Expressibility in the Lambda Calculus with μ

Clemens Grabmayer[1] and Jan Rochel[2]

1 **Department of Philosophy, Utrecht University**
PO Box 80126, 3508 TC Utrecht, The Netherlands
clemens@phil.uu.nl

2 **Department of Information and Computing Sciences**
PO Box 80089, 3508 TB Utrecht
jan@rochel.info

Abstract

We address a problem connected to the unfolding semantics of functional programming languages: give a useful characterization of those infinite λ-terms that are $\boldsymbol{\lambda}_{\mathsf{letrec}}$-expressible in the sense that they arise as infinite unfoldings of terms in $\boldsymbol{\lambda}_{\mathsf{letrec}}$, the λ-calculus with letrec. We provide two characterizations, using concepts we introduce for infinite λ-terms: regularity, strong regularity, and binding–capturing chains. It turns out that $\boldsymbol{\lambda}_{\mathsf{letrec}}$-expressible infinite λ-terms form a proper subclass of the regular infinite λ-terms. In this paper we establish these characterizations only for expressibility in $\boldsymbol{\lambda}_{\mu}$, the λ-calculus with explicit μ-recursion. We show that for all infinite λ-terms T the following are equivalent: (i): T is $\boldsymbol{\lambda}_{\mu}$-expressible; (ii): T is strongly regular; (iii): T is regular, and it only has finite binding–capturing chains.

We define regularity and strong regularity for infinite λ-terms as two different generalizations of regularity for infinite first-order terms: as the existence of only finitely many subterms that are defined as the reducts of two rewrite systems for decomposing λ-terms. These rewrite systems act on infinite λ-terms furnished with a bracketed prefix of abstractions for collecting decomposed λ-abstractions and keeping the terms closed under decomposition. They differ in which vacuous abstractions in the prefix are removed.

1998 ACM Subject Classification F.3.3 Studies of Program Constructs

Keywords and phrases lambda-calculus, lambda-calculus with letrec, unfolding semantics, regularity for infinite lambda-terms, binding-capturing chain

Digital Object Identifier 10.4230/LIPIcs.RTA.2013.206

1 Introduction

A syntactical core of functional programming languages is formed by $\boldsymbol{\lambda}_{\mathsf{letrec}}$, the λ-calculus with letrec, which can also be viewed as an abstract functional language. Formally, $\boldsymbol{\lambda}_{\mathsf{letrec}}$ is the extension of the λ-calculus by adding the construct letrec for expressing recursion as well as explicit substitution. In a slightly enriched form (of e.g. Haskell's Core language) it is used as an intermediate language for the compilation of functional programs, and as such it is the basis for optimizing program transformations. A calculus that in some respects is weaker than $\boldsymbol{\lambda}_{\mathsf{letrec}}$ is $\boldsymbol{\lambda}_{\mu}$, the λ-calculus with the binding construct μ for μ-recursion. Terms in $\boldsymbol{\lambda}_{\mu}$ can be interpreted directly as terms in $\boldsymbol{\lambda}_{\mathsf{letrec}}$ (expressions $\mu f.M(f)$ as **letrec** $f = M(f)$ **in** f), but translations in the other direction are more complicated, and have weaker properties.

For analyzing the execution behavior of functional programs, and for constructing program transformations, expressions in $\boldsymbol{\lambda}_{\mathsf{letrec}}$ or in $\boldsymbol{\lambda}_{\mu}$ are frequently viewed as finite representations of their unfolding semantics: the infinite λ-term that is obtained by completely unfolding all occurring recursive definitions, the letrec- or μ-bindings, in the expression.

In order to provide a theoretical foundation for such practical tasks, we aim to understand how infinite λ-terms look like that are expressible in $\boldsymbol{\lambda}_{\mathsf{letrec}}$ or in $\boldsymbol{\lambda}_{\mu}$ in the sense that they are infinite unfoldings of expressions from the respective calculus. In particular, we want to obtain useful characterizations of these classes of infinite λ-terms. Quite clearly, any such infinite λ-term must exhibit an, in some sense, repetitive structure that reflects the cyclic dependencies present in the finite description. This is because these dependencies are only 'rolled out', and so are preserved, by a typically infinite, stepwise unfolding process.

For infinite terms over a first-order signature there is a well-known concept of repetitive structure, namely regularity. An infinite term is called 'regular' if it has only a finite number of different subterms. Such infinite terms correspond to trees over ranked alphabets that are regular [4]. Like regular trees also regular terms can be expressed finitely by systems of recursion equations [4], by 'rational expressions' [4, Def. 4.5.3] which correspond to μ-terms (see e.g. [5]), or by terms using letrec-bindings. In this context finite expressions denote infinite terms either via a mathematical definition (a fixed-point construction, or induction on paths) or as the limit of a rewrite sequence consisting of unfolding steps. Regularity of infinite terms coincides, furthermore, with expressibility by finite terms enriched with either of the binding constructs μ or letrec. It is namely well-known that both representations are equally expressive with respect to denoting infinite terms, because a representation using letrec's can also be transformed into one using μ's while preserving the infinite unfolding.

For infinite λ-terms, however, the situation is different: A definition of regularity is less clear due to the presence of variable binding. And there are infinite λ-terms that are regular in an intuitive sense, yet apparently are not $\boldsymbol{\lambda}_{\mathsf{letrec}}$- or $\boldsymbol{\lambda}_{\mu}$-expressible. For example, the syntax trees of the infinite λ-terms T in Fig. 1 and U in Fig. 2 both exhibit a regular structure. But while T clearly is $\boldsymbol{\lambda}_{\mu}$- and $\boldsymbol{\lambda}_{\mathsf{letrec}}$-expressible (by $\mu f. \lambda xy. f\, y\, x$ and **letrec** $f = \lambda xy. f\, y\, x$ **in** f, respectively), this seems not to be the case for U: the λ-bindings in U are infinitely entangled, which suggests that it cannot be the result of just an unfolding process. Therefore it appears that the intuitive notion of regularity is too weak for capturing the properties of $\boldsymbol{\lambda}_{\mu}$- and of $\boldsymbol{\lambda}_{\mathsf{letrec}}$-expressibility. We note that actually these two properties coincide, because between λ_{μ}-terms and $\lambda_{\mathsf{letrec}}$-terms similar transformations are possible as between representations with μ and with letrec of infinite first-order terms (but this will not be proved here).

It is therefore desirable to obtain a precise, and conceptually satisfying, definition of regularity for infinite λ-terms that formalizes the intuitive notion, and that makes it possible to prove that $\boldsymbol{\lambda}_{\mu}$-/$\boldsymbol{\lambda}_{\mathsf{letrec}}$-expressible infinite λ-terms form only a proper subclass of the regular ones. Furthermore the question arises of whether the property of $\boldsymbol{\lambda}_{\mu}$-/$\boldsymbol{\lambda}_{\mathsf{letrec}}$-expressibility can be captured by a stronger concept of regularity that is still natural in some sense.

We tackle both desiderata at the same time, and provide solutions, but treat only the case of $\boldsymbol{\lambda}_{\mu}$-expressibility here. We introduce two concepts of regularity for infinite λ-terms. For this, we devise two closely related rewrite systems (infinitary Combinatory Reduction Systems) that allow to 'observe' infinite λ-terms by subjecting them to primitive decomposition steps and thereby obtaining 'generated subterms'. Then regular, and strongly regular infinite λ-terms are defined as those that give rise to only a finite number of generated subterms in the respective decomposition system. We establish the inclusion of the class of strongly regular in the class of regular infinite λ-terms, and the fact that this is a proper inclusion (by recognizing that the λ-term U in Fig. 2 is regular, but not strongly regular). As our main result we show that an infinite λ-term is $\boldsymbol{\lambda}_{\mu}$-expressible (that is, expressible by a term in $\boldsymbol{\lambda}_{\mu}$) if and only if it is strongly regular. Here we say that a term M in $\boldsymbol{\lambda}_{\mu}$ expresses an infinite λ-term V if V is the infinite unfolding of M. An infinite unfolding is unique if it exists, and it can be obtained as the limit of an infinite rewrite sequence of unfolding steps.

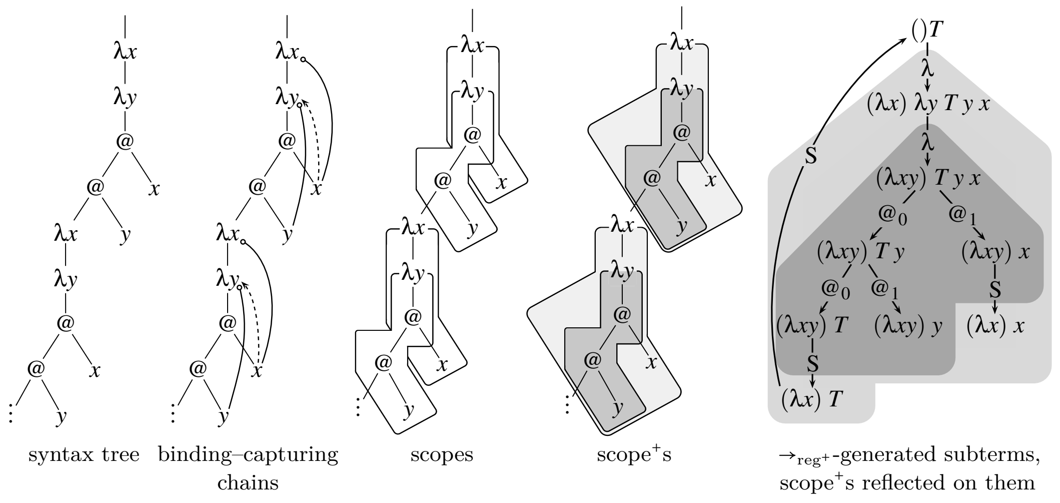

Figure 1 Strongly regular infinite λ-term T, which can be expressed by the λ_μ-term $\mu f.\lambda xy.f\,y\,x$.

This expressibility theorem is a special case of a result we reported in [6], which states that strong regularity coincides with $\boldsymbol{\lambda}_{\mathsf{letrec}}$-expressibility. That more general result settles a conjecture by Blom in [3, Sect. 1.2.4]. Its proof is closely connected to the proof of the result on $\boldsymbol{\lambda}_\mu$-expressibility we give here, which exhibits and highlights all the same features, but lacks the complexity that is inherent to the formal treatment of unfolding for terms in $\boldsymbol{\lambda}_{\mathsf{letrec}}$.

Additionally we give a result that explains the relationship between regularity and strong regularity by means of the concept of 'binding–capturing chain': a regular infinite λ-terms is strongly regular if and only if it does not contain an infinite binding–capturing chain.

Overview. In Section 2 we introduce rewriting systems (infinitary CRSs) for decomposing λ-terms into their generated subterms. By means of these systems we define regularity and strong regularity for infinite λ-terms. In Section 3 we provide sound and complete proof systems for these notions. In Section 4 we develop the notion of binding–capturing chain in infinite λ-terms, and show that strong regularity amounts to regularity plus the absence of infinite binding–capturing chains. In Section 5 we establish the correspondence between strong regularity and λ_μ-expressibility for infinite λ-terms. In the final Section 6 we place the results presented here in the context of our investigations about sharing in cyclic λ-terms.

2 Regular and strongly regular infinite λ-terms

In this section we motivate the introduction of higher-order versions of regularity, and subsequently introduce the concepts of regularity and strong regularity for infinite λ-terms.

For higher-order infinite terms such as infinite λ-terms, regularity has been used with as meaning the existence of a first-order syntax tree with named variables that is regular (e.g. in [2, 1]). For example, the infinite λ-terms T and U from Figures 1 and 2 are regular in this sense. However, such a definition of regularity has the drawback that it depends on a first-order representation (as syntax trees with named abstractions and variables) that is not invariant under α-conversion, the renaming of bound variables. Note that the syntax trees of T and U have renaming variants that contain infinitely many variables, and that for this reason are not regular as first-order trees. It is therefore desirable to obtain a definition of regularity that uses the condition for the first-order case but adapts the notion of subterm to λ-terms, and that pertains to a formulation of infinite λ-terms as higher-order terms.

Viable notions of subterm for λ-terms in a higher-order formalization require a stipulation on how to treat variable binding when stepping from a λ-abstraction $\lambda z.V$ into its body V. For this purpose we enrich the syntax of λ-terms with a bracketed prefix of abstractions (similar to a proof system for weak μ-equality in [5, Fig. 12]), and consider $(\lambda z)V$ as a 'generated subterm' of $\lambda z.V$, obtained by a λ-abstraction decomposition applied to $()\lambda z.V$, where $()$ is the empty prefix. An expression $(\lambda x_1 \ldots x_n)T$ represents a partially decomposed λ-term: the body T typically contains free occurrences of variables that in the original λ-term were bound by λ-abstractions but have since been split off by decomposition steps. The role of such abstractions has then been taken over by abstractions in the prefix $(\lambda x_1 \ldots x_n)$. In this way expressions with abstraction prefixes are kept closed under decomposition steps.

We formulate infinite λ-terms and their prefixed variants as terms in iCRSs (infinitary Combinatory Reduction Systems) for which we draw on the literature. By *iCRS-terms* we mean α-equivalence classes of iCRS-preterms that are defined by metric completion from finite CRS-terms [10]. For denoting and manipulating infinite terms we use customary notation for finite terms. In order to simplify our exposition we restrict to closed terms, but at one stage (a proof system in Section 5) we allow constants in our terms.

Note that we do not formalize β-reduction since we are only concerned with a static analysis of infinite λ-terms and later with finite expressions that express them via unfolding.

▶ **Definition 1** (iCRS-representation of $\boldsymbol{\lambda}^\infty$)**.** The CRS-signature for the λ-calculus $\boldsymbol{\lambda}$ and the infinitary λ-calculus $\boldsymbol{\lambda}^\infty$ consists of the set $\Sigma_\lambda = \{\mathsf{app}, \mathsf{abs}\}$ where app is a binary and abs a unary function symbol. By $Ter(\boldsymbol{\lambda}^\infty)$ we denote the set of infinite closed iCRS-terms over Σ_λ with the restriction that CRS-abstraction can only occur as an argument of an abs-symbol. Note that here and below we subsume finite λ-terms under the infinite ones.

▶ **Definition 2** (iCRS-representation of $(\boldsymbol{\lambda}^\infty)$)**.** The CRS-signature $\Sigma_{(\lambda)}$ for $(\boldsymbol{\lambda}^\infty)$, the version of $\boldsymbol{\lambda}^\infty$ with bracketed abstractions, extends Σ_λ by unary function symbols of arbitrary arity: $\Sigma_{(\lambda)} = \Sigma_\lambda \cup \{\mathsf{pre}_n \mid n \in \mathbb{N}\}$. Prefixed λ-terms $\mathsf{pre}_n([x_1]\ldots[x_n]T)$ will informally be denoted by $(\lambda x_1 \ldots x_n)T$, abbreviated as $(\lambda \vec{x})T$, or $()T$ in case of an empty prefix. By $Ter((\boldsymbol{\lambda}^\infty))$ we denote the set of closed iCRS-terms over $\Sigma_{(\lambda)}$ of the form $\mathsf{pre}_n([x_1]\ldots[x_n]T)$ for some $n \in \mathbb{N}$ and some term T over the signature Σ_λ with the restriction that a CRS-abstraction can only occur as an argument of an abs-symbol.

▶ **Example 3.** The λ-term $\lambda xy.y\,x$ in CRS-notation is $\mathsf{abs}([x]\mathsf{abs}([y]\mathsf{app}(y,x)))$. The prefixed λ-term $(\lambda x)\lambda y.y\,x$ is represented by $\mathsf{pre}_1([x]\mathsf{abs}([y]\mathsf{app}(y,x)))$.

On these prefixed λ-terms, we define two rewrite strategies $\to_{\text{reg}}$ and $\to_{\text{reg}^+}$ that deconstruct infinite λ-terms by steps that decompose applications and λ-abstractions, and take place just below the marked abstractions. They differ with respect to which vacuous prefix bindings they remove: while $\to_{\text{reg}}$-steps drop such bindings always before steps over applications and λ-abstractions, $\to_{\text{reg}^+}$-steps remove vacuous bindings only if they occur at the end of the abstraction prefix. These rewrite strategies will define respective notions of 'generated subterm', and will give rise to two concepts of regularity: a λ-term is called regular/strongly regular if its set of $\to_{\text{reg}}$-reachable/$\to_{\text{reg}^+}$-reachable generated subterms is finite.

▶ **Definition 4** (decomposing $(\boldsymbol{\lambda}^\infty)$-terms with rewrite strategies $\to_{\text{reg}}$ and $\to_{\text{reg}^+}$)**.** We consider the following CRS-rules over $\Sigma_{(\lambda)}$ in informal notation:[1]

$$\begin{aligned}
(\varrho^{@_i}):&\quad (\lambda x_1 \ldots x_n)T_0\,T_1 \to (\lambda x_1 \ldots x_n)T_i && (i \in \{0,1\})\\
(\varrho^{\lambda}):&\quad (\lambda x_1 \ldots x_n)\lambda x_{n+1}.T_0 \to (\lambda x_1 \ldots x_{n+1})T_0 &&\\
(\varrho^{\mathsf{S}}):&\quad (\lambda x_1 \ldots x_{n+1})T_0 \to (\lambda x_1 \ldots x_n)T_0 && (\text{if binding } \lambda x_{n+1} \text{ is vacuous})\\
(\varrho^{\text{del}}):&\quad (\lambda x_1 \ldots x_{n+1})T_0 \to (\lambda x_1 \ldots x_{i-1}x_{i+1} \ldots x_{n+1})T_0 && (\text{if bind. } \lambda x_i \text{ is vacuous})
\end{aligned}$$

[1] E.g. explicit form of scheme (ϱ^{S}): $\mathsf{pre}_{n+1}([x_1 \ldots x_{n+1}]Z(x_1,\ldots,x_n)) \to \mathsf{pre}_n([x_1 \ldots x_n]Z(x_1,\ldots,x_n))$.

We call an occurrence o of a binding like a λ-abstraction λz or a CRS-abstraction $[z]$ in a term V *vacuous* if V does not contain a variable occurrence of z that is bound by o.

The iCRS with these rules induces an ARS (abstract rewriting system) $\mathcal{A}$ on infinite terms over $\Sigma_{(\lambda)}$. By (Λ) we denote the sub-ARS of $\mathcal{A}$ with its set of objects restricted to $Ter((\boldsymbol{\lambda}))$. Note that $Ter((\boldsymbol{\lambda}))$ is closed under steps in (Λ). By $\rightarrow_{@_0}$, $\rightarrow_{@_1}$, $\rightarrow_{\lambda}$, $\rightarrow_{\mathsf{S}}$, $\rightarrow_{\mathrm{del}}$ we denote the rewrite relations induced by (Λ)-steps with respect to rules $\varrho^{@_0}$, $\varrho^{@_1}$, ϱ^{λ}, ϱ^{S}, ϱ^{del}. We define *Reg* (*Reg*$^+$) as the sub-ARS of (Λ) that arises from dropping steps that are:

- due to ϱ^{S} (ϱ^{del}), so that the prefix can be shortened only by ϱ^{del}-steps (ϱ^{S}-steps).
- due to rules other than ϱ^{del} (ϱ^{S}) but whose source is also a source of a ϱ^{del}-step (ϱ^{S}-step).

Reg (*Reg*$^+$) is ϱ^{del}*-eager* (ϱ^{S}-eager) in the sense that on each path ϱ^{del}-steps (ϱ^{S}-steps) occur as soon as possible. We denote by $\rightarrow_{\mathrm{reg}}$ ($\rightarrow_{\mathrm{reg}^+}$) the rewrite strategy induced by *Reg* (*Reg*$^+$).[2]

▶ **Example 5.** Using the recursive equation $T = \lambda xy.T\,y\,x$ as a description for the infinite λ-term T in Fig. 1, we find that decomposition by $\rightarrow_{\mathrm{reg}^+}$-steps proceeds as follows, repetitively:

$$\begin{array}{llllllll} & & & & (\lambda xy)\,T & (\lambda x)\,T & ()\,T & \ldots \\ ()\,T & (\lambda x)\,\lambda y.T\,y\,x & (\lambda xy)\,T\,y\,x & (\lambda xy)\,T\,y & (\lambda xy)\,y & & & \\ & & & (\lambda xy)\,x & (\lambda x)\,x & & & \end{array}$$

(in a tree that branches to the right). Note that removal steps for vacuous bindings take place only at the end of the prefix. See Fig. 1 right for the reduction graph of $()T$ with displayed sorts of decomposition steps. Although $\rightarrow_{\mathsf{S}}$-steps also are $\rightarrow_{\mathrm{del}}$-steps, this decomposition is not also one according to $\rightarrow_{\mathrm{reg}}$, because e.g. the step $(\lambda xy)\,T\,y \rightarrow_{@_1} (\lambda xy)\,y$ is not ϱ^{del}-eager.

The rules ϱ^{S} are related to the de Bruijn notation of λ-terms. Consider $\lambda x.(\lambda y.x\,x)\,x$ which in de Bruijn notation is $\lambda.(\lambda.1\,1)\,0$ and when using Peano numerals $\lambda.(\lambda.S(0)\,S(0))\,0$. Now if the symbols S are allowed to appear 'shared' and occur further up in the term as in $\lambda.(\lambda.S(0\,0))\,0$, then this term structure corresponds to the decomposition with $\rightarrow_{\mathrm{reg}^+}$.

To understand the difference between $\rightarrow_{\mathrm{reg}}$ and $\rightarrow_{\mathrm{reg}^+}$, consider the notions of scope and scope$^+$, illustrated in Figures 1 and 2. The scope of an abstraction is the smallest connected portion of a syntax tree that contains the abstraction itself as well as all of its bound variable occurrences. And scope$^+$s extend scopes minimally so that the resulting areas appear properly nested. For a precise definition we refer to [6, Sect. 4]. As can be seen in Figures 1 and 2, applications of ϱ^{del} (ϱ^{S}) coincide with the positions where scopes (scope$^+$s) are closed.

▶ **Definition 6** (regular/strongly regular λ-terms, generated subterms)**.** Let $T \in Ter(\boldsymbol{\lambda}^{\infty})$. We define the sets $ST(T)$ and $ST^+(T)$ of *generated subterms* of T *with respect to* $\rightarrow_{\mathrm{reg}}$ and $\rightarrow_{\mathrm{reg}^+}$:

$$ST(T) := \{U \in Ter((\boldsymbol{\lambda}^{\infty})) \mid ()T \twoheadrightarrow_{\mathrm{reg}} U\} \qquad ST^+(T) := \{U \in Ter((\boldsymbol{\lambda}^{\infty})) \mid ()T \twoheadrightarrow_{\mathrm{reg}^+} U\}$$

We say that T is *regular* (*strongly regular*) if T has only finitely many generated subterms with respect to $\rightarrow_{\mathrm{reg}}$ (respectively, with respect to $\rightarrow_{\mathrm{reg}^+}$).

▶ **Example 7.** From the $\rightarrow_{\mathrm{reg}^+}$-decomposition in Example 5 and Fig. 1 of the infinite λ-term T in Fig. 1 it follows that $ST^+(T)$ consists of 9 generated subterms. Hence T is strongly regular.

The situation is different for the infinite λ-term U in Fig. 2. When represented as the term $\lambda x.R(x)$ together with the CRS-rule $R(X) \rightarrow \lambda y.R(y)\,X$, its $\rightarrow_{\mathrm{reg}^+}$-decomposition is:

$$\begin{array}{llllllll} & & & & & (\lambda xyz)\,R(z) & (\lambda xyzu)\,R(u)\,z & \ldots \\ ()\,U & (\lambda x)\,R(x) & (\lambda xy)\,R(y)\,x & (\lambda xy)\,R(y) & (\lambda xyz)\,R(z)\,y & (\lambda xyx)\,y & (\lambda xy)\,y & \\ & & & (\lambda xy)\,x & (\lambda x)\,x & & & \end{array}$$

[2] We use 'rewrite strategy' for a relation on terms, and not for a sub-ARS of a CRS-induced ARS [13].

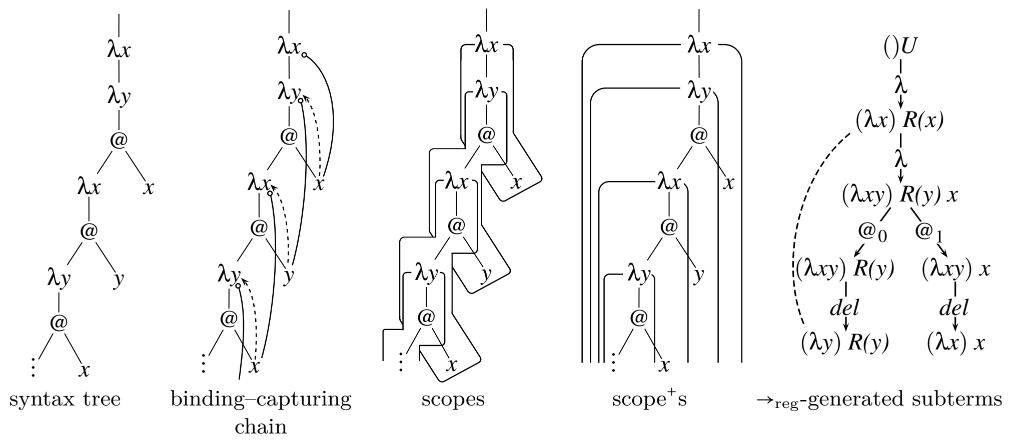

Figure 2 The regular infinite λ-term U that is not strongly regular, and not λ_μ-expressible.

Since here the prefixes grow unboundedly, U has infinitely many $\rightarrow_{\text{reg}^+}$-generated subterms, and hence U is not strongly regular. But its $\rightarrow_{\text{reg}}$-decomposition exhibits again a repetition as can be seen from the reduction graph in Fig. 2 on the right. Note that a vacuous binding from within a prefix is removed. $()U$ has 6 only different $\rightarrow_{\text{reg}}$-reducts. Hence U is regular.

For infinite λ-terms like $(\lambda x_1.x_1)\,(\lambda x_1.\lambda x_2.x_2)\,(\lambda x_1.\lambda x_2.\lambda x_3.x_3)\ldots$ that do not have any regular pseudoterm syntax-trees, both $\rightarrow_{\text{reg}^+}$-decomposition and $\rightarrow_{\text{reg}}$-decomposition yield infinitely many generated subterms, and hence they are neither regular nor strongly regular.

For a better understanding of the precise relationship between $\rightarrow_{\text{reg}}$ and $\rightarrow_{\text{reg}^+}$, and eventually of the two concepts of generated subterm and of regularity, we gather a number of basic properties of these rewrite strategies and their constituents.

▶ **Proposition 8.** The restrictions of the rewrite relations from Def. 4 to $\mathit{Ter}((\boldsymbol{\lambda}^\infty))$, the set of objects of *Reg* and *Reg*$^+$, have the following properties:

(i) $\rightarrow_{\text{del}}$ is confluent, and terminating.

(ii) $\rightarrow_{\mathsf{S}} \subseteq \rightarrow_{\text{del}}$. Furthermore, $\rightarrow_{\mathsf{S}}$ is deterministic, hence confluent, and terminating.

(iii) $\rightarrow_{\text{del}}$ one-step commutes with $\rightarrow_\lambda$, $\rightarrow_{@_0}$, $\rightarrow_{@_1}$, and one-step sub-commutes with $\rightarrow_{\mathsf{S}}$; $\rightarrow_{\text{del}}$ postpones over $\rightarrow_\lambda$, $\rightarrow_{@_0}$, $\rightarrow_{@_1}$ and $\rightarrow_{\mathsf{S}}$. Formulated symbolically, this means:

$$\begin{array}{lll}
\leftarrow_{\text{del}} \cdot \rightarrow_\lambda \subseteq \rightarrow_\lambda \cdot \leftarrow_{\text{del}} & \leftarrow_{\text{del}} \cdot \rightarrow_{@_i} \subseteq \rightarrow_{@_i} \cdot \leftarrow_{\text{del}} & \leftarrow_{\text{del}} \cdot \rightarrow_{\mathsf{S}} \subseteq \rightarrow_{\mathsf{S}}^{=} \cdot \leftarrow_{\text{del}}^{=} \\
\rightarrow_{\text{del}} \cdot \rightarrow_\lambda \subseteq \rightarrow_\lambda \cdot \rightarrow_{\text{del}} & \rightarrow_{\text{del}} \cdot \rightarrow_{@_i} \subseteq \rightarrow_{@_i} \cdot \rightarrow_{\text{del}} & \rightarrow_{\text{del}} \cdot \rightarrow_{\mathsf{S}} \subseteq \rightarrow_{\mathsf{S}} \cdot \rightarrow_{\text{del}}
\end{array}$$

(iv) Normal forms of $\rightarrow_{\text{reg}}$ and $\rightarrow_{\text{reg}^+}$ are of the form $(\lambda x)x$, and $(\lambda x_1 \ldots x_n)x_n$, respectively.

(v) $\rightarrow_{\text{reg}}$ and $\rightarrow_{\text{reg}^+}$ are finitely branching, and, on finite terms, terminating.

Proof. These properties, including those concerning commutation of steps, are easy to verify by analyzing the behavior of the rewrite rules in *Reg* on terms of $\mathit{Ter}((\boldsymbol{\lambda}^\infty))$. ◀

▶ **Proposition 9.** (i) Let $(\lambda\vec{x})T$ be a term in $\mathit{Ter}((\boldsymbol{\lambda}^\infty))$ with $|\vec{x}| = n \in \mathbb{N}$. Then the number of terms $(\lambda\vec{y})U$ in $\mathit{Ter}((\boldsymbol{\lambda}^\infty))$ with $(\lambda\vec{y})U \twoheadrightarrow_{\text{del}} (\lambda\vec{x})T$ and $|\vec{y}| = n + k \in \mathbb{N}$ is $\binom{n+k}{n}$.

(ii) Let $A \subseteq \mathit{Ter}((\boldsymbol{\lambda}^\infty))$ be a finite set, and $k \in \mathbb{N}$. Then also the set of terms in $\mathit{Ter}((\boldsymbol{\lambda}^\infty))$ that are the form $(\lambda\vec{y})U$ with $|\vec{y}| \le k$ and that have a $\twoheadrightarrow_{\text{del}}$-reduct in A is finite.

We state a lemma about a close connection between $\rightarrow_{\text{reg}}$- and $\rightarrow_{\text{reg}^+}$-rewrite sequences.

▶ **Lemma 10.** (i) *On* $\mathit{Ter}(\boldsymbol{\lambda}^\infty)$ *it holds:* $\twoheadleftarrow_{\text{del}} \cdot \rightarrow_{\text{reg}^+} \subseteq \rightarrow^{!}_{\text{del}} \cdot \rightarrow^{=}_{\text{reg}} \cdot \twoheadleftarrow_{\text{del}}$, *where* $\rightarrow^{!}_{\text{del}}$ *denotes many-step* $\rightarrow_{\text{del}}$*-reduction to* $\rightarrow_{\text{del}}$*-normal form. As a consequence of this and of* $\rightarrow^{!}_{\text{del}} \cdot \rightarrow^{=}_{\text{reg}} \subseteq \twoheadrightarrow_{\text{reg}}$*, every finite or infinite rewrite sequence in* $\mathit{Ter}(\boldsymbol{\lambda}^\infty)$ *of the form:*

$$\tau : (\lambda\vec{x}_0)T_0 \to_{\text{reg}^+} (\lambda\vec{x}_1)T_1 \to_{\text{reg}^+} \ldots \to_{\text{reg}^+} (\lambda\vec{x}_k)T_k \to_{\text{reg}^+} \ldots$$

projects over a sequence $\pi : (\lambda\vec{x}_0)T_0 \twoheadrightarrow_{\text{del}} (\lambda\vec{x}'_0)T_0$ *to a rewrite sequence of the form:*

$$\check{\tau} : (\lambda\vec{x}'_0)T_0 \twoheadrightarrow_{\text{reg}} (\lambda\vec{x}'_1)T_1 \twoheadrightarrow_{\text{reg}} \;\ldots\; \twoheadrightarrow_{\text{reg}} (\lambda\vec{x}'_k)T_k \twoheadrightarrow_{\text{reg}} \ldots$$

in the sense that $(\lambda\vec{x}_k)T_k \twoheadrightarrow_{\text{del}} (\lambda\vec{x}'_k)T_k$ *for all* $k \in \mathbb{N}$ *less or equal to the length of* τ.

(ii) *On* $Ter(\boldsymbol{\lambda}^{\infty})$ *it holds:* $\twoheadrightarrow_{\text{del}} \cdot \to_{\text{reg}} \subseteq \to^{!}_{\mathsf{S}} \cdot \to^{=}_{\text{reg}^+} \cdot \twoheadrightarrow_{\text{del}}$. *Due to this and* $\to^{!}_{\mathsf{S}} \cdot \to^{=}_{\text{reg}^+} \subseteq \twoheadrightarrow_{\text{reg}^+}$, *every rewrite sequence* $\tau : (\lambda\vec{x}'_0)T_0 \to_{\text{reg}} (\lambda\vec{x}'_1)T_1 \to_{\text{reg}} \ldots \to_{\text{reg}} (\lambda\vec{x}'_k)T_k \to_{\text{reg}} \ldots$ *in* $Ter(\boldsymbol{\lambda}^{\infty})$ *lifts over a sequence* $\pi : (\lambda\vec{x}_0)T_0 \twoheadrightarrow_{\text{del}} (\lambda\vec{x}'_0)T_0$ *to a* $\to_{\text{reg}^+}$*-rewrite sequence of the form:* $\hat{\tau} : (\lambda\vec{x}_0)T_0 \twoheadrightarrow_{\text{reg}^+} (\lambda\vec{x}_1)T_1 \twoheadrightarrow_{\text{reg}^+} \;\ldots\; \twoheadrightarrow_{\text{reg}^+} (\lambda\vec{x}_k)T_k \twoheadrightarrow_{\text{reg}^+} \ldots$ *in the sense that* $(\lambda\vec{x}_k)T_k \twoheadrightarrow_{\text{del}} (\lambda\vec{x}'_k)T_k$ *for all* $k \in \mathbb{N}$ *less or equal to the length of* τ.

Proof. The inclusion properties in (10) and (10) can be shown by easy arguments with diagrams using the commutation properties in Prop. 8, (iii), as well as (i) and (ii) from there. ◀

Now we are able to establish that strong regularity implies regularity for infinite λ-terms.

▶ **Proposition 11.** Every strongly regular infinite λ-term is also regular. Finite λ-terms are both regular and strongly regular.

Proof. Let T be a strongly regular infinite λ-term. Therefore $ST^{+}(T)$ is finite. Since every $\to_{\text{reg}}$-rewrite-sequence from $()T$ lifts to a $\to_{\text{reg}^+}$-rewrite-sequence from $()T$ over $\twoheadrightarrow_{\text{del}}$-compression due to Lemma 10, (10), every term in $ST(T)$ is the $\twoheadrightarrow_{\text{del}}$–compression of a term in $ST^{+}(T)$. Then it follows by Prop. 9, (i), that also $ST(T)$ is finite. Hence T is also regular.

Let T be a finite λ-term. Due to to Prop. 8, (v), Kőnig's Lemma can be applied to the reduction graph of $()T$ with respect to $\twoheadrightarrow_{\text{reg}^+}$ to yield that T has only finitely many generated subterms with respect to $\twoheadrightarrow_{\text{reg}^+}$. Hence T is strongly regular. ◀

3 Proving regularity and strong regularity

In this section we introduce proof systems for regularity and strong regularity: the systems $\mathbf{Reg}^{\infty}$ and $\mathbf{Reg}^{+,\infty}$ with typically infinite derivations, and the systems $\mathbf{Reg}$, $\mathbf{Reg}^{+}$, and $\mathbf{Reg}^{+}_{0}$ for provability by finite derivations. A completed derivation of $()U$ in $\mathbf{Reg}^{\infty}$ (in $\mathbf{Reg}^{+,\infty}$) corresponds to the 'tree unfolding' of the $\to_{\text{reg}}$-reduction graph (the $\to_{\text{reg}^+}$-reduction graph) of $()U$, which is a tree that describes all $\to_{\text{reg}}$-(resp. $\to_{\text{reg}^+}$-)rewrite sequences from $()U$. Closed derivations of $()U$ in $\mathbf{Reg}$ (in $\mathbf{Reg}^{+}$, or $\mathbf{Reg}^{+}_{0}$) correspond to finite unfoldings of the $\to_{\text{reg}}$-reduction graph (the $\to_{\text{reg}^+}$-reduction graph) into a graph with only vertical sharing.

We start by introducing proof systems for well-formed prefixed terms (terms in $Ter((\boldsymbol{\lambda}^{\infty}))$).

▶ **Definition 12** (proof systems $(\boldsymbol{\Lambda})^{\infty}$, $(\boldsymbol{\Lambda})^{+,\infty}$ for well-formed $\boldsymbol{\lambda}^{\infty}$-terms)**.** The proof systems defined here act on CRS-terms over signature $\Sigma_{(\lambda)}$ as formulas, and are Hilbert-style systems for potentially infinite prooftrees (of depth $\leq \omega$). The system $(\boldsymbol{\Lambda})^{+,\infty}$ has the axioms (0) and the rules (@), (λ), and (S) in Fig. 3. The system $(\boldsymbol{\Lambda})^{\infty}$ arises from $(\boldsymbol{\Lambda})^{+,\infty}$ by replacing the axioms (0) and the rule (S) with the axioms (0) and the rule (del) in Fig. 4, respectively.

A finite or infinite derivation $\mathcal{T}$ in $(\boldsymbol{\Lambda})^{\infty}$ (in $(\boldsymbol{\Lambda})^{+,\infty}$) is called *closed* if all terms in leafs of $\mathcal{T}$ are axioms. Derivability of a term $(\lambda\vec{x})T$ in $(\boldsymbol{\Lambda})^{\infty}$ (in $(\boldsymbol{\Lambda})^{+,\infty}$), symbolically $\vdash_{(\boldsymbol{\Lambda})^{\infty}} (\lambda\vec{x})T$ (resp. $\vdash_{(\boldsymbol{\Lambda})^{+,\infty}} (\lambda\vec{x})T$), means the existence of a closed derivation with conclusion $(\lambda\vec{x})T$.

▶ **Definition 13** (proof systems $\mathbf{Reg}^{\infty}$, $\mathbf{Reg}^{+,\infty}$)**.** The proof systems $\mathbf{Reg}^{\infty}$ and $\mathbf{Reg}^{+,\infty}$ have the same axioms and rules as $(\boldsymbol{\Lambda})^{\infty}$ and $(\boldsymbol{\Lambda})^{+,\infty}$, respectively, but they restrict the notion of derivability. A derivation $\mathcal{D}$ in $(\boldsymbol{\Lambda})^{\infty}$ (in $(\boldsymbol{\Lambda})^{+,\infty}$) is called *admissible in* $\mathbf{Reg}^{\infty}$ (*in* $\mathbf{Reg}^{+,\infty}$)

$$\frac{}{(\lambda\vec{x}y)\,y}\;\mathsf{0} \qquad \frac{(\lambda\vec{x}y)\,T_0}{(\lambda\vec{x})\,\lambda y.T_0}\;\lambda \qquad \frac{(\lambda\vec{x})\,T_0 \qquad (\lambda\vec{x})\,T_1}{(\lambda\vec{x})\,T_0\,T_1}\;@$$

$$\frac{(\lambda x_1\ldots x_{n-1})\,T}{(\lambda x_1\ldots x_n)\,T}\;\mathsf{S}\;\;\begin{array}{l}\text{(if the binding}\\ \lambda x_n \text{ is vacuous)}\end{array} \qquad \frac{\begin{array}{c}[(\lambda\vec{x})\,T]^l\\ \mathcal{D}_0\\ (\lambda\vec{x})\,T\end{array}}{(\lambda\vec{x})\,T}\;\text{FIX}, l \;\;\; (\text{if } |\mathcal{D}_0| \geq 1)$$

Figure 3 The proof system $\mathbf{Reg}^{+}$ for strongly regular λ-terms. In the variant system $\mathbf{Reg}^{+}_{0}$ of $\mathbf{Reg}^{+}$, instances of (FIX) are subject to the additional side-condition: for all $(\lambda\vec{y})\,U$ on threads in $\mathcal{D}_0$ from open marked assumptions $((\lambda\vec{x})\,T)^u$ downwards it holds that $|\vec{y}| \geq |\vec{x}|$. The systems $(\mathbf{\Lambda})^{+,\infty}$ and $\mathbf{Reg}^{+,\infty}$ do not contain the rule FIX. Derivations in $\mathbf{Reg}^{+}$, $\mathbf{Reg}^{+}_{0}$, and $\mathbf{Reg}^{\infty}$ must be (S)-eager.

$$\frac{}{(\lambda y)\,y}\;\mathsf{0} \qquad \frac{(\lambda x_1\ldots x_{i-1}x_{i+1}\ldots x_n)\,T}{(\lambda x_1\ldots x_n)\,T}\;\text{del}\;\;\begin{array}{l}\text{(if the binding}\\ \lambda x_i \text{ is vacuous)}\end{array}$$

Figure 4 The proof system $\mathbf{Reg}$ for regular λ-terms arises from $\mathbf{Reg}^{+}$ through replacing the rule (S) by the rule (del), and the axiom scheme ($\mathsf{0}$) by the more restricted version here. The systems $(\mathbf{\Lambda})^{\infty}$ and $\mathbf{Reg}^{\infty}$ do not contain the rule (FIX). Derivations in $\mathbf{Reg}$ and $\mathbf{Reg}^{\infty}$ must be (del)-eager.

if it contains only finitely many different terms, and if it is (del)-*eager* ((S)-*eager*), that is, if no conclusion of an instance of (@) or (λ) in $\mathcal{D}$ is the source of a $\rightarrow_{\text{del}}$-step (a $\rightarrow_{\mathsf{S}}$-step). Derivability in $\mathbf{Reg}^{\infty}$ (in $\mathbf{Reg}^{+,\infty}$) means the existence of a closed admissible derivation.

We say that a proof system $\mathcal{S}$ is *sound* (*complete*) for a property P of infinite λ-terms if $\vdash_{\mathcal{S}} ()T$ implies $P(T)$ (if $P(T)$ implies $\vdash_{\mathcal{S}} ()T$) for all infinite λ-terms T.

The systems $(\mathbf{\Lambda})^{\infty}$ and $(\mathbf{\Lambda})^{+,\infty}$ are sound and complete for all infinite λ-terms in $Ter(\boldsymbol{\lambda}^{\infty})$: for completeness note that every prefixed term $(\lambda\vec{y})\,U$ with U not a variable is the conclusion of an instance of a rule in these systems. This leads us to statements for $\mathbf{Reg}^{\infty}$ and $\mathbf{Reg}^{+,\infty}$.

▶ **Proposition 14.** (i) $\mathbf{Reg}^{\infty}$ is sound and complete for regularity of infinite λ-terms.
(ii) $\mathbf{Reg}^{+,\infty}$ is sound and complete for strong regularity of infinite λ-terms.

Proof. We argue only for (ii), since (i) can be seen analogously. Every (S)-eager derivation $\mathcal{T}$ in $(\mathbf{\Lambda})^{+,\infty}$ with conclusion $()T$ assembles the maximal $\rightarrow_{\text{reg}^{+}}$-rewrite sequences from $()T$ in the following sense: the steps of every such rewrite sequence correspond to the steps through $\mathcal{T}$ along a thread from the conclusion upwards. Therefore if $\mathcal{T}$ is an admissible derivation in $\mathbf{Reg}^{+,\infty}$, and hence contains only finitely many terms, then $ST^{+}(T)$ is finite. Since every term $()T$ in $Ter((\mathbf{\lambda}))$ has a (S)-eager derivation in $(\mathbf{\Lambda})^{+,\infty}$, the converse holds as well. ◀

▶ **Definition 15** (proof systems $\mathbf{Reg}$, $\mathbf{Reg}^{+}$, and $\mathbf{Reg}^{+}_{0}$). The natural-deduction style proof system $\mathbf{Reg}^{+}$ has the axioms and rules in Fig. 3. Its variant $\mathbf{Reg}^{+}_{0}$ demands an additional side-condition on instances of the rule (FIX) as described there. The system $\mathbf{Reg}$ arises from $\mathbf{Reg}^{+}$ by dropping the rule (S), and restricting the axioms to the axioms ($\mathsf{0}$) in Fig. 4.

A derivation in one of these systems is called *closed* if it does not contain any undischarged marker assumptions (discharging assumptions is indicated by assigning the appertaining assumption markers to instances of FIX, see Fig. 3). Derivability in $\mathbf{Reg}$ (in $\mathbf{Reg}^{+}$ or in $\mathbf{Reg}^{+}_{0}$) means the existence of a closed, (del)-eager ((S)-eager), finite derivation.

The proposition below explains that the side-condition '$|\mathcal{D}_0| \geq 1$' on subderivations of FIX-instances guarantees a 'guardedness' property for threads in derivations in these systems.

▶ **Proposition 16.** Let $\mathcal{D}$ be a derivation in $\mathbf{Reg}$, $\mathbf{Reg}^+$, or $\mathbf{Reg}^+_\mathbf{0}$. Then for every instance ι of the rule (FIX) in $\mathcal{D}$ it holds: every thread from ι upwards to a marked assumption that is discharged at ι passes at least one instance of a rule (λ) or (@).

Proof. Let $\mathcal{D}$ be a derivation in $\mathbf{Reg}$, as the argument is analogous for $\mathbf{Reg}^+$ and $\mathbf{Reg}^+_\mathbf{0}$. Let ι be an instance of (FIX) in $\mathcal{D}$, and π a thread from the conclusion $(\lambda\vec{y})U$ of ι to a marked assumption $((\lambda\vec{y})U)^l$ that is discharged at ι. Then due to the side-condition on the topmost instance κ of (FIX) passed on π there is at least one instance of a rule (λ), (@), or (del) passed on π above κ. We are done unless that is an instance of (del). But then there must also be an instance of (λ) on π, since (del) decreases the prefix length, only (λ) increases it, and the prefix lengths in the formula at the start and at the end of π are the same. ◀

▶ **Example 17.** (i) The following are two derivations in $\mathbf{Reg}^+$ of different efficiency of the infinite λ-term T from Fig. 1 when represented by the recursive equation $T = \lambda xy.T\,y\,x$:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{(()T)^l}{(\lambda x)T}\,\mathsf{S}}{(\lambda xy)T}\,\mathsf{S} \qquad \dfrac{}{(\lambda xy)y}\,\mathsf{0}}{(\lambda xy)T\,y}\,@ \qquad \dfrac{\dfrac{}{(\lambda x)x}\,\mathsf{0}}{(\lambda xy)x}\,\mathsf{S}}{(\lambda xy)T\,y\,x}\,@}{(\lambda x)\,\lambda y.T\,y\,x}\,\lambda}{()\,\lambda xy.T\,y\,x}\,\lambda}{()T}\,\text{FIX},l
\qquad
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{((\lambda x)\,\lambda y.T\,y\,x)^l}{()T}\,\lambda}{(\lambda x)T}\,\mathsf{S}}{(\lambda xy)T}\,\mathsf{S} \qquad \dfrac{}{(\lambda xy)y}\,\mathsf{0}}{(\lambda xy)T\,y}\,@ \qquad \dfrac{\dfrac{}{(\lambda x)x}\,\mathsf{0}}{(\lambda xy)x}\,\mathsf{S}}{(\lambda xy)T\,y\,x}\,@}{(\lambda x)\,\lambda y.T\,y\,x}\,\lambda}{(\lambda x)\,\lambda y.T\,y\,x}\,\text{FIX},l}{()T}\,\lambda$$

Note that only the left derivation is one in $\mathbf{Reg}^+_\mathbf{0}$, because the right one contains a term with shorter prefix than the discharged assumption on a thread to the instance of FIX.

(ii) The infinite λ-term from Fig. 2, denoted by the term $()\lambda x.R(x)$ and generated by the CRS-rule $R(X) \to \lambda x.R(x)\,X$ is derivable in $\mathbf{Reg}$ by the closed derivation on the left, but it is not derivable in $\mathbf{Reg}^+$:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\overbrace{((\lambda y)\,R(y))^l}^{=(\lambda x)\,R(x)}}{(\lambda xy)\,R(y)}\,\text{del} \qquad \dfrac{\dfrac{}{(\lambda x)\,x}\,\mathsf{0}}{(\lambda xy)\,x}\,\text{del}}{(\lambda xy)\,R(y)\,x}\,@}{(\lambda x)\,\lambda y.R(y)\,x}\,\lambda}{(\lambda x)\,R(x)}\,\text{FIX},\,l}{()\,\lambda x.R(x)}\,\lambda
\qquad
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{(\lambda xyzuv)\,R(v)\,u}}{(\lambda xyzu)\,\lambda v.R(v)\,u}\,\lambda \qquad \dfrac{\dfrac{}{(\lambda xyz)\,z}\,\mathsf{0}}{(\lambda xyzu)\,z}\,\mathsf{S}}{(\lambda xyzu)\,R(u)\,z}\,@}{(\lambda xyz)\,\lambda u.R(u)\,z}\,\lambda \qquad \dfrac{\dfrac{}{(\lambda xy)\,y}\,\mathsf{0}}{(\lambda xyz)\,y}\,\mathsf{S}}{(\lambda xyz)\,R(z)\,y}\,@}{(\lambda xy)\,\lambda z.R(z)\,y}\,\lambda \qquad \dfrac{\dfrac{}{(\lambda x)\,x}\,\mathsf{0}}{(\lambda xy)\,x}\,\mathsf{S}}{(\lambda xy)\,R(y)\,x}\,@}{(\lambda x)\,\lambda y.R(y)\,x}\,\lambda}{()\,\lambda x.R(x)}\,\lambda$$

The latter follows from the infinite prooftree on the right, the result of a bottom-up proof search in $\mathbf{Reg}^+$, which is a derivation in $(\mathbf{\Lambda})^{+,\infty}$ but not in $\mathbf{Reg}^{+,\infty}$, since, as it does not contain repetitions, the rule FIX cannot be used to cut off repetitive subderivations.

Finally, we can link derivability in $\mathbf{Reg}$ and $\mathbf{Reg}^+$ to regularity and strong regularity.

▶ **Theorem 18.** (i) $\mathbf{Reg}$ *is sound and complete for regularity of infinite λ-terms.*
(ii) $\mathbf{Reg}^+$ *and* $\mathbf{Reg}^+_\mathbf{0}$ *are sound and complete for strong regularity of infinite λ-terms.*

Proof. For (18), in view of Prop. 14, (i), it suffices to be able to transform closed, admissible derivations in $\mathbf{Reg}^\infty$ into closed derivations in $\mathbf{Reg}$, and vice versa. Every closed derivation $\mathcal{D}$ in $\mathbf{Reg}$ can be unfolded by a stepwise, typically infinite process into a closed derivation in $(\mathbf{\Lambda})^\infty$: in every step the subderivation of a bottommost instance ι of FIX is transferred to above each of the marked assumptions that are discharged at ι, and the original instance of FIX is removed. If this process is infinite, then due to Prop. 16 it always eventually increases the size of the part of the derivation below the bottommost occurrences of FIX. Hence in the limit it produces a closed, (del)-eager derivation in $(\mathbf{\Lambda})^{\infty}$ that contains only finitely many terms (only those in $\mathcal{D}$), and thus is admissible in $\mathbf{Reg}^{\infty}$. Conversely, every admissible, closed derivation $\mathcal{T}$ in $\mathbf{Reg}^{\infty}$ can be 'folded' into a finite closed derivation in $\mathbf{Reg}$ by introducing FIX-instances to cut off the derivation above the upper occurrence of a repetition. This yields a finite derivation since due to admissibility of $\mathcal{T}$ in $\mathbf{Reg}^{\infty}$ every sufficiently long thread contains a repetition, and then Kőnig's Lemma can be applied.

For $\mathbf{Reg}^{+}$ in (18) it can be argued analogously, using Prop. 14, (ii), and unfolding/folding between closed derivations in $\mathbf{Reg}^{+}$ and closed, admissible derivations in $\mathbf{Reg}^{+,\infty}$. Soundness of $\mathbf{Reg}^{+}_{\mathbf{0}}$ follows from soundness of $\mathbf{Reg}^{+}$. For completeness of $\mathbf{Reg}^{+}_{\mathbf{0}}$, note that every closed, admissible derivation $\mathcal{T}$ in $\mathbf{Reg}^{+,\infty}$ can be 'folded' into a closed derivation of $\mathbf{Reg}^{+}_{\mathbf{0}}$ by using a stricter version of repetition of terms: distinct occurrences of a term $(\lambda\vec{y})U$ on a thread of a prooftree form such a repetition only if all formulas in between have an equally long or longer abstraction prefix. Since $\mathcal{T}$ is admissible, on every infinite thread θ of $\mathcal{T}$ there must occur such a stricter form of repetition, namely of a term with the shortest abstraction prefix among the terms that occur infinitely often on θ. ◀

4 Binding–Capturing Chains

In this section we develop a characterization of strongly regular infinite λ-terms through a property of their term structure, concerning 'binding–capturing chains' on positions of the term. While not needed for obtaining the result concerning $\boldsymbol{\lambda}_{\mathsf{letrec}}$-expressibility in Section 5, we think that this characterization is of independent interest. However, we only outline its proof here, which can be found in [6] and in a report [7] that accompanies this article.

Binding–capturing chains originate from the notion of 'gripping' due to Melliès [11], and from techniques concerning the notion of 'holding' of redexes developed by van Oostrom [12]. In [5] they have been used to study α-conversion-avoiding μ-unfolding.

Technically, binding–capturing chains are alternations of two kinds of links between positions of variable occurrences and λ-abstractions (called binders below) in a λ-term: 'binding links' from a λ-abstraction downward to the variable occurrences it binds, and 'capturing links' from a variable occurrence upward to λ-abstractions that do not bind it, but are situated on the upward path to its binding λ-abstraction. We formalize these links by binding and capturing relations, which are then used to define binding–capturing chains.

▶ **Definition 19** (binding, capturing). For every $T \in \mathit{Ter}(\boldsymbol{\lambda}^{\infty})$ we define two binary relations on the set $\mathit{Pos}(T)$ of positions of T: the *binding relation* $\multimapdotinv$, and the *capturing relation* $\dashrightarrow$. (For positions in iCRS-terms, see [10].) For defining these relations let $p, q \in \mathit{Pos}(T)$.

$p \multimapdotinv q$ (in words: a binder, that is, a λ-abstraction, at p *binds* a variable occurrence at q) holds if p is a binder position, and q a variable position in T, and the binder at position p binds the variable occurrence at position q.

$q \dashrightarrow p$ (in words: a variable occurrence at q *is captured by* a binder at p), and conversely $p \dashleftarrow q$ (the binder at p *captures* a variable occurrence at q), hold if q is a variable position and $p < q$ a binder position in T, and there is no binder position q_0 in T with $p \leq q_0$ and $q_0 \multimapdotinv q$.

▶ **Definition 20** (binding–capturing chain). Let $T \in \mathit{Ter}(\boldsymbol{\lambda}^{\infty})$. A finite or infinite sequence $\langle p_0, q_1, p_1, q_2, p_2, \ldots \rangle$ in $\mathit{Pos}(T)$ is called a *binding–capturing chain in* T if it links positions alternatingly via binding and capturing: $p_1 \multimapdotinv q_2 \dashrightarrow p_2 \multimapdotinv q_3 \dashrightarrow p_3 \multimapdotinv \ldots$, starting with a binding and ending with a capturing. Its *length* is the number of 'is captured by' links.

See Figs. 1 and 2 for illustrations of binding–capturing chains in terms we have encountered. Next we introduce a position-annotated variant $\mathit{Reg}^{+}_{\mathit{pos}}$ of Reg^{+} in order to relate binding–capturing chains to rewrite sequences in Reg^{+}. The idea is that if a λ-term T has a generated subterm $(\lambda y_1 \dots y_n)U$ in Reg^{+}, then $(\lambda y_1 \dots y_n)^{q}_{p_1,\dots,p_n} U$ is a generated subterm in $\mathit{Reg}^{+}_{\mathit{pos}}$, where $p_1,\dots,p_n$ are the positions in T from which the bindings $\lambda y_1 \dots y_n$ in the abstraction prefix descend, and q is the position in T of the body U of this generated subterm.

▶ **Definition 21** (position-annotated variant $\mathit{Reg}^{+}_{\mathit{pos}}$ of Reg^{+})**.** On $Ter((\boldsymbol{\lambda}^{\infty}))$ we consider the following rewrite rules in informal notation:

$$(\varrho^{@_i}_{pos}): \ (\lambda x_1 \dots x_n)^{q}_{p_1,\dots,p_n} T_0\, T_1 \to (\lambda x_1 \dots x_n)^{qi}_{p_1,\dots,p_n} T_i \qquad \text{(for each } i \in \{0,1\}\text{)}$$
$$(\varrho^{\lambda}_{pos}): \ (\lambda x_1 \dots x_n)^{q}_{p_1,\dots,p_n} \lambda y. T_0 \to (\lambda x_1 \dots x_n y)^{q00}_{p_1,\dots,p_n,q} T_0$$
$$(\varrho^{\mathsf{S}}_{pos}): (\lambda x_1 \dots x_{n+1})^{q}_{p_1,\dots,p_{n+1}} T_0 \to (\lambda x_1 \dots x_n)^{q}_{p_1,\dots,p_n} T_0 \qquad \text{(if binding } \lambda x_{n+1} \text{ is vacuous)}$$

The change of the term-body position in a λ-decomposition step is motivated by the underlying CRS-notation for terms in $(\boldsymbol{\lambda}^{\infty})$: when a term $\mathsf{abs}([y]T_0)$ representing a λ-abstraction starts at position q, its binding is declared at position $q0$, and its body T_0 starts at position $q00$.

By Reg^{+}_{pos} we denote the abstract rewriting system induced, similar to the definition of Reg^{+} in Def. 4 earlier, by the rules above on position-annotated terms in $Ter((\boldsymbol{\lambda}^{\infty}))$.

Also analogously to Def. 4, by $\to_{\mathrm{reg}^{+}}$ we denote the $\varrho^{\mathsf{S}}_{pos}$-eager rewrite strategy for Reg^{+}_{pos}.

$\to_{\mathrm{reg}^{+}}$-rewrite sequences on terms in $Ter((\boldsymbol{\lambda}^{\infty}))$ are related to $\to_{\mathrm{reg}^{+}}$-rewrite sequences on position-annotated terms via lifting (adding annotations) and projecting (dropping annotations). The proposition and the lemma below describe the connection between position-annotated $\to_{\mathrm{reg}^{+}}$-rewrite sequences and the concepts of binding, capturing, and binding–capturing chains. Then a lemma and the main theorem of this section are given.

▶ **Proposition 22.** For all $T \in Ter(\boldsymbol{\lambda}^{\infty})$ and positions $p, q \in Pos(T)$ it holds:

$$p \multimapinv q \iff ()^{\epsilon}_{\langle\rangle} T \twoheadrightarrow_{\mathrm{reg}^{+}} (\lambda x_1 \dots x_n)^{q}_{p_1,\dots,p_n} x_n \ \wedge \ p = p_n$$
$$p \dashleftarrow q \iff ()^{\epsilon}_{\langle\rangle} T \twoheadrightarrow_{\mathrm{reg}^{+}} (\lambda x_1 \dots x_i \dots x_n)^{q'}_{p_1,\dots,p_i,\dots,p_n} U \twoheadrightarrow_{\mathrm{reg}^{+}} (\lambda x_1 \dots x_i)^{q}_{p_1,\dots,p_i} x_i$$
$$\text{for some } i < n \text{ such that } p \in \{p_{i+1}, \dots, p_n\}$$

▶ **Lemma 23** (binding–capturing chains)**.** *For all $T \in Ter^{\infty}(\boldsymbol{\lambda})$ it holds:*

(i) *If $()^{\epsilon} T \twoheadrightarrow_{\mathrm{reg}^{+}} (\lambda x_1 \dots x_n)^{q}_{p_1,\dots,p_n} U$, then $p_1 \multimapinv q_2 \dashrightarrow p_2 \multimapinv \dots \multimapinv q_n \dashrightarrow p_n$ holds for some $q_2, \dots, q_n \in Pos(T)$.*

(ii) *If $p_1 \multimapinv q_2 \dashrightarrow p_2 \multimapinv \dots \multimapinv q_n \dashrightarrow p_n$ is a binding–capturing chain in T, then there exist $r_1, \dots, r_m, s \in Pos(T)$ with $m \geq n$ such that $()^{\epsilon} T \twoheadrightarrow_{\mathrm{reg}^{+}} (\lambda x_1 \dots x_m)^{s}_{r_1,\dots,r_m} U$ and furthermore $p_1, \dots, p_n \in \{r_1, \dots, r_m\}$ such that $p_1 < p_2 < \dots < p_n = r_m$.*

▶ **Lemma 24** (infinite binding–capturing chains)**.** *Let T be an infinite λ-term, and let τ be an infinite $\to_{\mathrm{reg}^{+}}$-rewrite sequence $()T = (\lambda \vec{x}_0)T_0 \to_{\mathrm{reg}^{+}} (\lambda \vec{x}_1)T_1 \to_{\mathrm{reg}^{+}} \dots$ with the property $\lim_{i \to \infty} |\vec{x}_i| = \infty$. Then there exists an infinite binding–capturing chain in T.*

Proof (Sketch). The assumed infinite $\to_{\mathrm{reg}^{+}}$-rewrite sequence can be lifted to one with position annotations $()^{\epsilon} T = (\lambda \vec{x}_0)^{\epsilon}_{\vec{p}_0} T_0 \to_{\mathrm{reg}^{+}} (\lambda \vec{x}_1)^{q_1}_{\vec{p}_1} T_1 \to_{\mathrm{reg}^{+}} (\lambda \vec{x}_2)^{q_2}_{\vec{p}_2} T_2 \to_{\mathrm{reg}^{+}} \dots$ where q_i are positions and $\vec{p}_i = \langle p_1, \dots, p_{m_i} \rangle$ vectors of positions. Due to $\lim_{i \to \infty} |\vec{x}_i| = \liminf_{i \to \infty} |\vec{x}_i| = \infty$ the sequence is of the form: $()^{\epsilon} T = (\lambda \vec{x}_{i_0})^{q_{i_0}}_{\vec{p}_{i_0}} T_{i_0} \twoheadrightarrow_{\mathrm{reg}^{+}} (\lambda \vec{x}_{i_1})^{q_{i_1}}_{\vec{p}_{i_1}} T_{i_1} \twoheadrightarrow_{\mathrm{reg}^{+}} (\lambda \vec{x}_{i_2})^{q_{i_2}}_{\vec{p}_{i_2}} T_{i_2} \twoheadrightarrow_{\mathrm{reg}^{+}} \dots$ with $0 = |\vec{x}_{i_0}| < |\vec{x}_{i_1}| < |\vec{x}_{i_2}| < \dots$ and such that $|\vec{x}_{i_j}| \leq |\vec{x}_k|$ holds for all $j, k \in \mathbb{N}$ with $k \geq i_j$. Since steps in $\mathit{Reg}^{+}_{\mathit{pos}}$ remove position annotations only when the corresponding abstraction variable is dropped from the prefix in an $\to_{\mathsf{S}}$-step, it follows that $\vec{p}_{i_0} < \vec{p}_{i_1} < \vec{p}_{i_2} < \dots$ holds with respect to the prefix order $<$. Hence in the limit these vectors tend to an infinite sequence of positions $r = \langle r_1, r_2, r_3, \cdots \rangle$. Then Lemma 23, (23), can be used to show that the positions on r are the binder positions of an infinite binding–capturing chain in T. ◀

▶ **Theorem 25.** *An infinite λ-term is strongly regular if and only if it is regular and contains only finite binding–capturing chains.*

Proof (Sketch). Suppose that T is strongly regular. By Prop. 11, T is regular. Also, $ST^+(T)$ is finite. Let n be the length of the longest abstraction prefix in $ST^+(T)$. Then Lemma 23, (23), implies that the length of any binding–capturing chain in T is bounded by $n-1$.

Suppose that T is regular, but not strongly regular. Then $ST(T)$ is finite, while $ST^+(T)$ is infinite. Since the rewrite strategy $\to_{\text{reg}^+}$ has branching degree ≤ 2 (branching only happens at sources of $\to_{@_i}$-steps), Kőnig's Lemma implies that there is an infinite $\to_{\text{reg}^+}$-rewrite sequence $\tau \;:\; ()T = (\lambda\vec{x}_0)T_0 \to_{\text{reg}^+} (\lambda\vec{x}_1)T_1 \to_{\text{reg}^+} \ldots$ that passes through distinct terms. By Lemma 10, (10), τ projects to a $\to_{\text{reg}}$-rewrite sequence $\check{\tau} \;:\; ()T = (\lambda\vec{x}'_0)T_0 \twoheadrightarrow_{\text{reg}} (\lambda\vec{x}'_1)T_1 \twoheadrightarrow_{\text{reg}} \ldots$ under $\twoheadrightarrow_{\text{del}}$-rewrite sequences $(\lambda\vec{x}_i)T_i \twoheadrightarrow_{\text{del}} (\lambda\vec{x}'_i)T_i$, for all $i \in \mathbb{N}$, that respectively shortening the length of the abstraction prefix. As $ST(T)$ is finite, $\check{\tau}$ passes only through finitely many terms. This contrast with τ can be used to show that prefix lengths of the terms on τ must tend to infinity. Due to this, Lemma 24 is applicable to τ, and yields the existence of an infinite binding–capturing chain in T. ◀

5 Expressibility by terms of the λ-calculus with μ

Having adapted (in Section 2) the concept of regularity for infinite λ-terms in two ways, we now obtain an expressibility result for one of these adaptations that is analogous to that in [4] for regular first-order trees with respect to rational expressions (or equivalently, μ-terms). We show that an infinite λ-term is strongly regular if and only if it is $\boldsymbol{\lambda}_\mu$-expressible.

We first define terms of $\boldsymbol{\lambda}_\mu$, the unfolding rewrite relation, and $\boldsymbol{\lambda}_\mu$-expressibility.

▶ **Definition 26** (CRS-representation for $\boldsymbol{\lambda}_\mu$). The CRS-signature $\Sigma_{\lambda_\mu} = \Sigma_\lambda \cup \{\mathsf{mu}\}$ for $\boldsymbol{\lambda}_\mu$ extends Σ_λ by a unary function symbol mu. By $Ter(\boldsymbol{\lambda}_\mu)$ we denote the set of closed finite CRS-terms over Σ_{λ_μ} with the restriction that CRS-abstraction occurs only as an argument of the symbols abs or mu. By $Ter((\boldsymbol{\lambda}_\mu))$ we denote the analogously defined set of terms over the signature $\Sigma_{(\lambda)} \cup \{\mathsf{mu}\}$. We consider the *$\mu$-unfolding rule* in informal and formal notation:

$$(\varrho^\mu): \;\; \mu x.M(x) \to M(\mu x.M(x)) \qquad\qquad \varrho^\mu: \;\; \mathsf{mu}([x]Z(x)) \to Z(\mathsf{mu}([x]Z(x)))$$

This rule induces the *unfolding rewrite relation* $\to_\mu$ on $Ter(\boldsymbol{\lambda}_\mu)$ and $Ter((\boldsymbol{\lambda}_\mu))$. We say that a λ_μ-term M *expresses* an infinite λ-term V if $M \twoheadrightarrow\!\!\!\!\!\rightarrow_\mu V$ holds, that is, M unfolds to V via a typically infinite, strongly convergent $\to_\mu$-rewrite sequence (similar for terms in $Ter((\boldsymbol{\lambda}_\mu))$). And an infinite λ-term T is *λ_μ-expressible* if there is a λ_μ-term M that expresses T.

We sketch some intuition for the proof, which proceeds by a sequence of proof-theoretic transformations. We focus on the more difficult direction. Let T be a strongly regular infinite λ-term. We want to extract a λ_μ-term M that expresses T from the finite $\to_{\text{reg}^+}$-reduction graph G of T. We first obtain a closed derivation $\mathcal{D}$ of $()T$ in $\mathbf{Reg}^+_0$. The derivation $\mathcal{D}$ can be viewed as a finite term graph that has G as its homomorphic image, and that does not exhibit horizontal sharing ([3, Sec. 4.3]). Such term graphs correspond directly to λ_μ-terms (analogous to [3]). In order to extract the λ_μ-term M corresponding to $\mathcal{D}$ from this derivation, we annotate it inductively to a λ_μ-term-annotated derivation $\hat{\mathcal{D}}$ with conclusion $()\,M : T$ in a proof system $\mathbf{Expr}$ that is a variant of $\mathbf{Reg}^+_0$. Then it remains to show that M indeed

$$\frac{}{(\lambda\vec{x}y)\,y \overset{\text{unf}}{\Longrightarrow} (\lambda\vec{x}y)\,y}\;\mathsf{0} \qquad \frac{(\lambda\vec{x})\,M_0 \overset{\text{unf}}{\Longrightarrow} (\lambda\vec{x})\,T_0 \qquad (\lambda\vec{x})\,M_1 \overset{\text{unf}}{\Longrightarrow} (\lambda\vec{x})\,T_1}{(\lambda\vec{x})\,M_0\,M_1 \overset{\text{unf}}{\Longrightarrow} (\lambda\vec{x})\,T_0\,T_1}\;@$$

$$\frac{(\lambda\vec{x}y)\,M \overset{\text{unf}}{\Longrightarrow} (\lambda\vec{x}y)\,T}{(\lambda\vec{x})\,\lambda y.M \overset{\text{unf}}{\Longrightarrow} (\lambda\vec{x})\,\lambda y.T}\;\lambda \qquad \frac{(\lambda\vec{x})\,M \overset{\text{unf}}{\Longrightarrow} (\lambda\vec{x})\,T}{(\lambda\vec{x}y)\,M \overset{\text{unf}}{\Longrightarrow} (\lambda\vec{x}y)\,T}\;\mathsf{S}\ \text{(if the binding } \lambda y \text{ is vacuous in } M \text{ and } T\text{)}$$

$$\frac{(\lambda\vec{x})\,M(\mu f.M(f)) \overset{\text{unf}}{\Longrightarrow} (\lambda\vec{x})\,T}{(\lambda\vec{x})\,\mu f.M(f) \overset{\text{unf}}{\Longrightarrow} (\lambda\vec{x})\,T}\;\mu$$

Figure 5 Proof system $\mathbf{Unf}^{\infty}$ for completely unfolding of λ_μ-terms into infinite λ-terms.

unfolds to T. For this we prove that $\hat{\mathcal{D}}$ unfolds to/gives rise to infinite derivations the variant systems $\mathbf{Expr}^{\infty}$ and $\mathbf{Unf}^{\infty}$, which witnesses infinite outermost rewrite sequences $M \twoheadrightarrow\!\!\!\!\!\rightarrow_\mu T$.

The CRS consisting of the rule ϱ^μ is orthogonal and fully-extended [13]. As a consequence of the result in [9] that outermost-fair strategies in orthogonal, fully extended iCRSs are normalizing, we obtain the following proposition.

▶ **Proposition 27.** Let $M \in Ter(\boldsymbol{\lambda}_\mu)$ and $T \in Ter(\boldsymbol{\lambda}^{\infty})$. If M expresses T, then there is an outermost $\rightarrow_\mu$-rewrite sequence of length $\leq\omega$ that witnesses $M \twoheadrightarrow\!\!\!\!\!\rightarrow_\mu T$, and T is the unique λ-term expressed by M. Analogously for prefixed terms in $\boldsymbol{\lambda}_\mu$ that express prefixed λ-terms. Hence the infinite outermost unfolding rewrite relation $\overset{\text{out}}{\twoheadrightarrow\!\!\!\!\!\rightarrow}{}^!_\mu$ to infinite normal form defines a partial mapping from $Ter(\boldsymbol{\lambda}_\mu)$ to $Ter(\boldsymbol{\lambda}^{\infty})$, and from $Ter((\boldsymbol{\lambda}_\mu))$ to $Ter((\boldsymbol{\lambda}^{\infty}))$.

The relation $\overset{\text{out}}{\twoheadrightarrow\!\!\!\!\!\rightarrow}{}^!_\mu$ can be defined via derivability in the proof system $\mathbf{Unf}^{\infty}$ in Fig. 5: the existence of a possibly infinite derivation that is closed in the sense of Def. 12, and *admissible*, i.e. it is (S)-eager, and does not contain infinitely many consecutive instances of the rule (μ).

▶ **Proposition 28.** $\mathbf{Unf}^{\infty}$ is sound and complete w.r.t. $\overset{\text{out}}{\twoheadrightarrow\!\!\!\!\!\rightarrow}{}^!_\mu$: For all $(\lambda\vec{x})T \in Ter((\boldsymbol{\lambda}^{\infty}))$ and $(\lambda\vec{x})M \in Ter((\boldsymbol{\lambda}_\mu))$, $\vdash_{\mathbf{Unf}^{\infty}} (\lambda\vec{x})\,M \overset{\text{unf}}{\Longrightarrow} (\lambda\vec{x})\,T$ holds if and only if $(\lambda\vec{x})\,M \overset{\text{out}}{\twoheadrightarrow\!\!\!\!\!\rightarrow}{}^!_\mu (\lambda\vec{x})\,T$.

▶ **Definition 29** (proof systems $\mathbf{Expr}$, $\mathbf{Expr}^{\infty}$, and $\mathbf{Expr}_{\boldsymbol{\mu}}$, $\mathbf{Expr}^{\infty}_{\boldsymbol{\mu}}$)**.** The natural-deduction-style proof system $\mathbf{Expr}$ has as its formulas abstraction prefixed λ_μ-terms annotated by infinite λ-terms, and the rules in Fig. 6. The system $\mathbf{Expr}_{\boldsymbol{\mu}}$ has abstraction prefixed λ_μ-terms as formulas, and its rules arise from $\mathbf{Expr}_{\boldsymbol{\mu}}$ by dropping the λ-terms. Derivability in these systems means the existence of a closed (no open assumptions), (S)-eager, finite derivation.

The variant $\mathbf{Expr}^{\infty}$ of $\mathbf{Expr}$ arises by replacing the rule (FIX) with the rule (μ) in Fig. 7. $\mathbf{Expr}^{\infty}_{\boldsymbol{\mu}}$ arises from $\mathbf{Expr}_{\boldsymbol{\mu}}$ analogously. A derivation in either of these systems is *admissible* if it does not contain infinitely many consecutive instances of (μ), and if it is (S)-eager in the sense of Def. 13. Derivability in these systems means the existence of an admissible derivation that is closed in the sense of Def. 12.

▶ **Lemma 30.** (i) *For every λ_μ-term M: $\vdash_{\mathbf{Expr}_{\boldsymbol{\mu}}} ()M$ if and only if $\vdash_{\mathbf{Expr}^{\infty}_{\boldsymbol{\mu}}} ()M$.*

(ii) *Every closed derivation in $\mathbf{Expr}^{\infty}_{\boldsymbol{\mu}}$ contains only finitely many λ_μ-terms.*

(iii) *For every λ_μ-term M it holds: $\vdash_{\mathbf{Expr}_{\boldsymbol{\mu}}} ()M$ if and only if there is no $\rightarrow_{\text{reg}^+}$-generated subterm of M (in $ST^+(M)$) of the form $()\mu x_0 \ldots x_n.x_0$ for $n \in \mathbb{N}$.*

Proof. For (30), in order to show "$\Rightarrow$" let $\mathcal{D}$ be a finite, closed, (S)-eager derivation in $\mathbf{Expr}_{\boldsymbol{\mu}}$ with conclusion $()M$. By 'unfolding' this derivation through a process in which in each step:

$$\frac{}{(\lambda\vec{x}y)\; y : y}\;\mathsf{0} \qquad \frac{(\lambda\vec{x}y)\; M : T}{(\lambda\vec{x})\; \lambda y.M : \lambda y.T}\;\lambda \qquad \frac{(\lambda\vec{x})\; M_0 : T_0 \qquad (\lambda\vec{x})\; M_1 : T_1}{(\lambda\vec{x})\; M_0\, M_1 : T_0\, T_1}\;@$$

$$\frac{\begin{matrix}[(\lambda\vec{x})\; \mathsf{c}_l : T]^l \\ \mathcal{D}_0 \\ (\lambda\vec{x})\; M(\mathsf{c}_l) : T\end{matrix}}{(\lambda\vec{x})\; \mu f.M(f) : T}\;\mathrm{FIX}, l \qquad \frac{(\lambda\vec{x})\; M : T}{(\lambda\vec{x}y)\; M : T}\;\mathsf{S}\ \text{(if the binding } \lambda y \text{ is vacuous)}$$

(if $|\mathcal{D}_0| \geq 1$, and $|\vec{y}| \geq |\vec{x}|$ for all $(\lambda\vec{y})\; N : U$ on threads from open assumptions $((\lambda\vec{x})\; \mathsf{c}_l : T)^l$ down)

Figure 6 Natural-deduction style proof system **Expr** for expressibility of infinite λ-terms by λ_μ-terms. The proof system $\mathbf{Expr}_\mu$ for λ_μ-terms that express infinite λ-terms arises by dropping the colons ':' and the subsequent infinite λ-terms. Derivations in **Expr** and $\mathbf{Expr}_\mu$ must be (S)-eager.

$$\frac{(\lambda\vec{x})\; M(\mu f.M(f)) : T}{(\lambda\vec{x})\; \mu f.M(f) : T}\;\mu$$

Figure 7 The proof system $\mathbf{Expr}^\infty$ for expressibility of λ-terms by λ_μ-terms arises from **Expr** by replacing the rule FIX with the rule μ. The proof system $\mathbf{Expr}^\infty_\mu$ for λ_μ-terms that express λ-terms arises from $\mathbf{Expr}^\infty$ by dropping the colons ':' and the subsequent infinite λ-terms. Admissible derivations in $\mathbf{Expr}^\infty$ and in $\mathbf{Expr}^\infty_\mu$ do not have infinitely many consecutive instances of μ.

$$\text{a subderivation of a bottommost instance of FIX}\quad \frac{\begin{matrix}[(\lambda\vec{y})\mathsf{c}_l]^l \\ \mathcal{D}_0(\mathsf{c}_l) \\ (\lambda\vec{y})\, N(\mathsf{c}_l)\end{matrix}}{(\lambda\vec{y})\, \mu f.N(f)}\;\mathrm{FIX}, l \quad \text{is 'unfolded' into a subderivation} \quad \frac{\begin{matrix}\dfrac{\begin{matrix}[(\lambda\vec{y})\mathsf{c}_l]^l \\ \mathcal{D}_0(\mathsf{c}_l) \\ (\lambda\vec{y})\, N(\mathsf{c}_l)\end{matrix}}{[(\lambda\vec{y})\, \mu f.N(f)]}\;\mathrm{FIX}, l \\ \mathcal{D}_0(\mu f.N(f)) \\ (\lambda\vec{y})\, N(\mu f.N(f))\end{matrix}}{(\lambda\vec{y})\, \mu f.N(f)}\;\mu$$

in the limit a closed derivation $\mathcal{T}$ in $\mathbf{Expr}^\infty_\mu$ is obtained with the same conclusion as $\mathcal{D}$. Furthermore, $\mathcal{T}$ does not contain infinitely many consecutive instances of μ, since the side-condition on (FIX) guarantees a guardedness condition analogous to Prop. 16. Hence $\mathcal{T}$ is a closed admissible derivation in $\mathbf{Expr}^\infty_\mu$ with conclusion $()M$. For showing "$\Leftarrow$", suppose that $\mathcal{T}$ is a closed admissible derivation in $\mathbf{Expr}^\infty_\mu$ with conclusion $()M$. Then there is a finite closed derivation $\mathcal{D}$ with the same conclusion in the variant system $\mathbf{Expr}^\infty_{\mu,-}$ that does not require the side-condition part $|\mathcal{D}_0| \geq 1$ for instances of FIX. Via the process described above, $\mathcal{D}$ unfolds to a closed, (S)-eager derivation in $\mathbf{Expr}^\infty_\mu$, which has to be equal to $\mathcal{T}$, since closed (S)-eager derivations in $\mathbf{Expr}^\infty_\mu$ are unique (due to the rules of this system). If $\mathcal{D}$ would not satisfy the guardedness condition described in Prop. 16, and therefore would also violate the mentioned side-condition part, for any of its FIX-instances, then $\mathcal{T}$ would not be admissible. It follows that $\mathcal{D}$ is a closed derivation in $\mathbf{Expr}_\mu$ with conclusion $()M$.

For (30) note that by the argument for "$\Leftarrow$" in (30), every closed derivation in $\mathbf{Expr}^\infty_\mu$ is the unfolding of a closed derivation in $\mathbf{Expr}_\mu$, and that the unfolding process can produce only finitely many λ_μ-terms. Statement (30) follows by an easy analysis of closed derivations in $\mathbf{Expr}^\infty_{\mu,-}$ that violate the guardedness condition in Prop. 16 on any of its FIX-instances. ◀

▶ **Lemma 31.** *For all infinite prefixed λ-terms T, $\vdash_{\mathbf{Reg}^+_0} (\lambda\vec{x})T$ holds if and only if there exists a λ_μ-term M such that $\vdash_{\mathbf{Expr}} (\lambda\vec{x})\; M : T$.*

Proof (Sketch). Every derivation $\mathcal{D}$ in $\mathbf{Reg}^+_0$ with conclusion $(\lambda\vec{x})T$ can be annotated with appropriate λ_μ-terms, by induction on the derivation depth, thereby introducing appropriate constants and exploiting the form of the rules in **Expr**, to a derivation $\hat{\mathcal{D}}$ in **Expr** with conclusion $(\lambda\vec{x})\ M : T$ and corresponding assumptions, for some prefixed λ_μ-term $(\lambda\vec{x})M$.

Conversely, the result of dropping the λ_μ-terms and the subsequent colons in a derivation $\mathcal{D}'$ in **Expr** results in a derivation $\check{\mathcal{D}}'$ in $\mathbf{Reg}^+_\mathbf{0}$. ◀

▶ **Example 32.** The derivation $\mathcal{D}_l$ in $\mathbf{Reg}^+_\mathbf{0}$ from Example 17, (i), on the left can be annotated, as described by Lemma 31 to obtain the following derivation $\hat{\mathcal{D}}_l$ in **Expr** :

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{((\,)\,\mathsf{c}_l : T)^l}{(\lambda x)\ \mathsf{c}_l : T}\,\mathsf{S}}{(\lambda xy)\ \mathsf{c}_l : T}\,\mathsf{S} \qquad \dfrac{}{(\lambda xy)\ y : y}\,\mathsf{0}}{(\lambda xy)\ \mathsf{c}_l\, y : T\, y}\,@ \qquad \dfrac{\dfrac{}{(\lambda x)\ x : x}\,\mathsf{0}}{(\lambda xy)\ x : x}\,\mathsf{S}}{(\lambda xy)\ \mathsf{c}_l\, y\, x : T\, y\, x}\,@}{(\lambda x)\ \lambda y.\mathsf{c}_l\, y\, x : T\, y\, x}\,\lambda}{(\,)\ \lambda xy.\mathsf{c}_l\, y\, x : \lambda xy.T\, y\, x}\,\lambda}{(\,)\ \mu f.\lambda xy.f\, y\, x : T}\,\mathrm{FIX}, u$$

Note that the λ_μ-term in the conclusion unfolds to T, the infinite λ-term in Fig. 1.

▶ **Theorem 33.** *The proof system* **Expr** *is sound and complete with respect to* $\twoheadrightarrow^!_\mu$ *: for all expressions* $(\lambda\vec{x})\ M : T$ *of* λ_μ*-term-annotated, prefixed infinite* λ*-terms it holds that* $\vdash_{\mathbf{Expr}} (\lambda\vec{x})\ M : T$ *if and only if* $(\lambda\vec{x})M \twoheadrightarrow^!_\mu (\lambda\vec{x})T$ *.*

Proof. For "$\Rightarrow$" let $\mathcal{D}$ be a closed, (S)-eager derivation in **Expr** with conclusion $(\lambda\vec{x})\ M : T$. By an unfolding process and arguments analogous as described in the proof of Lemma 30, $\mathcal{D}$ unfolds to a closed admissible prooftree $\mathcal{T}$ in $\mathbf{Expr}^\infty$ with the same conclusion. By changing all symbols ":" in $\mathcal{T}$ to "$\overset{\text{unf}}{\Longrightarrow}$", and distributing the abstraction prefixes in the expressions of $\mathcal{T}$ over "$\overset{\text{unf}}{\Longrightarrow}$", a closed admissible prooftree $\mathcal{T}'$ in $\mathbf{Unf}^\infty$ is obtained that has the conclusion $(\lambda\vec{x})M \overset{\text{unf}}{\Longrightarrow} (\lambda\vec{x})T$. Then by Prop. 28 it follows that $(\lambda\vec{x})M \twoheadrightarrow^!_\mu (\lambda\vec{x})T$.

For "$\Leftarrow$", suppose that $(\lambda\vec{x})M \twoheadrightarrow^!_\mu (\lambda\vec{x})T$ holds. Then Prop. 28 entails that there is a closed admissible derivation $\mathcal{T}$ in $\mathbf{Unf}^\infty$ with the conclusion $(\lambda\vec{x})M \overset{\text{unf}}{\Longrightarrow} (\lambda\vec{x})T$. Since subderivations of closed admissible derivations in $\mathbf{Unf}^\infty$ are again such derivations, it follows by Prop. 28 and Prop. 27 that $\mathcal{T}$ does not contain more infinite λ-terms than λ_μ-terms. By dropping the symbols $\overset{\text{unf}}{\Longrightarrow}$ and the infinite λ-terms on the right, a closed admissible derivation $\mathcal{T}_\mu$ in $\mathbf{Expr}^\infty_\mu$ is obtained. Due to Lemma 30, (30), $\mathcal{T}_\mu$, and hence $\mathcal{T}$, contains only finitely many λ_μ-terms. As $\mathcal{T}$ does not contain more infinite λ-terms than λ_μ-terms, it follows that $\mathcal{T}$ contains only finitely many formulas. By changing all symbols "$\overset{\text{unf}}{\Longrightarrow}$" in $\mathcal{T}$ into ":", and gathering the abstraction prefixes in the expressions of $\mathcal{T}$, a closed admissible prooftree $\mathcal{T}'$ in $\mathbf{Expr}^\infty$ with the conclusion $(\lambda\vec{x})\ M : T$ and only finitely many formulas are obtained.

Finally, similar as in the proof of Theorem 18, $\mathcal{T}'$ can be 'folded' into a finite closed derivation $\mathcal{D}'$ in **Expr** with conclusion $(\lambda\vec{x})\ M : T$ by introducing FIX-instances to cut off the derivation above the upper occurrence of a repetition (the side-condition on such instances of FIX is guaranteed due to admissibility of $\mathcal{T}'$). ◀

▶ **Theorem 34.** *An infinite* λ*-term is* λ_μ*-expressible if and only if it is strongly regular.*

Proof. For all infinite λ-terms T it holds:

$$\begin{array}{rcll} T \text{ is } \lambda_\mu\text{-expressible} & \Longleftrightarrow & \exists M \in \mathit{Ter}(\boldsymbol{\lambda}_\mu).\ M \twoheadrightarrow^!_\mu T & \text{(by Prop. 27)} \\ & \Longleftrightarrow & \exists M \in \mathit{Ter}(\boldsymbol{\lambda}_\mu).\ \vdash_{\mathbf{Expr}} (\,)\ M : T & \text{(by Theorem 33)} \end{array}$$

$$\iff \ \vdash_{\mathbf{Reg}^+_0} ()T \qquad \text{(by Theorem 31)}$$
$$\iff \ T \text{ is strongly regular} \qquad \text{(by Theorem 18, (18)),}$$

which establishes the statement of the theorem. ◀

As a consequence of Theorems 34 and 25 we obtain a theorem with our main results.

▶ **Theorem 35.** *For all infinite λ-terms the following statements are equivalent:*

(i) *T is λ_μ-expressible.*
(ii) *T is strongly regular.*
(iii) *T is regular, and it only contains finite binding–capturing chains.*

6 Generalization to $\boldsymbol{\lambda}_{\mathsf{letrec}}$ and practical perspectives

In [6] we undertook an in-depth study of expressibility in $\boldsymbol{\lambda}_{\mathsf{letrec}}$, and obtained the more general, but analogous result for full $\boldsymbol{\lambda}_{\mathsf{letrec}}$ instead of only for $\boldsymbol{\lambda}_\mu$. While there are significantly more technicalities involved, the structure of the proofs is analogous to here. Instead of demanding eager application of the scope-delimiting rules ϱ^{del} and ϱ^{S}, respectively, there we study λ-term decomposition $\rightarrow^{\mathbb{S}}_{\text{reg}}$ and $\rightarrow^{\mathbb{S}}_{\text{reg}^+}$ for arbitrary scope-delimiting strategies $\mathbb{S}$.

Concepts introduced here and in [6] have the potential to be practically relevant for the implementation of functional programming languages. In [8] we study various higher-order and first-order term-graph representations of cyclic λ-terms. Their definitions draw heavily on the decomposition rewrite systems in this paper. That is, every term in $\boldsymbol{\lambda}_{\mathsf{letrec}}$ can be translated into a finite first-order 'λ-term-graph' by applying the rewrite strategy $\rightarrow_{\text{reg}^+}$ to the expressed strongly regular, infinite λ-term. Thereby vertices with the labels λ, @, S are created according to the kind of $\rightarrow_{\text{reg}^+}$-step observed (plus variable occurrence vertices with label $\mathsf{0}$). The degree of sharing exhibited by λ-term-graphs can be analyzed with functional bisimulation. In [8] we identify a class of first-order representations with eager application of scope closure that faithfully preserves and reflects the sharing order on higher-order term graphs. This leads to an algorithm for efficiently determining the maximally shared form of a term in $\boldsymbol{\lambda}_{\mathsf{letrec}}$, which can be put to use in a compiler as part of an optimizing transformation.

Associated with this article is the report [7], which contains more details on Section 4. Also closely related is the report [6] about the more general case of expressibility in $\boldsymbol{\lambda}_{\mathsf{letrec}}$.

References

1 Zena M. Ariola and Stefan Blom. Cyclic Lambda Calculi. In Martin Abadi and Takayasu Ito, editors, *Proceedings of TACS'97*, volume 1281 of *LNCS*, pages 77–106. Springer, 1997.
2 Zena M. Ariola and Jan Willem Klop. Lambda Calculus with Explicit Recursion. *Information and Computation*, 139(2):154–233, 1997.
3 Stefan Blom. *Term Graph Rewriting – Syntax and Semantics*. PhD thesis, Vrije Universiteit Amsterdam, 2001.
4 Bruno Courcelle. Fundamental Properties of Infinite Trees. *Theoretical Computer Science*, 25(2):95–169, 1983.
5 Jörg Endrullis, Clemens Grabmayer, Jan Willem Klop, and Vincent van Oostrom. On Equal μ-Terms. In I. Bethke, A. Ponse, and P.H. Rodenburg, editors, *Festschrift in Honour of Jan Bergstra*, Special Issue of TCS, 412 (28), pages 3175–3202. Elsevier, June 2011.
6 Clemens Grabmayer and Jan Rochel. Expressibility in the Lambda-Calculus with Letrec. Technical Report arXiv:1208.2383, `arxiv.org`, August 2012.
7 Clemens Grabmayer and Jan Rochel. Expressibility in the Lambda Calculus with μ. Technical Report arxiv:1304.6284, `arxiv.org`, 2013. Extends this article.

8 Clemens Grabmayer and Jan Rochel. Term Graph Representations for Cyclic Lambda Terms. In *Proc. of TERMGRAPH 2013*, number 110 in EPTCS, 2013. arXiv:1302.6338.
9 Jeroen Ketema and Jakob Grue Simonsen. Infinitary Combinatory Reduction Systems: Normalising Reduction Strategies. *Logical Methods in Computer Science*, 6(1:7):1–35, 2010.
10 Jeroen Ketema and Jakob Grue Simonsen. Infinitary Combinatory Reduction Systems. *Information and Computation*, 209(6):893 – 926, 2011.
11 Paul-André Melliès. *Description Abstraite des Systèmes de Réécriture (Thèse de doctorat)*. PhD thesis, l'Université Paris 7, December 1996.
12 Vincent van Oostrom. FD à la Melliès, February 1997. Vrije Universiteit Amsterdam.
13 Terese. *Term Rewriting Systems*, volume 55 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2003.