



An encoding of the λ -calculus in the String MultiSet Rewriting calculus

Attila Bagossy¹ · Péter Battyányi¹

Received: 27 August 2021 / Accepted: 17 November 2023
© The Author(s) 2024

Abstract

In this paper, we present an encoding of the λ -calculus in a multiset rewriting system and provide a few applications of the construction. For this purpose, we choose the calculus named String MultiSet Rewriting, which was introduced in Barbuti et al. (Electron Notes Theor Comput Sci 194:19–34, 2008) by Barbuti et al. With the help of our encoding, we give alternative proofs for the standardization and the finiteness of developments theorems in the λ -calculus.

1 Introduction

In past years, many formalisms have appeared in order to model biological systems based on interacting components. Most of these systems are high-level abstractions of biological phenomena aiming to describe the organization of the components of a biological system and the possible interactions between the components. For example, we would mention the κ -calculus of Danos and Laneve [11], which is a formal system modeling protein interaction, and the biochemical Stochastic π -calculus [16], which allows qualitative and quantitative investigations of composite systems. The Brane Calculus defined by Cardelli [8] is an efficient tool for the exact presentation of cell-level processes, and the same is true for the Calculus of Looping Sequences [3]. Many of the systems mentioned are based on multiset rewriting, and the same is true for membrane systems [10], chemical reaction systems [15] and Petri nets [9]. There are recent investigations concerning the introduction of regulating mechanism to reduce the extent of non-determinism and to enhance the expressive power of these systems [20]. A significant part of the attention devoted to these investigations is due to the effort of providing manageable but general enough modeling frameworks for biologically inspired computational systems [18].

The work is supported by the EFOP-3.6.3-VEKOP-16-2017-00002 project.

✉ Péter Battyányi
battyanyi.peter@inf.unideb.hu
Attila Bagossy
bagossyattila@mailbox.unideb.hu

¹ Department of Computer Science, Faculty of Informatics, University of Debrecen, Debrecen, Hungary

A form of multiset rewriting, which is the Calculus of Looping Sequences, was proposed in [14] offering a simple and easy to implement tool with a clear semantics to model biochemical processes. Translating higher-level systems to this calculus turned out to be difficult, however. Hence, the calculus of String MultiSet Rewriting (SMSR) was developed [4], which proved to be an appropriate intermediate solution: its main invention is the maximal matching operator, which facilitates the representation of higher-level languages. The underlying idea of that calculus is that, by grabbing the tree-like structures of terms of a formal system, a multiset of strings can describe the index paths from root to leaves in a given term and, consequently, substitutions with terms are encoded as replacement of a whole multiset of strings having a common prefix with another multiset in a single step.

In this paper, we introduce first the SMSR system following [4]. Then, we show how to represent λ -terms with multisets and define the multiset rewriting rule corresponding to β -reduction. We prove that this multiset calculus correctly simulates the λ -calculus equipped with β -reduction. With this in hand, we present two applications of the construction: We give a proof of the standardization theorem of the λ -calculus and a demonstration of the finiteness of developments theorem. The proofs benefit from the fact that strings in multisets can be considered index sequences identifying subterm occurrences in the corresponding λ -terms; hence, the statements concerning subterms, contractums and residuals can be formulated in a precise and natural way based on the multiset calculus. The proofs give some insight into the original λ -calculus notions: For example, we can address every subterm occurrence by some string and, hence, by giving an appropriate ordering to these strings, we can show that a β -reduction sequence is standard if and only if the sequence of index strings is non-decreasing. That is, the phrase “from left to right” used to explain standardness obtains its meaning. By the proof of the finiteness of developments theorem, we make use of a separation lemma applied in [21], the proof of which can possibly be presented in a more intuitive and clearer way when we resort to multisets. The present encoding can open the way for approaching theorems of the λ -calculus from a different perspective: It offers a computationally complete “index calculus” that may be the source of further investigations. It seems to be flexible enough to encode different evaluation semantics of the lambda calculus, or, possibly, resource lambda calculus [7], or symmetric lambda calculi [17] or non-deterministic calculi like, for example, the chemical programming model of Bânatre et al [2]. On the other hand, the calculus appears to be simple enough to serve as the foundation for implementations. There is already an ongoing research on the implementation of a multiset rewriting calculus [20] very similar to SMSR calculus. To sum it up, the multiset rewriting calculus provides a powerful tool in the study of rewriting systems, which is capable of both underpinning precise mathematical arguments and of permitting the creation of tools for real-life applications.

2 String MultiSet Rewriting

In this section, we provide a brief overview of the String MultiSet Rewriting (SMSR) calculus following the treatment in [4].

2.1 Strings and multisets

In what follows, we present the String MultiSet Rewriting (SMSR) calculus. Prior to this, we briefly recall the main notions regarding multisets.

Let \mathbb{N} denote the set of natural numbers and \mathbb{N}^+ denote the set of positive integers, respectively, and let U be a finite non-empty set. A multiset M over U is a pair $M = (U, f)$, where the mapping $f : U \rightarrow \mathbb{N}$ gives the multiplicity of each element $a \in U$. If $f(a) = 0$ for each $a \in U$, then M is the empty multiset. The number of elements in a multiset M , that is, the elements $a \in U$ with positive multiplicity, is denoted by $|M|$, and we refer to this number as the cardinality of M . In our case, the cardinality of a string multiset M is the number of strings in M .

Next, we define some elementary operations concerning multisets. Let $M_1 = (U, f_1)$, $M_2 = (U, f_2)$. Then, $(M_1 \sqcap M_2) = (O, f)$ where $f(a) = \min\{f_1(a), f_2(a)\}$; $(M_1 \sqcup M_2) = (O, f')$, where $f'(a) = \max\{f_1(a), f_2(a)\}$; $(M_1 \oplus M_2) = (O, f'')$, where $f''(a) = f_1(a) + f_2(a)$; $(M_1 \ominus M_2) = (O, f''')$ where $f'''(a) = \max\{f_1(a) - f_2(a), 0\}$; and $M_1 \sqsubseteq M_2$, if $f_1(a) \leq f_2(a)$ for all $a \in O$.

String MultiSet Rewriting (SMSR) is a term rewriting calculus. Firstly, we will define the syntax of terms, which are the strings and their multisets, and then a structural congruence relation on them. Afterward, a rewrite rule will be introduced which constitutes the instructions for the evolution of terms.

In the sequel, we will assume that an infinite alphabet \mathcal{E} is given.

Definition 1 (*Terms*) Multisets M and strings S over an alphabet \mathcal{E} are defined by the following Backus–Naur form:

$$\begin{aligned} M &= S \mid (M \mid M) \\ S &= \varepsilon \mid e \mid (S \cdot S) \end{aligned}$$

where ε is the empty string, e is a generic element of the alphabet \mathcal{E} , \cdot denotes string concatenation, and \mid stands for multiset union. The set of all multisets over \mathcal{E} is denoted by $\mathbf{M}(\mathcal{E})$, or by \mathbf{M} if \mathcal{E} is clear from the context. Similarly for $\mathbf{S}(\mathcal{E})$, which is the set of all strings over \mathcal{E} . We denote an arbitrary element of \mathbf{S} either by the letters P, Q, S, \dots possibly with indexes, or by smaller case Greek letter ξ, η, ν , etc. Let $|M|$ denote the cardinality of M as a multiset, while let $lh(M)$ denote the total number of occurrences of symbols of \mathcal{E} in M counted with multiplicities.

We introduce a structural congruence relation expressing the associativity of \cdot and \mid , commutativity of \mid , and we make sure that ε is an identity element either considered a string or a multiset.

Definition 2 Let ξ, ξ_1, ξ_2, ξ_3 be strings and M, M_1, M_2, M_3 be multisets. The structural congruence relation \equiv is the least congruence relation on multisets with the following properties:

$$\begin{aligned} (\xi_1 \cdot (\xi_2 \cdot \xi_3)) &\equiv ((\xi_1 \cdot \xi_2) \cdot \xi_3) & (\xi \cdot \varepsilon) &\equiv \xi \\ (M_1 \mid M_2) &\equiv (M_2 \mid M_1) & (M_1 \mid (M_2 \mid M_3)) &\equiv ((M_1 \mid M_2) \mid M_3) & (M \mid \varepsilon) &\equiv M \end{aligned}$$

Applying this congruence, we abbreviate expressions by omitting parentheses if possible. The outermost parentheses will be frequently omitted. In the sequel, we implicitly understand multiset equality modulo that congruence. We use standard equality to express that on the left- and right-hand side there stand the same objects (in most of the cases by virtue of some definition). Having the syntax and the structural congruence of terms at hand, we can now introduce the concept of patterns. We will assume an infinite set of variables $\mathcal{V} = \mathcal{V}_{\mathcal{E}} \cup \mathcal{V}_{\mathcal{S}} \cup \mathcal{V}_M$, where $\mathcal{V}_{\mathcal{E}} = \{x, y, z, \dots\}$ denotes the infinite set of *element variables*, $\mathcal{V}_{\mathcal{S}} = \{\tilde{x}, \tilde{y}, \tilde{z}, \dots\}$ denotes the infinite set of *string variables* and $\mathcal{V}_M = \{X, Y, Z, \dots\}$ stands for the infinite set of *multiset variables*. The definition below defines multiset patterns.

Definition 3 (*Patterns*) Multiset patterns and string patterns over an alphabet \mathcal{E} are defined applying the following Backus–Naur form:

$$\begin{aligned} MP &= SP \mid MP \mid MP \mid \{\!\{SP\}\!\}_X \mid \\ &\quad \{\!\{SP\}\!\}_X \triangleright_e \{\!\{SP\}\!\}_Y \\ SP &= \varepsilon \mid e \mid SP \cdot SP \mid \tilde{x} \mid x \end{aligned}$$

In the sequel, we denote by **MP** the set of multiset patterns and by **SP** the set of string patterns, respectively.

The above definition of multiset patterns differs from that applied in [4]. Namely, we augmented it with the construction $\{\!\{SP_1\}\!\}_X \triangleright \{\!\{SP_2\}\!\}_Y$ for arbitrary string patterns SP_1, SP_2 . This construction could be circumvented in the original string multiset calculus; however, by this, the encoding of λ -terms seemed to be more natural and smooth.

Multiset patterns are used to construct rewrite rules, which define the evolution of SMSR terms. A rewrite rule is a pair of multiset patterns, where the first pattern matches the term to be modified by the rule, while the second pattern matches the result of the rule application, which is the new term. Element variables of multiset patterns are substituted by elements of \mathcal{E} and string variables by strings during the reduction.

We call a function $\sigma : \mathcal{V}_{\mathcal{E}} \cup \mathcal{V}_{\mathcal{S}} \rightarrow \mathbf{S}$ an instantiation. Given an instantiation σ , we can extend it to a function $\sigma' : \mathbf{SP} \rightarrow \mathbf{S}$ in an obvious way, by letting σ' agree with σ on $\mathcal{V}_{\mathcal{E}} \cup \mathcal{V}_{\mathcal{S}}$ and by demanding that $\sigma'(SP_1 \cdot SP_2) = \sigma'(SP_1) \cdot \sigma'(SP_2)$ for any $SP_1, SP_2 \in \mathbf{SP}$ and $\sigma'(e) = e$ if $e \in \mathcal{E}$. In the sequel, we use the notation σ for σ' as well and call σ' an instantiation, also. Hence, to define an instantiation $\sigma : \mathbf{SP} \rightarrow \mathbf{S}$, it is enough to determine the values assigned by σ to the set of element and string variables, respectively. To define the meaning of a multiset pattern, besides the instantiation $\sigma : \mathbf{SP} \rightarrow \mathbf{S}$, we need a function $\rho : \mathcal{V}_{\mathbf{M}} \rightarrow \mathbb{N}^+$ and a pattern expansion function $\langle _ \rangle_{\rho}^- : \mathbf{MP} \times (\mathcal{V}_{\mathbf{M}} \rightarrow \mathbb{N}^+) \times (\mathbf{SP} \rightarrow \mathbf{S}) \rightarrow \mathbf{M}$.

The pattern expansion function transforms multiset patterns into multisets. Particularly, it assigns to a multiset pattern of the form $\{\!\{SP\}\!\}_X$ a finite union of multisets all having the same prefix, namely, the common prefix is obtained after evaluating SP with the help of ρ and σ , where $SP \in \mathbf{SP}$ is arbitrary. The patterns of the form $\{\!\{SP_1\}\!\}_X \triangleright_e \{\!\{SP_2\}\!\}_Y$ are evaluated to some “cross-product” of the multisets obtained by evaluating $\{\!\{SP_1\}\!\}_X$ and $\{\!\{SP_2\}\!\}_Y$ ($SP_1, SP_2 \in \mathbf{SP}$). The subsequent definitions will provide the exact notions.

Given multisets M_1, M_2 and element $e \in \mathcal{E}$, we define the operation of substituting the multiset M_2 in M_1 in place of e when the underlying occurrence of e is the rightmost element of a string in M_1 .

Definition 4 The multiset substitution function $.[\leftarrow \cdot] : \mathbf{M} \times \mathcal{E} \times \mathbf{M} \rightarrow \mathbf{M}$ acts in the following way.

1. $S_1[e \leftarrow S_2] = \begin{cases} S'_1 \cdot S_2, & \text{if } S_1 = S'_1 \cdot e, \\ S_1 & \text{otherwise.} \end{cases}$
2. $(M_1 \mid M_2)[e \leftarrow S] = (M_1[e \leftarrow S] \mid M_2[e \leftarrow S])$
3. $M_1[e \leftarrow (M_2 \mid M_3)] = (M_1[e \leftarrow M_2] \mid M_1[e \leftarrow M_3])$

Now, we are in a position to define the pattern expansion itself.

Definition 5 Let $SP, SP_1, SP_2 \in \mathbf{SP}$ and $MP_1, MP_2 \in \mathbf{MP}$. The pattern expansion function $\langle _ \rangle_{\rho}^- : \mathbf{MP} \times (\mathcal{V}_{\mathbf{M}} \rightarrow \mathbb{N}^+) \times (\mathbf{SP} \rightarrow \mathbf{S}) \rightarrow \mathbf{M}$ is recursively defined as follows:

1. $\langle SP \rangle_{\rho}^{\sigma} = \sigma(SP),$

2. $\langle \llbracket SP \rrbracket_X \rangle_\rho^\sigma = (\sigma(SP) \cdot \sigma(\tilde{x}_1) \mid \dots \mid \sigma(SP) \cdot \sigma(\tilde{x}_n))$ where $\tilde{x} \in \mathcal{V}_S$ and $\rho(X) = n$,
3. $\langle \llbracket SP_1 \rrbracket_X \triangleright_e \llbracket SP_2 \rrbracket_Y \rangle_\rho^\sigma = \langle \llbracket SP_1 \rrbracket_X \rangle_\rho^\sigma [e \leftarrow \langle \llbracket SP_2 \rrbracket_Y \rangle_\rho^\sigma]$,
4. $\langle MP_1 \mid MP_2 \rangle_\rho^\sigma = (\langle MP_1 \rangle_\rho^\sigma \mid \langle MP_2 \rangle_\rho^\sigma)$.

Observe that the operator \triangleright_e acts as a replacement with respect to the occurrences of e which are final occurrences of a string in $\langle \llbracket SP_1 \rrbracket_X \rangle_\rho^\sigma$. We illustrate this in the next example.

Example 6 Let $\mathcal{E} = \{a, b, c, d, e\}$, let $SP_1 = a, SP_2 = \varepsilon$. Let $\rho : \mathcal{V}_M \rightarrow \mathbb{N}^+$ and $\sigma : \mathbf{SP} \rightarrow \mathbf{S}$ in the multiset pattern $\llbracket SP_1 \rrbracket_X \triangleright_c \llbracket SP_2 \rrbracket_Y$ be defined as $\rho(X) = \rho(Y) = 2$ and $\sigma(\tilde{x}_1) = b, \sigma(\tilde{x}_2) = c$ and $\sigma(\tilde{y}_1) = d, \sigma(\tilde{y}_2) = e$. Then,

$$\begin{aligned} \langle \llbracket SP_1 \rrbracket_X \triangleright_c \llbracket SP_2 \rrbracket_Y \rangle_\rho^\sigma &\equiv (a \cdot b \mid a \cdot c) \triangleright_c (d \mid e) \\ &\equiv (a \cdot b \mid a \cdot c)[c \leftarrow (d \mid e)] \\ &\equiv ((a \cdot b)[c \leftarrow (d \mid e)] \mid (a \cdot c)[c \leftarrow (d \mid e)]) \\ &\equiv ((a \cdot b)[c \leftarrow d] \mid (a \cdot b)[c \leftarrow e] \mid \\ &\quad (a \cdot c)[c \leftarrow d] \mid (a \cdot c)[c \leftarrow e]) \\ &\equiv (a \cdot b \mid a \cdot b \mid a \cdot d \mid a \cdot e). \end{aligned}$$

2.2 The rewrite rules

We give an account, based on [4], of the reduction rule in the SMSR calculus. Before giving a full-fledged formal description of SMSR reduction, we try to grab the intuition behind it in the next definition.

Definition 7 1. Let $M, N \in \mathbf{M}$. We apply the notation $N \in M$ if $\exists P : M \equiv N \mid P$.

2. A rewrite rule is a pair (MP_1, MP_2) of multiset patterns such that MP_1 is non-empty. As before, we call a function $\sigma : \mathbf{SP} \rightarrow \mathbf{S}$ an instantiation. We denote the set of rewrite rules by \mathbf{R} .
3. Let M and $M' \in \mathbf{M}$. Then, M, M' can be matched by the rule (MP_1, MP_2) if $\langle MP_1 \rangle_\rho^\sigma \equiv M$ and $\langle MP_2 \rangle_\rho^\sigma \equiv M'$, where $\rho : \mathcal{V}_S \rightarrow \mathbb{N}$ and σ is an instantiation. Assume $M \in N$ and M, M' can be matched by $(MP_1, MP_2) \in \mathbf{R}$. Then, M can be exchanged for M' only if there is no string U in $N \setminus M$ such that there is a prefix U' of U which is a common prefix of all the strings in M .

We have to deal with some technical questions first. Namely, we make sure that rule applications always generate fresh names. With this aim, we introduce some auxiliary notions. The definition below is based on [4].

Definition 8 1. Let $Var(MP)$ denote the set of variables appearing in the multiset pattern MP with the proviso that $Var(\llbracket SP \rrbracket_X)$ is understood as $Var(SP) \cup \{\tilde{x}_i \mid i \in \mathbb{N}\}$. For example, $Var(a \cdot \tilde{x} \mid y \mid \llbracket \varepsilon \rrbracket_Y) = \{\tilde{x}, y\} \cup \{\tilde{y}_i \mid i \in \mathbb{N}\}$.

2. Let $Symbols(U)$ denote the set of elements of \mathcal{E} that appear in the multiset pattern or multiset U . For instance, $Symbols(a \cdot x \mid a \cdot b \mid \llbracket d \rrbracket_X) = \{a, b, d\}$. We assume that $Symbols$ extends to a set of multiset patterns or multisets and we define the set of fresh names for a multiset M as $\mathcal{E} \setminus Symbols(M)$.
3. A multiset pattern MP is ground if and only if $Var(MP) = \emptyset$. A rule $R = (MP, MP')$ is ground if and only if both MP and MP' are ground. We write $Var(R) = Var(MP) \cup Var(MP')$ and $Symbols(R) = Symbols(MP) \cup Symbols(MP')$.
4. Let $R = (MP, MP')$. Then, $Fv(R) = \{v \mid v \in Var(MP') \setminus Var(MP)\}$ and $Bv(R) = Var(R) \setminus Fv(R)$.

Now, we define the rewrite rules.

Definition 9 Let $R = (MP_1, MP_2) \in \mathbf{R}$. An application of R can be described as follows. Assume M, M' are multisets such that $\langle MP_1 \rangle_\rho^\sigma \equiv M$ and $\langle MP_2 \rangle_\rho^\sigma \equiv M'$ for some ρ and instantiation σ .

1. Assume $Symbols(\sigma(Bv(R)) \cup Symbols(R)) \cap Symbols(\sigma(Fv(R))) = \emptyset$. Then, we write

$$\frac{(\{\sigma(SP) | \{\!\!\{SP\}\!\!\}_X^\sigma \in M, Symbols(\sigma(Fv(R)))\})}{\longrightarrow} M'. \tag{1}$$

2. If $M \xrightarrow{(\xi, \zeta)} M'$ and $(\forall S \in \xi)(\nexists S' \in \mathbf{S})(S \cdot S') \in M''$ and $Symbols(M'') \cap \zeta = \emptyset$, then

$$M'' \mid M \xrightarrow{(\xi, \zeta)} M'' \mid M'. \tag{2}$$

The first rule expresses the fact that each submultiset of M with a common prefix must be taken into account in the course of the reduction. The second rule ensures that only maximal submultisets with a common prefix can be exchanged.

The next example illustrates the definition.

Example 10 Let $M \equiv b \mid b \cdot c \mid b \cdot d$. Let $R_1 = (\{\!\!\{b\}\!\!\}_X, a)$ and $R_2 = (b \cdot \tilde{x} \mid \{\!\!\{b\}\!\!\}_X, a)$ be two rules. Then,

1. $M \xrightarrow{(\xi_1, \zeta_1)} a$ is a correct application of R_1 with $\rho_1(X) = 3$ and $\sigma_1(\tilde{x}_1) = \epsilon, \sigma_1(\tilde{x}_2) = c$ and $\sigma_1(\tilde{x}_3) = d$. Regarding the label: $(\xi_1, \zeta_1) = (\{b\}, \emptyset)$.
2. Furthermore, $M \xrightarrow{(\xi_2, \zeta_2)} a$ with $\sigma_2(\tilde{x}) = \epsilon, \rho_2(X) = 2$ and $\sigma_2(\tilde{x}_1) = c, \sigma_2(\tilde{x}_2) = d$ is a correct derivation applying R_2 . Regarding the label: $(\xi_2, \zeta_2) = (\{b\}, \emptyset)$.
3. On the contrary,

$$b \mid b \cdot c \mid b \cdot d \rightarrow_{R_1} a \mid b \cdot d, \quad b \mid b \cdot c \mid b \cdot d \rightarrow_{R_2} b \mid a$$

would be incorrect derivations, since they violate the stipulation that $\{\!\!\{b\}\!\!\}_X^\sigma$ should be the maximal subterm of M the strings of which begin with the prefix b .

In general, rewriting affects the maximal submultisets beginning with the common prefix $\sigma(SP)$ for some SP . This ensures that in encodings, where submultisets correspond to subterms, multisets denoting subterms are replaced as a whole. The next subsection clarifies this idea.

3 Encoding of the λ -calculus in SMSR

In this section, we present an encoding of the λ -calculus in SMSR. The main idea behind the encoding is to make use of the tree-like representation of λ -terms. We create an SMSR string for every branch of the syntax tree of a λ -term. Leaf elements of strings represent variables or constants in the original λ -terms. We lean on the fact that in the SMSR calculus we can identify submultisets of strings by the pattern expansion function. The next function inserts a string in front of each element of a string multiset.

Definition 11 The injection function $\ggg : \mathbf{S} \times \mathbf{M} \rightarrow \mathbf{M}$ is defined recursively as follows:

1. $\xi_1 \ggg \xi_2 = \xi_1 \cdot \xi_2$

$$2. \xi \gg (M_1 \mid M_2) = (\xi \gg M_1) \mid (\xi \gg M_2)$$

Observe that \gg extends the \cdot operator on strings. In the sequel, by abusing notation, we write $\xi \cdot M$ instead of $\xi \gg M$ even when M is a multiset consisting of more than one string. We assume that \gg associates to the right.

Using the injection function, we can now easily construct SMSR strings representing paths in the syntactic tree. Prior to this, following [5], we provide the inevitable notions for formulating λ -calculus terms. Given a set of variables \mathcal{V}_λ , a λ -term is defined by the following Backus–Naur form:

$$\Lambda = \mathcal{V}_\lambda \mid (\lambda \mathcal{V}_\lambda \Lambda) \mid (\Lambda \Lambda).$$

We may agree on the usual ways of applying parentheses in the λ -calculus: Abstraction associates to the right, while application associates to the left. The dot notation indicates the weakest operation in the subterm surrounded by parentheses. The outermost parentheses can be omitted. Hence, $\lambda x \lambda y . xy \equiv (\lambda x (\lambda y (x y)))$, or $\lambda x . xy (\lambda z z) \equiv (\lambda x ((x y) (\lambda z z)))$. With this in hand, we can now encode terms by a multiset of strings as the next definition shows.

Definition 12 Let \mathcal{V}_λ be the set of λ -variables. Let $\mathcal{E}_\lambda = \mathcal{V}_\lambda \cup \{l, r\} \cup \{\lambda x \mid x \in \mathcal{V}_\lambda\}$. The encoding function $\lfloor _ \rfloor : \Lambda \rightarrow \mathbf{M}(\mathcal{E}_\lambda)$ is defined recursively as follows:

1. $\lfloor M N \rfloor = (l \cdot \lfloor M \rfloor \mid r \cdot \lfloor N \rfloor)$
2. $\lfloor \lambda x . M \rfloor = \lambda x \cdot \lfloor M \rfloor$
3. $\lfloor x \rfloor = x$

In what follows, we mean by strings the elements of the set $\mathbf{S}(\mathcal{E}_\lambda)$ and by multisets the elements of the set $\mathbf{M}(\mathcal{E}_\lambda)$, respectively. Since we identify congruent multisets according to Definition 2, several possible representations are equivalent for the same λ -term M . Definition 12, however, prescribes a certain order of the strings of a multiset. We call the multiset $\lfloor M \rfloor$ obtained by Definition 12 the canonical representation of the λ -term M or, simply, the representation of M . Moreover, if N is a multiset obtained from M by means of this encoding, then N is said to be a λ -multiset.

Example 13 As a short example, we present the encoding of the well-known term $(\lambda x . xx)(\lambda x . xx)$:

$$\lfloor (\lambda x . xx)(\lambda x . xx) \rfloor \equiv l \cdot \lambda x \cdot l \cdot x \mid l \cdot \lambda x \cdot r \cdot x \mid r \cdot \lambda x \cdot l \cdot x \mid r \cdot \lambda x \cdot r \cdot x$$

3.1 β -reduction

After establishing the encoding of λ -terms, we proceed with presenting the multiset rewrite rule corresponding to β -reduction in λ -calculus. Prior to this, we present the β -reduction in the λ -calculus. A Λ -term of the form $(\lambda U) V$ is called a β -redex. A β -reduction is the smallest binary relation over Λ compatible with the term formation rules and containing

$$(\lambda U) V \rightarrow_\beta U[x := V],$$

where $U[x := V]$ is obtained from U by obtaining every free occurrence of x in U by V . In the term $\lambda x . U$, the occurrences of the variable x in U are considered to be bound. If an occurrence of x is free in U , then we say that the abstraction λx binds x in $\lambda x . U$. We adopt Barendregt’s notational conventions here: Terms differing in proper renaming of bound

variables are considered equal, and if M_1, \dots, M_n occur in a certain mathematical context (e.g., definition, proof), then in these terms all bound variables are chosen to be different from the free variables. With all that being said, we can formulate b -reduction on multiset patterns, which corresponds to β -reduction.

Definition 14 (*b-rule, b_0 -rule*)

1. In order to define a reduction on multisets, we introduce the b -rule, which is a relation over multiset patterns.

$$(\langle \{l \cdot \lambda x\}_X \mid \{r\}_Y \rangle, (\{\varepsilon\}_X \triangleright_x \{\varepsilon\}_Y)) \quad (b\text{-rule})$$

We call the multiset pattern $(\{l \cdot \lambda x\}_X \mid \{r\}_Y)$ a b -redex pattern, and the multiset pattern $(\{\varepsilon\}_X \triangleright_x \{\varepsilon\}_Y)$ a b -contractum pattern. Note that $x \in \mathcal{E}_\lambda$ by Definition 12.

2. We call a given M a b_0 -redex, and M' its b_0 -contractum if $M, M' \in \mathbf{M}(\mathcal{E}_\lambda)$ is such that (M, M') can be matched with the b -rule, that is, $M \equiv \langle \{l \cdot \lambda x\}_X \mid \{r\}_Y \rangle_\rho^\sigma$ and $M' \equiv (\{\varepsilon\}_X \triangleright_x \{\varepsilon\}_Y)_\rho^\sigma$. In notation: $M \hookrightarrow_{b_0} M'$.

As usual, we define b -reduction as the smallest relation over the set of multisets $\mathbf{M}(\mathcal{E}_\lambda)$ that contains \hookrightarrow_{b_0} and is compatible with the term formation rules. In notation: $M \hookrightarrow_b M'$ if M b -reduces to M' .

Definition 15 (*b-reduction*) The b_0 -rule can be extended to \hookrightarrow_b -reduction over multisets in $\mathbf{M}(\mathcal{E}_\lambda)$. Let U, V and $W \in \mathbf{M}(\mathcal{E}_\lambda)$.

1. If $U \hookrightarrow_{b_0} V$, then $U \hookrightarrow_b V$.
2. If $U \hookrightarrow_b V$, then $\lambda x \cdot U \hookrightarrow_b \lambda x \cdot V$.
3. If $U \hookrightarrow_b V$, then $l \cdot U \mid r \cdot W \hookrightarrow_b l \cdot V \mid r \cdot W$ and $l \cdot W \mid r \cdot U \hookrightarrow_b l \cdot W \mid r \cdot V$.

We can observe that this is equivalent to the following approach: let U be a submultiset of M such that U is a b_0 -redex. Then, $M \hookrightarrow_b M'$ if either U is M and M' is the b_0 -contractum of M , or $\exists \xi$ such that $M \equiv \xi \cdot U$ and $M' \equiv \xi \cdot U'$ and U' is the b_0 -contractum of U , or $M \equiv N \mid V$ and $V \equiv \xi \cdot U$ and $M' \equiv \xi \cdot U'$ and U' is the b_0 -contractum of U and none of the strings in N begins with ξ . The latter part follows from Rule (2) of Definition 9, since it identifies redexes as special from multisets with maximal common prefixes. The key idea behind the rewrite rule is as follows. Given a λ -term M , assume $(\lambda x U) V$ is a β -redex in M . Then, the canonical encoding $\llbracket (\lambda x U) V \rrbracket$ is of the form $l \cdot (\lambda x \cdot \llbracket U \rrbracket) \mid r \cdot \llbracket V \rrbracket$. In this case, we are able to find ρ and σ such that $\langle \{l \cdot \lambda x\}_X \rangle_\rho^\sigma$ evaluates to $\llbracket \lambda x.U \rrbracket$ and $\langle \{r\}_Y \rangle_\rho^\sigma$ evaluates to $\llbracket V \rrbracket$. But then $(\{\varepsilon\}_X \triangleright_x \{\varepsilon\}_Y)$ correctly encodes the contractum $U[x := V]$. This is stated by Theorem 24 in the next subsection.

We term the calculus obtained as above the λ -SMSR calculus. In what follows, when we talk about a String MultiSet Rewriting calculus, we refer to the λ -SMSR calculus; hence, we may omit the prefix λ .

We recall that given $M \in \mathbf{M}$, the multiset patterns $(\{l \cdot \lambda x\}_X \mid \{r\}_Y)$ are matched against the largest possible submultisets of M with a certain common prefix. If M is the b_0 -redex itself, then Rule (2) of Definition 9 is trivially valid. Then, the values for $\rho(X)$, $\rho(Y)$ are chosen according to the number of strings of the corresponding submultisets representing U and V in $\llbracket (\lambda x U) V \rrbracket$. This is illustrated in the next example.

Example 16 Previously, in Example 13, we showed how the term $(\lambda x.xx)(\lambda x.xx)$ can be encoded. Here, we demonstrate the application of the rewrite rule:

$$\begin{aligned}
 & l \cdot \lambda x \cdot l \cdot x \mid l \cdot \lambda x \cdot r \cdot x \mid r \cdot \lambda x \cdot l \cdot x \mid r \cdot \lambda x \cdot r \cdot x \\
 & \equiv \langle (\|l \cdot \lambda x\|_X \triangleright_x \|r\|_Y) \rangle_\rho^\sigma \\
 & \hookrightarrow_b \langle (\|\varepsilon\|_X \triangleright_x \|\varepsilon\|_Y) \rangle_\rho^\sigma \\
 & \equiv l \cdot \lambda x \cdot l \cdot x \mid r \cdot \lambda x \cdot l \cdot x \mid l \cdot \lambda x \cdot r \cdot x \mid r \cdot \lambda x \cdot r \cdot x,
 \end{aligned}$$

where $\rho(X) = \rho(Y) = 2$ and $\sigma(\tilde{x}_1) = l \cdot x$, $\sigma(\tilde{x}_2) = r \cdot x$ and $\sigma(\tilde{y}_1) = \lambda x \cdot l \cdot x$ and $\sigma(\tilde{y}_2) = \lambda x \cdot r \cdot x$. Of course, the above rewriting can be continued, because of the non-normalizing property of $(\lambda x.x x)(\lambda x.x x)$.

A few remarks on the correct execution of \hookrightarrow_b -reduction are in order here. To preserve the correctness of the encoding between the λ -calculus and the String MultiSet Rewriting system obtained above, we must keep in mind that in the λ -calculus bound and free variable occurrences are distinguished and λ -terms are defined in a way such that the problem of variable collision should be avoided. In order to acquire the same ability for the λ -SMSR calculus, a solution can be to introduce the analogue of Barendregt’s variable conventions for λ -SMSR. This means that we identify multisets that are images of λ -terms differing in the correct renaming of bound variables by extending Definition 2 to these cases. Hence, substitution as defined in Definition 4 is understood via “renaming of bound variables”; therefore, collision of variables does not occur. We omit the technical details, however. In what follows, the notation IH stands for the phrase “the induction hypothesis.”

3.2 Correctness of the encoding

Before stating the theorem on the correctness of the encoding, we introduce some necessary notation and lemmas. We first show that b -redexes in multisets correspond to β -redexes in λ -terms. Prior to this, we examine the relation between λ -terms and their representing multisets. From now on, we mean by multisets the elements of the set $\mathbf{M}(\mathcal{E}_\lambda)$. We write \mathbf{M} instead of $\mathbf{M}(\mathcal{E}_\lambda)$ and \mathbf{S} instead of $\mathbf{S}(\mathcal{E}_\lambda)$.

Definition 17 (*Maximal submultisets, multiset prefixes*)

1. Let $\xi, \zeta \in \mathbf{S}$. Then, $\xi < \zeta$ if $\exists v$ such that $\zeta = \xi \cdot v$ and $v \neq \varepsilon$.
2. Let M, N be multisets. Then, $\xi \in \mathbf{S}$ is a multiset prefix of N in M , or, in short, a multiset prefix of N , if, for every $\chi \in N$, $\xi \cdot \chi \in M$. In other words, $(\xi \cdot N) \sqsubseteq M$. in notation: $N \sqsubseteq_\xi M$. Let $\text{pref}(M)$ denote the set of multiset prefixes for M , that is, $\text{pref}(M) = \{\xi \mid \exists N : N \sqsubseteq_\xi M\}$.
3. Let M, U be multisets. Let $\xi \in \mathbf{S}$ and assume ξ is a prefix for U . We call U maximal in M with respect to ξ , or, in short, maximal, if $\nexists \zeta$ for which $(\zeta \mid \xi \cdot U) \sqsubseteq M$ and $\xi < \zeta$. In notation: $U \not\sqsubseteq_\xi M$. We may omit ξ if it is clear from the context. If we want to emphasize that ξ is the multiset prefix of the maximal multiset U , we write $\xi = \text{pref}_M(U)$. We may omit the subscript M . We collect the maximal multisets U of M with respect to some ξ by writing $U \in \text{mm}(M)$. Note that if $U \in \text{mm}(M)$, U is not a submultiset of M (except for $U \equiv M$); however, there exists ξ such that $(\xi \cdot U) \sqsubseteq M$.

Note that, given $N \sqsubseteq_\xi M$, the multiset prefix ξ is unique if it exists. On the other hand, if $N \sqsubseteq_\xi M$ such that N is maximal, then N is the unique such multiset. The following example illustrates the above notions.

Example 18 Let $M \equiv ((\lambda x.x y x) y)$. Consider the corresponding multiset $U \equiv [M] \equiv (l \cdot \lambda x \cdot l \cdot l \cdot x \mid l \cdot \lambda x \cdot l \cdot r \cdot y \mid l \cdot \lambda x \cdot l \cdot x \mid r \cdot y)$.

1. The multiset $(\lambda x \cdot l \cdot l \cdot x \mid \lambda x \cdot l \cdot r \cdot y)$ has l as prefix and the multiset $(l \cdot l \cdot x \mid l \cdot r \cdot y)$ has prefix $l \cdot \lambda x$, respectively.
2. The multiset $V \equiv (l \cdot x \mid x)$ has prefix $l \cdot \lambda x \cdot l$. However, V is not maximal in M with respect to $l \cdot \lambda x \cdot l$. The multiset $(l \cdot x \mid r \cdot y \mid x)$ is maximal in M with respect to $l \cdot \lambda x \cdot l$, and it is the only such multiset.

Intuitively, U is maximal in N , provided N is the image of some λ -term M by the encoding, if and only if U collects all the strings of N starting with a certain common prefix. The next lemma identifies the maximal multisets in $\lfloor M \rfloor$ in terms of the encoding. First of all, we give an informal description of what we mean by subterm occurrences. A subterm occurrence of N in M is a pair $(N, C[\])$ such that $app(C[\], N) = M$, where $C[\]$ a one-holed context and $app(C[\], N)$ yields the result of the substitution of N in $C[\]$. We recall the fairly straightforward definitions.

Definition 19 Let $[\]$ be a given symbol called a *hole* and assume \mathcal{V}_λ are the set of λ -variables.

1. Then, the one-holed λ -contexts are defined by the Backus–Naur form below:

$$C[\] = [\] \mid (\lambda \mathcal{V}_\lambda C[\]) \mid (\Lambda C[\]) \mid (C[\] \Lambda),$$

where Λ denotes the λ -terms.

2. Let $C[\]$ be a one-holed λ -context and $N \in \Lambda$. Then, $app(C[\], N)$ is the result of writing N in place of the hole:

- (a) $app([\], N) = N$,
- (b) $app(\lambda x.C[\], N) = \lambda x.app(C[\], N)$,
- (c) $app((U C[\]), N) = (U app(C[\], N))$,
- (d) $app((C[\] U), N) = (app(C[\], N) U)$, where $U \in \Lambda$.

Furthermore, by abuse of notation, we may ignore references to contexts when we mention subterms. Namely, we may write $N \leq M$ instead of $app(C[\], N) = M$. Furthermore, we assume in what follows that $[\]$ is an element of the alphabet \mathcal{E}_λ underlying the encoding of the λ -calculus. The main advantage of the encoding is that the notion of subterm occurrence can be made precise through it and, hence, we can reason on them in exact terms. As we will see later, we can determine the address of a subterm occurrence explicitly by examining its representing strings, hence identifying the subterm occurrence in a unique way. We denote by $N \leq M$ the fact that N is a subterm (occurrence) of M , where M is a λ -term. Given a finite sequence σ , let the notation $a \in \sigma$ denote the fact that a is one of the elements of σ .

Lemma 20 Let $\xi \in pref(\lfloor M \rfloor)$, where $M \in \Lambda$. Then, if $\xi' = (\xi \cdot e) \in pref(\lfloor M \rfloor)$, for $e \in \mathcal{E}$, then either $e \in \{l, r\}$ or $e = \lambda x$ or $e = x$ for some $x \in \mathcal{V}_\lambda$ and these possibilities are mutually exclusive.

Proof Straightforward. □

Lemma 21 Let M be a λ -term. Then, U is a maximal multiset of $\lfloor M \rfloor$ iff there exists a subterm N of M such that $U \equiv \lfloor N \rfloor$. Moreover, $app(C[\], N) = M$ iff $U \preceq_\xi \lfloor M \rfloor$ such that $[\] \preceq_\xi C[\]$.

Proof (\Rightarrow) Let M be a λ -term, assume U is a maximal multiset of $\lfloor M \rfloor$. Then, $U \preceq_\xi M$ for some ξ . We argue by induction on ξ .

- $|\xi| = 0$. Then, $U \equiv M$ and the statement is obvious.

- $\xi = \xi' \cdot \lambda x$ for some ξ' and λx . Let $U' = mm(M, \xi')$. By IH, $\exists V' \leq M$ such that $U' \equiv \lfloor V' \rfloor$. By Lemma 20, $V' \equiv \lambda x \cdot V$. Regarding the maximality, let $\xi' \cdot \lambda x < \zeta$ for some ζ . This means $\xi' < \zeta$, and, by the maximality of $\lfloor V' \rfloor$, $\zeta \in \xi' \cdot \lfloor V' \rfloor$. Since $\lfloor V' \rfloor \equiv \lambda x \cdot \lfloor V \rfloor$, we have $\zeta \in \xi \cdot \lfloor V \rfloor$. Moreover, let $app(C'[\], V') = M$ and let $U' \preceq_{\xi}^{\prime} \lfloor M \rfloor$. Then, $app(C[\], V) = M$, where $C[\]$ is obtained from $C'[\]$ when we exchange $\lfloor \]$ is $C'[\]$ for $\lambda x \cdot \lfloor \]$.
- $\xi = \xi' \cdot l$ or $\xi = \xi' \cdot r$. Analogous to the above case.

(\Leftarrow) Let $N \leq M$. Assume $app(C[\], N) = M$. We argue by induction on $C[\]$.

- $C[\] = \lfloor \]$. Then, $N \equiv M$ and the statement is clear.
- $C[\] = (U C_1[\])$. Then, $M = (U app(C_1[\], N))$. By IH, $U \equiv \lfloor N \rfloor$ is a maximal multiset of $\lfloor app(C_1[\], N) \rfloor$. Let ξ be its multiset prefix. Then, $r \cdot \xi \cdot \lfloor app(C_1[\], N) \rfloor \sqsubseteq M$. Moreover, let $r \cdot \xi < \zeta$. Then, $\zeta = r \cdot \zeta'$. By the maximality of U in $\lfloor app(C_1[\], N) \rfloor$, $\zeta' \in \xi \cdot U$, which means, $\zeta \in r \cdot \xi \cdot U$. Hence, $U \in mm(M)$. The remaining cases are similar.

□

We observe that since a b -redex is a maximal submultiset of $\lfloor M \rfloor$ for any M , it corresponds to a subterm of M . Moreover, the above lemma shows that a b -redex must be the image of a β -redex by the encoding. Hence, b -redexes are the counterparts of β -redexes. The following lemma is a preparation for Lemma 23.

Lemma 22 *Let $M, N \in \Lambda$. Then, $\lfloor M \rfloor[x \leftarrow \lfloor N \rfloor] \equiv \lfloor M[x := N] \rfloor$.*

Proof By induction on M . □

The next lemma states that if we start from a b -redex, the result of the reduction is the encoding of the contractum of the corresponding β -redex.

Lemma 23 *Let $M = (\lambda x.M_1) M_2$, assume $\lfloor M \rfloor \hookrightarrow_b U$ such that $\lfloor M \rfloor$ is the redex reduced. Then, $U \equiv \lfloor M_1[x := M_2] \rfloor$.*

Proof Assume $M = (\lambda x.M_1) M_2$ and $\lfloor M \rfloor \hookrightarrow_b^{\lfloor M \rfloor} U$ as above. Since $\lfloor M \rfloor = l \cdot \lambda x \cdot \lfloor M_1 \rfloor \mid r \cdot \lfloor M_2 \rfloor$, by the formulation of b -reduction, we must have $\langle \{\varepsilon\}_x \rangle_{\rho}^{\sigma} \equiv \lfloor M_1 \rfloor$ and $\langle \{\varepsilon\}_y \rangle_{\rho}^{\sigma} \equiv \lfloor M_2 \rfloor$. We argue by induction on M_1 .

- $M_1 \in \mathcal{V}_{\lambda}$. If $M_1 = x$, then, by Definition 4, $\lfloor M_1 \rfloor[x \leftarrow \lfloor M_2 \rfloor] \equiv \lfloor M_2 \rfloor \equiv \lfloor M_1[x := M_2] \rfloor$. When $M_1 = y$, then $\lfloor M_1[x := M_2] \rfloor \equiv y \equiv \lfloor y \rfloor$.
- $M_1 = \lambda y.M_3$. Then, $\lfloor M_1 \rfloor \equiv \lambda y \cdot \lfloor M_3 \rfloor$ and, straightforwardly, $(\lambda y \cdot \lfloor M_3 \rfloor)[x \leftarrow \lfloor M_2 \rfloor] \equiv \lambda y \cdot (\lfloor M_3 \rfloor[x \leftarrow \lfloor M_2 \rfloor])$. Applying Lemma 22 ($\lfloor M_3 \rfloor[x \leftarrow \lfloor M_2 \rfloor]$), we obtain the result.
- $M_1 = (M_3 M_4)$. In this case, $\lfloor M_1 \rfloor \equiv l \cdot \lfloor M_3 \rfloor \mid r \cdot \lfloor M_4 \rfloor$ and $(l \cdot \lfloor M_3 \rfloor \mid r \cdot \lfloor M_4 \rfloor)[x \leftarrow \lfloor M_2 \rfloor] \equiv l \cdot \lfloor M_3 \rfloor[x \leftarrow \lfloor M_2 \rfloor] \mid r \cdot \lfloor M_4 \rfloor[x \leftarrow \lfloor M_2 \rfloor]$. The statement follows from Lemma 22 and the fact $(M_3 M_4)[x := M_2] = (M_3[x := M_2] M_4[x := M_2])$.

□

Now, we are in a position to prove the correctness theorem.

Theorem 24 *The encoding correctly simulates β -reduction in the λ -calculus. That is, the following assertions are valid.*

1. Let $M \rightarrow_{\beta} M'$. Then, $\llbracket M \rrbracket \hookrightarrow_b \llbracket M' \rrbracket$.
2. Let $\llbracket M \rrbracket \hookrightarrow_b U$. Then, there exists M' such that $U \equiv \llbracket M' \rrbracket$ and $M \rightarrow_{\beta} M'$.

Proof 1. We argue by induction on M .

- $M = \lambda x.M_1$. Then $M = \lambda x.M_1 \rightarrow_{\beta} \lambda x.M'_1 = M'$ and $\llbracket M \rrbracket \equiv \lambda x \cdot \llbracket M_1 \rrbracket$. By the induction hypothesis, $\llbracket M_1 \rrbracket \hookrightarrow_b \llbracket M'_1 \rrbracket$. Then, since \hookrightarrow_b is compatible with the term formation rules, we obtain $\llbracket M \rrbracket \equiv \lambda x \cdot \llbracket M_1 \rrbracket \hookrightarrow_b \lambda x \cdot \llbracket M'_1 \rrbracket \equiv \llbracket M' \rrbracket$.
- $M = (M_1 M_2)$. Assume $M_1 \rightarrow_{\beta} M'_1$. The case for $M_2 \rightarrow_{\beta} M'_2$ being similar. Then we apply IH to M_1 to obtain $\llbracket M_1 \rrbracket \hookrightarrow_b \llbracket M'_1 \rrbracket$. The result follows from the compatibility of \hookrightarrow_b with the term formation rules. The case $M = (\lambda x.M_1) M_2 \rightarrow_{\beta} M' = M_1[x := M_2]$ is treated by Lemma 23.

2. Let $\llbracket M \rrbracket \hookrightarrow_b U$. We proceed by induction on M .

- $M = \lambda x.M_1$. Then, $\llbracket M \rrbracket \equiv \lambda x \cdot \llbracket M_1 \rrbracket$. Definitions 14 and 15 imply that $\llbracket M_1 \rrbracket \hookrightarrow_b V$ such that $U \equiv \lambda x \cdot V$. Then, by IH, we obtain $M_1 \rightarrow_{\beta} M'_1$ with $V \equiv \llbracket M'_1 \rrbracket$. Then, clearly, $M = \lambda x.M_1 \rightarrow_{\beta} \lambda x.M'_1 = M'$ and $U \equiv \llbracket M' \rrbracket$.
- $M = (M_1 M_2)$. Then $\llbracket M \rrbracket \equiv l \cdot \llbracket M_1 \rrbracket \mid r \cdot \llbracket M_2 \rrbracket$. Let $\llbracket M_1 \rrbracket \hookrightarrow_b U'$, the case $M_2 \hookrightarrow_b U''$ being similar. Then, by IH, $\exists M'_1$ such that $M_1 \rightarrow_{\beta} M'_1$ and $U' \equiv \llbracket M'_1 \rrbracket$. Hence, $M \rightarrow_{\beta} (M'_1 M_2)$ and $U \equiv \llbracket (M'_1 M_2) \rrbracket$. In the case when $\llbracket M \rrbracket \hookrightarrow_b M'$ and $\llbracket M \rrbracket$ is the redex itself, we necessarily have $\llbracket M \rrbracket \equiv \langle \langle \{l \cdot \lambda x\}_X \mid \{r\}_Y \rangle \rangle_{\rho}^{\sigma}$ with some ρ and σ . Hence, $\llbracket M \rrbracket \equiv (l \cdot \lambda x) \cdot U_1 \mid r \cdot U_2$ for some multisets U_1 and U_2 . Lemma 21 provides M_1 and M_2 such that $\llbracket M_i \rrbracket \equiv U_i$ ($i \in \{1, 2\}$). Then, Lemma 23 yields $U \equiv \llbracket M_1[x := M_2] \rrbracket$.

□

4 Applications: proofs of the standardization and finiteness of developments theorems

We demonstrate the applicability and flexibility of the new tool obtained by presenting proofs for two fundamental theorems. We prove the standardization and the finiteness of developments theorems by means of the encoding. The main advantage of the encoding that, by this, we obtain a tool for identifying subterm occurrences in λ -terms, more precisely, multiset occurrences in the corresponding multisets. The multiset prefixes not only provide an index pointing to that occurrence, but we are also able to manipulate them with the help of the underlying calculus. Firstly, we formulate the necessary notions for the λ -calculus, then we discuss how these notions transfer to string multisets. We present the usual, more or less, intuitive definitions concerning residuals and developments. Then we turn to the proof of the standardization theorem. The most widespread method for standardization is that introduced in [5]. Several authors have produced somewhat similar variants of Barendregt's method [1, 13]. In [1], Amadio and Curien define subterm occurrence by giving a sequence of indexes leading to the underlying subterm in a way very similar to that applied here. Their construction, however, does not expand to a full calculus, they make use of it to give the precise interpretation of subterm occurrences. In Sects. 2.1 and 2.2 they present, among the others, a proof of the standardization theorem. The proof when M is strongly normalizing, is relatively easy, on the contrary, the general case is obtained by introducing labeled lambda terms, which makes their treatment a little more involved than that in [5]. In this paper, we chose a method

presented in [12]. The present proof is facilitated by Proposition 37 of this section, which gives a characterization of standard reduction sequences.

Next, we briefly recall the notions of residuals and developments in such a form that is customary in the λ -calculus. If s is an arbitrary sequence of some elements, other than a sequence in the SMSR calculus, a is an element from the same set, we denote by $a :: s$ the concatenation of the one element sequence (a) and s .

Definition 25 (Residuals in λ -calculus) Let M be a term, $R = (\lambda x.U) V$ be a redex in M . Let $P \leq M$. The residual set $P[R]$ of P with respect to R (or $P[R]_M$ if we intend to indicate M) is defined by the following cases.

1.
 - $P \leq U$: $P[R] = \{P[x := V]\}$.
 - $P \leq V$: $P' \in P[R]$ iff P' is an occurrence P in one of the substituted instances of V .
 - $R \leq P$: $P[R] = \{P'\}$, where P' is obtained from P by replacing $(\lambda x.U) V$ with $U[x := V]$.
 - P and R are disjoint: $P[R] = \{P\}$.

In all the other cases P has no residuals.

2. Let $\sigma : M \rightarrow M_1 \rightarrow \dots \rightarrow M_n$ be a reduction sequence starting from M . Let $P \leq M$. Then, the residuals of P with respect to σ are defined by induction on $|\sigma|$ and are denoted by $P[\sigma]$ or by $P[\sigma]_M$ if we intend to emphasize that the original term is M . Assume we have already determined the set $P[\sigma']$ for each $|\sigma'| < |\sigma|$.

- $|\sigma| = 0$: every term is a residual of itself.
- $\sigma = \sigma' :: R$: then $P'' \in P[\sigma]$ if, for some $P' \in P[\sigma']$, P'' is a residual of P' with respect to R . That is, $P'' \in P'[R]_{M_{n-1}}$.

We can define the residuals of a set of subterms \mathbf{N} with respect to σ in an analogous way, which is denoted by $\mathbf{N}[\sigma]$.

3. Let \mathbf{R} be a set of redexes. A development is a reduction sequence $[R_1, \dots, R_n]$, where $R_i \in \mathbf{R}[R_1, \dots, R_{i-1}]$. A development is complete if there are no more residuals of \mathbf{R} left.

We define standard reduction sequences in the usual manner ([5]).

Definition 26 A β -redex $R_1 = (\lambda x.U_1) U_2$ of M is to the left of $R_2 = (\lambda y.V_1) V_2$ if λx precedes λy when we read M from left to right. In other terms, $R_1 = (\lambda x.U_1) U_2$ being to the left of $R_2 = (\lambda y.V_1) V_2$ this means the fulfillment of one of the following cases:

- There are $(M_1 M_2) \leq M$ such that $R_1 \leq M_1$ and $R_2 \leq M_2$, or
- $R_2 \leq U_1$, or
- $R_2 \leq U_2$.

Definition 27 (Standard reduction sequence) Let $\sigma : M_1 \xrightarrow{R_1} M_2 \xrightarrow{R_2} \dots \xrightarrow{R_n} M_{n+1}$ be a reduction sequence. Then σ is standard if there are no $1 \leq i < j \leq n + 1$ such that R_j is a residual of some R standing to the left of R_i . We use the notation $\sigma \in St$ to claim that σ is a standard reduction sequence.

We might have the impression that the definition of standard reduction sequences is somewhat imprecise and difficult to handle. We intend to remedy this shortcoming by introducing standard reduction sequences in connection with multisets. We present the necessary notions first.

Definition 28 Let M be a λ -multiset, assume $P \sqsubseteq_{\xi} M$, $Q \sqsubseteq_{\zeta} M$ as defined in Definition 17. We write $P \sqsubseteq_M Q$ if $\zeta \leq \xi$. We denote this containment by writing $P \preceq_M Q$ if, in addition, $P, Q \in mm(M)$.

In the sequel, we write $N \preceq M$ if ξ is not known or not interesting to us. We do not omit M from the relation $P \preceq_M Q$. Observe that, if M is such that $M \equiv \lfloor U \rfloor$, and $N \preceq_{\xi} M$ with an appropriate ξ , then, by Lemma 21, there exists $V \leq U$ such that $N \equiv \lfloor V \rfloor$. Hence, multiset occurrences in M match subterm occurrences in U provided $M \equiv \lfloor U \rfloor$. We sum up these properties in the next lemma.

Lemma 29 Let U, V, W be multisets. Assume $U \equiv \lfloor M \rfloor$ for some λ -term M . Then, the following statements hold.

1. Let $V \in mm(U)$. Then, $V \preceq_{\xi} U$ iff $\exists N \leq M$ such that $V \equiv \lfloor N \rfloor$.
2. Let $V, W \in mm(U)$. Then, $V \preceq_U W$ iff $\exists N, P \leq M$ such that $V \equiv \lfloor N \rfloor$ and $W \equiv \lfloor P \rfloor$ and $N \leq P$.

Proof

1. The first part is a simple restatement of Lemma 21.
2. Assume $V, W \in mm(U)$ as above. Let $V \preceq_U W$. Then, there exist ξ, ζ such that $V \preceq_{\xi} U$, $W \preceq_{\zeta} U$ and $\zeta \leq \xi$. Since, in addition, $V, W \in mm(U)$, then, by Lemma 21, we obtain $N, P \leq M$ such that $V \equiv \lfloor N \rfloor$ and $W \equiv \lfloor P \rfloor$. What remains to prove is that $\zeta \leq \xi$ iff $N \leq P$. By Lemma 21, there exist $C'[\], C''[\] : app(C'[\], N) = M$ and $app(C''[\], P) = M$ and $[\] \preceq_{\zeta} C''[\]$ and $[\] \preceq_{\xi} C'[\]$. An easy proof by induction on $|\xi| - |\zeta|$ shows that, from $\zeta \leq \xi$ and Lemma 20, it follows that $\exists C'''[\]$ such that $app(C'''[\], N) = P$, which is the result.

□

Remark 30 Observe that there is a delicate connection between λ -multisets, that is, multisets that are the images of λ -terms, and maximal multisets of λ -multisets. Namely, if U is a maximal multiset of $\lfloor M \rfloor$, that is, $U \preceq_{\xi} \lfloor M \rfloor$ for some ξ , then $U \equiv \lfloor V \rfloor$ such that $V \leq M$ as a λ -term. Moreover, ξ determines the position of $[\]$ in $C[\]$, where $app(C[\], V) = M$. In what follows, if we write “we proceed by induction on P ” and $P \equiv \lfloor M \rfloor$ is a λ -multiset, we understand induction on the number of steps needed to produce P from M by reason of Definition 12.

In order to talk about the relative positions of multisets in a multiset in a more precise way, we define an order on the strings. The next definition extends the relation $<$ on strings.

Definition 31 Let ξ, η be strings. Then, ξ precedes η , in notation $\xi < \eta$ if, either

1. $\xi < \eta$, or
2. $\xi = \nu \cdot l \cdot \xi'$ and $\eta = \nu \cdot r \cdot \eta'$ for some $\nu, \xi', \eta' \in \mathbf{S}$.

We write $\xi \preceq \eta$, if either $\xi < \eta$ or $\xi = \eta$.

Before stating the proposition that expresses standardness in terms of maximal prefixes, we formulate some auxiliary lemmas. The next lemma gives the notion of being to the left in the context of multisets.

Lemma 32 Let $R_1, R_2 \leq M$ be two different redex occurrences of M . Then, R_1 is to the left of R_2 iff $pref(\lfloor R_1 \rfloor) < pref(\lfloor R_2 \rfloor)$ in $\lfloor M \rfloor$.

Proof Let $R_1 = (\lambda x_1.P_1) Q_1$ and $R_2 = (\lambda x_2.P_2) Q_2$.

- (\Rightarrow) Assume R_1 is to the left of R_2 . We take into account the various cases of Definition 26. Applying Lemma 21, it is easy to check that in each of the cases the assertion holds.
- (\Leftarrow) Let $pref(R_i) = \xi_i$ ($i \in \{1, 2\}$). If $\xi_1 < \xi_2$ holds by reason of $\xi_2 = \xi_1 \cdot \nu$, then $\lfloor R_2 \rfloor \leq_M \lfloor R_1 \rfloor$, hence $R_2 \leq R_1$ is valid by Lemma 29. Otherwise, $\xi_1 = \nu \cdot l \cdot \xi'_1$ and $\xi_2 = \nu \cdot r \cdot \xi'_2$. Let $U_1 \leq \lfloor M \rfloor$ such that $pref(U_1) = \nu \cdot l$ and $U_2 \leq \lfloor M \rfloor$ such that $pref(U_2) = \nu \cdot r$. By Lemma 21, there exist N_1, N_2 such that $U_i \equiv \lfloor N_i \rfloor$ ($i \in \{1, 2\}$). By virtue of Lemma 29, we obtain that $R_i \leq N_i$ ($i \in \{1, 2\}$), which means that R_1 is to the left of R_2 . □

We remark that since there is a bijection between λ -terms and λ -multisets, which are the images of λ -terms in the encoding, we do not introduce a different font style for λ -multisets. We indicate it in most of the cases, however, whether we are dealing with λ -terms or λ -multisets.

We can transfer the notions of residuals and standardness to multisets. By the identification of terms with λ -multisets, β -redexes can also be matched to b -redexes and the notions like residuals, being to the left and standardness apply mutatis mutandis. In order to define the property of a b -redex lying to the left of another one, we can resort to Lemma 32. We define residuals only for maximal submultisets of the given multiset. Considering the strong analogy, we apply the same notation to multisets in connection with residuals as those introduced for λ -terms in Definition 25. We repeat the necessary definitions.

Definition 33 (*Residuals in the multiset calculus*) Let U be a λ -multiset, assume $R = (l \cdot \lambda x \cdot S \mid r \cdot T)$ is a b -redex in U . Let $P \leq M$ be a λ -multiset. The b -residual set $P[R]$ of P with respect to R (or $P[R]_U$ if we intend to indicate U) is defined by the following cases.

1.
 - $P \leq S$: $P[R] = \{P[x \leftarrow T]\}$.
 - $P \leq T$: $P' \in P[R]$ iff P' is a multiset occurrence of P in one of the substituted instances of T .
 - $R \leq P$: $P[R] = \{P'\}$, where P' is obtained from P by replacing R with $S[x \leftarrow T]$.
 - P and R are disjoint: $P[R] = \{P\}$.

In all the other cases P has no residuals.

2. Let $\sigma : M \rightarrow M_1 \rightarrow \dots \rightarrow M_n$ be a b -reduction sequence starting from M . Let $P \sqsubseteq M$ be a λ -multiset. Then, the b -residuals of P with respect to σ are defined by induction on $|\sigma|$ and are denoted by $P[\sigma]$ or by $P[\sigma]_M$ if we intend to emphasize that the original term is M . Assume we have already determined the set $P[\sigma']$ for each $|\sigma'| < |\sigma|$.

- $|\sigma| = 0$: every λ -multiset is a b -residual of itself.
- $\sigma = \sigma' \cdot R$: then $P'' \in P[\sigma]$ if, for some $P' \in P[\sigma']$, P'' is a b -residual of P' with respect to R . That is, $P'' \in P'[R]_{M_{n-1}}$.

We can define the b -residuals of a set of submultisets \mathbf{N} with respect to σ in an analogous way, which is denoted by $\mathbf{N}[\sigma]$.

3. Let \mathbf{R} be a set of b -redexes. A b -development is a b -reduction sequence $[R_1, \dots, R_n]$, where $R_i \in \mathbf{R}[R_1, \dots, R_{i-1}]$. A b -development is complete if there are no more b -residuals of \mathbf{R} left.

We omit the prefix “ b ” and write “residual” and “development” when it is clear from the context that those are the multiset residuals and multiset developments that are under discussion.

The next lemma describes the previous notions in terms of maximal prefixes.

Lemma 34 *Let R, M, N be λ -multisets. Assume $R \equiv (l \cdot \lambda x \cdot P \mid r \cdot Q)$, $N \sqsubseteq M$ and assume $M \hookrightarrow_b^R M'$. Suppose N' is a residual of N and let $\text{pref}_M(R) = \eta$, $\text{pref}_M(N) = \xi$ and $\text{pref}_{M'}(N') = \chi$. Then, the following assertions hold.*

1. *Let $\xi = \eta \cdot l \cdot \lambda x \cdot \xi'$ for some ξ' . Then, $\chi = \eta \cdot \xi'$.*
2. *Let $\xi = \eta \cdot r \cdot \xi''$ for some ξ'' . Then, $\chi \in \{\eta \cdot v \cdot \xi'' \mid \text{pref}_R(x) = l \cdot \lambda x \cdot v \text{ for some string } v\}$.*
3. *Let $\xi < \eta$ and ξ be a substring of η . Then, $\chi = \xi$.*
4. *Let $\xi < \eta$ and assume ξ is not a substring of η or $\eta < \xi$ and assume η is not a substring of ξ . Then, $\chi = \xi$.*

Proof We consider the cases of Definition 33 and apply Lemma 29. □

Lemma 35 *Let M be a λ -multiset and assume R is a b -redex. Let $M \hookrightarrow^R M'$ and assume R' is a b -redex such that R' is not a residual of any b -redex in M . Then, $\text{pref}_M(R) \preceq \text{pref}_{M'}(R')$.*

Proof We observe that when we turn to λ -terms instead of multisets, if $P = (\lambda y.U) V$ is a redex in a λ -term N and $N \rightarrow^P N'$, then Q can be a new redex in N' only if it is of the form $Q = (V W[y := V])$. Hence, $(y W)$ is a subterm of U and $V = \lambda z.S$ for some S .

Let M, R be as above. Since M is a λ -multiset, $\exists N$ such that $M \equiv [N]$. Assume $R \equiv [(\lambda y.U) V]$. Let $W \equiv [U[y := V]]$. By the previous discussion we derive that $R' \preceq_{M'} W$, and we know by Lemma 29 that $\text{pref}_{M'}(W) \leq \text{pref}_{M'}(R')$. Moreover, by Lemma 34, $\text{pref}_M(R) = \text{pref}_{M'}(W)$; hence, we can conclude that $\text{pref}_M(R) \leq \text{pref}_{M'}(R')$. □

The next corollary states that the “order” of redexes will not change after a reduction step.

Corollary 36 *Let M be a λ -multiset and $N \preceq M$ be a b -redex.*

1. *Suppose R is a b -redex of M and assume $M \hookrightarrow^R M'$. Suppose N' is a residual of N , R' is the contractum of R and assume N is to the left of R (R is to the left of N , respectively). Then N' is to the left of R' (R' is to the left of N' , respectively).*
2. *Let $\sigma = [R_1, R_2, \dots, R_n]$ and assume $M \equiv M_1 \hookrightarrow^{R_1} M_2 \hookrightarrow^{R_2} \dots \hookrightarrow^{R_n} M_{n+1} \equiv M'$. Suppose N' is a residual of N with respect to σ and assume that, for every $1 \leq i \leq n$, N is to the left of R_i (N is to the right of R_i , resp.). Then, for every $1 \leq i \leq n$, $\text{pref}_{M'}(N') < \text{pref}_M(R_i)$ ($\text{pref}_M(R_i) < \text{pref}_{M'}(N')$, resp.).*

Proof

1. By Lemmas 32 and 34.
2. Follows from Point 1 taking into account Lemma 35. □

The next proposition gives a surprising characterization of standard reduction sequences: It is enough to check whether the maximal prefixes of the actual redexes form a monotone increasing sequence.

Proposition 37 *Let $\sigma : M_1 \hookrightarrow^{R_1} M_2 \hookrightarrow^{R_2} \dots \hookrightarrow^{R_n} M_{n+1}$ be a reduction sequence. Then, σ is standard iff $\text{pref}_{M_1}(R_1) \leq \text{pref}_{M_2}(R_2) \leq \dots \leq \text{pref}_{M_n}(R_n)$.*

Proof Let $\sigma : M_1 \hookrightarrow^{R_1} M_2 \hookrightarrow^{R_2} \dots \hookrightarrow^{R_n} M_{n+1}$ be a reduction sequence. Assume $\xi_i = \text{pref}_{M_i}(R_i)$ ($1 \leq i \leq n$).

(\Rightarrow) Let σ be a standard reduction sequence. Assume $\xi_i = \text{pref}_{M_i}(R_i)$ and $\xi_1 \preceq \xi_2 \preceq \dots \preceq \xi_j \succ \xi_{j+1}$ for some $1 \leq j \leq n - 1$.

- Suppose $R_{j+1} \in R[R_j]$ for some R . Then, Corollary 36 implies that R is to the left of R_j , which contradicts the standardness assumption.
- Suppose R_{j+1} is a newly created redex, that is, R_{j+1} is not a residual of some redex R with respect to R_j . In this case, Lemma 35 involves that $\xi_j \preceq \xi_{j+1}$, contrary to our assumption.

(\Leftarrow) Suppose $\xi_1 \preceq \xi_2 \preceq \dots \preceq \xi_j$ and there are $1 \leq i < j \leq n$ such that R_j is a residual of some R standing to the left of R_i . Then, by Corollary 36, we would necessarily have $\xi_j \prec \xi_i$ contradicting our assumption.

□

We are now in a position to prove the standardization theorem for λ -multisets. Thus, by reason of the encoding, we obtain the standardization theorem for the λ -calculus. Prior to this, however, we have to state some more lemmas. The next lemma prepares the proof of the standardization theorem.

Lemma 38 *Let P, Q be λ -multisets. Then, the following assertions are valid.*

1. If $P \equiv \lambda y \cdot P_1$, then $P[x \leftarrow Q] \equiv \lambda y \cdot P_1[x \leftarrow Q]$.
2. If $P \equiv (P_1 \mid P_2)$, then $P[x \leftarrow Q] \equiv (P_1[x \leftarrow Q] \mid P_2[x \leftarrow Q])$.

Proof The proof proceeds by induction on $lh(P) + lh(Q)$ using the previous lemma. □

The analogue of the next lemma is one of the main tools for demonstrating the standardization theorem in the λ -calculus. The proof exploiting multiset representation might probably seem more precise than the traditional one owing to the characterization of standardness in Proposition 37.

Lemma 39 *Let P, Q be λ -multisets and assume $P \xrightarrow{\sigma}_{st} P'$ and $Q \xrightarrow{\nu}_{st} Q'$. Then, there exists $P[x \leftarrow Q] \xrightarrow{\tau}_{st} P'[x \leftarrow Q']$.*

Proof Let $P \xrightarrow{\sigma}_{st} P', Q \xrightarrow{*}_{st} Q'$. We prove that $P[x \leftarrow Q] \xrightarrow{\xi}_{st} P'[x \leftarrow Q']$ by induction on the lexicographically order pair $\langle \|\sigma\|, lh(P) \rangle$.

- $\|\sigma\| = 0$. We proceed by induction P .
 - P is a variable. If $P \equiv x$, then we have the result by our assumption on Q . Otherwise, the statement is trivial.
 - $P \equiv \lambda y \cdot P_1$. IH applies together with Lemma 38.
 - $P \equiv (P_1 \mid P_2)$. By Lemma 38 and IH, we obtain the result.
- $\|\sigma\| = k > 0$.
 - $P \equiv \lambda y \cdot P_1$. Then $P_1 \xrightarrow{\sigma} P'_1$, and we can use IH.
 - $P \equiv (P_1 \mid P_2)$.
 - * Assume $\exists \sigma_1, \sigma_2$ such that $P_i \xrightarrow{\sigma_i} P'_i$ ($i \in \{1, 2\}$). Then, by IH, $P_1[x \leftarrow Q] \xrightarrow{\tau_1} P'_1[x \leftarrow Q']$ and $P_2[x \leftarrow Q] \xrightarrow{\tau_2}_{st} P'_2[x \leftarrow Q']$. Applying Proposition 37 we obtain that $\tau = \tau_1 \# \tau_2 \in St$.
 - * Let $P \xrightarrow{\sigma_1} (I \cdot \lambda y \cdot U \mid r \cdot V) \xrightarrow{\sigma} U[y \leftarrow V]$ be an initial reduction sequence of σ . Then σ_1 is necessarily empty, otherwise, by Proposition 37, the reduction

sequence σ would not be standard. Hence, σ is $P \equiv (l \cdot \lambda y \cdot U \mid r \cdot V) \hookrightarrow U[y \leftarrow V] \hookrightarrow^{\sigma'} P'$. Let $P'' = U[y \leftarrow V]$. Then, by applying IH, we obtain $P''[x \leftarrow Q] \hookrightarrow_{st}^{\tau'} P'[x \leftarrow Q']$. Since P is the redex itself, applying Proposition 37 again, we obtain that $\tau = [P] :: \tau' \in St$. □

We can now state and prove the standardization theorem for λ -multisets.

Theorem 40 (Standardization) *Let $M \hookrightarrow^* M'$. Then, there exists a standard reduction sequence $M \hookrightarrow_{st}^* M'$.*

Proof Let $\sigma : M_1 \hookrightarrow^{R_1} M_2 \hookrightarrow^{R_2} \dots \hookrightarrow^{R_n} M_{n+1} \equiv M'$, and let us write $pref_{M_i}(R_i) = \xi_i$ ($1 \leq i \leq n$). Suppose $\xi_1 \preceq \dots \preceq \xi_{n-1} \succ \xi_n$. Let i be the first index $1 \leq i \leq n$ such that $R_n \in R[R_i, \dots, R_{n-1}]$, that is, R_n is a residual of some $R \leq M_i$ and, if $pref_{M_i}(R) = \xi$, then $\xi_i < \xi < \xi_{i+1}$. We prove that there exists a standard reduction sequence τ such that $M_1 \hookrightarrow^{R_1} M_2 \hookrightarrow^{R_2} \dots \hookrightarrow^{R_{i-1}} M_i \xrightarrow{R} M' \hookrightarrow^* M_{n+1}$.

Let $R \equiv [(\lambda x.P) Q]$. Then, the last redex R_n of σ is $[(\lambda x.P') Q']$, where, by IH, $P \hookrightarrow_{st}^* P'$ and $Q \hookrightarrow_{st}^* Q'$. Lemma 39 provides us with a $\theta \in St$ such that $P[x \leftarrow Q] \hookrightarrow_{st}^{\theta} P'[x \leftarrow Q']$. Putting together Lemma 34 and Corollary 36, if $R' \in \sigma$ is such that $\xi' = pref(R') < \xi = pref_{M_i}(R)$ or $\xi < \xi'$ and $\xi \not\prec \xi'$, that is, ξ is not a substring of ξ' , then $\xi' \preceq \xi''$ or $\xi'' \preceq \xi'$, resp., for any ξ'' , where ξ'' is multiset the prefix of some $R'' \in \theta$. Hence, the standardness of the new reduction sequence is guaranteed. □

We turn to the proof of the finiteness of developments theorem. Considering its main idea, our proof is similar to the one given in [21]; however, the development separation theorem, which was the main tool in [21], seems to be more intuitive and easier to handle in our formalism. In order to deal with developments, we apply a well-known construction for distinguishing the abstractions that do not count in the development. The definition of redexes remains the same, we define applications though which do not count as redexes. We introduce λ^* -abstractions and stipulate that, as before, only ordinary λ -abstractions can form a redex. Hence, new redexes do not emerge in the reduction sequence. First of all, we introduce some auxiliary notions and prove some lemmas.

Definition 41 (Starred terms)

1. The set of λ^* -terms is defined as follows.

$$T^* = x \mid (\lambda^* x.T^*) \mid ((\lambda x.T^*) T^*) \mid (T^* T^*).$$

The terms of the form $(\lambda x.U) V$ are considered β -redexes, as usual.

In what follows, we understand by a term a starred term, unless otherwise stated.

The multiset encoding of λ^* -terms goes along the same way as that of λ -terms. The new elements of the underlying set \mathcal{E} compared to the λ -terms are the elements λ^*x , where x is an arbitrary λ -variable. As before, we call the multisets forming the image set of λ^* -terms λ^* -multisets. The following theorem is a version of the development separation theorem in [21]. Before that, we formulate a definition that generalizes the $M[x \leftarrow N]$ notation. We recall relation \preceq on strings from a total order as defined in Definition 28. With the help of this, we define a term that is obtained by exchanging the occurrences of x in it with certain multisets. Intuitively, the notation $P[Q_1, \dots, Q_k]_x$ means that P contains k multisets ending with x and all such occurrences of x are replaced by terms Q_1, \dots, Q_k respecting the multiset order. Obviously, if we let $Q_i \equiv Q$ ($1 \leq i \leq k$), then $P[Q_1, \dots, Q_k]_x$ is the same as $P[x \leftarrow Q]$. We repeat this construction in the next definition.

Definition 42 Let P, Q_1, \dots, Q_k be multisets and assume $\xi_1 < \dots < \xi_k$ are all the strings in P which end with the element x . Then, let $P[Q_1, \dots, Q_k]_x$ denote the multiset which is obtained from P when we exchange $\xi_i = \xi'_i \cdot x$ with $\xi'_i \cdot Q_i$ ($1 \leq i \leq k$).

The next lemma is intuitive and straightforward. It helps us with proving the development separation theorem for multisets.

Lemma 43 Let P, Q_1, \dots, Q_k be λ -multisets such that $U \equiv P[Q_1, \dots, Q_k]_x$. Then, either of the following statements holds.

1. $U \equiv \lambda^*x \cdot U_1$, where $U_1 \equiv P_1[Q_1, \dots, Q_k]_x$ for some $P \equiv \lambda^*x \cdot P_1$.
2. $U \equiv (l \cdot U_1 \mid r \cdot U_2)$, where $U_1 \equiv P_1[Q_1, \dots, Q_i]_x$ and $U_2 \equiv P_2[Q_{i+1}, \dots, Q_k]_x$ for some $1 \leq i \leq k$ with $P \equiv (l \cdot P_1 \mid r \cdot P_2)$.

Proof Let $U \equiv P[Q_1, \dots, Q_k]_x$. We proceed by induction on P .

1. $P \equiv \lambda^*x \cdot P_1$. We note that every string in $P \equiv \lambda^*x \cdot P_1$ is of the form $\lambda^*x \cdot \xi_1$, where ξ_1 is a string of P_1 . From this, Point 1 follows.
2. $P \equiv (l \cdot P_1 \mid r \cdot P_2)$. Since P is in the canonical representation, the strings of P are in the order prescribed by $<$. Hence, there is an $1 \leq i \leq k$ such that $\xi_1 < \dots < \xi_i$ are all the strings of P falling in P_1 and ending in x . By this, we obtain Point 2. Since P must take exactly one of these two forms, these cases exhaust the possibilities for U .

□

Now, we can turn to the proof of the multiset counterpart of the development separation theorem.

Theorem 44 Let $R \equiv [(\lambda x.P) Q]$ be a b -redex. Assume $R \hookrightarrow_b^* R'$. Then, $R' \equiv P'[Q_1, \dots, Q_k]_x$ such that $P \hookrightarrow_b^* P'$ and $Q_i \hookrightarrow_b^* Q'_i$ for $1 \leq i \leq k$, where k is the number of occurrences of x in R' .

Proof Let $R \hookrightarrow_b^k R'$. We proceed by induction on k .

- $k = 1$. $R' \equiv P[x \leftarrow Q]$. By this, the assertion is true.
- $k = l + 1$. Assume $R \hookrightarrow_b^l R' \equiv P''[Q''_1, \dots, Q''_k]_x \hookrightarrow_b R'$ with appropriate P'' and Q''_1, \dots, Q''_k . We prove by induction on P'' that, if $P''[Q''_1, \dots, Q''_k]_x \hookrightarrow_b^T S$, then S is of the desired form. From this, we can derive our assertion.
 - $P'' \equiv x$. Then, $Q''_j \hookrightarrow S$ for some $1 \leq j \leq k$. Then, the case $P'' \equiv y$, where $x \neq y$, is impossible.
 - $P'' \equiv \lambda^*y.U$. Then, Definition 15 and Lemma 43 imply that, from $P''[Q''_1, \dots, Q''_k]_x \hookrightarrow_b S$, it follows that $U[Q''_1, \dots, Q''_k]_x \hookrightarrow_b S_1$ for some $S \equiv \lambda^*y.S_1$. Then we use IH.
 - $P'' \equiv (l \cdot U_1 \mid r \cdot U_2)$. Then, by Lemma 43, $P''[Q''_1, \dots, Q''_k]_x \equiv (l \cdot U_1[Q''_1, \dots, Q''_i]_x \mid r \cdot U_2[Q''_{i+1}, \dots, Q''_k]_x)$ for some $1 \leq i \leq k$. If either $U_1[Q''_1, \dots, Q''_i]_x \hookrightarrow_b^T S_1$ or $U_2[Q''_{i+1}, \dots, Q''_k]_x \hookrightarrow_b^T S_2$, where $S \equiv (S_1 \mid S_2)$, then we can apply the IH. Otherwise, $U_1[Q''_1, \dots, Q''_i]_x \equiv \lambda y.U'_1$. By Lemma 43, $U_1 \equiv \lambda y.U'_1 < P''$. Hence, $P''[Q''_1, \dots, Q''_k]_x \hookrightarrow U'_1[Q''_1, \dots, Q''_i]_x[y \leftarrow U_2[Q''_{i+1}, \dots, Q''_k]_x] \equiv U'_1[y \leftarrow U_2][Q''_1, \dots, Q''_i]_x$, where l depends on the number of strings belonging to $U'_1[y \leftarrow U_2]$ ending in x and $Q''_j \in \{Q''_1, \dots, Q''_k\}$ ($1 \leq j \leq l$).

Now, we can demonstrate the finiteness of developments theorem for multisets. □

Theorem 45 Let \mathbf{R} be a set of b -redexes of the λ -multiset M . Assume $M \hookrightarrow_b^\sigma M'$ is a development of \mathbf{R} . Then, the length of σ cannot be arbitrarily large.

Proof We argue by induction on $n = lh(M)$, which is the total number of the elements of \mathcal{E} in M with multiplicities. Let $M \hookrightarrow_b^\sigma M'$, where σ is a development of \mathbf{R} . Let us denote by l_n the maximal number of strings of M' and s_n the maximal length of σ if any one of them is finite. We assume that l_t and s_t are finite for every $1 \leq t < n$. Furthermore, let $|M|$ stand for the number of strings in the multiset M , that is, the cardinality of M .

- $M \equiv \lambda^*y \cdot M_1$. Then $M_1 \hookrightarrow_b^\sigma M'_1$ by Definition 15, hence, IH applies.
- $M \equiv (l \cdot M_1 \mid r \cdot M_2)$. If $M' \equiv (l \cdot M'_1 \mid r \cdot M_2)$ such that $M_1 \hookrightarrow_b^\sigma M'_1$ or $M' \equiv (l \cdot M_1 \mid r \cdot M'_2)$ such that $M_2 \hookrightarrow_b^\sigma M'_2$, respectively, then we can apply IH. Otherwise, $M \equiv (l \cdot \lambda y \cdot M_1 \mid r \cdot M_2)$ and $M \equiv (l \cdot \lambda x \cdot M_1 \mid r \cdot M_2) \hookrightarrow_b^{\sigma_1} (l \cdot \lambda x \cdot M'_1 \mid r \cdot M'_2) \hookrightarrow_b^{\mathcal{R}} M'_1[x \leftarrow M'_2] \hookrightarrow_b^{\sigma_2} M''$ for some σ_1 and σ_2 . By Theorem 44, $M'' \equiv M'_1[M'_2, \dots, M'_2]_x$, where $M'_1 \hookrightarrow_b^* M''_1$ and $M'_2 \hookrightarrow_b^* M''_j$ ($1 \leq j \leq k$). Hence, $|\sigma| \leq s_k + s_m \cdot l_k + 1$, where $lh(M_1) = k$, $lh(M_2) = m$ with $1 \leq k, m < n$ and $|M''| \leq l_k + l_k \cdot l_m$. The right-hand side of both inequalities has a finite maximum; therefore, since M was arbitrary, s_n and l_n are finite. □

Corollary 46 Let \mathbf{R} be a set of redexes in the λ -term M . Let σ be a development of \mathbf{R} . Then, the length of σ cannot be arbitrarily large.

Proof It follows from the previous theorem and Theorem 24. □

Remark 47 Theorem 45 gives us a bound for the length of developments and for the size of the result obtained by completing a development. It is not intended in this paper to try to set up an explicit bound for the lengths of reduction sequences, but it could be performed by utilizing the bound provided by Theorem 45 (cf. [6]).

5 Conclusion

Regarding the calculi developed for the simulation of biological processes, the String MultiSet Rewriting calculus proposed by Barbuti et al. is outstanding in the sense that it possesses the full computational strength of a Turing machine [4] while, on the other hand, it preserves a simple structure owing to which it appears to be quite handy on the side of implementation. In order to demonstrate that computational strength and flexibility, we encoded the λ -calculus in the frames of the SMSR calculus. The encoding allowed us to address each subterm of a λ -term directly, hence, making the proof of such classical theorems precise and, at the same time, intuitive where the argumentation depends on the behavior of particular subterms. We demonstrated this by proving the standardization and finiteness of developments theorems of the λ -calculus. Besides shedding light on crucial points of the demonstrations of these well-known theorems, this method can offer an illustrative tool for formalizing and implementing λ -calculus theorems.

Funding Open access funding provided by University of Debrecen.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence,

and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Amadio, R., Curien, P.: Domains and Lambda-Calculi." Cambridge Tracts in Theoretical Computer Science 46, Cambridge. Cambridge University Press (1998). <https://doi.org/10.1017/CBO9780511983504>
2. Banâtre, J.-P., Fradet, P., Radenac, Y.: Principles of chemical programming. *Electron. Notes Theor. Comput. Sci.* **124**(1), 133–147 (2005). <https://doi.org/10.1016/j.entcs.2004.07.019>
3. Barbuti, R., Maggiolo-Schettini, A., Milazzo, P., Troina, A.: A Calculus of looping sequences for modelling microbiological systems. *Fund. Inform.* **72**, 21–35 (2006)
4. Barbuti, R., Caravagna, G., Maggiolo-Schettini, A., Milazzo, P.: An intermediate language for the simulation of biological systems. *Electron. Notes Theor. Comput. Sci.* **194**, 19–34 (2008)
5. Barendregt, H.P.: "The Lambda Calculus: Its Syntax and Semantics," *Studies in Logic and the Foundations of Mathematics* **103**, North Holland (1985)
6. Battyányi, P., Nour, K.: An estimation for the lengths of reduction sequences of the $\lambda\mu\rho\theta$ -calculus, *Logical Methods in Computer Science* **14**(2) (2018)
7. Boudol, G., Curien, P., Lavatelli, C.: A semantics for lambda calculi with resources. *Math. Struct. Comput. Sci.* **9**(4), 437–482. <https://doi.org/10.1017/S0960129599002893> (1999)
8. Cardelli, L.: Brane Calculi. Interactions of Biological Membranes, In: Danos, V., Schachter, V. (eds.), CMSB'04: Proceedings of the 20 International Conference on Computational Methods in Systems Biology, *Lecture Notes in Computer Science* **3082**, Springer (2005)
9. Cervesato, I.: Petri Nets as Multiset Rewriting Systems in a Linear Framework, Unpublished notes (1994)
10. Ciobanu, G.: Scalable distributed implementation of a biologically inspired parallel model. *Complex Intell. Syst.* **1**(1–4), 69–80 (2015)
11. Danos, V., Laneve, C.: Formal molecular biology. *Theoret. Comput. Sci.* **325**, 69–110 (2004)
12. David, R., Nour, K.: Why the usual candidates of reducibility do not work for the symmetric $\lambda\mu$ -calculus. *Electron. Notes Theor. Comput. Sci.* **140**, 101–111 (2005)
13. Krivine, J.L.: *Lambda-Calculus, Types and Models*. Ellis Horwood (1993)
14. Martinelli, F., Bistarelli, S., Cervesato, I., Lenzini, G., Marangoni, R.: Representing Biological Systems Through Multiset Rewriting, In: Moreno Diaz, R., Pichler, F. (Eds.): *Computer Aided Systems Theory (EUROCAST'03)*, *Lecture Notes in Computer Science* **2809**, Springer (2004)
15. Plotkin, G.D.: A Calculus of Chemical Systems, In: Tannen, V., Wong, L., Libkin, L., Fan, W., Tan, W.C., Fourman, M. (Eds.): *In Search of Elegance in the Theory and Practice of Computation*, *Lecture Notes in Computer Science*, **8000**, Springer (2013)
16. Priami, C.: Stochastic π -calculus. *Comput. J.* **38**(7), 578–589 (1995)
17. Rehof, N.J., Sørensen, M.H. The $\lambda\Delta$ -calculus. In: Hagiya, M., Mitchell, J.C. (eds) *Theoretical Aspects of Computer Software, TACS 1994*. *Lecture Notes in Computer Science*, vol 789. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-57887-0_113 (1994)
18. Troják, M., Šafránek, D., Brim, L.: Biochemical Space Language in Relation to Multiset Rewriting Systems, Technical Report (2022) [arXiv:2201.08817](https://arxiv.org/abs/2201.08817)
19. Troják, M., Samuel, P., Šafránek, D., Brim, L.: Regulated Multiset Rewriting Systems, <https://doi.org/10.48550/arXiv.2111.13036>
20. Troják, M., Šafránek, D., Brim, L., Šnalagovič, J., Červený, J.: Executable biochemical space for specification and analysis of biochemical systems. *Electron. Notes Theor. Comput. Sci.* **350**, 91–116 (2020)
21. Xi, H.: Development separation in lambda-calculus. *Electron. Notes Theor. Comput. Sci.* **143**(6), 207–221 (2006)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.