Mathematical Structures in Computer Science (2021), 1–17
doi:10.1017/S0960129521000256

CAMBRIDGE UNIVERSITY PRESS

PAPER

A quantitative model for simply typed λ-calculus

Martin Hofmann[1,†] and Jérémy Ledent[2*]

[1]Institut für Informatik, Ludwig-Maximilians-Universität München, Germany and [2]Mathematically Structured Programming Group, University of Strathclyde, Glasgow, Scotland
*Corresponding author. Email: jeremy.ledent@strath.ac.uk

(Received 10 August 2020; revised 23 June 2021; accepted 19 August 2021)

Abstract

We use a simplified version of the framework of *resource monoids*, introduced by Dal Lago and Hofmann to interpret simply typed λ-calculus with constants zero and successor. We then use this model to prove a simple quantitative result about bounding the size of the normal form of λ-terms. While the bound itself is already known, this is to our knowledge the first *semantic* proof of this fact. Our use of resource monoids differs from the other instances found in the literature, in that it measures the size of λ-terms rather than time complexity.

Keywords: Resource monoid; length spaces; quantitative semantics; lambda-calculus.

1. Introduction

We start by formulating a simple problem about the computational power of simply typed λ-calculus. The result itself is well known; the aim of this paper will be to provide a *semantic* proof of it, using Dal Lago and Hofmann's framework based on resource monoids.

A simply typed busy beaver. Consider simply typed λ-calculus, equipped with one base type o, and two constants $0 : o$ and $S : o \to o$. Under the standard set-theoretic interpretation of λ-terms (with $[\![o]\!] = \mathbb{N}$, $[\![0]\!] = 0$ and $[\![S]\!] = n \mapsto n + 1$), closed terms of type o correspond to natural numbers. A natural question to ask is how compactly we can write large natural numbers. More precisely, given a closed term $t : o$ of size $|t|$, can we find an upper bound on its denotation $[\![t]\!] \in \mathbb{N}$?

As an example, compare the following three λ-terms of type o. They use (roughly) the same number of characters, but their denotations are increasingly large natural numbers. We write $\bar{2}$ for the church encoding of the number two, $\bar{2} := \lambda f.\, \lambda x.\, f(fx)$.

(a) $t = S(S(S(\ldots(S0))))$ with n occurrences of S.
(b) $u = \bar{2}(\bar{2}(\bar{2}(\ldots(\bar{2}S))))0$ with n occurrences of $\bar{2}$.
(c) $v = (((\bar{2}\bar{2})\bar{2})\ldots\bar{2})S0$ with n occurrences of $\bar{2}$.

[†]Martin Hofmann died on January 23, 2018. This work was carried out in 2016, while the second author was doing a Masters internship supervised by Martin at the Ludwig Maximilian University of Munich. While he could not take part in writing this article, Martin is undoubtedly an author of the work presented here.

Their denotations are, respectively, $[\![t]\!] = n$, $[\![u]\!] = 2^n$, and $[\![v]\!] = 2^{2^{\cdot^{\cdot^{\cdot^2}}}}$, a tower of exponentials of height n. Our goal is to show that one cannot do better than such a tower of exponentials. More precisely, we will prove the following upper bound due to Beckmann (2001):

Claim 1. *For every closed term $t : o$, we have $[\![t]\!] \leq 2^{2^{\cdot^{\cdot^{\cdot^{|t|}}}}}$, where the height of the tower of exponentials is one plus the maximal order of a nonlinear λ-abstraction occurring in t.*

A related question is to ask which set-theoretic functions arise as the denotation of some closed term of type $o \to o$. Intuitively, all these terms can do is add some fixed number of S's to their argument. Thus,

Claim 2. *For every closed term $f : o \to o$, there exists a constant $C \in \mathbb{N}$ such that its denotation $[\![f]\!] : \mathbb{N} \to \mathbb{N}$ is bounded by $n \mapsto n + C$.*

A mild generalization of this result is the case of functions with several arguments. Intuitively, only one of these arguments can be used, and the other ones are discarded:

Claim 3. *For every closed term $f : o^k \to o$, there exists a constant $C \in \mathbb{N}$ such that its denotation $[\![f]\!] : \mathbb{N}^k \to \mathbb{N}$ is bounded by $(n_1, \ldots, n_k) \mapsto max(n_1, \ldots, n_k) + C$.*

Claim 3, and consequently Claim 2 as a special case of it, appear, for example, in Simmons (2005). Once again, we stress that these three results are well known, and not very difficult to prove by syntactic means. The originality of our proofs reside in their semantic nature: we construct a model for λ-calculus where fast-growing functions do not exist.

Resource monoids and length spaces. Resource monoids[1] were introduced by Dal Lago and Hofmann as part of a semantic framework to prove soundness theorems in the field of Implicit Computational Complexity (Dal Lago and Hofmann, 2011). The main idea of this setting is to consider a modification of realizability, where bounded-time algorithms are used as realizers. The bounds are expressed abstractly as elements of an algebraic structure called a *resource monoid*. This allows for more flexibility in the framework. Given a particular resource monoid $\mathcal{M}$, one can define the notion of *length spaces* on $\mathcal{M}$, that is, sets of data that can be bounded by elements of $\mathcal{M}$ in a coherent way. As it turns out, the category of length spaces on any resource monoid $\mathcal{M}$ is symmetric monoidal closed, making it suitable to interpret second-order multiplicative affine logic. This is the common base of this semantic framework, which is independent of the choice of $\mathcal{M}$. To be able to model logical systems with additional features, one must choose an appropriate resource monoid $\mathcal{M}$, so that the associated category of length spaces has the required additional structure.

To our knowledge, all instances of resource monoids previously found in the literature were used to measure the time complexity of programs. Moreover, they all deal with systems based on linear logic, where duplication of data is forbidden or restricted. In this paper, we present a new use-case of resource monoids. We define a particular resource monoid $\mathcal{M}$ whose elements represent bounds the potential of a λ-term to produce large normal forms. The main difficulty to interpret simply typed λ-calculus is then to prove that the category of length spaces over $\mathcal{M}$ actually has duplication morphisms, allowing us to model the contraction rule.

Lastly, we use this model to prove Claims 1 to 3. The interpretation of a λ-term in our model can be computed effectively, by induction on the structure of the term. By doing so, one can get an upper bound on the size of the normal form of a term, without needing to actually β-reduce it. A careful analysis of this process yields the tower of exponentials of Claim 1.

[1]The term "resource monoid" also appears in the context of bunched logics with a different definition and purpose; this is unrelated and should not be confused with the resource monoids that we use here.

https://doi.org/10.1017/S0960129521000256 Published online by Cambridge University Press

Related work. The first upper bound on the number of reduction steps necessary to compute the normal form of a simply typed λ-term was given by Schwichtenberg (1982), who proved that it lies in the class $\mathcal{E}^4$ of the Grzegorczyk hierarchy. Later, Beckmann (2001) improved on the result of Schwichtenberg and gave an upper bound akin to the one in Claim 1. Note that both Schwichtenberg and Beckmann measure the length of a reduction sequence to reach a normal form; while we measure the size of the normal form itself. That explains the additional level that we get in our tower of exponentials. Moreover, our notion of size $|t|$ of a term t (defined formally in Section 5) slightly differs from the one used by Beckmann: we count the number of variables and successors in the term t; while Beckmann's formula counts the variables and λ-abstractions.

The two Claim 2 and Claim 3 should not be confused with other similar results about the so-called λ-*definability* of functions $\mathbb{N} \to \mathbb{N}$. This notion of definability refers to terms of type $\mathsf{Nat}_o \to \mathsf{Nat}_o$, where $\mathsf{Nat}_o := (o \to o) \to (o \to o)$ is the type of Church numerals over some base type o. More precisely, let $\overline{n} \in \mathsf{Nat}_o$ denote the Church encoding of $n \in \mathbb{N}$. Then a function $f : \mathbb{N} \to \mathbb{N}$ is λ-definable if there is a λ-term $t : \mathsf{Nat}_o \to \mathsf{Nat}_o$ such that for all $n \in \mathbb{N}$, $t\overline{n} =_\beta \overline{f(n)}$. The famous theorem of Schwichtenberg says that those functions are exactly the *extended polynomials* (Schwichtenberg, 1975).

This is not the end of the story, however. One can greatly augment the class of λ-definable functions by considering terms $t : \mathsf{Nat}_\tau \to \mathsf{Nat}_o$, where τ can be any arbitrary type. In this case, fast-growing functions up to towers of exponentials can be defined, but other very simple functions such as subtraction cannot be expressed (Fortune et al., 1983). For this notion of λ-definability, some partial descriptions of the class of definable functions can be found in the literature (Joly, 2001), but no exact characterization is currently known (Nguyên, 2019). Recently, new interest in the notion of λ-definability was sparked by the discovery of links between linear logic and automata theory. Indeed, the class of regular languages can be characterized as the class of functions definable by λ-terms of type $\mathsf{Str}_\tau \to \mathsf{Bool}_o$ (Hillebrand and Kanellakis, 1996). By imposing additional constraints on the λ-terms (e.g. using *non-commutative affine types*), one can characterize subclasses of regular languages, such as *star-free* languages (Nguyên and Pradic, 2020; Nguyên et al., 2020).

Dal Lago and Hofmann's work on resource monoids originated in the field of *implicit complexity*. In this context, the aim is usually to prove soundness results of the form: "every definable function lies in a given complexity class." In this way, various complexity classes can be characterized as the classes of all functions definable in a certain logical system. By instantiating their general framework in different ways, Dal Lago and Hofmann proved such soundness results for various programming languages: Elementary Affine Logic, LFPL and Soft Affine Logic (Dal Lago and Hofmann, 2011); Light Affine Logic (Dal Lago and Hofmann, 2010*b*); and Bounded Affine Logic (Dal Lago and Hofmann, 2010*a*). The same technique was used by Brunel and Terui (2010) to prove the polytime soundness of the system $\mathrm{DIAL}_{\mathrm{lin}}$. A modification of the resource monoid framework dubbed *quantitative realizability* was introduced by Brunel (2015).

Plan of the paper. The preliminary Section 2 defines the notions of resource monoids and length spaces, and shows how they give rise to a model of multiplicative affine logic. This is the common base given by the framework developed by Dal Lago and Hofmann.

Then, in Section 3, we define our resource monoid $\mathcal{M}$ of interest, whose elements are finite sequences of natural numbers. Such sequences will be used to bound the growth potential of a λ-term. For instance, a term bounded by the sequence $[a,b,c,d]$ can have a normal form of size at most $a + b \cdot 2^{c \cdot 2^d}$. Section 3.2 introduces this conversion between sequences of numbers and towers of exponents, that we call "collapsing" a sequence.

In Section 4, we define our interpretation of simply typed λ-calculus in the category of length spaces over $\mathcal{M}$. The main technical difficulty is to model the contraction rule, where a variable in the typing context of a term gets duplicated. This is where the notion of *order* of a type becomes crucial: the element of $\mathcal{M}$ bounding the duplication morphism will depend on the order of the

https://doi.org/10.1017/S0960129521000256 Published online by Cambridge University Press

type of the variable being duplicated. At the end of Section 4, we can already prove Claim 2 and 3, simply by considering the denotation of a term in our model.

Finally, in Section 5, we prove the remaining Claim 1. To achieve this, we need to analyze more carefully how the length of the bounding sequence grows when we compute the interpretation of a term in our model.

2. Preliminaries

2.1. Simply typed λ-calculus with constants

We briefly recall syntax and typing rules of simply typed λ-calculus, equipped with one base type o, and two constants $0 : o$ and $S : o \to o$. The types are given by the following grammar:

$$\sigma, \tau ::= o \mid \sigma \to \tau \mid \sigma \times \tau$$

The terms are given by the grammar below, where x belongs to some infinite set of variables.

$$t, u ::= 0 \mid S \mid x \mid tu \mid \lambda x.\, t$$

Finally, we give the typing rules with explicit contraction and weakening. This presentation highlights the contraction rule (CONTR), where a variable in the context gets duplicated. The main technical difficulty of the paper will be to deal with this duplication operation.

$$\textsc{Var}\ \frac{}{x : \sigma \vdash x : \sigma} \qquad \textsc{Lam}\ \frac{\Gamma, x : \sigma \vdash t : \tau}{\Gamma \vdash \lambda x.\, t : \sigma \to \tau} \qquad \textsc{App}\ \frac{\Gamma \vdash t : \sigma \to \tau \qquad \Delta \vdash u : \sigma}{\Gamma, \Delta \vdash t\, u : \tau}$$

$$\textsc{Zero}\ \frac{}{\vdash 0 : o} \qquad \textsc{Succ}\ \frac{}{\vdash S : o \to o} \qquad \textsc{Weak}\ \frac{\Gamma \vdash t : \tau}{\Gamma, x : \sigma \vdash t : \tau} \qquad \textsc{Contr}\ \frac{\Gamma, x : \sigma, y : \sigma \vdash t : \tau}{\Gamma, z : \sigma \vdash t[x, y \leftarrow z] : \tau}$$

2.2. Resource monoids and length spaces

In this paper, we use a simplified version of the notion of *resource monoid*, introduced by Dal Lago and Hofmann (2011). The elements of a resource monoid can be used to bound various quantitative properties of programs, such as runtimes or the size of data. To allow more flexibility, these bounds need not be numbers: they can be elements of any commutative monoid, equipped with a suitable pre-order.

Definition 1. *A resource monoid*[2] *is a triple* $\mathcal{M} = (|\mathcal{M}|, +, \leq)$ *such that:*

- $(|\mathcal{M}|, +)$ *is a commutative monoid, and*
- $\leq$ *is a pre-order on* $|\mathcal{M}|$ *compatible with* $+$*, that is,* $\alpha \leq \beta$ *implies* $\alpha + \gamma \leq \beta + \gamma$.

As shown in Dal Lago and Hofmann (2011), any resource monoid $\mathcal{M}$ gives rise to a notion of *length spaces* on $\mathcal{M}$, which form a symmetric monoidal closed category. Thus, this yields a model of multiplicative linear logic. In order to interpret programming languages with more features, one needs to choose a particular resource monoid of interest, whose associated category of length spaces has the desired additional structure. This is the approach taken in Dal Lago and Hofmann (2011, 2010b,a) to provide models of various programming languages: Elementary Affine Logic, Light Affine Logic, Soft Affine Logic, Bounded Affine Logic, and LFPL. In the remainder of the section, we assume given an arbitrary resource monoid $\mathcal{M}$.

[2] In Dal Lago and Hofmann (2011), resource monoids also contain a fourth component $\mathcal{D}$ that can be interpreted as the *difference* $\mathcal{D}(\alpha, \beta)$ between two elements $\alpha \leq \beta$ of the monoid. This allows to measure the cost of a computation (e.g., time complexity) as a trade-off between the size of the data and the time taken to process it. For the application in this paper, this component is not required.

https://doi.org/10.1017/S0960129521000256 Published online by Cambridge University Press

Definition 2. *A length space*[3] *on a resource monoid $\mathcal{M}$ is a pair $\mathcal{A} = (|\mathcal{A}|, \Vdash)$ where $|\mathcal{A}|$ is a set, and $\Vdash \subseteq |\mathcal{M}| \times |\mathcal{A}|$ satisfies the following properties:*

- *for all a, there exists α such that $\alpha \Vdash a$, and*
- *if $\alpha \Vdash a$ and $\alpha \leq \beta$, then $\beta \Vdash a$.*

When $\alpha \Vdash a$, we say that α is a *majorizer* of a, or that α *majorizes* a. Intuitively, a is the denotation of a program, and α represents an upper bound on some quantitative property of a. When we need to distinguish between several length spaces, we will write the majorizer relation $\Vdash_{\mathcal{A}}$, with the name of the length space as a subscript.

Definition 3. *Fix a resource monoid $\mathcal{M}$. A morphism of length spaces from $\mathcal{A}$ to $\mathcal{B}$ is a function $f : |\mathcal{A}| \to |\mathcal{B}|$ satisfying the following property:*

- *there exists $\varphi \in |\mathcal{M}|$, such that $\alpha \Vdash_{\mathcal{A}} a$ implies $\varphi + \alpha \Vdash_{\mathcal{B}} f(a)$.*

When this property holds, we call φ a majorizer of f, and we denote it by $\varphi \Vdash_{\mathcal{A} \multimap \mathcal{B}} f$.

As the above notation suggests, given two length spaces $\mathcal{A}$ and $\mathcal{B}$, we can define a new length space $\mathcal{A} \multimap \mathcal{B} := (\mathrm{Hom}(\mathcal{A}, \mathcal{B}), \Vdash_{\mathcal{A} \multimap \mathcal{B}})$, where $\mathrm{Hom}(\mathcal{A}, \mathcal{B})$ is the set of morphisms from $\mathcal{A}$ to $\mathcal{B}$. It is easily checked that $\mathcal{A} \multimap \mathcal{B}$ is indeed a length space. Similarly, we can define the tensor product of two length spaces, $\mathcal{A} \otimes \mathcal{B} := (|\mathcal{A}| \times |\mathcal{B}|, \Vdash_{\mathcal{A} \otimes \mathcal{B}})$, where the relation $\Vdash_{\mathcal{A} \otimes \mathcal{B}}$ is defined by $\gamma \Vdash_{\mathcal{A} \otimes \mathcal{B}} (a, b)$ iff there exist $\alpha, \beta \in |\mathcal{M}|$ such that:

$$\alpha \Vdash_{\mathcal{A}} a \quad \text{and} \quad \beta \Vdash_{\mathcal{B}} b \quad \text{and} \quad \gamma \geq \alpha + \beta.$$

Again, we can check that this is a well-defined length space on $\mathcal{M}$.

As mentioned earlier, length spaces together with the two operators $\multimap$ and $\otimes$ assemble into a symmetric monoidal closed category. The next Theorem was first proved by Dal Lago and Hofmann (2011). We reproduce the proof in details below; first because we slightly modified the definitions of resource monoids and length spaces, and second because we will need to know how the various maps are defined in order to compute the interpretation of a λ-term in this category.

Theorem. *(Dal Lago and Hofmann, 2011). Given any resource monoid $\mathcal{M}$, the category of length spaces on $\mathcal{M}$ is symmetric monoidal closed with respect to the tensor product and linear map defined above.*

Proof. Recall that morphisms of length spaces simply consist of set-theoretic functions which can be majorized by some element of $\mathcal{M}$. Thus, the symmetric monoidal closed structure is derived from the one in **Set**, and all we need to do is find appropriate majorizers.

For example, to define composition of two morphisms $f : \mathcal{A} \to \mathcal{B}$ and $g : \mathcal{B} \to \mathcal{C}$, majorized, respectively, by φ and ψ, we check that the function $g \circ f := x \mapsto g(f(x)) : |\mathcal{A}| \to |\mathcal{C}|$ is majorized by $\varphi + \psi$. Assume $\alpha \Vdash_{\mathcal{A}} a$, we want to show that $\varphi + \psi + \alpha \Vdash_{\mathcal{C}} g(f(a))$. Since φ majorizes f, we know that $\varphi + \alpha \Vdash_{\mathcal{B}} f(a)$, and therefore since ψ majorizes g, $\psi + \varphi + \alpha \Vdash_{\mathcal{C}} g(f(a))$. The identity function $\mathsf{id} : \mathcal{A} \to \mathcal{A}$ is majorized by $0_{\mathcal{M}}$, the neutral element of $\mathcal{M}$.

The unit object is defined as $\mathcal{I} = (\{\star\}, \Vdash_{\mathcal{I}})$ where $\alpha \Vdash_{\mathcal{I}} \star$ for all α. This is trivially a length space. The unitors, associator and braiding are all majorized by $0_{\mathcal{M}}$. Let us check that this is the case for the associator $\mathrm{assoc}_{\mathcal{A},\mathcal{B},\mathcal{C}} : (\mathcal{A} \otimes \mathcal{B}) \otimes \mathcal{C} \to \mathcal{A} \otimes (\mathcal{B} \otimes \mathcal{C})$, whose underlying function is $((a, b), c) \mapsto (a, (b, c))$. We assume that $\mu \Vdash_{(\mathcal{A} \otimes \mathcal{B}) \otimes \mathcal{C}} ((a, b), c)$, i.e., that $\mu \geq \nu + \gamma$ with $\nu \Vdash_{\mathcal{A}} \otimes \mathcal{B}(a, b)$ and $\gamma \Vdash_{\mathcal{C}} c$. Unfolding the definition of $\Vdash_{\mathcal{A} \otimes \mathcal{B}}$, we get that $\nu \geq \alpha + \beta$, with $\alpha \Vdash_{\mathcal{A}} a$ and $\beta \Vdash_{\mathcal{B}} b$. Then we can show $0_{\mathcal{M}} + \mu \Vdash_{\mathcal{A} \otimes (\mathcal{B} \otimes \mathcal{C})} (a, (b, c))$, using the fact that $\mu \geq \alpha + (\beta + \gamma)$.

[3] In Dal Lago and Hofmann (2011), the relation $\Vdash$ also contains a third component e, called a *realizer* of a. It plays a role to measure the computation time of programs, and for cardinality issues when interpreting the second order universal quantification. We do not need these features, hence we also drop these realizers from our definitions.

https://doi.org/10.1017/S0960129521000256 Published online by Cambridge University Press

Given two parallel morphisms $f : \mathcal{A} \to \mathcal{A}'$ and $g : \mathcal{B} \to \mathcal{B}'$, majorized by φ and ψ, respectively, the functorial action of $\otimes$ yields the morphism $f \otimes g : \mathcal{A} \otimes \mathcal{B} \to \mathcal{A}' \otimes \mathcal{B}'$. The underlying function $(a, b) \mapsto (f(a), g(b))$ can be majorized by $\varphi + \psi$. Indeed, given $\gamma \Vdash_{\mathcal{A}\otimes\mathcal{B}} (a, b)$, we need to prove that $\varphi + \psi + \gamma \Vdash_{\mathcal{A}'\otimes\mathcal{B}'} (f(a), g(b))$. By definition of $\Vdash_{\mathcal{A}\otimes\mathcal{B}}$, there are α, β such that $\gamma \geq \alpha + \beta$, $\alpha \Vdash_{\mathcal{A}} a$ and $\beta \Vdash_{\mathcal{B}} b$. Then we know that $\varphi + \alpha \Vdash_{\mathcal{A}'} f(a)$ and $\psi + \beta \Vdash_{\mathcal{B}'} g(b)$ because f and g are morphisms. Hence, $\varphi + \psi + \alpha + \beta \Vdash_{\mathcal{A}'\otimes\mathcal{B}'} (f(a), g(b))$, and we can conclude by the second property of length spaces.

The evaluation morphism $\mathrm{eval} : \mathcal{A} \otimes (\mathcal{A} \multimap \mathcal{B}) \to \mathcal{B}$, defined by $\mathrm{eval}(a, f) = f(a)$, can be majorized by $0_{\mathcal{M}}$. Indeed, assume $\varepsilon \Vdash_{\mathcal{A}\otimes(\mathcal{A}\multimap\mathcal{B})} (a, f)$. By definition, there are α, φ such that $\alpha \Vdash_{\mathcal{A}} a$, $\varphi \Vdash_{\mathcal{A}\multimap\mathcal{B}} f$ and $\varepsilon \geq \alpha + \varphi$. Since φ majorizes f we have $\alpha + \varphi \Vdash_{\mathcal{B}} f(a)$, and since $\varepsilon \geq \alpha + \varphi$, we conclude that $\varepsilon \Vdash_{\mathcal{B}} f(a)$ as required.

Finally to define currying, given any morphism $f : \mathcal{A} \otimes \mathcal{B} \to \mathcal{C}$ majorized by φ, the morphism $\mathrm{curry}(f) : \mathcal{A} \to (\mathcal{B} \multimap \mathcal{C})$ whose underlying function is $a \mapsto b \mapsto f(a, b)$ can also be majorized by φ. Indeed, assuming $\alpha \Vdash_{\mathcal{A}} a$, we want to show that $\varphi + \alpha \Vdash_{\mathcal{B}\multimap\mathcal{C}}]b \mapsto f(a, b)$. So suppose $\beta \Vdash_{\mathcal{B}} b$, we need to prove that $\varphi + \alpha + \beta \Vdash_{\mathcal{C}} f(a, b)$. But this is the case because $\alpha + \beta \Vdash_{\mathcal{A}\otimes\mathcal{B}} (a, b)$ and f is a morphism. □

Remark 4. We actually have a little bit more than stated in Section 2.2: for all $\mathcal{A}$, there is a unique weakening map from $\mathcal{A}$ to $\mathcal{I}$ majorized by $0_{\mathcal{M}}$. Thus, length spaces over a resource monoid actually provide a model of multiplicative *affine* logic. The missing ingredient to interpret λ-calculus is the duplication morphism $\Delta : \mathcal{A} \to \mathcal{A} \otimes \mathcal{A}$. Such a morphism might not exist in general, so we need to choose a suitable resource monoid.

3. A Resource Monoid Measuring the Size of Terms

We now define our resource monoid of interest, $\mathcal{M}$, that we will use later to prove the three Claims 1 to 3 of the introduction.

Intuition. The elements of $\mathcal{M}$ are finite lists of numbers, e.g. [3, 4, 5, 6]. Such a list can be thought of as representing a (large) natural number, obtained by computing a tower of exponents of 2. For instance, the list [3, 4, 5, 6] represents the number $3 + 4 \cdot 2^{5 \cdot 2^6}$. Usually, longer lists tend to produce very large numbers. Indeed, the size of the list corresponds to the height of the tower of exponentials. The ordering relation $\leq_{\mathcal{M}}$ on these lists of numbers *almost* amounts to comparing the associated numbers, but it also takes into account the size of the lists.

In Section 4, we will interpret λ-terms in the category of length spaces over $\mathcal{M}$. Thus, every λ-term can be associated with an element of $\mathcal{M}$, which should be thought of as an upper bound on the "potential" of this term for producing large natural numbers. Alternatively, one can think of it as measuring the size of the normal form of the associated λ-term. Indeed, for a closed term t of type o, the denotation $[\![t]\!] \in \mathbb{N}$ is equal to the size of its normal form. Crucially, the size of the list associated with a λ-term t is equal to the maximal order of a nonlinear λ-abstraction that occurs in t. This key observation will allow us to prove Claim 1 in Section 5.

3.1. The resource monoid

The elements of $\mathcal{M}$ are sequences of natural numbers with finite support, that is, ending with infinitely many 0's. For ease of notation, we write them as finite lists of natural numbers: an element $\alpha = (a_n)_{n\in\mathbb{N}}$ of $|\mathcal{M}|$ such that $a_k \neq 0$ and $\forall n > k, a_n = 0$ is denoted by $\alpha = [a_0, \ldots, a_k]$. We write $|\alpha|$ the length of that list (in this case, $k + 1$). Note that we always assume that the last element of a list is non-zero, to avoid having several lists representing the same sequence. The constant zero sequence (corresponding to the empty list) is written $0_{\mathcal{M}}$.

https://doi.org/10.1017/S0960129521000256 Published online by Cambridge University Press

Definition 5. *Let $\mathcal{M} = (|\mathcal{M}|, +_{\mathcal{M}}, \leq_{\mathcal{M}})$ be the following structure:*

- *$|\mathcal{M}|$ is the set of sequences of natural numbers with finite support.*
- *$+_{\mathcal{M}}$ is defined componentwise: we take the max on the first component, and the sum on the others. For ease of notation, we denote max(a, b) by $a \vee b$.*

$$(a_n) +_{\mathcal{M}} (b_n) = (c_n) \quad \textit{where} \begin{cases} c_0 = a_0 \vee b_0 \\ c_n = a_n + b_n \ \textit{for } n > 0 \end{cases}$$

- *$(a_n) \leq_{\mathcal{M}} (b_n)$ iff there exists $(d_n) \in \mathcal{M}$ such that:*

$$\begin{cases} a_0 \leq b_0 + d_0 \\ a_n + d_{n-1} \leq b_n \cdot 2^{d_n} \quad \textit{for all } n > 0 \end{cases}$$

The intuition behind this ordering is the following. First, notice that if we take (d_n) to be the constant zero sequence $0_{\mathcal{M}}$, then the condition above reduces to the componentwise ordering, $a_n \leq b_n$ for all n. However, if a_0 happens to be greater than b_0, we can accommodate for it by taking a "debt" d_0. This debt must be paid back in the next step, where we compare a_1 and b_1: if it is not the case that $a_1 + d_0 \leq b_1$, then we must accumulate some more debt d_1, and so on. However, notice that the debt must eventually be paid back: since the sequence (b_n) ultimately becomes null, so must (d_n). Also note that the debt decreases exponentially at each step (but never reaches 0 unless $a_i < b_i$ at some point).

Example 1.

(a) To check that $[7, 12, 1] \leq_{\mathcal{M}} [0, 2, 5]$, first we take a debt $d_0 = 7$ so that $7 \leq 0 + d_0$. Then the condition on the second component is $12 + 7 \leq 2 \cdot 2^{d_1}$; take $d_1 = 4$. Finally, the last condition is $1 + 4 \leq 5 \cdot 2^{d_2}$; we can take $d_2 = 0$ and the debt is paid off.

(b) Similarly, one can check that $[65537] \leq_{\mathcal{M}} [1, 1, 1, 1, 1, 1]$.

(c) However, $[65538] \not\leq_{\mathcal{M}} [1, 1, 1, 1, 1, 1]$.

Lemma 1. *If $\alpha \leq_{\mathcal{M}} \beta$, then $|\alpha| \leq |\beta|$.*

Proof. Let $\beta = [b_0, \ldots, b_k]$ (the case $\beta = 0_{\mathcal{M}}$ is trivial). Assuming $\alpha \leq_{\mathcal{M}} \beta$, for every index $n > k$, the condition $a_n + d_{n-1} \leq 0$ is verified because $b_n = 0$. Since both a_n and d_{n-1} are natural numbers, we must have $a_n = d_{n-1} = 0$. □

In the rest of the paper, we often use Lemma 1 implicitly: whenever there is an assumption of the form $\alpha \leq_{\mathcal{M}} [b_0, \ldots, b_k]$, only the first $(k+1)$ inequalities matter. In particular, the last inequality is just $a_k + d_{k-1} \leq b_k$, since we must have $d_k = 0$.

Proposition 6. *$\mathcal{M}$ is a resource monoid.*

Proof. First, we check that $(|\mathcal{M}|, +_{\mathcal{M}})$ is a commutative monoid. The neutral element is $0_{\mathcal{M}}$. Associativity and commutativity are clear since $+_{\mathcal{M}}$ is defined componentwise, and $+$ and $\vee$ on natural number are both associative and commutative.

Next, we check that $\leq_{\mathcal{M}}$ is a pre-order. Reflexivity is clear by taking $(d_n) = 0_{\mathcal{M}}$. For transitivity, assume $(a_n) \leq_{\mathcal{M}} (b_n)$ with debt (d_n), and $(b_n) \leq_{\mathcal{M}} (c_n)$ with debt (d'_n). We show that $(a_n) \leq_{\mathcal{M}} (c_n)$ with debt $(d_n + d'_n)$.

- $a_0 \leq b_0 + d_0$ and $b_0 \leq c_0 + d'_0$, thus $a_0 \leq c_0 + d_0 + d'_0$.

https://doi.org/10.1017/S0960129521000256 Published online by Cambridge University Press

- For all $n > 0$, we have $a_n + d_{n-1} \leq b_n \cdot 2^{d_n}$ and $b_n + d'_{n-1} \leq c_n \cdot 2^{d_n}$. Then,

$$\begin{aligned} a_n + d_{n-1} + d'_{n-1} &\leq b_n \cdot 2^{d_n} + d'_{n-1} \\ &\leq (b_n + d'_{n-1}) \cdot 2^{d_n} \\ &\leq c_n \cdot 2^{d'_n} \cdot 2^{d_n} \\ &\leq c_n \cdot 2^{d_n + d'_n} \end{aligned}$$

Finally, $\leq_{\mathcal{M}}$ is compatible with $+_{\mathcal{M}}$. Assume that $(a_n) \leq_{\mathcal{M}} (b_n)$ with debt (d_n). We show that $(a_n) +_{\mathcal{M}} (c_n) \leq_{\mathcal{M}} (b_n) +_{\mathcal{M}} (c_n)$ with debt (d_n).

- $a_0 \vee c_0 \leq (b_0 + d_0) \vee c_0 \leq (b_0 \vee c_0) + d_0$.
- For all $n > 0$, $(a_n + c_n) + d_{n-1} \leq b_n \cdot 2^{d_n} + c_n \leq (b_n + c_n) \cdot 2^{d_n}$. □

Remark 7. The relation $\leq_{\mathcal{M}}$ is actually an order. Indeed, if $\alpha \leq_{\mathcal{M}} \beta$ and $\beta \leq_{\mathcal{M}} \alpha$, then $|\alpha| = |\beta|$ and we can then prove by induction on their size that $\alpha = \beta$. However, this property is not required to be able to use the framework of resource monoids.

3.2. The collapse functions

An element $\alpha = [a_0, \ldots, a_k]$ of $\mathcal{M}$ can be associated with a natural number $a_0 + a_1 \cdot 2^{a_2 \cdot 2^{\cdot^{\cdot^{a_k}}}}$. We call this operation "collapsing" the sequence α. The order relation $\leq_{\mathcal{M}}$ on these sequences almost amounts to comparing these associated numbers, but not exactly. In this section, we make precise the relationship between the two. To this end, we define a function $\text{collapse}_n : |\mathcal{M}| \to |\mathcal{M}|$ for every $n \in \mathbb{N}$. The idea is that, given an element $\alpha = [a_0, \ldots, a_k]$ of $\mathcal{M}$, we decrease its length by one by plugging the last component into the penultimate one (either by addition or by multiplying by a power of 2, depending on the case). We then repeat this process until the size of the list becomes $n + 1$ or smaller. Formally:

$$\begin{aligned} &\text{collapse}_0([a_0, a_1]) && = [a_0 + a_1] && \\ &\text{collapse}_n(\alpha) && = \alpha && \text{if } |\alpha| \leq n + 1 \\ &\text{collapse}_n([a_0, \ldots, a_k]) && = \text{collapse}_n([a_0, \ldots, a_{k-2}, a_{k-1} \cdot 2^{a_k}]) && \text{otherwise} \end{aligned}$$

Example 2. *For instance,* $collapse_3([0, 1, 2, 3, 4, 5, 6]) = [0, 1, 2, 3 \cdot 2^{4 \cdot 2^{5 \cdot 2^6}}]$ *is a list of size 4, while* $collapse_0([6, 3, 11, 4]) = [6 + 3 \cdot 2^{11 \cdot 2^4}]$ *is a list of size 1.*

Proposition 8. *The collapse functions have the following properties:*

(i) For all n, $\left|collapse_n(\alpha)\right| \leq n + 1$.
(ii) For all n, $collapse_n(\alpha) \leq_{\mathcal{M}} \alpha$.
(iii) If $\alpha \leq_{\mathcal{M}} \beta$ *and* $|\alpha| \leq n + 1$, *then* $\alpha \leq_{\mathcal{M}} collapse_n(\beta)$.
(iv) For all n, if $\alpha \leq_{\mathcal{M}} \beta$, *then* $collapse_n(\alpha) \leq_{\mathcal{M}} collapse_n(\beta)$.
(v) If $n \leq m$, *then* $collapse_n(\alpha) \leq_{\mathcal{M}} collapse_m(\alpha)$.
(vi) If $|\alpha| \leq n$, *then* $collapse_n(\alpha + \beta) = \alpha + collapse_n(\beta)$.

Proof.

(i) The first property is straightforward by induction on the size of the list.
(ii) Proceed by induction on the size of α. We distinguish the three cases of the definition.

https://doi.org/10.1017/S0960129521000256 Published online by Cambridge University Press

- If $|\alpha| \leq n+1$, then $\text{collapse}_n(\alpha) = \alpha \leq_{\mathcal{M}} \alpha$.
- If $\alpha = [a_0, a_1]$ and $n = 0$, then $[a_0 + a_1] \leq_{\mathcal{M}} [a_0, a_1]$ by taking the debt $d_0 = a_1$.
- Otherwise $\alpha = [a_0, \ldots, a_k]$ with $k > n$, and the induction hypothesis gives us:

$$\text{collapse}_n([a_0, \ldots, a_{k-2}, a_{k-1} \cdot 2^{a_k}]) \leq_{\mathcal{M}} [a_0, \ldots, a_{k-2}, a_{k-1} \cdot 2^{a_k}]$$

Moreover,

$$[a_0, \ldots, a_{k-2}, a_{k-1} \cdot 2^{a_k}] \leq_{\mathcal{M}} [a_0, \ldots, a_k]$$

by taking the debt $d_{k-1} = a_k$ and $d_i = 0$ otherwise. By transitivity,

$$\text{collapse}_n([a_0, \ldots, a_k]) \leq_{\mathcal{M}} [a_0, \ldots, a_k].$$

(iii) By induction on the size of β. Let (d_i) be the debt that proves $\alpha \leq_{\mathcal{M}} \beta$.

- Case $|\beta| \leq n+1$: trivial since $\text{collapse}_n(\beta) = \beta$.
- Case $\beta = [b_0, b_1]$ and $n = 0$: we have $a_0 \leq b_0 + d_0$ and $a_1 + d_0 \leq b_1$. Since $|\alpha| \leq 1$, we have $a_1 = 0$ and therefore $d_0 \leq b_1$. So $a_0 \leq b_0 + b_1$, from which we can deduce that $\alpha \leq_{\mathcal{M}} [b_0 + b_1] = \text{collapse}_0(\beta)$ without debt.
- Case $\beta = [b_0, \ldots, b_k]$ for $k > n$. By induction hypothesis, we just need to prove:

$$\alpha \leq_{\mathcal{M}} [b_0, \ldots, b_{k-2}, b_{k-1} \cdot 2^{b_k}].$$

We use the debt (d'_i) defined by $d'_k = 0$ and $d'_i = d_i$ otherwise. The conditions at ranks 0 to $k-2$ stay the same. We need to check the condition at rank $k-1$, which is $a_{k-1} + d_{k-1} \leq b_{k-1} \cdot 2^{b_k}$. From the hypothesis $\alpha \leq_{\mathcal{M}} \beta$, we get $a_{k-1} + d_{k-1} \leq b_{k-1} \cdot 2^{d_k}$ and $a_k + d_k \leq b_k$. But we assumed $|\alpha| \leq n+1$ and $k > n$, so $a_k = 0$. Hence, $d_k \leq b_k$ and we are done.

(iv) By property (ii), $\text{collapse}_n(\alpha) \leq_{\mathcal{M}} \alpha \leq_{\mathcal{M}} \beta$. But $\left|\text{collapse}_n(\alpha)\right| \leq n+1$, so by property (iii) we conclude that $\text{collapse}_n(\alpha) \leq_{\mathcal{M}} \text{collapse}_n(\beta)$.

(v) By (ii), we have $\text{collapse}_n(\alpha) \leq_{\mathcal{M}} \alpha$. Moreover, since $\left|\text{collapse}_n(\alpha)\right| \leq n+1 \leq m+1$, we can apply (iii) to get $\text{collapse}_n(\alpha) \leq_{\mathcal{M}} \text{collapse}_m(\alpha)$.

(vi) This is obvious since the collapse_n function does not alter the first n components of a list. Formally, a proof by induction on the size of β is straightforward. □

4. A Quantitative Model for λ-Calculus

We can now interpret the λ-calculus with constants of Section 2.1 in the category of length spaces over $\mathcal{M}$. To do this, we need to define the interpretation of the constants; and most importantly, define a duplication morphism to interpret the contraction rule.

4.1. Interpretations of o, 0, S

In our model, we would like the underlying sets of the length spaces, and the underlying functions of the morphisms, to be the usual set-theoretic semantics. Thus, we need a length space whose underlying set is $\mathbb{N}$ in order to interpret the type o.

Definition 9. *Define* $\mathbf{N} = (\mathbb{N}, \Vdash_{\mathbf{N}})$, *where* $\alpha \Vdash_{\mathbf{N}} n$ *iff* $[n] \leq_{\mathcal{M}} \alpha$.

https://doi.org/10.1017/S0960129521000256 Published online by Cambridge University Press

This is easily seen to be a length space on $\mathcal{M}$. Indeed, for all $n \in \mathbb{N}$ we have $[n] \Vdash_{\mathbf{N}} n$. Moreover, if $\alpha \Vdash_{\mathbf{N}} n$ and $\alpha \leq_{\mathcal{M}} \beta$, by transitivity $[n] \leq_{\mathcal{M}} \beta$ so $\beta \Vdash_{\mathbf{N}} n$.

Lemma 2. *We have another characterization of* $\Vdash_{\mathbf{N}}$*:* $\alpha \Vdash_{\mathbf{N}} n$ *iff* $[n] \leq_{\mathcal{M}} collapse_0(\alpha)$*. If we write* $collapse_0(\alpha) = [a_0]$*, then* $\alpha \Vdash_{\mathbf{N}} n$ *iff* $n \leq a_0$*.*

Proof. If $[n] \leq_{\mathcal{M}} \alpha$, then by Proposition 8 (iii), $[n] \leq_{\mathcal{M}} \mathrm{collapse}_0(\alpha)$. Conversely, if $[n] \leq_{\mathcal{M}} \mathrm{collapse}_0(\alpha)$, then $[n] \leq_{\mathcal{M}} \alpha$ by Proposition 8 (ii) and transitivity.

Next, we want to interpret the function symbols 0 and S. Recall that the unit of the category of length spaces is $\mathcal{I} = (\{\star\}, \Vdash_{\mathcal{I}})$ where $\alpha \Vdash_{\mathcal{I}} \star$ for all α.

- We interpret $0 : o$ as the morphism in $\mathcal{I} \multimap \mathbf{N}$ that sends $\star$ to $0 \in \mathbb{N}$. It is easy to check that this morphism is majorized by $0_{\mathcal{M}}$.
- $S : o \to o$ is a little bit less trivial: we want to interpret it as the successor function $n \mapsto n + 1 : \mathbb{N} \to \mathbb{N}$. To prove that it is indeed a morphism in $\mathbf{N} \multimap \mathbf{N}$, we need to find a majorizer. Let us check that $[0, 1] \Vdash_{\mathbf{N} \multimap \mathbf{N}} n \mapsto n + 1$.

Suppose that $\alpha \Vdash_{\mathbf{N}} n$, we want to prove that $\alpha + [0, 1] \Vdash_{\mathbf{N}} n + 1$. By assumption, we have $[n] \leq_{\mathcal{M}} \alpha$, so $[n] + [0, 1] \leq_{\mathcal{M}} \alpha + [0, 1]$. And by Proposition 8 (ii), we get $[n + 1] \leq_{\mathcal{M}} \alpha + [0, 1]$.

4.2. The duplication morphism

In Section 2.2, we have seen that we can interpret multiplicative linear logic with full weakening in the category of length spaces on $\mathcal{M}$. What remains to be done is to define a *duplication* morphism of type $\mathcal{A} \to \mathcal{A} \otimes \mathcal{A}$ whose underlying function is $a \mapsto (a, a)$. Unfortunately, this is not possible in general: we would have to find an element $\varphi \in \mathcal{M}$ such that $\alpha + \alpha \leq_{\mathcal{M}} \varphi + \alpha$ for all α, and no such element exists (take $|\alpha| > |\varphi|$).

However, finding such a φ becomes possible if we know in advance that the size of α is bounded (see Lemma 4). To obtain such a bound on the size of majorizers, we need to have a notion of *order* of a length space, akin to the order of a type in λ-calculus. Thus, let us restrict ourselves to the full subcategory $\mathcal{S}$ of the category of length spaces on $\mathcal{M}$, whose objects are the length spaces built inductively from $\mathbf{N}$, $\mathcal{I}$, $\multimap$ and $\otimes$. The objects of $\mathcal{S}$ are the length spaces generated by the following grammar:

$$\mathcal{A}, \mathcal{B} ::= \mathcal{I} \mid \mathbf{N} \mid \mathcal{A} \multimap \mathcal{B} \mid \mathcal{A} \otimes \mathcal{B}$$

Definition 10. *The order* $Ord(\tau)$ *of a type* τ *is defined inductively as follows:*

$$\begin{aligned} Ord(o) &= 0 \\ Ord(\tau_1 \to \tau_2) &= max(Ord(\tau_1) + 1, Ord(\tau_2)) \\ Ord(\tau_1 \times \tau_2) &= max(Ord(\tau_1), Ord(\tau_2)) \end{aligned}$$

For a length space $\mathcal{A}$ *in* $\mathcal{S}$*, we also define* $Ord(\mathcal{A})$ *in a similar way (with* $Ord(\mathcal{I}) = 0$*).*

The next Lemma is the main motivation for the definition of the collapse functions. Intuitively, it says that the function denoted by a term $t : \tau$ can always be majorized by an element of $\mathcal{M}$ of size at most $\mathrm{Ord}(\tau) + 1$. Such a majorizer is obtained from any other majorizer by applying the function $\mathrm{collapse}_{[\mathrm{Ord}(\tau)]}$.

Lemma 3. *Let* $\mathcal{A} \in \mathcal{S}$*, and let* $n = Ord(\mathcal{A})$*. If* $\alpha \Vdash_{\mathcal{A}} a$*, then* $collapse_n(\alpha) \Vdash_{\mathcal{A}} a$*.*

Proof. By induction on the structure of $\mathcal{A}$.

- The case $\mathcal{A} = \mathcal{I}$ is trivial.

https://doi.org/10.1017/S0960129521000256 Published online by Cambridge University Press

- The case $\mathcal{A} = \mathbf{N}$ follows from Lemma 2.
- Case $\mathcal{A} = \mathcal{B} \multimap \mathcal{C}$: assume $\varphi \Vdash_{\mathcal{B}\multimap\mathcal{C}} f$. By definition of the order of $\mathcal{B} \multimap \mathcal{C}$, we must have $\mathrm{Ord}(\mathcal{B}) \leq n-1$ and $\mathrm{Ord}(\mathcal{C}) \leq n$.
  Assume $\beta \Vdash_{\mathcal{B}} b$. By induction hypothesis, $\mathrm{collapse}_{\mathrm{Ord}(\mathcal{B})}(\beta) \Vdash_{\mathcal{B}} b$, and by Proposition 8 (v) and upward-closure of $\Vdash$, $\mathrm{collapse}_{n-1}(\beta) \Vdash_{\mathcal{B}} b$.
  Since φ majorizes f, we have $\varphi + \mathrm{collapse}_{n-1}(\beta) \Vdash_{\mathcal{C}} f(b)$. By the second induction hypothesis (and upward-closure), $\mathrm{collapse}_n(\varphi + \mathrm{collapse}_{n-1}(\beta)) \Vdash_{\mathcal{C}} f(b)$. Then:

$$\begin{aligned} &\mathrm{collapse}_n(\varphi + \mathrm{collapse}_{n-1}(\beta)) \\ &\leq_{\mathcal{M}} \mathrm{collapse}_n(\varphi) + \mathrm{collapse}_{n-1}(\beta) && \text{by Proposition 8 (vi)} \\ &\leq_{\mathcal{M}} \mathrm{collapse}_n(\varphi) + \beta && \text{by Proposition 8 (ii)} \end{aligned}$$

  Hence $\mathrm{collapse}_n(\varphi) + \beta \Vdash_{\mathcal{C}} f(b)$, so $\mathrm{collapse}_n(\varphi) \Vdash_{\mathcal{B}\multimap\mathcal{C}} f$.
- Case $\mathcal{A} = \mathcal{B} \otimes \mathcal{C}$: assume $\alpha \Vdash_{\mathcal{B}\otimes\mathcal{C}} (b, c)$. There are β and γ such that $\beta \Vdash_{\mathcal{B}} b$, $\gamma \Vdash_{\mathcal{C}} c$, and $\beta + \gamma \leq_{\mathcal{M}} \alpha$. Since $\mathrm{Ord}(\mathcal{B}) \leq n$ and $\mathrm{Ord}(\mathcal{C}) \leq n$, we can use the induction hypothesis (and upward-closure of $\Vdash$) to obtain $\mathrm{collapse}_n(\beta) \Vdash_{\mathcal{B}} b$ and $\mathrm{collapse}_n(\gamma) \Vdash_{\mathcal{C}} c$.
  Moreover, $\mathrm{collapse}_n(\beta) + \mathrm{collapse}_n(\gamma) \leq_{\mathcal{M}} \beta + \gamma \leq_{\mathcal{M}} \alpha$. By Proposition 8 (iii), $\mathrm{collapse}_n(\beta) + \mathrm{collapse}_n(\gamma) \leq_{\mathcal{M}} \mathrm{collapse}_n(\alpha)$. Therefore, $\mathrm{collapse}_n(\alpha) \Vdash_{\mathcal{B}\otimes\mathcal{C}} (b, c)$. □

Lemma 4 *Let* $\varphi = [0, 0, 1, \ldots, 1]$ *where* $|\varphi| = n+1$. *Then, for all* α *such that* $|\alpha| \leq n$, $\alpha + \alpha \leq_{\mathcal{M}} \varphi + \alpha$.

Proof. Write $\alpha = (a_i)$ and take the debt $d_i = 1$ when $1 \leq i \leq n-1$, and $d_i = 0$ otherwise. On the first component, $a_0 \vee a_0 \leq (a_0 \vee 0) + 0$ is verified. On the second component, $a_1 + a_1 \leq (a_1 + 0) \cdot 2^1$ is verified. Then for $2 \leq i \leq n-1$, $a_i + a_i + 1 \leq (a_i + 1) \cdot 2^1$ is verified. The last condition is just $1 \leq 1$ since $a_n = 0$. □

We can now define the duplication in $\mathcal{S}$:

Proposition 11. *Let* $\mathcal{A} \in \mathcal{S}$ *with* $Ord(\mathcal{A}) = n$, *the duplication function* $a \mapsto (a, a)$ *is majorized by* $\varphi_n = [0, 0, 1, \ldots, 1]$ *where* $|\varphi_n| = n+2$.

Proof. Assume $\alpha \Vdash_{\mathcal{A}} a$, we must prove that $\varphi_n + \alpha \Vdash_{\mathcal{A}\otimes\mathcal{A}} (a, a)$. By Lemma 3, we know that $\mathrm{collapse}_n(\alpha) \Vdash_{\mathcal{A}} a$, and since $\left|\mathrm{collapse}_n(\alpha)\right| \leq n+1$, we can apply Lemma 4 to get $\mathrm{collapse}_n(\alpha) + \mathrm{collapse}_n(\alpha) \leq_{\mathcal{M}} \varphi_n + \mathrm{collapse}_n(\alpha)$. Then by Proposition 8 (ii), we obtain $\mathrm{collapse}_n(\alpha) + \mathrm{collapse}_n(\alpha) \leq_{\mathcal{M}} \varphi_n + \alpha$, which concludes the proof. □

4.3. *Interpreting λ-calculus*

Putting together the results of Sections 2.2, 4.1 and 4.2, we get the next Theorem, allowing us to interpret our simply typed λ-calculus in the category of length spaces on $\mathcal{M}$.

Theorem. *The category* $\mathcal{S}$ *is cartesian closed.*

We denote by $[\![-]\!]_{\mathcal{S}}$ the interpretation of types and terms in $\mathcal{S}$, and by $[\![-]\!]$ the standard set-theoretic interpretation. Thus,

$$[\![o]\!]_{\mathcal{S}} := \mathbf{N} \qquad [\![\sigma \to \tau]\!]_{\mathcal{S}} := [\![\sigma]\!]_{\mathcal{S}} \multimap [\![\tau]\!]_{\mathcal{S}} \qquad [\![\sigma \times \tau]\!]_{\mathcal{S}} := [\![\sigma]\!]_{\mathcal{S}} \otimes [\![\tau]\!]_{\mathcal{S}}$$

Recall that $[\![0]\!]_{\mathcal{S}} : \mathcal{I} \to \mathbf{N}$ is majorized by $0_{\mathcal{M}}$; $[\![S]\!]_{\mathcal{S}} : \mathbf{N} \to \mathbf{N}$ is majorized by [0,1]; and the duplication morphism $\Delta_{\mathcal{A}} : \mathcal{A} \to \mathcal{A} \otimes \mathcal{A}$ with $\mathrm{Ord}(\mathcal{A}) = n$ is majorized by $[0, 0, 1, \ldots, 1]$ of size $n+2$.

https://doi.org/10.1017/S0960129521000256 Published online by Cambridge University Press

Moreover, recall from the proof of Section 2.2 that if f and g are majorized by φ and ψ, then both $f \circ g$ and $f \times g$ are majorized by $\varphi + \psi$. All the other structure maps of $\mathcal{S}$ are majorized by 0_M.

Thus, given a term $t : \tau$, we can effectively compute a majorizer of the element $[\![t]\!]_{\mathcal{S}}$ of the length space $[\![\tau]\!]_{\mathcal{S}}$. It is straightforward to check that the underlying set of $[\![\tau]\!]_{\mathcal{S}}$ is $[\![\tau]\!]$, and the underlying function of $[\![t]\!]_{\mathcal{S}}$ is $[\![t]\!]$.

Example 3. Let us compute the majorizers of a few λ-terms.

(a) Let $t = S(S(\ldots(S0)))$, with k occurrences of S. Then $[\![t]\!]_{\mathcal{S}} : \mathcal{I} \to \mathbf{N}$ is majorized by $[0,k]$. By Lemma 2, this yields the upper bound $[\![t]\!] \leq k$, which is a tight bound since the denotation of t is $[\![t]\!] = k$.

(b) The Church encoding $\bar{2} = \lambda f.\, \lambda x.\, f(fx) : (o \to o) \to o \to o$ has a contraction on the variable f of order 1. Thus, its denotation $[\![\bar{2}]\!]_{\mathcal{S}}$ is majorized by $[0,0,1]$.

(c) Let $u = \bar{2}(\bar{2}(\ldots(\bar{2}S)))0$, with k occurrences of $\bar{2}$. Its denotation $[\![u]\!]_{\mathcal{S}} : \mathcal{I} \to \mathbf{N}$ is majorized by $[0,1,k]$. By Lemma 2, we can collapse it to get the bound $[\![u]\!] \leq 2^k$, which is once again a tight bound.

(d) Let $v = (((\bar{2}\bar{2})\bar{2})\ldots\bar{2})S0$, with k occurrences of $\bar{2}$. Here, each occurrence of $\bar{2}$ has a different type, $\bar{2} : (\tau \to \tau) \to \tau \to \tau$, where the order of τ increases by 1 at each occurrence. Therefore, $[\![v]\!]_{\mathcal{S}}$ is majorized by $[0, 1, k, k-1, \ldots, 2, 1]$ of size $k+2$. By collapsing it using Lemma 2, we get the upper bound $[\![v]\!] \leq 1 \cdot 2^{k \cdot 2^{\cdot^{\cdot^{\cdot^{2 \cdot 2^1}}}}}$. This time the bound is not tight: the actual value of $[\![v]\!]$ is a tower of exponentials of 2 with the same height, but without the multiplicative constants k, $k-1$, etc. For example, with $k = 3$, $[\![v]\!] = 2^{2^2} = 16$, while the majorizer $[0, 1, 3, 2, 1]$ collapses to $[2^{48}]$.

We can already prove Claims 2 and 3 of the introduction:

Claim 2. *For every closed term $f : o \to o$, there exists a constant $C \in \mathbb{N}$ such that its denotation $[\![f]\!] : \mathbb{N} \to \mathbb{N}$ is bounded by $n \mapsto n + C$.*

Proof. Let $f : o \to o$, and $[\![f]\!] : \mathbb{N} \to \mathbb{N}$ its set-theoretic denotation. Thus, there is a morphism $[\![f]\!]_{\mathcal{S}} : \mathbf{N} \multimap \mathbf{N}$ in $\mathcal{S}$ whose underlying function is $[\![f]\!]$. Let φ be a majorizer of $[\![f]\!]$. By Lemma 3, $\mathrm{collapse}_1(\varphi) = [f_0, f_1]$ is also a majorizer. Then, for any $n \in \mathbb{N}$, since $[n] \Vdash_{\mathbf{N}} n$, we have $[n] + [f_0, f_1] \Vdash_{\mathbf{N}} f(n)$. By Lemma 2, $f(n) \leq (n \vee f_0) + f_1 \leq n + (f_0 + f_1) = n + C_f$. □

Claim 3. *For every closed term $f : o^k \to o$, there exists a constant $C \in \mathbb{N}$ such that its denotation $[\![f]\!] : \mathbb{N}^k \to \mathbb{N}$ is bounded by $(n_1, \ldots, n_k) \mapsto max(n_1, \ldots, n_k) + C$.*

Proof. Let $f : o^k \to o$, and $[\![f]\!]_{\mathcal{S}} : \mathbf{N}^k \multimap \mathbf{N}$ in $\mathcal{S}$ whose underlying function is $[\![f]\!]$. By the same reasoning, $[\![f]\!]_{\mathcal{S}}$ has a majorizer of size 2, say $\varphi = [f_0, f_1]$. Given k arguments $n_1, \ldots, n_k$ majorized by $[n_1], \ldots, [n_k]$, respectively, the tuple $(n_1, \ldots, n_k)$ is majorized by $[n_1] +_{\mathcal{M}} \ldots +_{\mathcal{M}} [n_k] = [\max(n_1, \ldots, n_k)]$. Then the same reasoning as before yields $f(n_1, \ldots, n_k) \leq \max(n_1, \ldots, n_k) + C_f$. □

5. Bounding the Majorizers

As demonstrated in Example 3, our model associates every λ-term t with a majorizer $\alpha \Vdash [\![t]\!]$. For a closed term of type o, this majorizer can be collapsed to give an upper bound on the denotation of the term. To prove the remaining Claim 1, we want to express this majorizer in terms of two parameters: (i) the size $|t|$ of the term t and (ii) the maximal order of a nonlinear λ-abstraction that occurs in t, $\mathsf{rank}(t)$. This is done in Lemma 6, which says that the size of the majorizer (which determines the height of the tower of exponentials) is $\mathsf{rank}(t) + 2$, and each coefficient of the majorizer is smaller than $|t|$.

https://doi.org/10.1017/S0960129521000256 Published online by Cambridge University Press

Definition 12. *First we introduce notations to write towers of exponentials more compactly:*

- $\mathbf{2}_n^a = 2^{2^{\cdot^{\cdot^{\cdot^a}}}}$ *with n occurrences of 2, i.e.,* $\begin{cases} \mathbf{2}_0^a = a \\ \mathbf{2}_{n+1}^a = 2^{\mathbf{2}_n^a} \end{cases}$
- $\mathscr{C}([a_1, \ldots, a_n]) = a_1 \cdot 2^{a_2 \cdot 2^{\cdot^{\cdot^{\cdot^{a_n}}}}}$, *i.e.,* $\begin{cases} \mathscr{C}([]) = 0 \\ \mathscr{C}([a_1, \ldots, a_n]) = a_1 \cdot 2^{\mathscr{C}([a_2, \ldots, a_n])} \end{cases}$

With this notation, we have $\text{collapse}_0([a_0, \ldots, a_n]) = [a_0 + \mathscr{C}([a_1, \ldots, a_n])]$.

Lemma 5. *Let* $\alpha = [a_0, \ldots, a_n] \in \mathcal{M}$, *and let* $a = \max(a_0, \ldots, a_n)$. *Then* $\text{collapse}_0(\alpha) \leq \mathbf{2}_n^a$.

Proof. Since $a_i \leq a$ for all i, $[a_0, \ldots, a_n] \leq [a, \ldots, a]$ with $(n+1)$-many a's, and by Proposition 8(iv), it is enough to prove that $\text{collapse}_0([a, \ldots, a]) \leq \mathbf{2}_n^a$. To keep track of the number of a's, we write $\mathscr{C}_n^a = \mathscr{C}([a, \ldots, a])$ with n-many a's. Let us show that $a + \mathscr{C}_n^a \leq \mathbf{2}_n^a$. We proceed by induction on n.

- $n = 0$ is trivial: $\mathscr{C}_0^a = 0$ and $\mathbf{2}_0^a = a$.
- $\mathbf{2}_{n+1}^a = 2^{\mathbf{2}_n^a} \geq 2^{a + \mathscr{C}_n^a} = 2^a \cdot 2^{\mathscr{C}_n^a}$ by induction hypothesis. Moreover, since a is a natural number, we have $2^a \geq a + a$. So $2^a \cdot 2^{\mathscr{C}_n^a} \geq (a+a) \cdot 2^{\mathscr{C}_n^a} \geq a + a \cdot 2^{\mathscr{C}_n^a} = a + \mathscr{C}_{n+1}^a$. □

With Lemma 5 in mind, we are going to bound the size of $\text{collapse}_0(\varphi)$ by relying on two parameters: the size of φ and its maximal coefficient. When φ is the majorizer of some λ-term f, these two parameters correspond respectively to the *rank* and the size of f. Proving this fact will be the aim of Lemma 6.

Definition 13. *We define the rank of a well-typed term t. The idea is that* $\mathsf{rank}(t)$ *is the maximal order of a contraction that occurs in the typing derivation of t. When t is closed, the rank is defined as follows:*

$$\begin{aligned} \mathsf{rank}(x) &= 0 \\ \mathsf{rank}(0) &= 0 \\ \mathsf{rank}(S) &= 0 \\ \mathsf{rank}(tu) &= \mathit{max}(\mathsf{rank}(t), \mathsf{rank}(u)) \\ \mathsf{rank}(\lambda x^\tau . t) &= \begin{cases} \mathsf{rank}(t) & \text{if } x \text{ appears at most once in } t \\ \mathit{max}(\mathit{Ord}(\tau), \mathsf{rank}(t)) & \text{otherwise} \end{cases} \end{aligned}$$

When t has free variables $x_1, \ldots, x_n$ *and is typed in context* $\Gamma = x_1 : \tau_1, \ldots, x_n : \tau_n$,

$$\mathsf{rank}(\Gamma \vdash t : \tau) = \mathit{max}(\{\mathit{Ord}(\tau_i) \mid x_i \text{ appears at least twice in } t\} \cup \{\mathsf{rank}(t)\})$$

For $\Gamma = x_1 : \tau_1, \ldots, x_n : \tau_n$, we write $|\Gamma| = n$ and $\mathsf{var}(\Gamma) = \{x_1, \ldots, x_n\}$. For a λ-term t, we denote by $\mathsf{FV}(t)$ the set of free variables of t, and its size $|t|$ is defined inductively by

$$|0| = 0 \qquad |S| = 1 \qquad |x| = 1 \qquad |t\ u| = |t| + |u| \qquad |\lambda x.\, t| = |t| + 1.$$

https://doi.org/10.1017/S0960129521000256 Published online by Cambridge University Press

For $\alpha = [a_0, \ldots, a_n] \in \mathcal{M}$, we write $\max(\alpha) = \max(a_0, \ldots, a_n)$. The key result of this section is the following Lemma.

Lemma 6. *If* $\Gamma \vdash t : \tau$ *and* $\mathsf{var}(\Gamma) \subseteq \mathsf{FV}(t)$,[4] *then there is a majorizer* α *of* $[\![t]\!]_{\mathcal{S}}$ *such that* $|\alpha| \leq \mathsf{rank}(\Gamma \vdash t : \tau) + 2$ *and* $\max(\alpha) + |\Gamma| \leq |t|$.

Before we can prove Lemma 6, we need to make sure that the derivation of $\Gamma \vdash t : \tau$ does not contain unnecessary contractions, i.e., contractions that introduce a new variable which is later weakened. This is the aim of two Facts below.

Fact 1. For every derivation of $\Gamma, x : \sigma \vdash t : \tau$ with $x \notin \mathsf{FV}(t)$, there is a derivation of $\Gamma \vdash t : \tau$ whose height is smaller or equal.

Proof. This is a straightforward induction on the derivation of $\Gamma, x : \sigma \vdash t : \tau$. □

Fact 2. If $\Gamma \vdash t : \tau$ with $\mathsf{var}(\Gamma) \subseteq \mathsf{FV}(t)$, then there is a derivation of $\Gamma \vdash t : \tau$ where every occurrence of a weakening rule is immediately above the lambda rule introducing the weakened variable, as shown below □

$$\dfrac{\dfrac{\Gamma \vdash t : \tau}{\Gamma, x : \sigma \vdash t : \tau}\ \textsc{Weak}}{\Gamma \vdash \lambda x.\, t : \sigma \to \tau}\ \textsc{Lam}$$

Proof. By induction on the height of the derivation.

- Var, Zero, Succ: these derivations already satisfy the property.
- App: $\Gamma, \Delta \vdash tu : \tau$ comes from $\Gamma \vdash t : \sigma \to \tau$ and $\Delta \vdash u : \sigma$.
  Since $\mathsf{var}(\Gamma, \Delta) \subseteq \mathsf{FV}(tu)$ by assumption and $\Gamma \cap \Delta = \varnothing$, we have $\mathsf{var}(\Gamma) \subseteq \mathsf{FV}(t)$ and $\mathsf{var}(\Delta) \subseteq \mathsf{FV}(u)$, so we can use the induction hypothesis on both premises. Then by applying the App rule, we get a derivation that satisfies the property.
- Lam: $\Gamma \vdash \lambda x.\, t : \sigma \to \tau$ comes from $\Gamma, x : \sigma \vdash t : \tau$.
  Either $x \in FV(t)$ or $x \notin \mathsf{FV}(t)$. In the first case, we can use our induction hypothesis and we are done. In the second case, Fact 1 gives us a derivation of $\Gamma \vdash t : \tau$ whose height is smaller, so we can use the induction hypothesis to get a derivation of $\Gamma \vdash t : \tau$ satisfying the desired property. Using weakening on x and the Lam rule, we get a derivation of $\Gamma \vdash \lambda x.\, t : \sigma \to \tau$ that satisfies the property.
- Contr: $\Gamma, z : \sigma \vdash t[x, y \leftarrow z] : \tau$ comes from $\Gamma, x : \sigma, y : \sigma \vdash t : \tau$.
  At least one of the variables x and y is in $\mathsf{FV}(t)$, otherwise z would not be either. If they both are in $\mathsf{FV}(t)$, we use the induction hypothesis and we are done. Otherwise, if $x \notin \mathsf{FV}(t)$, then Fact 1 gives us a smaller derivation of $\Gamma, y : \sigma \vdash t : \tau$. We can then use the induction hypothesis on this derivation, and by renaming y into z, we are done. Same reasoning if $y \notin \mathsf{FV}(t)$ instead.
- Weak: $\Gamma, x : \sigma \vdash t : \tau$ comes from $\Gamma \vdash t : \tau$.
  This case is not possible since by assumption $x \in \mathsf{FV}(t)$.

We can now prove Lemma 6, with the extra assumption that the derivation does not contain weakening rules, except right above the corresponding lambda rule.

Proof of Lemma 6. By induction on the typing derivation of $\Gamma \vdash t : \tau$.

- Var, Zero, Succ: the morphisms $\mathsf{id} : [\![\tau]\!]_{\mathcal{S}} \to [\![\tau]\!]_{\mathcal{S}}$, $[\![0]\!]_{\mathcal{S}} : \mathcal{I} \to \mathbf{N}$ and $[\![S]\!]_{\mathcal{S}} : \mathbf{N} \to \mathbf{N}$ are majorized by $0_{\mathcal{M}}$, $0_{\mathcal{M}}$ and [0,1] respectively, which satisfy the two conditions.

[4] Note that $\mathsf{FV}(t) \subseteq \mathsf{var}(\Gamma)$ is always true for a well-typed term, so in fact $\mathsf{FV}(t) = \mathsf{var}(\Gamma)$.

https://doi.org/10.1017/S0960129521000256 Published online by Cambridge University Press

- APP: $\Gamma, \Delta \vdash tu : \tau$ comes from $\Gamma \vdash t : \sigma \to \tau$ and $\Delta \vdash u : \sigma$.
  Since $\Gamma \cap \Delta = \varnothing$, we have $\mathsf{var}(\Gamma) \subseteq \mathsf{FV}(t)$ and $\mathsf{var}(\Delta) \subseteq \mathsf{FV}(u)$, so we can use the induction hypothesis on both premises to get two majorizers α and β of $[\![t]\!]_{\mathcal{S}}$ and $[\![u]\!]_{\mathcal{S}}$ which satisfy the property. Then, the morphism $[\![tu]\!]_{\mathcal{S}}$ is obtained by composing $[\![t]\!]_{\mathcal{S}} \otimes [\![u]\!]_{\mathcal{S}}$ with the evaluation morphism. Since eval is majorized by $0_{\mathcal{M}}$, $[\![tu]\!]_{\mathcal{S}}$ is majorized by $\alpha + \beta$. Then,

$$\begin{aligned} |\alpha + \beta| &\leq \max(|\alpha|, |\beta|) \\ &\leq \max(\mathsf{rank}(\Gamma \vdash t : \sigma \to \tau) + 2, \mathsf{rank}(\Delta \vdash u : \sigma) + 2) \\ &= \max(\mathsf{rank}(\Gamma \vdash t : \sigma \to \tau), \mathsf{rank}(\Delta \vdash u : \sigma)) + 2 \\ &= \mathsf{rank}(\Gamma, \Delta \vdash tu : \tau) + 2 \end{aligned}$$

  and

$$\begin{aligned} \max(\alpha + \beta) + |\Gamma, \Delta| &\leq \max(\alpha) + |\Gamma| + \max(\beta) + |\Delta| \\ &\leq |t| + |u| \\ &\leq |tu| \end{aligned}$$

- LAM: $\Gamma \vdash \lambda x.\, t : \sigma \to \tau$ comes from $\Gamma, x : \sigma \vdash t : \tau$.
  If $x \in \mathsf{FV}(t)$, the induction hypothesis gives a majorizer α for $[\![t]\!]_{\mathcal{S}} : [\![\Gamma]\!]_{\mathcal{S}} \otimes [\![\sigma]\!]_{\mathcal{S}} \to [\![\tau]\!]_{\mathcal{S}}$. By currying, we obtain a morphism $[\![\lambda x.\, \tau]\!]_{\mathcal{S}} : [\![\Gamma]\!]_{\mathcal{S}} \to [\![\sigma]\!]_{\mathcal{S}} \multimap [\![\tau]\!]_{\mathcal{S}}$ which is also majorized by α. Then $|\alpha| \leq \mathsf{rank}(\Gamma, x : \sigma \vdash t : \tau) + 2 = \mathsf{rank}(\Gamma \vdash \lambda x.\, t : \sigma \to \tau) + 2$, and $\max(\alpha) + |\Gamma| \leq \max(\alpha) + |\Gamma, x : \sigma| - 1 \leq |t| - 1 \leq |\lambda x.\, t|$, as required.
  If $x \notin \mathsf{FV}(t)$, the rule is immediately followed by a weakening rule whose premise is $\Gamma \vdash t : \tau$. The induction hypothesis on this premise gives us a majorizer α for $[\![t]\!]_{\mathcal{S}} : [\![\Gamma]\!]_{\mathcal{S}} \to [\![\tau]\!]_{\mathcal{S}}$. By weakening and currying, we get a morphism $[\![\lambda x.\, \tau]\!]_{\mathcal{S}} : [\![\Gamma]\!]_{\mathcal{S}} \to [\![\sigma]\!]_{\mathcal{S}} \multimap [\![\tau]\!]_{\mathcal{S}}$ which is also majorized by α. The two conditions on the size and the max still hold: $|\alpha| \leq \mathsf{rank}(\Gamma \vdash t : \tau) + 2 = \mathsf{rank}(\Gamma \vdash \lambda x.\, t : \sigma \to \tau) + 2$ and $\max(\alpha) + |\Gamma| \leq |t| \leq |\lambda x.\, t|$.
- CONTR: $\Gamma, z : \sigma \vdash t[x, y \leftarrow z] : \tau$ comes from $\Gamma, x : \sigma, y : \sigma \vdash t : \tau$.
  Both x and y are necessarily in $\mathsf{FV}(t)$: otherwise, it would need to be weakened later, and we assumed this does not happen. So we can use the induction hypothesis to get a majorizer α of $[\![t]\!]_{\mathcal{S}} : [\![\Gamma]\!]_{\mathcal{S}} \otimes [\![\sigma]\!]_{\mathcal{S}} \otimes [\![\sigma]\!]_{\mathcal{S}} \to [\![\tau]\!]_{\mathcal{S}}$. To obtain $[\![t[x, y \leftarrow z]]\!]_{\mathcal{S}}$, we compose $[\![t]\!]_{\mathcal{S}}$ with the duplication morphism $[\![\sigma]\!]_{\mathcal{S}} \to [\![\sigma]\!]_{\mathcal{S}} \otimes [\![\sigma]\!]_{\mathcal{S}}$. Thus, it is majorized by $\alpha + \varphi_n$, where $n = \mathrm{Ord}(\sigma)$ and z $\varphi_n = [0, 0, 1, \ldots, 1]$ of size $n + 2$.
  Then,

$$\begin{aligned} |\alpha + \varphi_n| &\leq \max(|\alpha|, |\varphi_n|) \\ &\leq \max(\mathsf{rank}(\Gamma, x : \sigma, y : \sigma \vdash t : \tau) + 2, n + 2) \\ &= \max(\mathsf{rank}(\Gamma, x : \sigma, y : \sigma \vdash t : \tau), \mathrm{Ord}(\sigma)) + 2 \\ &= \mathsf{rank}(\Gamma, x : \sigma \vdash t[x, y \leftarrow z] : \tau) + 2 \end{aligned}$$

  and

$$\max(\alpha + \varphi_n) + |\Gamma, z : \sigma| \leq \max(\alpha) + 1 + \left|\Gamma, x : \sigma, y : \sigma\right| - 1 \leq |t|$$

- WEAK: this case is not possible since weakenings only occur after a lambda rule: it cannot be at the root. □

https://doi.org/10.1017/S0960129521000256 Published online by Cambridge University Press

We can finally apply Lemma 6 to prove Claim 1:

Claim 1. For every closed term $t : o$, we have $[\![t]\!] \leq \mathbf{2}^{|t|}_{\mathsf{rank}(t)+1}$.

Proof. Applying Lemma 6 to a closed term $t : o$ gives a majorizer α of $[\![t]\!]_{\mathcal{S}}$ with $\max(\alpha) \leq |t|$ and $|\alpha| \leq \mathsf{rank}(t) + 2$. By Lemma 2, this means that $[\![t]\!] \leq a_0$, where $\mathrm{collapse}_0(\alpha) = [a_0]$. And by Lemma 5, we deduce the desired upper bound, $[\![t]\!] \leq \mathbf{2}^{|t|}_{\mathsf{rank}(t)+1}$. □

Note that with the same reasoning, this bound also applies to the constants that appear in Claims 1 and 2. As can be seen in the Examples below, this bound is far from being tight: there is roughly one spare level in the tower of exponents of 2. This is because powers of 2 are hardwired into our model, while λ-terms using the church encoding $\bar{3}$ might compute a tower of 3's instead.

Example 4.

(a) $t = S(S(\ldots(S0)))$, with k occurrences of S. The size of t is $|t| = k$ and its rank is $\mathsf{rank}(t) = 0$, so our bound gives $[\![t]\!] \leq 2^{k+1}$, while its real denotation is $[\![t]\!] = k$.

(b) $u = \bar{2}(\bar{2}(\ldots(\bar{2}S)))0$, with k occurrences of $\bar{2}$. The size of u is $|u| = 3k + 1$ and its rank is $\mathsf{rank}(u) = 1$, so our bound gives $[\![u]\!] \leq 2^{2^{3k+1}}$, while the denotation is $[\![u]\!] = 2^k$.

(c) $v = \bar{2}\bar{2}\ldots\bar{2}S0$, with k occurrences of $\bar{2}$. The size of v is $|v| = 3k + 1$ and its rank is $\mathsf{rank}(v) = k$, so our bound gives $[\![v]\!] \leq \mathbf{2}^{3k+1}_{k+1}$, while the denotation is $[\![v]\!] = \mathbf{2}^1_k$.

(d) $w = \bar{3}\bar{3}\ldots\bar{3}S0$, with k occurrences of $\bar{3}$. The size of w is $|w| = 4k + 1$ and its rank is $\mathsf{rank}(w) = k$, so our bound gives $[\![w]\!] \leq \mathbf{2}^{4k+1}_{k+1}$, while the denotation is $[\![w]\!] = \mathbf{3}^1_k$.

6. Conclusion

We have shown a new use-case of Dal Lago and Hofmann's semantic framework based on resource monoids. Our resource monoid is new in two aspects: unlike other resource monoids found in the literature, which are concerned with time complexity bounds, our model measures the potential of a λ-terms for producing large natural numbers. The second difference with previous instances of resource monoids is that it is able to model the contraction rule of λ-calculus, by bounding the combinatorial explosion caused by duplication of variables.

It would be interesting to try to extend this model to tackle programming languages with more computational power such as Gödel's System T. Another motivating research direction would be to find other use-cases of resource monoids, measuring yet another kind of quantitative property of programs, such as space complexity or probability.

Acknowledgements. The authors would like to thank Robert Atkey, Lê Thành Dũng (Tito) Nguyên, and Igor Walukiewicz for helpful discussions and comments. We also thank the anonymous referees for their valuable comments and pointers to the literature.

Financial support. This research received no specific grant from any funding agency, commercial, or not-for-profit sectors.

Competing interests. The authors declare none.

References

Beckmann, A. (2001). Exact bounds for lengths of reductions in typed λ-calculus. *Journal of Symbolic Logic* **66** (3) 1277–1285.

Brunel, A. (2015). Quantitative classical realizability. *Information and Computation* **241** 62–95.

Brunel, A. and Terui, K. (2010). Church => scott = ptime: An application of resource sensitive realizability. In: *Proceedings International Workshop on Developments in Implicit Computational complExity, DICE 2010*, 31–46.

https://doi.org/10.1017/S0960129521000256 Published online by Cambridge University Press

Dal Lago, U. and Hofmann, M. (2005). Quantitative models and implicit complexity. In: *Proceedings of Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2005*, 189–200.
Dal Lago, U. and Hofmann, M. (2010a). Bounded linear logic, revisited. *Logical Methods in Computer Science* **6** (4).
Dal Lago, U. and Hofmann, M. (2010b). A semantic proof of polytime soundness of light affine logic. *Theory of Computing Systems* **46** (4) 673–689.
Dal Lago, U. and Hofmann, M. (2011). Realizability models and implicit complexity. *Theoretical Computer Science* **412** (20) 2029–2047.
Fortune, S., Leivant, D. and O'Donnell, M. (1983). The expressiveness of simple and second-order type structures. *Journal of ACM* **30** (1) 151–185.
Hillebrand, G. G. and Kanellakis, P. C. (1996). On the expressive power of simply typed and let-polymorphic lambda calculi. In: *Proceedings, 11th Annual IEEE Symposium on Logic in Computer Science, New Brunswick, New Jersey, USA, July 27-30, 1996*, IEEE Computer Society, 253–263.
Joly, T. (2001). Constant time parallel computations in lambda-calculus. *Theoretical Computer Science* **266** (1–2) 975–985.
Nguyên, L. T. D. 2019. Typed lambda-calculi and superclasses of regular functions.
Nguyên, L. T. D., NoÛs, C. and Pradic, P. (2020). Implicit automata in typed λ-calculi II: Streaming transducers vs categorical semantics.
Nguyên, L. T. D. and Pradic, P. (2020). Implicit automata in typed λ-calculi I: aperiodicity in a non-commutative logic. In: *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020*, 135:1–135:20.
Schwichtenberg, H. (1975). Definierbare funktionen im λ-kalkÜl mit typen. *Archive for Mathematical Logic* **17** (3–4) 113–114.
Schwichtenberg, H. (1982). Complexity of normalization in the pure typed lambda-calculus. In: Troelstra, A. and van Dalen, D. (eds.) *The L. E. J. Brouwer Centenary Symposium*, vol. 110, Studies in Logic and the Foundations of Mathematics, Elsevier, 453–457.
Simmons, H. (2005). Tiering as a recursion technique. *The Bulletin of Symbolic Logic* **11** (3) 321–350.

Cite this article: Hofmann M and Ledent J (2021). A quantitative model for simply typed **λ**-calculus. *Mathematical Structures in Computer Science*. https://doi.org/10.1017/S0960129521000256