Example 42. (i) $(\mu x)\mathsf{F}(\mu y.y, \mathsf{c})$ is an annotated μ-term.

(ii) For every μ-term M, $(\mu\vec{x}\vec{y})M$ is an annotated μ-term.

We employ the annotation vectors $(\mu\vec{z})$ to keep a record of the μ-bindings that have been removed (μ-removal), or in another view, that we have passed in selecting the redex to be contracted in the standard reduction which is implicit in the proof. For the annotated proof system $\mathcal{S}$, see Figure 12.

A priori, the annotations would grow unboundedly. In order to obtain a finite proof system, we introduce a *compression rule* that removes 'two-sided vacuous' μ-binders from the annotations. To enforce compression, we restrict application of the other inference rules to fully compressed formulas.

Definition 43. An occurrence of a binder μx in a term $M \in \mathit{Ter}(\mu)$ at position p is called *active* if $x \in \mathsf{FV}(M|_{p1})$, the set of *free variables* of $M|_{p1}$. A non-active occurrence of a binder μx we will also call *vacuous*.

For annotated terms $(\mu x_1 \dots x_n)M$ we say that x_i is *active* or *vacuous* if the respective property holds for μx_i at position 1^{i-1} in $\mu x_1 \dots x_n.M$.

Let $M = (\mu x_1 \dots x_n)M'$ and $N = (\mu y_1 \dots y_n)N'$, and consider the annotated equation $e : M = N$. Then an index $1 \leq i \leq n$ is called *two-sided vacuous in* e if the displayed occurrence of x_i is vacuous in M, and y_i is vacuous in N. The equation e is called *compressed* if it has no two-sided vacuous indexes.

Example 44. All μ-unsolvables are weakly μ-equivalent to $\Omega = \mu y.y$; for example $M \equiv \mu x_3 x_2 x_1.x_2 \overset{x_3}{\to}_\mu \mu x_2 x_1.x_2 \overset{x_1}{\to}_\mu \mu x_2.x_2 \equiv_\alpha \Omega$. See Figure 13 for a systematic proof search for the equation $M = \Omega$ in the proof system $\mathcal{S}$. We stop the proof search when encountering a *repetition* along a branch.

Example 45. For a proof employing the compression rule, we consider the equation:

$$\mu xyz.\mathsf{F}(\mu u.x, z) = \mu x'y'.\mathsf{F}(x', \mu z'.\mathsf{F}(x', z'))$$

The successful part of the proof search is displayed in Figure 14. Note that the graph contains two different edge types. The dashed edges '- - -' represent the non-deterministic choices in the proof search, while the solid edges '—' stand for splits due to F-decomposition. A node in a search tree is successful if:

(i) An inner node with outgoing '- - -' edges is successful if at least one of its children is successful.

(ii) An inner node with outgoing '—' edges is successful if all its children are successful (such nodes have exactly two children).

(iii) A leaf (that is, a node without outgoing edges) is successful if and only if it is an axiom.

A proof search is successful if the root node of the proof search tree is successful. We mark the successful nodes with $\top$, and the unsuccessful nodes with $\bot$.

Lemma 46 (**Standard reductions represented by $\mathcal{S}$-derivations**). *The following statements are equivalent for capture-avoiding μ-terms M and N:*

The *compression* rule:

$$\frac{e : (\mu x_1 \dots x_n)M = (\mu y_1 \dots y_n)N}{(\mu x_1 \dots x_{i-1}x_{i+1} \dots x_n)M = (\mu y_1 \dots y_{i-1}y_{i+1} \dots y_n)N}$$

whenever i is a two-sided vacuous index in e.

The following inference rules are restricted to such instances in which the conclusion (the formula on top) is a *compressed* equation:

$$\frac{(\mu\vec{z})\mu x.M(x) = (\mu\vec{u})N}{(\mu\vec{z})M(\mu x.M(x)) = (\mu\vec{u})N} \quad \textit{annotated } \mu\textit{-step in lhs}$$

$$\frac{(\mu\vec{z})M = (\mu\vec{u})\mu y.N(y)}{(\mu\vec{z})M = (\mu\vec{u})N(\mu y.N(y))} \quad \textit{annotated } \mu\textit{-step in rhs}$$

$$\frac{(\mu\vec{z})\mu x.M = (\mu\vec{u})\mu y.N}{(\mu\vec{z}x)M = (\mu\vec{u}y)N} \quad \textit{annotated } \mu\textit{-removal}$$

$$\frac{(\mu\vec{z})\mathsf{F}(M_1, M_2) = (\mu\vec{u})\mathsf{F}(N_1, N_2)}{(\mu\vec{z})M_1 = (\mu\vec{u})N_1 \quad (\mu\vec{z})M_2 = (\mu\vec{u})N_2} \quad \textit{annotated } \mathsf{F}\textit{-decomposition}$$

$$\frac{(\mu x)x = (\mu y)y}{} \qquad \frac{()x = ()x}{} \qquad \frac{()\mathsf{c} = ()\mathsf{c}}{} \quad \textit{axioms (end points, matches)}$$

Figure 12: Proof system $\mathcal{S}$ with annotations.

(i) $\vdash_{\mathcal{S}} (\mu\vec{x})M = (\mu\vec{y})N$.

(ii) There exist α-free standard reductions $M \twoheadrightarrow_{\mathsf{std}} M'$ and $N \twoheadrightarrow_{\mathsf{std}} N'$ such that for variable vectors $\vec{x}$ and $\vec{y}$ with $|\vec{x}| = |\vec{y}|$ it holds: $\mu\vec{x}.M' \equiv_{\alpha} \mu\vec{y}.N'$.

Let us first comment on the intuition behind the lemma. A proof $\Delta : M = N$ can be seen as the result of an *interleaving* of two standard reductions for M and N, respectively. Actually, the interleaving also contains at some points a synchronization between the two 'processes' that are the respective standard reductions of M and N. Namely, note that the proof system requires the *simultaneous removal* of μ-binders, μx and μy, respectively. Indeed, the definition of standard reduction 'freezes' the μ's that are passed; thereby they are turned into fixed, immutable *constructors*. Also F is a constructor, this one binary; and also F's are peeled off simultaneously in lhs and rhs, that is, the removal (which can be seen as an *observation*) is simultaneous, synchronized.

Proof of Lemma 46. For showing '(ii) $\Rightarrow$ (i)', we first argue that it suffices to show this implication only in situations in which a compressed equation has

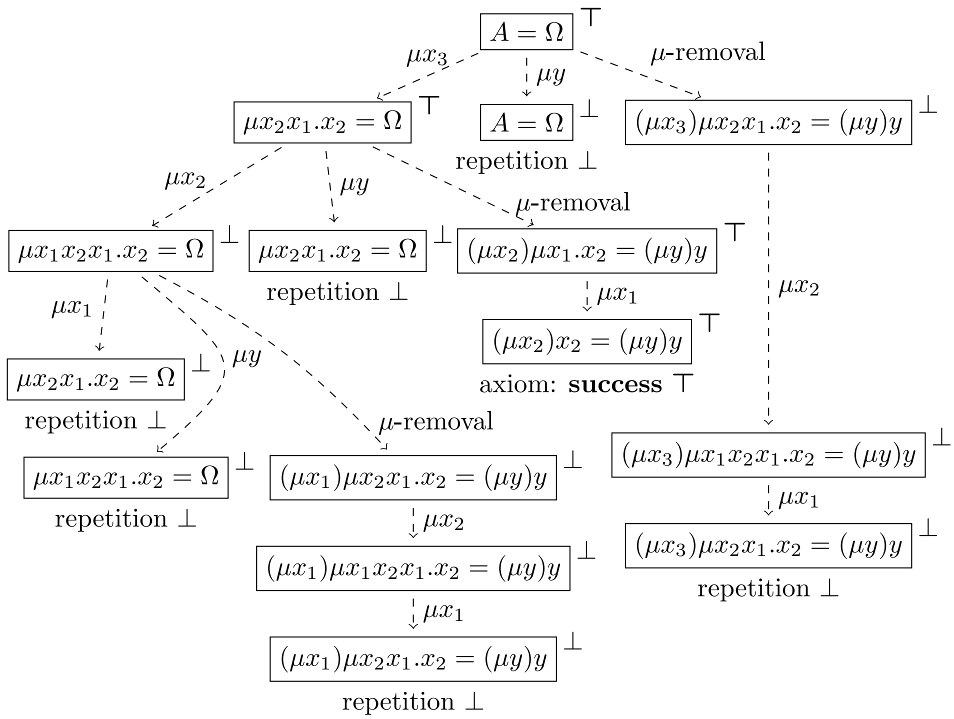


Figure 13: A proof of $\mu x_3 x_2 x_1.x_2 = \mu y.y$.

to derived in $\mathcal{S}$: For this, suppose that that (ii) holds for some $\vec{x}$, $\vec{y}$, M, and N such that the equation $(\mu\vec{x})M = (\mu\vec{y})N$ is not compressed. Then the equation $(\mu\vec{x})M = (\mu\vec{y})N$ contains a two-sided vacuous index i. Now an application of the compression rule yields the equation $(\mu\vec{x}')M = (\mu\vec{y}')N$ where $\vec{x}' = x_1 \ldots x_{i-1}x_{i+1}x_n$ and $\vec{y}' = x_1 \ldots y_{i-1}y_{i+1}y_n$. And furthermore, $\mu\vec{x}.M \equiv_\alpha \mu\vec{y}.N$ implies $\mu\vec{x}'.M' \equiv_\alpha \mu\vec{y}'.N'$. By assumption (ii) we still have α-free standard reductions $M \twoheadrightarrow_{\mathsf{std}} M'$ and $N \twoheadrightarrow_{\mathsf{std}} N'$. If $(\mu\vec{x}')M = (\mu\vec{y}')N$ is compressed, then the restricted version of (ii) $\Rightarrow$ (i)' can be applied to obtain a derivation $\mathcal{D}$ in $\mathcal{S}$ with conclusion $(\mu\vec{x}')M = (\mu\vec{y}')N$, and, by extending $\mathcal{D}$ by an application of the compression rule, a derivation with conclusion $(\mu\vec{x})M = (\mu\vec{y})N$. If $(\mu\vec{x}')M = (\mu\vec{y}')N$ is not compressed, then this argument can be repeated until, after finitely many steps, the restricted version of '(ii) $\Rightarrow$ (i)' can be applied.

Now we proceed to show '(ii) $\Rightarrow$ (i)' by induction on the inductive definition of the inductive standard reductions $\sigma : M \twoheadrightarrow_{\mathsf{std}} M'$ and $\tau : N \twoheadrightarrow_{\mathsf{std}} N'$ (induction on the lexicographic order on $\langle\sigma,\tau\rangle$) with respect to Definition 36. By the argument above, we may assume that $\vec{x}$, $\vec{y}$, M, and N in (ii) are such that the equation $(\mu\vec{x})M = (\mu\vec{y})N$ is compressed.

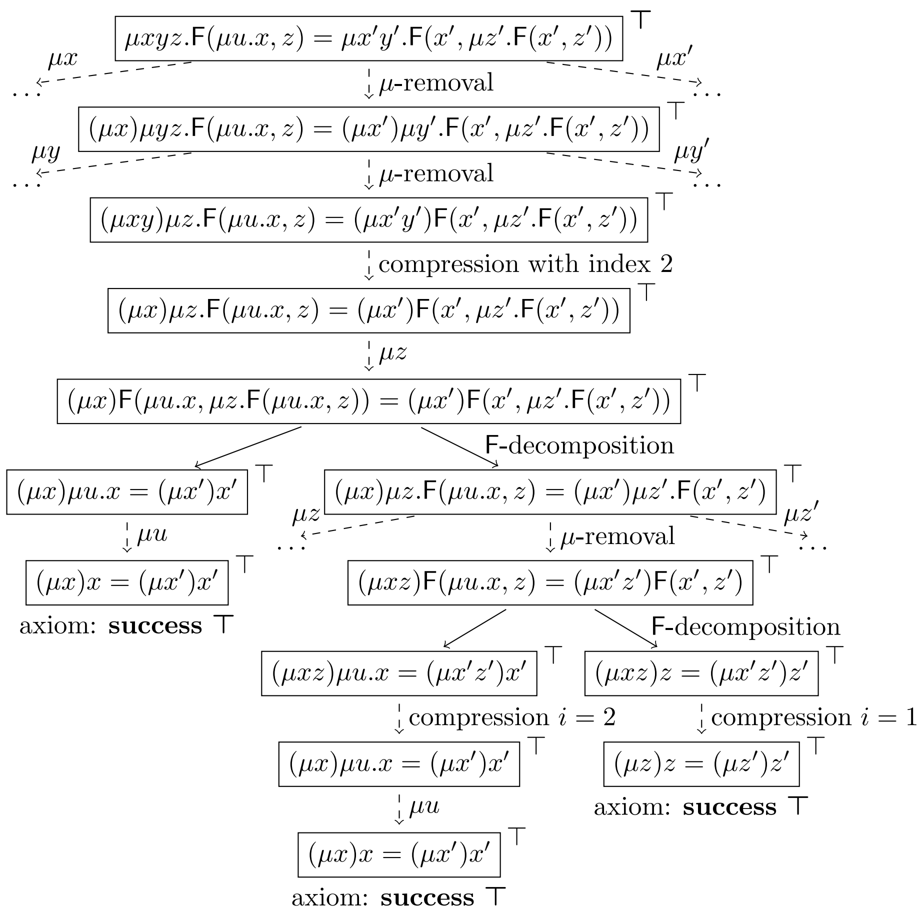


Figure 14: A proof of $\mu xyz.\mathsf{F}(\mu u.x, z) = \mu x'y'.\mathsf{F}(x', \mu z'.\mathsf{F}(x', z'))$.

- The base case: σ and τ are with respect to case (i) in Definition 36. Then $M \equiv M'$ and $N \equiv N'$ are variables or constants. Since $(\mu\vec{x})M = (\mu\vec{y})N$ is compressed, the assumption leaves room for only the following three cases: (i) $M \equiv M' \equiv N \equiv N'$ is a constant, and both of $\vec{x}$ and $\vec{y}$ are empty; (ii) $M \equiv M \equiv N \equiv N' \equiv z$ for a variable z, and both of $\vec{x}$ and $\vec{y}$ are empty; (iii) $M \equiv M \equiv z$ and $N \equiv N' \equiv u$ for a variables z and u, and $\vec{x} = z$ and $\vec{y} = u$. In all three cases $(\mu\vec{x})M = (\mu\vec{y})N$ is an axiom, which demonstrates $\vdash_{\mathcal{S}} (\mu\vec{x})M = (\mu\vec{y})N$.
- Assume that σ or τ starts with a μ-step at the root (that is, case (ii) in Definition 36). By symmetry let it be σ. Then $M \equiv \mu z.M''$ and $M''[z{:=}M] \twoheadrightarrow_{\mathsf{std}} M'$. Then a μ-step in the left-hand side of the compressed equation $(\mu\vec{x})M = (\mu\vec{y})N$ yields $(\mu\vec{x})M''[z{:=}M] = (\mu\vec{y})N$ to which the induction hypothesis is applicable.

– If σ and τ are with respect to case (iii) in Definition 36, then $M \equiv \mu z.M_2$ and $M' \equiv \mu z.M_2'$ with $M_2 \twoheadrightarrow_{\mathsf{std}} M_2'$, and $N \equiv \mu u.N_2$, $N' \equiv \mu u.N_2'$ with $N_2 \twoheadrightarrow_{\mathsf{std}} N_2'$. Then a μ-removal in $(\mu\vec{x})M = (\mu\vec{y})N$, which is compressed, yields $(\mu\vec{x}z)M_2 = (\mu\vec{y}u)N_2$; again the induction hypothesis is applicable.

– If σ and τ are with respect to case (iv) in Definition 36, then $M \equiv \mathsf{F}(M_1, M_2)$, $M' \equiv \mathsf{F}(M_1', M_2')$, $N \equiv \mathsf{F}(N_1, N_2)$, and $N' \equiv \mathsf{F}(N_1', N_2')$ with $M_1 \twoheadrightarrow_{\mathsf{std}} M_1'$, $M_2 \twoheadrightarrow_{\mathsf{std}} M_2'$, $N_1 \twoheadrightarrow_{\mathsf{std}} N_1'$, and $N_2 \twoheadrightarrow_{\mathsf{std}} N_2'$. Now observe that the assumption $\mu\vec{x}.M' \equiv_\alpha \mu\vec{y}.N'$ implies $\mu\vec{x}.M_1' \equiv_\alpha \mu\vec{y}.N_1'$, and $\mu\vec{x}.M_2' \equiv_\alpha \mu\vec{y}.N_2'$. Then the induction hypothesis can be applied to obtain derivations of $(\mu\vec{x})M_1 = (\mu\vec{y})N_1$ and $(\mu\vec{x})M_2 = (\mu\vec{y})N_2$, respectively, from which a derivation with as conclusion the compressed equation $(\mu\vec{x})M = (\mu\vec{y})N$ can be constructed by applying the F-decomposition rule.

This concludes the proof of direction '(ii) $\Rightarrow$ (i)'.

The proof of the direction direction '(i) $\Rightarrow$ (ii)' works by straightforward induction on the proof of an equation in $\mathcal{S}$, employing the following key observation: if $\mu\vec{x}.M_1 \equiv_\alpha \mu\vec{y}.N_1$ and $\mu\vec{x}.M_2 \equiv_\alpha \mu\vec{y}.N_2$ then it follows $\mu\vec{x}.\mathsf{F}(M_1, M_2) \equiv_\alpha \mu\vec{y}.\mathsf{F}(N_1, N_2)$. $\square$

Proposition 47. *Provability in $\mathcal{S}$ is decidable.*

Proof. By induction on the depth of derivations it can be shown that if the formula $(\mu\vec{z})M' = (\mu\vec{u})N'$ occurs in a derivation $\mathcal{D}$ in $\mathcal{S}$ with $(\mu\vec{x})M = (\mu\vec{y})N$ as its conclusion, then:

(i) $M' \in \mathsf{SC}(M)$ and $N' \in \mathsf{SC}(N)$ (for this the rewrite relation $\xrightarrow{\varepsilon}_{\mu\pi}$ in the definition of SC can be used);

(ii) $\vec{z} = \vec{v}\vec{x}$ and $\vec{u} = \vec{w}\vec{y}$ for some vectors $\vec{v}$ and $\vec{w}$ of variables, which are contained among the bound variables of M, and of N, respectively;

(iii) $\mathsf{FV}(M') \subseteq \mathsf{FV}(M) \cup \mathsf{BV}(M)$, and $\mathsf{FV}(N') \subseteq \mathsf{FV}(N) \cup \mathsf{BV}(N)$.

and furthermore, as a consequence of ((iii)):

(iv) For the lengths $|\vec{z}|$ and $|\vec{u}|$ of the annotation vectors $\vec{z}$ and $\vec{u}$ it holds: $|\vec{z}| \leq \max\{|\vec{x}|, n_M + n_N + 1\}$ and $|\vec{u}| \leq \max\{|\vec{y}|, n_M + n_N + 1\}$, where $n_M \in \mathbb{N}$ ($n_N \in \mathbb{N}$) is the sum of the number of free variables of M (of N) and the number of bound variables in M (in N). (Due to ((iii)) there cannot be more than n_M (more than n_N) non-vacuous μ-bindings among the visible μ-bindings of $\mu\vec{z}.M'$ (of $\mu\vec{u}.N'$), which implies that whenever $|\vec{z}|, |\vec{u}| \geq n_M + n_N + 1$, then the equation $(\mu\vec{z})M' = (\mu\vec{u})N'$ contains a two-sided vacuous index, which can be removed, and together with all other two-sided vacuous indices has to be removed, by the compression rule of $\mathcal{S}$ if the $\mathcal{D}$ extends below the considered occurrence of $(\mu\vec{z})M' = (\mu\vec{u})N'$.)

The facts (i), (ii), and (iv) imply that for every given conclusion there can be, due to Theorem 27, only a finite number of irredundant derivations, that is, derivations without formula repetitions. Now observe that every proof $\mathcal{D}$ in $\mathcal{S}$ can be transformed into an irredundant one by a finite number of size-decreasing steps in which, respectively, some subderivation $\mathcal{D}_0$ that contains a proper subderivation $\mathcal{D}'_0$ with the same conclusion is replaced by $\mathcal{D}'_0$. It follows that the problem of deciding whether a given equation between annotated μ-terms is provable in $\mathcal{S}$ can be reduced to the problem of finding an irredundant proof for this equation, and that the latter problem is decidable because the search space for it is always finite. $\square$

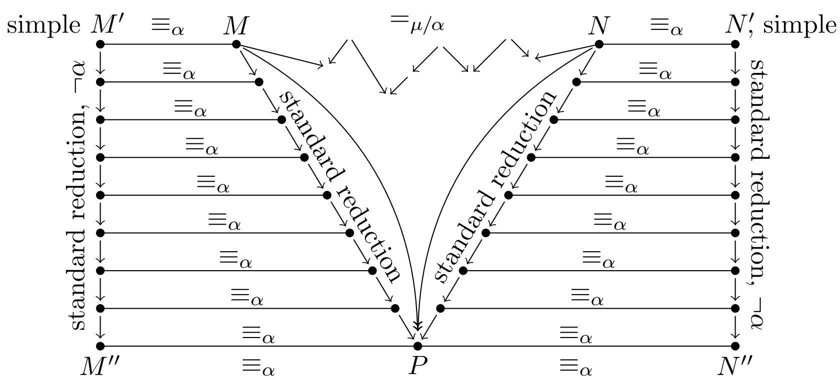


Figure 15: Structure of the proof of the Theorem 48 in this section.

Theorem 48. *Weak μ-equality is decidable.*

Proof. In view of Lemma 46 and Proposition 47 it suffices to prove that, for all μ-terms M and N, $M =_{\mu/\alpha} N$ holds if and only if there exist α-variants M' and N' of M and N, respectively, such that there are α-free standard reductions $M' \twoheadrightarrow_{\mathsf{std}} M''$ and $N' \twoheadrightarrow_{\mathsf{std}} N''$ with $M'' \equiv_\alpha N''$. We proceed to show this.

"$\Leftarrow$" follows from $M \equiv_\alpha M' \twoheadrightarrow_\mu M'' \equiv_\alpha N'' \twoheadleftarrow_\mu N' \equiv_\alpha N$. For showing "$\Rightarrow$", we suppose that $M =_{\mu/\alpha} N$, and argue in three steps that are illustrated in Figure 15 to obtain the desired α-free standard reductions:

(i) Since $\rightarrow_{\mu/\alpha}$ has the Church–Rosser property, there exists a μ-term P and a pair $M \twoheadrightarrow_{\mu/\alpha} P \twoheadleftarrow_{\mu/\alpha} N$ of joining $\rightarrow_{\mu/\alpha}$-rewrite sequences.

(ii) By standardization of these rewrite sequences (which is possible as a consequence of the fact that standardization holds for all local (linear and fully extended) HRSs (higher-order rewriting systems), see [**?** , Ch.5]), one obtains standard reductions $M \twoheadrightarrow_{\mathsf{std}} P$ and $N \twoheadrightarrow_{\mathsf{std}} P$ that join M and N.

(iii) Let M' and N' be capture-avoiding α-variants of M and N, respectively. Since α-conversion can be postponed over $\rightarrow_\mu$-steps, it follows that there exist α-free rewrite sequences $M' \twoheadrightarrow_\mu M''$ and $N' \twoheadrightarrow_\mu N''$ that are stepwisely linked via α-conversion to the standard reductions $M \twoheadrightarrow_{\mathsf{std}} P$

and $N \twoheadrightarrow_{\mathsf{std}} P$, contracting redexes at corresponding positions. Hence these rewrite sequences are α-free standard reductions $M' \twoheadrightarrow_{\mathsf{std}} M''$ and $N' \twoheadrightarrow_{\mathsf{std}} N''$, and it holds: $M'' \equiv_\alpha P \equiv_\alpha N''$, $M' \equiv_\alpha M$, and $N' \equiv_\alpha N$.

□

7.3. More efficient proof search

We can improve the efficiency of the decision procedure for provability in $\mathcal{S}$ by devising (efficient) criteria for non-provability of equations. In this subsection we consider such a criterion that discerns whether a binder μx is vacuous (not binding an actual occurrence of x) or not.

Definition 49. Let $M = (\mu x_1 \dots x_n)M'$ and $N = (\mu y_1 \dots y_n)N'$. The annotated equation $M = N$ is said to have a *binder mismatch* if there exists an index $1 \leq i \leq n$ such that either

- x_i is vacuous in M, and y_i is non-vacuous in N, or
- x_i is non-vacuous in M, and y_i is vacuous in N.

Note that μ-reduction preserves the set of free variables:

Proposition 50. *Let $M \twoheadrightarrow_{\mu/\alpha} N$. Then $\mathsf{FV}(M) = \mathsf{FV}(N)$.* □

Note that if e has a binder mismatch, then it cannot be proved in the annotated proof system. This follows immediately from Lemma 46 and Proposition 50.

Proposition 51. *Annotated equations $M = N$ with binder mismatch cannot be proven in $\mathcal{S}$.* □

So when we encounter an annotated equation e with binder mismatch, in the 'meta-search tree' in which we try to find a proof of $M = N$, we can stop that branch in the meta-search tree with failure. For example, in Figure 13 we could have stopped at the equations $(\mu x_3)\mu x_2 x_1.x_2 = (\mu y)y$ and $(\mu x_1)\mu x_2 x_1.x_2 = (\mu y)y$, pruning the search tree by 4 nodes.

8. Deciding μ-convertibility by higher-order means

Thus far we treated μ-terms as far as possible as first-order terms. In this section we show that μ-convertibility is decidable by higher-order means. That means that here we will, unlike what is the case in the rest of the paper, always consider terms as α-equivalence classes. Higher-order syntax has the disadvantage of being less concrete than first-order syntax, but the advantage of not having to care about binding and renaming issues, and also of enabling the use of the extant body of theory on higher-order term rewriting ([**?** , Chapter 11] and [**?**]). In the higher-order setting μ-convertibility is simply the convertibility relation of the HRS μ

$$\mu x.M(x) \quad \rightarrow \quad M(\mu x.M(x))$$

with the signature of μ comprising the binding symbol $\mu{:}(o{\to}o){\to}o$, the function symbol $\mathsf{F}{:}o \to o \to o$, and constants $\mathsf{c}, \mathsf{d}, \ldots$ and numerals $\underline{n}$, for each natural number n, all of type o.

Remark 52. Note that the HRS has a *single* rule, whereas in the first-order rendering μ-convertibility is generated from an *infinity* of rules.

We write $\vdash M = N$ to indicate that $M = N$ can be derived using the higher-order version of the proof system of Cardone and Coppo, as displayed in Figure 16.

$$\frac{\mu x.M(x) = N}{M(\mu x.M(x)) = N} \quad \textit{μ-step in lhs}$$

$$\frac{M = \mu y.N(y)}{M = N(\mu y.N(y))} \quad \textit{μ-step in rhs}$$

$$\frac{\mu x.M(x) = \mu y.N(y)}{M(\underline{n}) = N(\underline{n})}\ n \text{ least not in } M, N \quad \textit{μ-removal}$$

$$\frac{\mathsf{F}(M_1, M_2) = \mathsf{F}(N_1, N_2)}{M_1 = N_1 \qquad M_2 = N_2} \quad \mathsf{F}\textit{-decomposition}$$

$$\frac{\underline{n} = \underline{n}}{} \qquad \frac{\mathsf{c} = \mathsf{c}}{} \quad \textit{axiom}$$

Figure 16: Higher-order Cardone and Coppo system.

Remark 53. To overcome the problem that the inference rule $\dfrac{\mu x.M = \mu x.N}{M = N}$ as employed by Cardone and Coppo turns bound occurrences of x into free ones, we have chosen here to substitute a (fresh) numeral for it (preserving closedness). This is a common technique; one may think of De Bruijn indices or of Schroer's technique for formalizing α-equivalence by substituting a fresh symbol, cf. [**?**].

Furthermore note that in the μ-removal rule the numeral $\underline{n}$ is chosen such that, apart from $\underline{n}$ being fresh for M and N, the natural number n is also the least one among all those numbers m with the property that $\underline{m}$ does not occur in M nor in N. The reason is that, for showing decidability of μ-convertibility, we will use the statement that, for every term M, only finitely many numerals occur in the 'subterm closure' $\mathcal{SC}(M)$ of M (cf. the definition of $\mathcal{SC}(M)$ based on $\mu\pi$-root-reduction below, and ultimately, Lemma 55).

Figure 17 displays an example of a proof in the system of Figure 16. The system of Figure 16 is sound and complete for deciding μ-convertibility.

To highlight the difference with the preceding part of the paper, and that we now work with α-equivalence classes, we will denote the convertibility relation with respect to $\to$ by $\leftrightarrow^*$.

$$
\begin{array}{c}
\mu xyz.\mathsf{F}(\mu u.x, z) = \mu x'y'.\mathsf{F}(x', \mu z'.\mathsf{F}(x', z')) \\ \hline
\mu yz.\mathsf{F}(\mu u.\underline{0}, z) = \mu y'.\mathsf{F}(\underline{0}, \mu z'.\mathsf{F}(\underline{0}, z')) \\ \hline
\mu z.\mathsf{F}(\mu u.\underline{0}, z) = \mathsf{F}(\underline{0}, \mu z'.\mathsf{F}(\underline{0}, z')) \\ \hline
\mathsf{F}(\mu u.\underline{0}, \mu z.\mathsf{F}(\mu u.\underline{0}, z)) = \mathsf{F}(\underline{0}, \mu z'.\mathsf{F}(\underline{0}, z')) \\ \hline
\begin{array}{cc}
\begin{array}{c} \mu u.\underline{0} = \underline{0} \\ \hline \underline{0} = \underline{0} \\ \hline \end{array}
&
\begin{array}{c}
\mu z.\mathsf{F}((\mu u.\underline{0}), z) = \mu z'.\mathsf{F}(\underline{0}, z') \\ \hline
\mathsf{F}(\mu u.\underline{0}, \underline{1}) = \mathsf{F}(\underline{0}, \underline{1}) \\ \hline
\begin{array}{cc}
\begin{array}{c} \mu u.\underline{0} = \underline{0} \\ \hline \underline{0} = \underline{0} \\ \hline \end{array}
&
\begin{array}{c} \underline{1} = \underline{1} \\ \hline \end{array}
\end{array}
\end{array}
\end{array}
\end{array}
$$

Figure 17: A higher-order proof of $\mu xyz.\mathsf{F}(\mu u.x, z) = \mu x'y'.\mathsf{F}(x', \mu z'.\mathsf{F}(x', y'))$.

Lemma 54. *$M \leftrightarrow^* N$ if and only if $\vdash M = N$, for closed terms M,N.*

Proof. The proof of this lemma is analogous to that of Lemma 25, employing the higher-order equivalents of the notions and results used there.

(only if) Suppose $M \leftrightarrow^* N$. Since the higher-order rewrite system is orthogonal, its rewrite relation is confluent [**?** , Chapter 11], hence μ-convertibility coincides with μ-joinability, and by the standardization theorem [**?**], we may assume that the witnessing rewrite sequences are standard. Hence it suffices to show $\vdash M = N$ for any pair $M \to^n L, N \to^m L$ of standard reductions. This we prove by induction on $n + m$.

In case $n = 0 = m$, then $M = L = N$ and the result follows since derivability of a pair of equal (α-equivalent) closed terms is easily inferred by induction on the size.

In case either of the reductions starts with a root step, say w.l.o.g. $\mu x.M' \xrightarrow{\varepsilon}_{\mu\pi} M'[\![x{:=}\mu x.M']\!] \twoheadrightarrow L$, then we have $\vdash M'[\![x{:=}\mu x.M']\!] = N$ by the induction hypothesis, from which we conclude by either of the first two rules of Figure 16.

In case neither of the reductions starts with a root step, then by standardness none of the steps in either reduction is a root step. Hence M and N must have the same root symbol.

In case the root symbol is μ, then $M = \mu x.M' \twoheadrightarrow \mu x.L' = L$ and $N = \mu x.N' \twoheadrightarrow \mu x.L' = L$, and we have $\vdash M'[\![x{:=}\underline{n}]\!] = N'[\![x{:=}\underline{n}]\!]$ by the induction hypothesis and closure of standard reductions under projection and substitution of terms of base type, from which we conclude by the μ-congruence rule of Figure 16.

In case the root symbol is F, then $M = \mathsf{F}(M_1, M_2) \twoheadrightarrow \mathsf{F}(L_1, L_2) = L$ and $N = \mathsf{F}(N_1, N_2) \twoheadrightarrow \mathsf{F}(L_1, L_2) = L$, and we have $\vdash M_1 = N_1$ and $\vdash M_2 = N_2$ by the induction hypothesis and closure of standard reductions under projection, from which we conclude by the last rule of Figure 16.

(if) By induction on the derivation of $\vdash M = N$ and by cases on the inference rules.

Suppose $\vdash \mu x.M(x) = N$ because $\vdash M(\mu x.M(x)) = N$. Then by the induction hypothesis $M(\mu x.M(x)) \leftrightarrow^* N$ and we conclude from $\mu x.M(x) \to M(\mu x.M(x))$. For the symmetric inference rule, the reasoning is analogous.

Suppose $\vdash \mu x.M(x) = \mu y.N(y)$ because $\vdash M(\underline{n}) = N(\underline{n})$ with n least not in M, N. By the induction hypothesis $M(\underline{n}) \leftrightarrow^* N(\underline{n})$, hence, by properties of α-equivalence [**?**], $M(z) \leftrightarrow^* N(z)$ for a fresh variable z, and by freshness $\mu x.M(x) = \mu z.M(z) \leftrightarrow^* \mu z.N(z) = \mu y.N(y)$.

The case for F follows straightforwardly from the induction hypothesis (twice), closure of convertibility under contexts, and transitivity.

Of course, $\underline{n} \leftrightarrow^* \underline{n}$ and $\mathsf{c} \leftrightarrow^* \mathsf{c}$ follow from reflexivity of $\leftrightarrow^*$. □

To show decidability of the proof system of Figure 16, we show that there are only finitely many distinct terms along a path while searching for a proof of $M = N$. To that end, we consider the set of terms reachable from M and N by means of *root*-reduction in the HRS $\mu\pi$, comprising besides the μ-rule also the π (projection) rules $\mu x.M(x) \to M(\underline{n})$ with n not in M and $\mathsf{F}(M_1, M_2) \to M_i$. Clearly, all terms in a proof search of $M = N$ are reachable by $\mu\pi$-root-reduction from either M or N[9] and we will show that although $\mu\pi$-root-reduction need not be terminating (e.g. $\mu x.x$ reduces to itself), any infinite such reduction must contain a cycle.

First, observe that after a μ-step $\mu x.M \to M[\![x{:=}\mu x.M]\!]$ the substitution part $[\![x{:=}\mu x.M]\!]$ only plays a 'passive' role: any reduct of $M[\![x{:=}\mu x.M]\!]$ is a $[\![x{:=}\mu x.M]\!]$-instance of a reduct of M. The only way the substitution can be 'activated' is when the reduction 'collapses' the term M to the single variable x, which would then give rise to a cycle $\mu x.M \overset{\varepsilon}{\to}_{\mu\pi} M[\![x{:=}\mu x.M]\!] \overset{\varepsilon}{\twoheadrightarrow}_{\mu\pi} x[\![x{:=}\mu x.M]\!] = \mu x.M$.

This leads to the higher-order version of Lemma 25 (note that the problematic second case of the lemma can be dispensed with here).

Lemma 55 ($\boldsymbol{\mu}$-finiteness). *If $M[\![x{:=}N]\!] \overset{\varepsilon}{\twoheadrightarrow}_{\mu\pi} K$ is a $\mu\pi$-root-reduction, then the reduction either*

(i) has a prefix which is a $[\![x{:=}N]\!]$-instance of a root-reduction $M \overset{\varepsilon}{\twoheadrightarrow}_{\mu\pi} x$; or

(ii) is a $[\![x{:=}N]\!]$-instance of a root-reduction $M \overset{\varepsilon}{\twoheadrightarrow}_{\mu\pi} L$.

Proof. The proof is by induction on the length of the reduction. If the length is 0, then case (ii) holds by setting $L = M$. Otherwise, $M[\![x{:=}N]\!] \overset{\varepsilon}{\twoheadrightarrow}_{\mu\pi} K' \overset{\varepsilon}{\to}_{\mu\pi} K$ for some term K' and we have by the induction hypothesis for the reduction without its final step, either

[9] This would even hold when n were required to be least in the μ-projection rule, as in Figure 16. For technical reasons which will become clear in the proof of Lemma 55, we do not require that just yet.

(i) it has a prefix which is a $[\![x{:=}N]\!]$-instance of a reduction $M \overset{\varepsilon}{\twoheadrightarrow}_{\mu\pi} x$. Then the same holds for the reduction extended by the step $K' \overset{\varepsilon}{\rightarrow}_{\mu\pi} K$; or

(ii) it is a $[\![x{:=}N]\!]$-instance of a root-reduction $M \overset{\varepsilon}{\twoheadrightarrow}_{\mu\pi} L'$. Distinguish cases on L' (which cannot be a constant or numeral since these are irreducible):

   (y) Then, since variables are normal forms, we must have $x = y$ and $N = K'$, and we are in case (i);

   ($\mu y.L_1$) By the variable convention we may assume $x \neq y$ and y not free in N, so $K' = \mu y.L_1[\![x{:=}N]\!]$.
   If $K' \overset{\varepsilon}{\rightarrow}_{\mu\pi} L_1[\![x{:=}N]\!][\![y{:=}\underline{n}]\!] = K$ with n not in $L_1[\![x{:=}N]\!]$ then $K = L_1[\![y{:=}\underline{n}]\!][\![x{:=}N]\!]$ by the assumption that y not free in N, and taking $L = L_1$ brings us in case (ii) as $\mu y.L_1 \overset{\varepsilon}{\rightarrow}_{\mu\pi} L_1[\![y{:=}\underline{n}]\!]$ since n not in L_1.[10]
   If $K' \overset{\varepsilon}{\rightarrow}_{\mu\pi} L_1[\![x{:=}N]\!][\![y{:=}K']\!] = K$ then by the substitution lemma $K = L_1[\![y{:=}L']\!][\![x{:=}N]\!]$ and setting $L = L_1[\![y{:=}L']\!]$ yields case (ii) again;

   ($\mathsf{F}(L_1, L_2)$) Then $K' = \mathsf{F}(L_1[\![x{:=}N]\!], L_2[\![x{:=}N]\!]) \overset{\varepsilon}{\rightarrow}_{\mu\pi} L_i[\![x{:=}N]\!] = K$ and we are in case (ii) setting $L = L_i$. □

Remark 56. It is easy to see that this factorization lemma for root-reduction holds in fact for all (higher-order) recursive program schemes [**?**], i.e. higher-order rewrite systems such that each left-hand side of a rule consists of a single (higher-order) function symbol applied to (higher-order) variables, of which $\mu\pi$ is an example.

Next, observe that although each π-step $\mu x.M \rightarrow M(\underline{n})$ introduces a numeral n which is fresh for M, only a bounded number of such is ever needed.

Lemma 57 (π-finiteness). *For every closed μ-term there is a bound on the number of numerals in its $\mu\pi$-reducts.*

Proof. First, note that if $M \overset{\varepsilon}{\twoheadrightarrow}_{\mu\pi} N$, then $M \twoheadrightarrow_{\mu} L \twoheadrightarrow_{\pi} N$ where the first reduction is *arbitrary* μ-reduction and the second a π-root-reduction. This holds since in any HRS, projection onto subterms can be postponed until after ordinary (here: μ) reduction steps. The condition on n for the projection of a μ-binder is preserved since ordinary reduction steps do not introduce numerals (nor erase them for that matter).

Next, note that any L as above is a closed μ-term again, as μ-reduct of the closed μ-term M. Hence each numeral $\underline{n}$ in the term N is either already present in M or generated by some projection step from $\mu x_n.M(x_n)$ onto $M(n)$ along $L \twoheadrightarrow_{\pi} N$, and these projections give rise to a chain $\mu x_{n_1}, x_{n_1}, \ldots, \mu x_{n_k}, x_{n_k}$ in L. Since the length of the chains in M is bounded and by (the proof of) Lemma 17 the length of chains does not increase along the μ-reduction to L, we conclude. □

[10] If n were least not in $L_1[\![x{:=}N]\!]$, then it would not need to be least not in L_1.

Theorem 58. $\leftrightarrow_\mu^*$ *is decidable, for closed μ-terms.*

Proof. By Lemma 54 it suffices to show that we can decide whether $\vdash M = N$ for closed μ-terms M, N. We claim that the sets of $\mu\pi$-root-reducts of M and N, where n is required to be *least* in the condition on the μ-projection rule, are finite, from which the result follows by performing an exhaustive search through these sets in the proof system of Figure 16.

We will prove the more general claim that the set of $\mu\pi$-root-reducts of a closed term M, denoted by $\mathcal{SC}(M)$, is finite, where n is required to be *below* k in the condition on the μ-projection rule, with k the bound obtained from Lemma 57. This claim is more general indeed as k is an upper bound on the number of distinct numerals in $\mu\pi$-root-reducts of M, so an upper bound on the least numeral not in any such given reduct. The proof is by induction on the size of M and cases on its shape.

($\underline{n}$) $\mathcal{SC}(\underline{n}) = \{\underline{n}\}$.

($\mu x.M$) $\mathcal{SC}(\mu x.M) \subseteq \{\mu x.M\} \cup \mathcal{SC}(M)[x{:=}\mu x.M] \cup \mathcal{SC}(M[\![x{:=}\underline{n}]\!])$ with n below k and not occurring in M. To see this consider a $\mu\pi$-root-reduction of minimal length $\mu x.M \overset{\varepsilon}{\twoheadrightarrow}_{\mu\pi} N$ witnessing $N \in \mathcal{SC}(\mu x.M)$.

If the reduction is empty, then clearly its final term is an element of the first disjunct.

If the reduction starts with a μ-step $\mu x.M \overset{\varepsilon}{\to}_{\mu\pi} M[x{:=}\mu x.M]$, then we conclude from applying Lemma 55 to its suffix, that all reducts are in the second disjunct. Note that case (i) of the lemma cannot occur as it would give rise to a cycle on $\mu x.M$, contradicting minimality of the witnessing reduction.

If the reduction starts with a π-step, $\mu x.M \overset{\varepsilon}{\to}_{\mu\pi} M[\![x{:=}\underline{n}]\!]$ with n below k and not in M, then all reducts are in the third disjunct.

In each case we conclude by the induction hypothesis for M.

($\mathsf{F}(M_1, M_2)$) $\mathcal{SC}(\mathsf{F}(M_1, M_2)) = \{\mathsf{F}(M_1, M_2)\} \cup \mathcal{SC}(M_1) \cup \mathcal{SC}(M_2)$ and we conclude by the induction hypothesis for the M_i. □

Decidability on open μ-terms is obtained as an easy corollary, closing the terms first by substituting suitable fresh numerals for the free variables.

Remark 59. This higher-order proof of decidability of μ-convertibility is again (only) based on the notion of *chain* (Lemma 57 relies on Lemma 17), underscoring the importance of the latter notion. Its connexion to the notions of *paths* in λ-calculus and *games* in semantics seems worthwhile to investigate.

9. Deciding μ-convertibility using regular languages

We now present an alternative proof of the decidability of weak μ-equality, based on a totally different intuition, namely, the regular nature of the set of reducts of a μ-term. The proof proceeds in the following steps:

(i) For every capture-avoiding μ-term M we construct a regular tree grammar $\mathcal{G}_M$ generating the set of reducts of M (with respect to standard reduction without α-conversion).

(ii) Given a regular tree grammar $\mathcal{G}$ generating a set of μ-terms $T \subseteq \mathit{Ter}(\mu)$ over a finite set of binders $\mathbb{B}$, we construct a regular tree grammar $\mathcal{G}^\alpha$ generating the closure of T under α-equivalence over the binders $\mathbb{B}$.

(iii) Then weak μ-equality of two μ-terms M and N boils down to the question: $\mathcal{L}(\mathcal{G}_M^\alpha) \cap \mathcal{L}(\mathcal{G}_N^\alpha) \neq \varnothing$? This problem is known to be decidable [**?**]. Actually it suffices to apply α-conversion to one of the terms: $\mathcal{L}(\mathcal{G}_M^\alpha) \cap \mathcal{L}(\mathcal{G}_N) \neq \varnothing$?

We begin with a definition of regular tree grammars.

Definition 60. A *regular tree grammar* $\mathcal{G}$ is a quadruple $\mathcal{G} = (V, \Sigma, S, R)$ where

- V is a set of nonterminals,
- Σ is a finite ranked alphabet (disjoint form N),
- $S \in V$ is the *start symbol*, and
- R is a set of rules of the form $v \to t$ with $v \in V$ and $t \in \mathit{Ter}(\Sigma \cup V, \varnothing)$.

Here $\mathit{Ter}(\Sigma \cup V, \varnothing)$ is the set of ground terms over $\Sigma \cup V$, see further [**?**].

The *language* $\mathcal{L}(\mathcal{G})$ *of* $\mathcal{G}$ is the set of terms $\mathcal{L}(\mathcal{G}) \subseteq \mathit{Ter}(\Sigma \cup V)$ defined by:

$$\mathcal{L}(\mathcal{G}) = \{t \mid t \in \mathit{Ter}(\Sigma, \varnothing),\ S \to_R^* t\}\ ,$$

that is, the set of nonterminal-free reducts of S. Here $\to_R$ is standard term rewriting with respect to the TRS (Σ, R), that is, we may replace left-hand sides of rules with the corresponding right-hand sides within arbitrary contexts.

We use regular tree grammars only for languages of μ-terms. For this reason, for the remainder of this section, we fix Σ to be the signature of μ-terms as defined in Definition 1 with the difference, that we here regard μ-term variables x, y, ... as constant symbols in Σ. Moreover, we extend Σ with the nonterminal symbols V. Since we consider only term languages, and no word languages, we will speak of (regular) languages and grammars as shorthand for (regular) tree languages and tree grammars, respectively.

Step (i): a regular grammar for μ-reducts

For capture-avoiding terms $M \in \mathit{Ter}(\mu)$, we construct a regular grammar $\mathcal{G}_M$ that generates the language of μ-reducts of M (with respect to α-free standard reduction):

Definition 61. Let $M \in \mathit{Ter}(\mu)$ be a capture-avoiding μ-term. We define the regular grammar $\mathcal{G}_M = (V, \Sigma, S, R)$ as follows. For every $N \in \mathsf{SC}(M)$, let V_N

be a fresh constant symbol. Let V be the set of symbols V_N with $N \in \mathsf{SC}(M)$, $S = \mathsf{V}_M$ the start symbol, and let R consist of the following rules:

$$
\begin{aligned}
\mathsf{V}_{\mu x.N} &\to \mathsf{V}_{N[x:=\mu x.N]} && \text{whenever } \mu x.N \in \mathsf{SC}(M) && (2)\\
\mathsf{V}_{\mu x.N} &\to \mu x.\mathsf{V}_N && \text{whenever } \mu x.N \in \mathsf{SC}(M) && (3)\\
\mathsf{V}_{\mathsf{F}(N,N')} &\to \mathsf{F}(\mathsf{V}_N, \mathsf{V}_{N'}) && \text{whenever } \mathsf{F}(N,N') \in \mathsf{SC}(M) && (4)\\
\mathsf{V}_x &\to x && \text{for variables and constants } x \in \mathsf{SC}(M) && (5)
\end{aligned}
$$

Observe that the rules model standard reduction: rule (2) allows for μ-steps at the current position, (3) and (4) move the rewriting activity (or snow line) downwards, that is, extending the frozen prefix. The rules 5 generate μ-term constants or variables (which are both regarded as constants from the first-order point of view of tree grammars).

Example 62. Let $M \equiv \mu x.\mu y.\mathsf{F}(x,y)$, then $\mathcal{G}_M$ consists of the rules:

$$
\begin{aligned}
\mathsf{V}_{\mu x.\mu y.\mathsf{F}(x,y)} &\to_{(2)} \mathsf{V}_{\mu y.\mathsf{F}(\mu x.\mu y.\mathsf{F}(x,y),y)} & \mathsf{V}_{\mu y.\mathsf{F}(x,y)} &\to_{(2)} \mathsf{V}_{\mathsf{F}(x,\mu y.\mathsf{F}(x,y))}\\
\mathsf{V}_{\mu x.\mu y.\mathsf{F}(x,y)} &\to_{(3)} \mu x.\mathsf{V}_{\mu y.\mathsf{F}(x,y)} & \mathsf{V}_{\mu y.\mathsf{F}(x,y)} &\to_{(3)} \mu y.\mathsf{V}_{\mathsf{F}(x,y)}\\
\mathsf{V}_{\mu y.\mathsf{F}(\mu x.\mu y.\mathsf{F}(x,y),y)} &\to_{(3)} \mu y.\mathsf{V}_{\mathsf{F}(\mu x.\mu y.\mathsf{F}(x,y),y)} & \mathsf{V}_{\mathsf{F}(x,y)} &\to_{(4)} \mathsf{F}(\mathsf{V}_x, \mathsf{V}_y)\\
\mathsf{V}_{\mathsf{F}(\mu x.\mu y.\mathsf{F}(x,y),y)} &\to_{(4)} \mathsf{F}(\mathsf{V}_{\mu x.\mu y.\mathsf{F}(x,y)}, \mathsf{V}_y) & \mathsf{V}_x &\to_{(5)} x\\
\mathsf{V}_{\mathsf{F}(x,\mu y.\mathsf{F}(x,y))} &\to_{(4)} \mathsf{F}(\mathsf{V}_x, \mathsf{V}_{\mu y.\mathsf{F}(x,y)}) & \mathsf{V}_y &\to_{(5)} y
\end{aligned}
$$

$$
\begin{aligned}
\mathsf{V}_{\mu y.\mathsf{F}(\mu x.\mu y.\mathsf{F}(x,y),y)} &\to_{(2)} \mathsf{V}_{\mathsf{F}(\mu x.\mu y.\mathsf{F}(x,y),\mu y.\mathsf{F}(\mu x.\mu y.\mathsf{F}(x,y),y))}\\
\mathsf{V}_{\mathsf{F}(\mu x.\mu y.\mathsf{F}(x,y),\mu y.\mathsf{F}(\mu x.\mu y.\mathsf{F}(x,y),y))} &\to_{(4)} \mathsf{F}(\mathsf{V}_{\mu x.\mu y.\mathsf{F}(x,y)}, \mathsf{V}_{\mu y.\mathsf{F}(\mu x.\mu y.\mathsf{F}(x,y),y)})
\end{aligned}
$$

where the start symbol of $\mathcal{G}_M$ is $\mathsf{V}_{\mu x.\mu y.\mathsf{F}(x,y)}$.

Then consider the following standard reduction:

$$\underline{\mu x}.\mu y.\mathsf{F}(x,y) \to \mu y.\mathsf{F}(\mu x.\underline{\mu y}.\mathsf{F}(x,y),y) \to \mu y.\mathsf{F}(\mu x.\mathsf{F}(x,\mu y.\mathsf{F}(x,y)),y)$$

We can generate the reduct using the grammar $\mathcal{G}_M$ as follows:

$$
\begin{aligned}
\mathsf{V}_{\mu x.\mu y.\mathsf{F}(x,y)} &\to \mathsf{V}_{\mu y.\mathsf{F}(\mu x.\mu y.\mathsf{F}(x,y),y)} \to \mu y.\mathsf{V}_{\mathsf{F}(\mu x.\mu y.\mathsf{F}(x,y),y)}\\
&\to \mu y.\mathsf{F}(\mathsf{V}_{\mu x.\mu y.\mathsf{F}(x,y)}, \mathsf{V}_y) \to \mu y.\mathsf{F}(\mu x.\mathsf{V}_{\mu y.\mathsf{F}(x,y)}, \mathsf{V}_y)\\
&\to \mu y.\mathsf{F}(\mu x.\mathsf{V}_{\mu y.\mathsf{F}(x,y)}, y) \to \mu y.\mathsf{F}(\mu x.\mathsf{V}_{\mathsf{F}(x,\mu y.\mathsf{F}(x,y))}, y)\\
&\to \mu y.\mathsf{F}(\mu x.\mathsf{F}(\mathsf{V}_x, \mathsf{V}_{\mu y.\mathsf{F}(x,y)}), y) \to \mu y.\mathsf{F}(\mu x.\mathsf{F}(x, \mathsf{V}_{\mu y.\mathsf{F}(x,y)}), y)\\
&\to \mu y.\mathsf{F}(\mu x.\mathsf{F}(x, \mu y.\mathsf{V}_{\mathsf{F}(x,y)}), y) \to \mu y.\mathsf{F}(\mu x.\mathsf{F}(x, \mu y.\mathsf{F}(\mathsf{V}_x, \mathsf{V}_y)), y)\\
&\to \mu y.\mathsf{F}(\mu x.\mathsf{F}(x, \mu y.\mathsf{F}(x, \mathsf{V}_y)), y) \to \mu y.\mathsf{F}(\mu x.\mathsf{F}(x, \mu y.\mathsf{F}(x,y)), y)
\end{aligned}
$$

Note that the derivation simulates the standard reduction, and makes the steps for freezing the prefix (pushing the snow line down) explicit.

Proposition 63. *For every capture-avoiding μ-term $M \in Ter(\mu)$, the language $\mathcal{L}(\mathcal{G}_M)$ coincides with the set of all μ-reducts of M modulo α-equivalence.*

Proof. We define an interpretation $[\cdot]$, mapping grammar terms to μ-terms:

$$\begin{aligned}
[\mathsf{V}_N] &= N && \text{for all } N \in \mathsf{SC}(M)\\
[\mu x.N] &= \mu x.[N] &&\\
[\mathsf{F}(N,N')] &= \mathsf{F}([N],[N']) &&\\
[x] &= x && \text{for variables and constants } x
\end{aligned}$$

A position p in a μ-term $[N]$ is called *active* if $N|_M$ is a nonterminal symbol.

We show that the grammar rules exactly simulate standard reduction on the interpretation. Note that the grammar rules of form 3 and 4 preserve the interpretation while moving the activity (snow line) downwards. In particular, we have $\mathsf{V}_N \to^* N$ for every $N \in \mathsf{SC}(M)$ using the rules of the form (3), (4), and (5). The rules of shape 2 enable μ-steps at an active position. That is, assume we have a grammar term N such that $N|_p = \mathsf{V}_{\mu x.P}$, (then $[N]|_p = \mu x.P$). An application of rule 2 at position p in N yields a term N' with $[N']|_q = P[x{:=}\mu x.P]$; exactly modeling the ($\alpha$-free) μ-unfolding in $[N]$ at position p. $\square$

Step (ii): a regular grammar for α-conversion

We show that the closure of a regular language of μ-terms, under α-conversion over a finite set of binder names $\mathbb{B}$, is a regular language again. The idea of the construction is as follows. Let L be a regular language of μ-terms over a finite set of binder names $\mathbb{B}$ given in form of a regular tree grammar $\mathcal{G}_L$. Then we construct a regular grammar $\mathcal{G}_L^\alpha$ for the closure of L under α-conversion. For this purpose we label the grammar variables with a map $\sigma : \mathbb{B} \to \mathbb{B}$ representing the renaming, and a set $\dagger \subseteq \mathbb{B}$ of 'forbidden' variables whose occurrence would cause a name clash (capturing by a wrong binder).

Without loss of generality we may assume that regular grammars are *normalized*, that is, all rules are of the form $\mathsf{V} \to f(\mathsf{V}_1, \ldots, \mathsf{V}_n)$; see [**?** , Prop 2.1.4].

Definition 64. Let $\mathcal{G} = (V, \Sigma, S, R)$ be a normalized regular tree grammar for a language of μ-terms over a finite set of binder $\mathbb{B}$. We define the regular grammar $\mathcal{G}^\alpha = (V_\alpha, \Sigma, S_\alpha, R_\alpha)$ as follows. For every symbol $\mathsf{V} \in V$, map $\sigma : \mathbb{B} \to \mathbb{B}$, and set $\dagger \subseteq \mathbb{B}$, let $\mathsf{V}_{\sigma,\dagger}$ be a fresh symbol. Let V_α be the set of these symbols. We choose $S_\alpha = S_{id,\varnothing}$ as the start symbol of $\mathcal{G}^\alpha$, and let R_α consist of the rules:

(i) $\mathsf{V}_{\sigma,\dagger} \to \sigma(x)$ (*renaming*) whenever $\mathsf{V} \to x \in R$ and $x \notin \dagger$,

(ii) $\mathsf{V}_{\sigma,\dagger} \to \mathsf{F}(\mathsf{V}'_{\sigma,\dagger}, \mathsf{V}''_{\sigma,\dagger})$ (*propagation*) whenever $\mathsf{V} \to \mathsf{F}(\mathsf{V}', \mathsf{V}'') \in R$, and

(iii) $\mathsf{V}_{\sigma,\dagger} \to \mu y(\mathsf{V}'_{\sigma',\dagger'})$ (*picking*) whenever $\mathsf{V} \to \mu x(\mathsf{V}') \in R$, and $y \in \mathbb{B}$, where $\sigma'(z) = \sigma(z)$ for all $z \neq x$, $\sigma'(x) = y$, and $\dagger' = (\dagger \cup \sigma^{-1}(y)) \setminus \{x\}$.

Example 65. Let G consist of the rules:

$$\mathsf{V}_1 \to \mu x.\mathsf{V}_2 \qquad \mathsf{V}_2 \to \mu y.\mathsf{V}_3 \qquad \mathsf{V}_3 \to \mathsf{F}(\mathsf{V}_4, \mathsf{V}_5) \qquad \mathsf{V}_4 \to x \qquad \mathsf{V}_5 \to y$$

with start variable V_1. Note that G generates the term $\mu x.\mu y.\mathsf{F}(x, y)$.

Let $\mathbb{B} = \{x, y\}$ be the set of binders. Then the reachable rules of $\mathcal{G}^\alpha$ are:

$$
\begin{aligned}
\mathsf{V}_{1,\{x\mapsto x,y\mapsto y\},\varnothing} &\to \mu x.\mathsf{V}_{2,\{x\mapsto x,y\mapsto y\},\varnothing} \\
\mathsf{V}_{1,\{x\mapsto x,y\mapsto y\},\varnothing} &\to \mu y.\mathsf{V}_{2,\{x\mapsto y,y\mapsto y\},\{y\}} \\
\mathsf{V}_{2,\{x\mapsto x,y\mapsto y\},\varnothing} &\to \mu y.\mathsf{V}_{3,\{x\mapsto x,y\mapsto y\},\varnothing} \\
\mathsf{V}_{2,\{x\mapsto x,y\mapsto y\},\varnothing} &\to \mu x.\mathsf{V}_{3,\{x\mapsto x,y\mapsto x\},\{x\}} \\
\mathsf{V}_{2,\{x\mapsto y,y\mapsto y\},\{y\}} &\to \mu y.\mathsf{V}_{3,\{x\mapsto y,y\mapsto y\},\{x\}} \\
\mathsf{V}_{2,\{x\mapsto y,y\mapsto y\},\{y\}} &\to \mu x.\mathsf{V}_{3,\{x\mapsto y,y\mapsto x\},\varnothing} \\
\mathsf{V}_{3,\sigma,\dagger} &\to \mathsf{F}(\mathsf{V}_{4,\sigma,\dagger}, \mathsf{V}_{5,\sigma,\dagger}) && \text{for the above } \sigma \text{ and } \dagger \\
\mathsf{V}_{4,\{x\mapsto x,y\mapsto y\},\varnothing} &\to x & \mathsf{V}_{5,\{x\mapsto x,y\mapsto y\},\varnothing} &\to y \\
& & \mathsf{V}_{5,\{x\mapsto x,y\mapsto x\},\{x\}} &\to x \\
& & \mathsf{V}_{5,\{x\mapsto y,y\mapsto y\},\{x\}} &\to y \\
\mathsf{V}_{4,\{x\mapsto y,y\mapsto x\},\varnothing} &\to y & \mathsf{V}_{5,\{x\mapsto y,y\mapsto x\},\varnothing} &\to x
\end{aligned}
$$

Note that the grammar $\mathcal{G}^\alpha$ can generate both valid α-conversions $\mu x.\mu y.\mathsf{F}(x, y)$ and $\mu y.\mu x.\mathsf{F}(y, x)$. However, if we map both variables x and y to the same value, then we obtain a term that contains either the nonterminal $\mathsf{V}_{4,\{x\mapsto x,y\mapsto x\},\{x\}}$ or $\mathsf{V}_{4,\{x\mapsto y,y\mapsto y\},\{x\}}$ for which there exist no production rules.

Proposition 66. *$\mathcal{L}(\mathcal{G}^\alpha)$ is exactly the closure of $\mathcal{L}(G)$ under α-conversion over the finite set of binders $\mathbb{B}$.*

Proof. Whenever V is start variable of G, then $\mathsf{V}_{id,\varnothing}$ is start variable of $\mathcal{G}^\alpha$. Here, id expresses that the free variables should not be renamed, and $\varnothing$ that there are initially no name clashes.

The rule (ii) simply propagates the α-conversion down. The interesting case is rule (iii), the case of μx. We pick the renaming y for x ($\mathcal{G}^\alpha$ contains one rule for every $y \in \mathbb{B}$), and accordingly update the renaming function σ for the subterm. In case there are other variables mapped to y, then we need to make sure that these variables do not occur free in the subterm (since otherwise they would be captured by the wrong binder). We update the set of 'forbidden' variables $\dagger$ to include all variables mapped to y, that is, $\dagger \cup \sigma^{-1}(y)$; from this set we remove x since it is bound by the currently innermost binder (and may occur free). The rule (ii) executes the renaming. Note that the renaming rules require that $x \notin \dagger$, that is, there are no production rules in case of a name clash. □

Step (iii): deciding weak μ-equality

Combining the Propositions 63 and 66, we can reduce weak μ-equality to the emptiness problem of the intersection of regular tree languages. In this way we obtain a decision procedure for weak μ-equality:

Theorem 67. *The following problem is decidable:*

– *Input: two μ-terms M and N.*

– *Answer: are M and N convertible?*

Proof. The decision procedure proceeds in the following steps:

(i) use α-conversion to obtain capture-avoiding μ-terms $M' \equiv_\alpha M$, $N' \equiv_\alpha N$,

(ii) construct the grammars $\mathcal{G}_{M'}$ and $\mathcal{G}_{N'}$ generating the language of μ-reducts of M' and N', respectively,

(iii) construct the grammar $\mathcal{G}^{\alpha}_{M'}$ over the set of binders of M' and N', and

(iv) answer *yes* if $\mathcal{L}(\mathcal{G}^{\alpha}_{M'}) \cap \mathcal{L}(\mathcal{G}_{N'}) \neq \varnothing$, and *no*, otherwise.

Thus, we have reduced the problem of deciding whether two μ-terms are convertible to a problem on regular tree languages. For regular tree languages, the intersection is regular (a regular tree grammar can be computed), and the emptiness is decidable (given a regular tree grammar), see [**?**]. Note that we also use that μ-unfolding is an orthogonal higher-order term rewrite system, and thereby is confluent. That is, for deciding whether two terms are convertible it suffices to check whether they have a common reduct, see [**?**]. □

10. Further Results

The following result states that reachability of μ-reduction with α-conversion is decidable:

Theorem 68. *Reachability with respect to $\twoheadrightarrow_{\mu/\alpha}$ is decidable, that is, on the input of two μ-terms M and N it is decidable whether $M \twoheadrightarrow_{\mu/\alpha} N$.*

Proof. Choose a capture-avoiding term $M' \equiv_\alpha M$. Then $M \twoheadrightarrow_{\mu/\alpha} N$ if and only if $N \in \mathcal{L}(\mathcal{G}^{\alpha}_{M'})$, the set of reducts of M', which is a decidable property. □

We remark that the theorem can also be proven using an easy adaptation of the proof system from Figure 12 by starting from the equation $M = N$ and restricting the proof system by disallowing μ-steps in the right-hand side of the equations.

We also obtain decidability of α-free μ-convertibility, that is, convertibility with respect to $\to_\mu$. We briefly sketch the proof. Not every occurrence of the symbol μ corresponds to a $\to_\mu$-redex position since $\to_\mu$-steps are forbidden if a variable would be captured by the substitution. The following example shows that μ-steps may 'activate' redex positions:

Example 69. We consider $M \equiv \mu x.\mathsf{F}(z, \mu z.x)$. The term M does not allow for a $\to_\mu$-step at the root as the variable z would be captured. However, the inner rewrite step $\mu x.\mathsf{F}(z, \mu z.x) \to_\mu \mu x.\mathsf{F}(z, x)$ 'enables' the redex at the root. As a consequence, the following rewrite sequence:

$$\mu x.\mathsf{F}(z, \mu z.x) \to_\mu \mu x.\mathsf{F}(z, x) \to_\mu \mathsf{F}(z, \mu x.\mathsf{F}(z, x))$$

cannot be transformed into a standard reduction.

However, redex 'activation' is not arbitrary. The following lemma states that redexes can be 'activated' but not 'deactivated', that is, whenever a μ-position is a redex occurrence, then all its residuals are redex occurrences as well:

Lemma 70. *Let M, N be μ-terms with a step $\sigma : M \to_\mu N$ and $\mathcal{R}$ a redex occurrence in M. Then all residuals of $\mathcal{R}$ in N after σ are $\to_\mu$-redex occurrences.* □

As a consequence $\to_\mu$ can be viewed as conditional higher order rewriting system with stable conditions (redexes stay redexes unless reduced). Then confluence of $\to_\mu$ follows from known results on higher order rewriting [**?**]:

Proposition 71. *The α-conversion-free μ-reduction $\to_\mu$ is confluent.* □

Remark 72. We remark that since the system $\to_\mu$ is orthogonal and residuals of redex occurrences are redex occurrences again, we can alternatively establish the diamond property for multi-steps (developments). The diamond property then immediately implies confluence of $\to_\mu$.

To apply the proof system from Sections 7, or the decision procedure based on regular languages from Section 9, we need the property that reductions can be turned into standard reductions. As we have seen in Example 69, for $\to_\mu$ this is in general not possible. However, we can recover the property using the following lemma:

Lemma 73. *Let M be a μ-term, and let τ be the occurrence in M of a subterm with root μ that is not a redex. Moreover, let $\sigma : M \to_\mu N$ be a step at position p. If a descendant of τ after σ* is a redex, *then $M|_p \equiv \mu x.M'$ with $x \notin \mathsf{FV}(M')$.* □

Roughly speaking, the lemma states that only steps of the form $\mu x.M' \to_\mu M'$ (that is, with $x \notin \mathsf{FV}(M')$), that is, the removal of vacuous μ-binders can activate redex positions. The proof of the lemma is a simple exercise. Then we can recover standardization for those terms that do not contain such redexes:

Proposition 74. *Let M be a μ-term not containing subterms of the from $\mu x.N$ with $x \notin \mathsf{FV}(N)$. Then for every reduction $M \to_\mu{}^* M'$ there exists a standard reduction $M \to_\mu{}^* M'$.*

Proof. As M does not contain subterms of the form $\mu x.N$ with $x \notin \mathsf{FV}(N)$ the same property holds also for all reducts of M. Hence all ancestors and descendants of redexes are redexes. Therefore standardization can be proved with the customary argument of swapping steps [**?**], [**?** , Sect. 8.5.3, p. 371]. □

Finally, we obtain decidability of $\to_\mu$-conversion as follows. Let M and N be given. Let the terms M' and N' be obtained from M and N by removing vacuous μ-binders. Then $M \leftrightarrow_\mu^* N$ if and only if $M' \leftrightarrow_\mu^* N'$. Now we can decide $M' \leftrightarrow_\mu^* N'$ by using slightly adapted systems from Section 7 or Section 9. That is, we need to restrict the μ-unfolding rule in these systems to $\to_\mu$-unfoldings, and hence in particular, without variable capture. Consequently we obtain:

Theorem 75. *Convertibility with respect to $\rightarrow_\mu$-reduction is decidable.* $\square$

We remark that the above results (decidability of $\rightarrow_{\mu/\alpha}$-reachability as well as $\rightarrow_\mu$-convertibility) do not immediately imply decidability of $\rightarrow_\mu$-reachability. The reason is that for reachability $M \rightarrow_\mu{}^* N$ we cannot simply drop all vacuous μ-binders from M and N.

10.1. Upward joinability for μ-reduction

In this subsection we establish decidability for the problem of upward-joinability with respect to $\rightarrow_{\mu/\alpha}$: Given two μ-terms M and N, does there exist a μ-term P such that $M \twoheadleftarrow_{\mu/\alpha} P \twoheadrightarrow_{\mu/\alpha} N$?

Upward-joinability with respect to $\rightarrow_{\mu/\alpha}$ is equivalent to joinability with respect to $\rightarrow_{\mathrm{fold}/\alpha}$, the converse rewrite relation of $\rightarrow_{\mu/\alpha}$. Note that $\rightarrow_{\mathrm{fold}/\alpha} = \equiv_\alpha \cdot \rightarrow_{\mathrm{fold}} \cdot \equiv_\alpha$ where $\rightarrow_{\mathrm{fold}}$ is induced by the conditional rewrite rule:

$$M[x{:=}\mu x.M] \;\rightarrow\; \mu x.M \qquad (\text{if } \mu x.M \text{ is free for } x \text{ in } M).$$

We show that joinability with respect to this rewrite relation is decidable, while it is not confluent. (Note that convertibility with respect to $\rightarrow_{\mathrm{fold}/\alpha}$ coincides with convertibility with respect to $\rightarrow_{\mu/\alpha}$, and that therefore decidability of this relation between μ-terms follows from the results in Sections 7, 8, and 9.)

Let $\rightarrow$ be a rewrite relation. We write $M \uparrow N$ (and respectively, $M \downarrow N$) for the statement that *M and N are upward-joinable* (*M and N are joinable*) *with respect to* $\rightarrow$: there exists P such that $M \twoheadleftarrow P \twoheadrightarrow N$ (such that $M \twoheadrightarrow P \twoheadleftarrow N$).

Both of the rewrite relations $\rightarrow_{\mu/\alpha}$ and $\rightarrow_{\mathrm{fold}/\alpha}$ can be split into a 'proper' part, steps in which actual unfoldings or, respectively, foldings take place, and into an 'improper' part, steps in which vacuous bindings are eliminated or introduced, respectively. For this we first split the rewrite relations $\rightarrow_\mu$ and $\rightarrow_{\mathrm{fold}}$ according to:

$$\rightarrow_\mu \;=\; \rightarrow_{\mu\text{-p}} \uplus \rightarrow_{\mathrm{vbE}} \qquad\qquad \rightarrow_{\mathrm{fold}} \;=\; \rightarrow_{\mathrm{fold\text{-}p}} \uplus \rightarrow_{\mathrm{vbI}} \tag{6}$$

into proper μ-reduction and 'vacuous-binder elimination' relations, and respectively, into proper folding reduction and 'vacuous-binder introduction' relations, which are induced by the following conditional rewrite rules:

$$\begin{array}{lll} \mu x.M \;\rightarrow_{\mu\text{-p}}\; M[x{:=}\mu x.M] & M[x{:=}\mu x.M] \;\rightarrow_{\mathrm{fold\text{-}p}}\; \mu x.M & (\text{if } x \in \mathsf{FV}(M)) \\ \mu x.M \;\rightarrow_{\mathrm{vbE}}\; M & M \;\rightarrow_{\mathrm{vbI}}\; \mu x.M & (\text{if } x \notin \mathsf{FV}(M)) \end{array}$$

(In the rules for $\rightarrow_{\mu\text{-p}}$ and $\rightarrow_{\mathrm{fold\text{-}p}}$ furthermore $\mu x.M$ has to be free for x in M.) Note that $\rightarrow_{\mathrm{fold\text{-}p}}$ and $\rightarrow_{\mathrm{vbI}}$ are the converses of $\rightarrow_{\mu\text{-p}}$ and $\rightarrow_{\mathrm{vbE}}$, respectively. The splittings (6) also induce splittings of $\rightarrow_{\mu/\alpha}$ into $\rightarrow_{\mu\text{-p}/\alpha}$ and $\rightarrow_{\mathrm{vbE}/\alpha}$, and of $\rightarrow_{\mathrm{fold}/\alpha}$ into $\rightarrow_{\mathrm{fold\text{-}p}/\alpha}$ and $\rightarrow_{\mathrm{vbI}/\alpha}$, the extensions of the rewrite relations $\rightarrow_{\mu\text{-p}}$, $\rightarrow_{\mathrm{vbE}}$, $\rightarrow_{\mathrm{fold\text{-}p}}$, and $\rightarrow_{\mathrm{vbI}}$ by α-conversion steps on both sides.

While we are interested in a property of the rewrite relations $\rightarrow_{\mu/\alpha}$ and $\rightarrow_{\mathrm{fold}/\alpha}$ here, the results in this subsection lend themselves better to formulations for rewrite relations on α-equivalence classes, which have smoother properties. For example, $\rightarrow_{\mathrm{vbE}/\alpha}$ is not confluent, because of reduction forks like

$\mu x.x \leftarrow_{\mathrm{vbE}/\alpha} \mu y.\mu x.x \rightarrow_{\mathrm{vbE}/\alpha} \mu z.z$ that lead to α-equivalent terms, but that cannot be joined by $\rightarrow_{\mathrm{vbE}/\alpha}$-steps; but a version of $\rightarrow_{\mathrm{vbE}/\alpha}$ on α-equivalence classes turns out to be confluent. Furthermore, $\rightarrow_{\text{fold-p}/\alpha}$ is non-confluent for the simple reason that there are 'trivial' reduction forks such as $M \leftarrow_{\text{fold-p}/\alpha} \mathsf{F}(M, M) \rightarrow_{\text{fold-p}/\alpha} M'$ leading to the α-equivalent $\rightarrow_{\text{fold-p}/\alpha}$-normal forms $M = \mu x.\mathsf{F}(x, x)$ and $M' = \mu y.\mathsf{F}(y, y)$ that cannot be joined by $\rightarrow_{\text{fold-p}/\alpha}$-rewrite steps. Such a trivial reason for non-confluence disappears for a version of $\rightarrow_{\text{fold-p}/\alpha}$ on α-equivalence classes. These examples may serve as indication why we will use auxiliary results on versions of the rewrite relations above defined for α-equivalence classes.

For every μ-term M, we denote by $[M] := \{N \mid N \equiv_\alpha M\}$ *the α-equivalence class of M*. The rewrite relations $\rightarrow_{\mu/\alpha}$, $\rightarrow_{\mu\text{-p}/\alpha}$, $\rightarrow_{\mathrm{vbE}/\alpha}$ as well as $\rightarrow_{\text{fold}/\alpha}$, $\rightarrow_{\text{fold-p}/\alpha}$ and $\rightarrow_{\mathrm{vbI}/\alpha}$ induce rewrite relations $\rightarrow_{[\mu]}$, $\rightarrow_{[\mu\text{-p}]}$, $\rightarrow_{[\mathrm{vbE}]}$ as well as $\rightarrow_{[\text{fold}]}$, $\rightarrow_{[\text{fold-p}]}$ and $\rightarrow_{[\mathrm{vbI}]}$ on α-equivalence classes: for example, $\rightarrow_{\mu/\alpha}$ induces the rewrite relation $\rightarrow_{[\mu]}$ by:

$$[M] \rightarrow_{[\mu]} [N] \;\; :\Longleftrightarrow \;\; M \rightarrow_{\mu/\alpha} N \qquad (\text{for all } M,\, N)\,.$$

Note that $\rightarrow_{\mathrm{vbE}}$ is strictly size-decreasing, and therefore terminating, which implies the same properties for $\rightarrow_{\mathrm{vbE}/\alpha}$ and $\rightarrow_{[\mathrm{vbE}]}$. Steps in the rewrite relation $\rightarrow_{\text{fold-p}/\alpha}$ turn out to be either strictly size-decreasing or to have the same source and target. This observation leads to the following proposition. (For its formulation, note that, for a rewrite relation $\rightarrow$, by $\rightarrow^+$ we mean the transitive closure of $\rightarrow$.)

Proposition 76. *If $[M] \rightarrow^+_{[\text{fold-p}]} [N]$, then either $\mathrm{size}(M) > \mathrm{size}(N)$ holds, or $M \equiv_\alpha N$.*

Proof. It suffices to prove that, for all μ-terms M and N, $M \rightarrow_{\text{fold-p}} N$ implies $\mathrm{size}(M) > \mathrm{size}(N)$ or $M \equiv_\alpha N$, because this makes it possible to show the statement of the proposition by induction on $\rightarrow_{[\text{fold-p}]}$-rewrite sequences.

Since $\rightarrow_{\text{fold-p}}$-steps are of the form $C[A[x{:=}\mu x.A]] \rightarrow_{\text{fold-p}} C[\mu x.A]$ with $x \in \mathsf{FV}(A)$, it remains to show that $\mathrm{size}(A[x{:=}\mu x.A]) > \mathrm{size}(\mu x.A)$ or $A[x{:=}\mu x.A] \equiv_\alpha \mu x.A$ whenever $x \in \mathsf{FV}(A)$. To show the latter, let A and x be arbitrary, but such that $x \in \mathsf{FV}(A)$. Note that $\mathrm{size}(A[x{:=}\mu x.A]) = 2 \cdot \mathrm{size}(A)$, and $\mathrm{size}(\mu x.A) = 1 + \mathrm{size}(A)$. Now if $\mathrm{size}(A) > 1$, then clearly $\mathrm{size}(A[x{:=}\mu x.A]) > \mathrm{size}(\mu x.A)$ holds; and if $\mathrm{size}(A) = 1$, then $\mu x.A = \mu x.x$ because x has to occur in A, and hence $A[x{:=}\mu x.A] = A \equiv_\alpha A$. $\square$

Lemma 77. *Upward-joinability $\uparrow_{[\mu\text{-p}]}$ w.r.t. $\rightarrow_{[\mu\text{-p}]}$ is decidable.*

Proof. In order to decide, for given μ-terms M and N, whether $[M] \uparrow_{[\mu\text{-p}]} [N]$ holds, it suffices to decide whether the sets A and B of reducts of, respectively, $[M]$ and $[N]$ under $\rightarrow_{[\text{fold-p}]} = \leftarrow_{[\mu\text{-p}]}$ have a non-empty intersection. Since, due to Proposition 76, A and B are finite, and for each member of A and B a term representative can be chosen, this decision can be taken by comparing the representatives of A with those of B modulo α-equivalence. $\square$

Proposition 78. $\to_{[\text{fold-p}]}$ *is not locally confluent, and hence, not confluent.*

Proof. Let $M = \mu y.\mathsf{F}(c, \mu x.\mathsf{F}(y, \mathsf{F}(c, x)))$ and $N = \mathsf{F}(M, \mathsf{F}(c, \mu x.\mathsf{F}(M, \mathsf{F}(c, x))))$. Then:

$$\mathsf{F}(M, M) \leftarrow_{\text{fold-p}} N \to_{\text{fold-p}} \mu x.\mathsf{F}(M, \mathsf{F}(c, x)) ,$$

and both of $\mathsf{F}(M, M)$ and $\mu x.\mathsf{F}(M, \mathsf{F}(c, x))$ are $\to_{\text{fold-p}/\alpha}$-normal forms, which are not α-equivalent. Consequently we obtain:

$$[\mathsf{F}(M, M)] \leftarrow_{[\text{fold-p}]} [N] \to_{[\text{fold-p}]} [\mu x.\mathsf{F}(M, \mathsf{F}(c, x))] ,$$

and the α-equivalence classes left and right are different $\to_{[\text{fold-p}]}$-normal forms. $\square$

Proposition 79. $\downarrow_{[\text{vbI}]} \;=\; \uparrow_{[\text{vbI}]}$*, and hence:* $\uparrow_{[\text{vbE}]} = \downarrow_{[\text{vbI}]} = \uparrow_{[\text{vbI}]} = \downarrow_{[\text{vbE}]}$*. Furthermore,* $\to_{[\text{vbI}]}$ *and* $\to_{[\text{vbE}]}$ *are confluent.*

Proof. Since $\to_{[\text{vbI}]}$ and $\to_{[\text{vbE}]}$ are each other's converse relations, it suffices to show $\downarrow_{[\text{vbI}]} \;=\; \uparrow_{[\text{vbI}]}$ to establish the first sentence.

Note that, for every rewrite relation $\to$ it holds that $\uparrow \subseteq \downarrow$ if and only if $\to$ is confluent; consequently, $\uparrow = \downarrow$ holds if and only if both $\to$ and $\leftarrow$ are confluent. Hence it remains to show that $\to_{[\text{vbI}]}$ and $\leftarrow_{[\text{vbI}]} \;=\; \to_{[\text{vbE}]}$ are confluent. But is easy to establish that both of these rewrite relations are sub-commutative[11], and hence confluent. $\square$

For every μ-term M, we denote by $\underline{M}_{\text{vbE}}$ the normal form of M with respect to $\to_{\text{vbE}}$-steps, that is, the result of removing all vacuous binders from M. Note that a μ-term M is a $\to_{\text{vbE}}$-normal form if and only if M is a $\to_{\text{vbE}/\alpha}$-normal form, and hence also, if and only if $[M]$ is a $\to_{[\text{vbE}]}$-normal form.

Lemma 80. *For all* μ*-terms* M*,* N *:*

$$[M] \uparrow_{[\mu]} [N] \iff [\underline{M}_{\text{vbE}}] \uparrow_{[\mu\text{-p}]} [\underline{N}_{\text{vbE}}] . \tag{7}$$

Proof. Since $\to_{[\text{fold}]} = \leftarrow_{[\mu]}$ and $\to_{[\text{fold-p}]} = \leftarrow_{[\mu\text{-p}]}$, (7) is equivalent to:

$$[M] \downarrow_{[\text{fold}]} [N] \iff [\underline{M}_{\text{vbE}}] \downarrow_{[\text{fold-p}]} [\underline{N}_{\text{vbE}}] . \tag{8}$$

it suffices to show (8).

"$\Rightarrow$": This direction can be shown by projecting a pair of joining $\to_{[\text{fold}]}$-reductions $[M] \twoheadrightarrow_{[\text{fold}]} [P] \twoheadleftarrow_{[\text{fold}]} [N]$ down to a pair of joining $\to_{[\text{fold-p}]}$-reductions $[\underline{M}_{\text{vbE}}] \twoheadrightarrow_{[\text{fold-p}]} [\underline{P}_{\text{vbE}}] \twoheadleftarrow_{[\text{fold-p}]} [\underline{N}_{\text{vbE}}]$ on α-equivalence classes of $\to_{\text{vbE}}$-normal forms.

[11] A rewrite relation $\to$ is called 'sub-commutative' if every pair of branching $\to$-steps can be joined by steps or empty steps: for all M, N, P, if $N \leftarrow M \to P$, then there exists Q such that $N \to^{=} Q \leftarrow^{=} P$, where $\to^{=} := \to \cup =$ is the reflexive closure of $\to$.

"$\Leftarrow$": This direction can be shown by lifting, for given μ-terms M and N, an assumed pair of joining $\to_{[\text{fold-p}]}$-reductions $[\underline{M}_{\text{vbE}}] \twoheadrightarrow_{[\text{fold-p}]} [P] \twoheadleftarrow_{[\text{fold-p}]} [\underline{N}_{\text{vbE}}]$ on $\to_{\text{vbE}/\alpha}$-normal forms up to a pair of reductions, $[M] \twoheadrightarrow_{[\text{fold}]} [\hat{P}_1]$ and $[N] \twoheadrightarrow_{[\text{fold}]} [\hat{P}_2]$ with $[\underline{\hat{P}_1}_{\text{vbE}}] = [\underline{\hat{P}_2}_{\text{vbE}}] = [P]$, which by using confluence of $\to_{[\text{vbI}]} = \leftarrow_{[\text{vbE}]}$ (see Proposition 79) can be extended to a joining pair $[M] \twoheadrightarrow_{[\text{fold}]} [\hat{P}_1] \twoheadrightarrow_{[\text{fold}]} [\hat{P}] \twoheadleftarrow_{[\text{fold}]} [\hat{P}_1] \twoheadleftarrow_{[\text{fold}]} [N]$ of $\to_{[\text{fold}]}$-reductions between $[M]$ and $[N]$.

$\square$

Proposition 81. $\to_{[\text{fold}]}$ *is not locally confluent, and hence, not confluent.*

Proof. In view of Lemma 80, the counterexample to local confluence of $\to_{[\text{fold-p}]}$ given in the proof of Proposition 78 can be used here as well. $\square$

Theorem 82. *Upward-joinability* $\uparrow_{[\mu]}$ *w.r.t.* $\to_{[\mu]}$ *is decidable.*

Proof. In order to decide, for given μ-terms M, N, whether $[M] \uparrow_{[\mu]} [N]$ holds, it suffices, due to Lemma 80, to decide whether $[\underline{M}_{\text{vbE}}] \uparrow_{[\mu\text{-p}]} [\underline{N}_{\text{vbE}}]$ holds. Since $\to_{\text{vbE}}$ is terminating, $\underline{M}_{\text{vbE}}$ and $\underline{N}_{\text{vbE}}$ can be produced effectively, and hence this decision can be to obtained by applying a decision algorithm for $\uparrow_{[\mu\text{-p}]}$, which exists due to Lemma 77. $\square$

Corollary 83. *Upward-joinability* $\uparrow_{\mu/\alpha}$ *with respect to* $\to_{\mu/\alpha}$ *is decidable.*

Proof. The corollary follows from the theorem observing that there is a correspondence via projection and lifting between $\to_{\text{fold}/\alpha}$- and $\to_{[\text{fold}]}$-rewrite sequences, and the fact that α-equivalent μ-terms are upward-joinable with respect to $\to_{\mu/\alpha}$: if $M_1 \equiv_\alpha M_2$, then we have $M_1 \to_{\text{vbI}/\alpha} \mu x.M_1 \leftarrow_{\text{vbI}/\alpha} M_2$ for some $x \notin \mathsf{FV}(M_1) = \mathsf{FV}(M_2)$. $\square$

Remark 84. Note that Proposition 81 blocks the way to a quick proof of weak μ-equality $=_{\mu/\alpha}$ that would be facilitated by Theorem 82 if $\to_{[\text{fold}]}$ were confluent.

Let us, for the sake of the argument, *assume* that $\to_{[\text{fold}]}$ is confluent. Let $=_{[\mu]}$ be the convertibility relation with respect to $\to_{[\mu]}$, or equivalently, with respect to $\to_{[\text{fold}]}$. Since $\to_{[\text{fold}]}$ is confluent, it follows that it is also Church–Rosser, which means that $=_{[\mu]} \subseteq \downarrow_{[\text{fold}]}$ holds, and implies $=_{[\mu]} = \downarrow_{[\text{fold}]}$. Since $\downarrow_{[\text{fold}]} = \uparrow_{[\mu]}$ is decidable by Theorem 82, it follows that $=_{[\mu]}$ is decidable as well. Since for all μ-terms M and N, $M =_{\mu/\alpha} N$ holds if and only if $[M] =_{[\mu]} [N]$ holds, decidability of $=_{\mu/\alpha}$ follows.

But since the assumption used is actually wrong, this proof approach fails.

Acknowledgment. The raison d'être of this paper is found in an extended discussion and correspondence with Henk Barendregt, Wil Dekkers, Felice Cardone and Mario Coppo, in particular during the last two years. Two of our decidability proofs for weak μ-equality are essentially based on the idea worked out by Cardone and Coppo in [**?**]. We thank all four for many stimulating personal communications and questions prompting us to our α-analysis and to the addition of virtual binders as annotations to the original proof system of Cardone and Coppo. Felice Cardone provided us with the ARS-rules underlying the picture of the μ-unsolvables.

References

[1] A. Arnold and D. Niwiński. *Rudiments of μ-Calculus*, volume 146 of *Studies in Logic and the Foundations of Mathematics*. North-Holland, 2001.

[2] H.P. Barendregt. *The Lambda Calculus: Its Syntax and Semantics*, volume 103 of *SLFM*. Elsevier, 2nd edition, 1984.

[3] H.P. Barendregt, W. J. M. Dekkers, and R. Statman. *Lambda Calculus with Types*. Perspectives in Logic. Cambridge University Press, 2011.

[4] Michael Brandt and Fritz Henglein. Coinductive Axiomatization of Recursive Type Equality and Subtyping. In Philippe de Groote and J. Roger Hindley, editors, *Typed Lambda Calculi and Applications*, volume 1210 of *Lecture Notes in Computer Science*, pages 63–81. Springer, 1997.

[5] H.J. Sander Bruggink. *Equivalence of Reductions in Higher-Order Rewriting*. PhD thesis, Utrecht University, 2008.

[6] Felice Cardone and Mario Coppo. Decidability Properties of Recursive Types. In C. Blundo and C. Laneve, editors, *ICTCS 2003*, volume 2841 of *LNCS*, pages 242–255. Springer, 2003.

[7] H. Comon, M. Dauchet, R. Gilleron, C. Löding, F. Jacquemard, D. Lugiez, S. Tison, and M. Tommasi. Tree Automata Techniques and Applications. `http://www.grappa.univ-lille3.fr/tata`, 2007.

[8] A. Corradini and F. Gadducci. Rational Term Rewriting. In Maurice Nivat, editor, *Foundations of Software Science and Computation Structures*, volume 1378 of *Lecture Notes in Computer Science*, pages 156–171. Springer, 1998.

[9] Marcelo P. Fiore and Gordon D. Plotkin. An Axiomatisation of Computationally Adequate Domain Theoretic Models of FPC. In *Proceedings of the Ninth Annual IEEE Symposium on Logic in Computer Science*, page 92–102. IEEE Computer Society Press, Los Alamitos, 1994.

[10] Clemens Grabmayer. *Relating Proof Systems for Recursive Types*. PhD thesis, Vrije Universiteit Amsterdam, 2005.

[11] Clemens Grabmayer. A Duality between Proof Systems for Cyclic Term Graphs. *Mathematical Structures in Computer Science*, 17:439–484, 2007.

[12] Dimitri Hendriks and Vincent van Oostrom. ʎ. In Franz Baader, editor, *Proceedings of CADE 19*, volume 2741 of *Lecture Notes in Artificial Intelligence*, pages 136–150. Springer, 2003.

[13] Paola Inverardi and Marisa Zilli. Rational Rewriting. In Igor Prívara, Branislav Rovan, and Peter Ruzicka, editors, *Mathematical Foundations of Computer Science 1994*, volume 841 of *Lecture Notes in Computer Science*, pages 433–442. Springer, 1994.

[14] B. Jacobs and J. J. M. M. Rutten. A Tutorial on (Co)Algebras and (Co)Induction. *EATCS Bulletin*, 62:62–222, 1997.

[15] S. Kahrs. Context Rewriting. In M. Rusinowitch and J.-L. Rémy, editors, *CTRS'92*, volume 656 of *LNCS*, pages 21–35. Springer, 1993.

[16] Richard Kennaway, Vincent van Oostrom, and Fer-Jan de Vries. Meaningless terms in rewriting. *Journal of Functional and Logic Programming*, 1999(1), February 1999.

[17] Jeroen Ketema. *Böhm-like Trees for Rewriting.* PhD thesis, Vrije Universiteit Amsterdam, 2006.

[18] Z. Khasidashvili. On Higher-Order Recursive Program Schemes. In Sophie Tison, editor, *Proceedings of the 19th International Colloquium on Trees in Algebra and Programming, CAAP 1994*, volume 787 of *lncs*, pages 172–186. Springer, 1994.

[19] Jan Willem Klop, Vincent van Oostrom, and Roel C. de Vrijer. Iterative Lexicographic Path Orders. In Kokichi Futatsugi, Jean-Pierre Jouannaud, and José Meseguer, editors, *Algebra, Meaning, and Computation, Essays Dedicated to Joseph A. Goguen on the Occasion of His 65th Birthday*, volume 4060 of *LNCS*, pages 541–554. Springer, 2006.

[20] J.W. Klop. *Combinatory Reduction Systems.* PhD thesis, Utrecht University, 1980.

[21] P.-A. Melliès. *Description Abstraite des Systèmes de Réécriture.* PhD thesis, Université Paris VII, December 1996.

[22] V. van Oostrom. FD à la Melliès, February 1997. VU University Amsterdam.

[23] D.E. Schroer. *The Church–Rosser Theorem.* PhD thesis, Cornell University, 1965.

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