HAL

archives-ouvertes.fr

Counting Terms in the Binary Lambda Calculus

Katarzyna Grygiel, Pierre Lescanne

COUNTING TERMS IN THE BINARY LAMBDA CALCULUS

KATARZYNA GRYGIEL[†]
AND
PIERRE LESCANNE[†,‡]

[†]JAGIELLONIAN UNIVERSITY,
FACULTY OF MATHEMATICS AND COMPUTER SCIENCE,
THEORETICAL COMPUTER SCIENCE DEPARTMENT,
UL. PROF. ŁOJASIEWICZA 6, 30-348 KRAKÓW, POLAND

[‡]UNIVERSITY OF LYON,
ÉCOLE NORMALE SUPÉRIEURE DE LYON,
LIP (UMR 5668 CNRS ENS LYON UCBL INRIA)
46 ALLÉE D'ITALIE, 69364 LYON, FRANCE

ABSTRACT. In a paper entitled *Binary lambda calculus and combinatory logic*, John Tromp presents a simple way of encoding lambda calculus terms as binary sequences. In what follows, we study the numbers of binary strings of a given size that represent lambda terms and derive results from their generating functions, especially that the number of terms of size n grows roughly like 1.963447954^n.

Keywords: lambda calculus, combinatorics, functional programming, test, random generator, ranking, unranking

1. INTRODUCTION

In recent years growing attention has been given to quantitative research in logic and computational models. Investigated objects (e.g., propositional formulae, tautologies, proofs, programs) can be seen as combinatorial structures, providing therefore the inspiration for combinatorists and computer scientists. In particular, several works have been devoted to studying properties of lambda calculus terms. On a practical point of view, generation of random lambda terms is the core of debugging functional programs using random tests [3] and the present paper offers an answer to a open question (see introduction of [3]) since we are able to generate closed typable terms following a uniform distribution. This work applies beyond λ-calculus to any system with bound variables, like first order predicate calculus (quantifiers are binders like λ) or block structures in programming languages.

First traces of the combinatorial approach to lambda calculus date back to the work of Jue Wang [19], who initiated the idea of enumerating λ-terms. In her report, Wang defined the size of a term as the total number of abstractions, applications

and occurrences of variables, which corresponds to the number of all vertices in the tree representing the given term.

This size model, although natural from the combinatorial viewpoint, turned out to be difficult to handle. The question that arises immediately concerns the number of λ-terms of a given size. This non-trivial task has been done by Bodini, Gardy, and Gittenberger in [1, 2] and Lescanne in [14].

The approach applied in the latter paper has been extended in [9] by the authors of the current paper to the model in which applications and abstractions are the only ones that contribute to the size of a λ-term. The same model has been studied in [4] by David et al., where several properties satisfied by random λ-terms are provided.

When dealing with the two described models, it is not difficult to define recurrence relations for the number of λ-terms of a given size. However, by applying standard tools of the theory of generating functions one obtains generating functions that are expressed as infinite sequences of radicals. Moreover, the radii of convergence are in both cases equal to zero, which makes the analysis of those functions very difficult to cope with.

In this paper, we study the binary encoding of lambda calculus introduced by John Tromp in [18]. This representation results in another size model. It comes from the binary lambda calculus he defined in which he builds a minimal self interpreter of lambda calculus[1] as a basis of algorithmic complexity theory [15]. Set as a central question of theoretical computer science and mathematics, this approach is also more realistic for functional programming. Indeed for compiler builders it is counter-intuitive to assign the same size to all the variables, because in the translation of a program written in Haskell, Ocaml or LISP variables are put in a stack. A variable deep in the stack is not as easily reachable as a variable shallow in the stack. Therefore the weight of the former should be larger than the weight of the latter. Hence it makes sense to associate a size with a variable proportional to its distance to its binder. In this model, recurrence relations for the number of terms of a given size are built using this specific notion of size. From that, we derive corresponding generating functions defined as infinitely nested radicals. However, this time the radius of convergence is positive and allows us for further analysis of the functions. We are able to compute the exact asymptotics for the number of all (not necessarily closed) terms and we also prove the approximate asymptotics for the number of closed ones. Moreover, we define an unranking function, i.e., a generator of terms from their indices from which we derive a uniform generator of λ-terms (general and typable). This allows us to provide outcomes of computer experiments in which we estimate the number of simply typable λ-terms of a given size.

2. Lambda calculus and its binary representation

Lambda calculus is a model of computation that is equivalent to Turing machines or recursive functions, serving as a powerful tool in the development of the programming theory [16]. Furthermore, it constitutes the basis for functional programming languages and has many applications in automated theorem provers.

Basic objects of the lambda calculus are λ-terms, which are regarded as denotation for functions or computer programs. Given a countable infinite set of variables

---

[1]an alternative to universal Turing machine

V, we define lambda terms by the following grammar:

$$M := V \mid \lambda V.M \mid (MM).$$

A term of the form $\lambda x.M$ is called an abstraction. Each occurrence of x in M is called bound. We say that a variable x is free in a term N if it is not bound by an enclosing abstraction. A term with no free variable is called closed. Two terms are considered equivalent if they are identical up to renaming of bound variables.

In order to eliminate names of variables from the notation of a λ-term, de Bruijn introduced an alternative way of representing equivalent terms. Instead of variables we are given now a set of de Bruijn indices $\{\underline{1}, \underline{2}, \underline{3}, \ldots\}$. Given a closed λ-term, we form the corresponding de Bruijn term as follows: an abstraction $\lambda x.M$ is now written as $\lambda\underline{M}$, where $\underline{M}$ is the result of substituting each occurrence of x by the index $\underline{n}$, where n is the number of λ's enclosing the given occurrence of x; an application MN is simply replaced by $\underline{MN}$.

Following John Tromp, we define the binary representation of de Bruijn indices in the following way:

$$
\begin{aligned}
\widehat{\lambda\underline{M}} &= 00\widehat{\underline{M}}, \\
\widehat{\underline{M}\,\underline{N}} &= 01\widehat{\underline{M}}\widehat{\underline{N}}, \\
\widehat{\underline{i}} &= 1^i 0.
\end{aligned}
$$

However, notice that unlike Tromp [18] and Lescanne [13], we start the de Bruijn indices at 1 like de Bruijn [5]. Given a λ-term, we define its size as the length of the corresponding binary sequence, i.e.,

$$
\begin{aligned}
|\underline{n}| &= n+1, \\
|\lambda M| &= |M| + 2, \\
|M\,N| &= |M| + |N| + 2.
\end{aligned}
$$

In contrast to previously studied models, the number of all (not necessarily closed) λ-terms of a given size is always finite. This is due to the fact that the size of each variable depends on the distance from its binder.

3. Combinatorial facts

In order to determine the asymptotics of the number of all/closed λ-terms of a given size, we will use the following combinatorial notions and results.

We say that a sequence $(F_n)_{n \geq 0}$ is of

- order G_n, for some sequence $(G_n)_{n \geq 0}$ (with $G_n \neq 0$), if

$$\lim_{n \to \infty} F_n / G_n = 1,$$

and we denote this fact by $F_n \sim G_n$;
- exponential order A^n, for some constant A, if

$$\limsup_{n \to \infty} |F_n|^{1/n} = A,$$

and we denote this fact by $F_n \bowtie A^n$.

Given the generating function $F(z)$ for the sequence $(F_n)_{n \geq 0}$, we write $[z^n]F(z)$ to denote the n-th coefficient of the Taylor expansion of $F(z)$, therefore $[z^n]F(z) = F_n$.

The theorems below (Theorem IV.7 and Theorem VI.1 of [8]) serve as powerful tools that allow to estimate coefficients of certain functions that frequently appear in combinatorial considerations.

Fact 1. *If $F(z)$ is analytic at 0 and R is the modulus of a singularity nearest to the origin, then*

$$[z^n]F(z) \bowtie (1/R)^n.$$

Fact 2. *Let α be an arbitrary complex number in $\mathbb{C} \setminus \mathbb{Z}_{\leq 0}$. The coefficient of z^n in*

$$f(z) = (1 - z)^\alpha$$

admits the following asymptotic expansion:

$$[z^n]f(z) \quad \sim \quad \frac{n^{\alpha-1}}{\Gamma(\alpha)} \left(1 + \frac{\alpha(\alpha-1)}{2n} + \frac{\alpha(\alpha-1)(\alpha-2)(3\alpha-1)}{24n^2} + O\left(\frac{1}{n^3}\right) \right),$$

where Γ is the Euler Gamma function.

4. THE SEQUENCES $S_{m,n}$

Let us denote the number of λ-terms of size n with at most m distinct free indices by $S_{m,n}$.

First, let us notice that there are no terms of size 0 and 1. Let us consider a λ-term of size $n + 2$ with at most m distinct free variables. Then we have one of the following cases.

- The term is a de Bruijn index $\underline{n + 1}$, provided m is greater than or equal to $n + 1$.
- The term is an abstraction whose binary representation is given by $00\widehat{M}$, where the size of M is n and M has at most $m + 1$ distinct free variables.
- The term is an application whose binary representation is given by $01\widehat{M}\widehat{N}$, where M is of size i and N is of size $n - i$, with $i \in \{0, \ldots, n\}$, and both terms have at most m distinct free variables.

This leads to the following recursive formula[2]:

$$\text{(1)} \qquad S_{m,0} \;=\; S_{m,1} \;=\; 0,$$

$$\text{(2)} \qquad S_{m,n+2} \;=\; [m \geq n + 1] + S_{m+1,n} + \sum_{k=0}^{n} S_{m,k} S_{m,n-k}.$$

The sequence $S_{0,n}$, i.e., the sequence of numbers of closed λ-terms of size n, can be found in the *On-line Encyclopedia of Integer Sequences* under the number **A114852**. Its first 20 values are as follows:

$$0, \; 0, \; 0, \; 0, \; 1, \; 0, \; 1, \; 1, \; 2, \; 1, \; 6, \; 5, \; 13, \; 14, \; 37, \; 44, \; 101, \; 134, \; 298, \; 431.$$

Now let us define the family of generating functions for sequences $(S_{m,n})_{n \geq 0}$:

$$\mathbb{S}_m(z) \quad = \quad \sum_{n=0}^{\infty} S_{m,n} \, z^n.$$

---

[2]Given a predicate P, $[P(\vec{x})]$ denotes the Iverson symbol, i.e., $[P(\vec{x})] = 1$ if $P(\vec{x})$ and $[P(\vec{x})] = 0$ if $\neg P(\vec{x})$.

Most of all, we are interested in the generating function for the number of closed terms, i.e.,

$$\mathbb{S}_0(z) = \sum_{n=0}^{\infty} S_{0,n} z^n.$$

Applying the recurrence on $S_{m,n}$, we get

$$
\begin{aligned}
\mathbb{S}_m(z) &= z^2 \sum_{n=0}^{\infty} S_{m,n+2} z^n \\
&= z^2 \sum_{n=0}^{\infty} [m \geq n+1] z^n + z^2 \sum_{n=0}^{\infty} S_{m+1,n} z^n + z^2 \sum_{n=0}^{\infty} \sum_{k=0}^{n} S_{m,k} S_{m,n-k} z^n \\
&= z^2 \sum_{k=0}^{m-1} z^k + z^2 \mathbb{S}_{m+1}(z) + z^2 \mathbb{S}_m(z)^2 \\
&= \frac{z^2 (1-z^m)}{1-z} + z^2 \mathbb{S}_{m+1}(z) + z^2 \mathbb{S}_m(z)^2.
\end{aligned}
$$

Solving the equation

$$(3) \qquad z^2 \mathbb{S}_m(z)^2 - \mathbb{S}_m(z) + \frac{z^2 (1-z^m)}{1-z} + z^2 \mathbb{S}_{m+1}(z) = 0$$

gives us

$$(4) \qquad \mathbb{S}_m(z) = \frac{1 - \sqrt{1 - 4z^4 \left(\frac{1-z^m}{1-z} + \mathbb{S}_{m+1}(z) \right)}}{2z^2}.$$

This means that the generating function $\mathbb{S}_m(z)$ is expressed by means of infinitely many nested radicals, a phenomenon which has already been encountered in previous research papers on enumeration of lambda terms, see e.g., [1]. However, in Tromp's binary lambda calculus we are able to provide more results than in other representations of lambda terms.

First of all, let us notice that the number of lambda terms of size n has to be less than 2^n, the number of all binary sequences of size n. This means that in the considered model of lambda terms the radius of convergence of the generating function enumerating closed lambda terms is positive (even larger that $1/2$), which is not the case in other models, where the radius of convergence is equal to zero.

5. The number of all λ-terms

Let us now consider the sequence enumerating all binary λ-terms, i.e., including terms that are not closed. Let $S_{\infty,n}$ denote the number of all such terms of size n. Repeating the reasoning from the previous section, we obtain the following recurrence relation:

$$
\begin{aligned}
S_{\infty,0} &= S_{\infty,1} = 0, \\
S_{\infty,n+2} &= 1 + S_{\infty,n} + \sum_{k=0}^{n} S_{\infty,k} S_{\infty,n-k}.
\end{aligned}
$$

The sequence $(S_{\infty,n})_{n\in\mathbb{N}}$ can be found in *On-line Encyclopedia of Integer Sequences* with the entry number **A114851**. Its first 20 values are as follows:

0, 0, 1, 1, 2, 2, 4, 5, 10, 14, 27, 41, 78, 126, 237, 399, 745, 1292, 2404, 4259.

Obviously, we have $S_{m,n} \leq S_{\infty,n}$ for every $m,n \in \mathbb{N}$. Moreover, $\lim_{m\to\infty} S_{m,n} = S_{\infty,n}$.

Let $\mathbb{S}_\infty(z)$ denote the generating function for the sequence $(S_{\infty,n})_{n\in\mathbb{N}}$, that is

$$\mathbb{S}_\infty(z) = \sum_{n=0}^{\infty} S_{\infty,n} z^n.$$

Notice that for $m \geq n-1$ we have $S_{m,n} = S_{\infty,n}$. Therefore

$$\mathbb{S}_\infty(z) = \sum_{n=1}^{\infty} S_{n,n} z^n,$$

which yields that $[z^n]\mathbb{S}_{n,n} = [z^n]\mathbb{S}_{\infty,n}$. Furthermore, $\mathbb{S}_\infty(z) = \lim_{m\to\infty} \mathbb{S}_m(z)$.

Theorem 1. *The number of all binary λ-terms of size n satisfies*

$$S_{\infty,n} \sim (1/\rho)^n \cdot \frac{C}{n^{3/2}},$$

where $\rho \doteq 0.509308127$ and $C \doteq 1.021874073$.

Proof. The generating function $\mathbb{S}_\infty(z)$ fulfills the equation

$$\mathbb{S}_\infty(z) = \frac{z^2}{1-z} + z^2\mathbb{S}_\infty(z) + z^2\mathbb{S}_\infty(z).$$

Solving the above equation gives us

$$\mathbb{S}_\infty(z) = \frac{z^3 - z^2 - z + 1 - \sqrt{z^6 + 2\,z^5 - 5\,z^4 + 4\,z^3 - z^2 - 2\,z + 1}}{2z^2(1-z)}.$$

The dominant singularity of the function $\mathbb{S}_\infty(z)$ is given by the root of smallest modulus of the polynomial

$$R_\infty(z) = z^6 + 2\,z^5 - 5\,z^4 + 4\,z^3 - z^2 - 2\,z + 1.$$

The polynomial has four real roots:

$$0.509308127, \quad -0.623845142, \quad 1, \quad -3.668100004,$$

and two complex ones that are approximately equal to $0.4 + 0.8i$ and $0.4 - 0.8i$.

Therefore $\rho \doteq 0.509308127$ is the singularity of $\mathbb{S}_\infty$ nearest to the origin. Let us write $\mathbb{S}_\infty(z)$ in the following form:

$$\mathbb{S}_\infty(z) = \frac{1 - z^2 - \sqrt{\rho(1 - \frac{z}{\rho}) \cdot \frac{Q(z)}{1-z}}}{2z^2},$$

where $Q(z) = \frac{R_\infty(z)}{(\rho-z)(1-z)}$ is the polynomial defined for all $|z| \leq \rho$.

We get that the radius of convergence of $\mathbb{S}_\infty(z)$ is equal to ρ and its inverse $\frac{1}{\rho} \doteq 1.963447954$ gives the growth of $S_{\infty,n}$. Hence, $S_{\infty,n} \bowtie (1/\rho)^n$.

Fact 2 allows us to determine the subexponential factor of the asymptotic estimation of the number of terms. Applying it, we obtain that

$$[z^n]\mathbb{S}_\infty(z) \sim \left(\frac{1}{\rho}\right)^n \cdot \frac{n^{-3/2}}{\Gamma(-\frac{1}{2})} \cdot \widetilde{C},$$

where the constant $\widetilde{C}$ is given by

$$\widetilde{C} = \frac{-\sqrt{\rho \cdot \frac{Q(\rho)}{1-\rho}}}{2\rho^2} \doteq -0.288265354.$$

Since $\dfrac{\widetilde{C}}{\Gamma(-\frac{1}{2})} \doteq 1.021874073$, the theorem is proved. $\qquad\square$

6. THE NUMBER OF CLOSED λ-TERMS

Proposition 1. *Let ρ_m denote the dominant singularity of $\mathbb{S}_m(z)$. Then for every natural number m we have*

$$\rho_m = \rho_0,$$

which means that all functions $\mathbb{S}_m(z)$ have the same dominant singularity.

Proof. First, let us notice that for every $m, n \in \mathbb{N}$ we have $S_{m,n} \leq S_{m+1,n}$. This means that the radius of convergence of the generating function for the sequence $(S_{m,n})_{n \in \mathbb{N}}$ is not smaller that the radius of convergence of the generating function for $(S_{m+1,n})_{n \in \mathbb{N}}$. Therefore, for every natural number m, we have

$$\rho_m \geq \rho_{m+1}.$$

On the other hand, from Equation 4 we see that every singularity of $\mathbb{S}_{m+1}(z)$ is also a singularity of $\mathbb{S}_m(z)$. Hence, the dominant singularity of $\mathbb{S}_m(z)$ is less than or equal to the dominant singularity of $\mathbb{S}_{m+1}(z)$, i.e., we have

$$\rho_m \leq \rho_{m+1}.$$

These two inequalities show that dominant singularities of all functions $\mathbb{S}_m(z)$ are the same. In particular, for every m we have $\rho_m = \rho_0$. $\qquad\square$

Proposition 2. *The dominant singularity of $\mathbb{S}_0(z)$ is equal to the dominant singularity of $\mathbb{S}_\infty(z)$, i.e.,*

$$\rho_0 \;=\; \rho \;\doteq\; 0.509308127.$$

Proof. Since the number of closed binary λ-terms is not greater than the number of all binary terms of the same size, we conclude immediately that $\rho_0 \geq \rho$.

Let us now consider the functionals

$$\Phi_m(F) \;=\; \frac{1 - \sqrt{1 - 4z^4(\frac{1-z^m}{1-z} + F)}}{2z^2},$$

$$\Phi_\infty(F) \;=\; \frac{1 - \sqrt{1 - 4z^4(\frac{1}{1-z} + F)}}{2z^2}.$$

In particular, when $m = 0$, we have

$$\Phi_0(F) \;=\; \frac{1 - \sqrt{1 + 4z^4 F}}{2z^2}.$$

We have also

$$\mathbb{S}_m(z) \;=\; \Phi_m(\mathbb{S}_{m+1}(z)).$$

The Φ_m's and Φ_∞ are increasing over functions over $[0, 1)$, which means that

$$F \leq G \;\Rightarrow\; \Phi_m(F) \leq \Phi_m(G),$$
$$F \leq G \;\Rightarrow\; \Phi_\infty(F) \leq \Phi_\infty(G).$$

For each $m \in \mathbb{N}$, let us consider the function $\widetilde{\mathbb{S}}_m(z)$ defined as the fixed point of Φ_m. In other words, $\widetilde{\mathbb{S}}_m(z)$ is defined as the solution of the following equation:

$$\widetilde{\mathbb{S}}_m(z) = \Phi_m(\widetilde{\mathbb{S}}_m(z)).$$

Notice that since $S_{m,n} \leq S_{m+1,n} \leq S_{\infty,n}$ we can claim that $\mathbb{S}_m(z) \leq \mathbb{S}_{m+1}(z) \leq \mathbb{S}_\infty(z)$. Therefore, we have

$$(5) \qquad \Phi_m(\mathbb{S}_m(z)) \leq \mathbb{S}_m(z),$$

$$(6) \qquad \widetilde{\mathbb{S}}_m(z) \leq \mathbb{S}_m(z) \leq \mathbb{S}_\infty(z).$$

Since $\widetilde{\mathbb{S}}_m(z)$ satisfies

$$2z^2\widetilde{\mathbb{S}}_m(z) = 1 - \sqrt{1 - 4z^4\left(\frac{1-z^m}{1-z} + \widetilde{\mathbb{S}}_m(z)\right)},$$

we get

$$z^2\widetilde{\mathbb{S}}_m^2(z) - (1-z^2)^2\widetilde{\mathbb{S}}_m(z) + \frac{z^2(1-z^m)}{1-z} = 0.$$

The discriminant of this equation is:

$$\Delta_m = (1-z^2)^2 - \frac{4z^4(1-z^m)}{1-z}.$$

The values for which $\Delta_m = 0$ are the singularities of $\widetilde{\mathbb{S}}_m(z)$. Let us denote the main singularity of $\widetilde{\mathbb{S}}_m(z)$ by σ_m. From Equation (6) we see that

$$\sigma_m \geq \rho_m \geq \rho.$$

The value of σ_m is equal to the root of smallest modulus of the following polynomial:

$$P_m(z) := (z-1)\Delta_m = 4z^4(1-z^m) - (1-z)^3(1+z)^2.$$

In the case of the function $\widetilde{\mathbb{S}}_\infty(z)$, we get the polynomial

$$P_\infty(z) = z^5 + 3z^4 - 2z^3 + 2z^2 + z - 1 = \frac{R_\infty(z)}{z-1},$$

whose root of smallest modulus is, like in the case of $R_\infty(z)$, equal to ρ.

Now let us show that the sequence $(\sigma_m)_{m\in\mathbb{N}}$ of roots of polynomials $P_m(z)$ is decreasing (see Figure 1) and that it converges to ρ.

Notice that $P_m(z) = P_\infty(z) - 4z^{m+4}$. Given a value ζ such that $\rho < \zeta < 1$ (for instance $\zeta = 0.8$), $P_m(z)$ converges uniformly to $P_\infty(z)$ in the interval $[0, \zeta]$. Therefore $\sigma_m \to \rho$ when $m \to \infty$. By $\sigma_m \leq \rho_m \leq \rho$, we get $\rho_m \to \rho$, as well. Since all the ρ_m's are equal, we obtain that $\rho_m = \rho$ for every natural m. $\qquad \square$

The above proposition leads immediately to the following result.

Corollary 1. *The number of closed binary λ-terms of size n is of exponential order $(1/\rho)^n$, i.e.,*

$$S_{0,n} \bowtie 1.963448^n.$$

The number of closed terms of a given size cannot be greater than the number of all terms. Therefore, we obtain what follows.

FIGURE 1. Roots of the P_m's

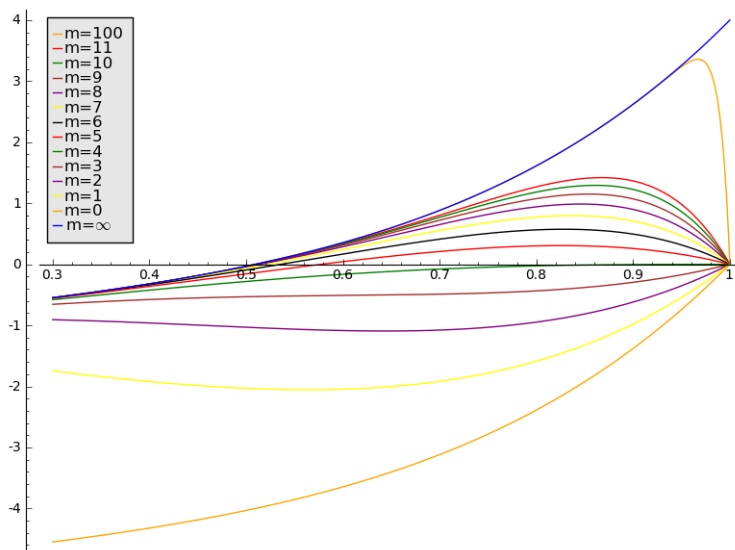**Theorem 2.** *The number of closed binary λ-terms of size n is asymptotically of order*

$$S_{0,n} \sim \left(\frac{1}{\rho}\right)^n \cdot O\big(n^{-3/2}\big) \doteq 1.963448^n \cdot O\big(n^{-3/2}\big).$$

Figure 2 shows values $S_{m,n} \cdot \rho^n \cdot n^{3/2}$ for a few initial values of m and n up to 600.
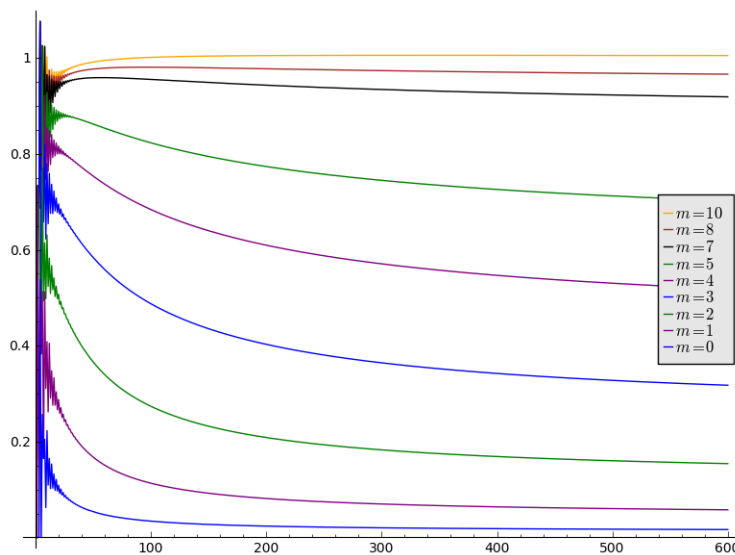

FIGURE 2. $S_{m,n}\rho^n n^{3/2}$ up to $n = 600$ for $m = 0$ to 10

These numerical experiments allow us to state the following conjecture.

Conjecture 1. *For every natural number m, we have*

$$S_{m,n} \sim 1.963448^n \cdot o\big(n^{-3/2}\big).$$

7. Unrankings

The recurrence relation (2) for $S_{m,n}$ allows us to define the function generating λ-terms. More precisely, we construct bijections $s_{m,n}$, called *unranking* functions, between all non-negative integers not greater than $S_{m,n}$ and binary λ-terms of size n with at most m distinct free variables [7]. This approach is also known as the *recursive method*, originating with Nijenhuis and Wilf [17] (see especially Chapter 13). In order to describe unranking functions, we make use of the Cantor pairing function.

Let us recall that for $n \geq 2$ we have, by (2),

$$S_{m,n} \;\;=\;\; S_{m+1,n-2} + \sum_{j=0}^{n-2} S_{m,j} S_{m,n-2-j} + [m \geq n-1].$$

The encoding function $s_{m,n}$ takes an integer $k \in \{1, \ldots, S_{m,n}\}$ and returns the term built in the following way.

- If $m \geq n-1$ and k is equal to $S_{m,n}$, the function returns the string $1^{n-1}0$.
- If k is less than or equal to $S_{m+1,n-2}$, then the corresponding term is in the form of abstraction $00\widehat{\mathsf{M}}$, where $\widehat{\mathsf{M}}$ is the value of the unranking function $s_{m+1,n-2}$ on k.
- Otherwise (i.e., k is greater than $S_{m+1,n-2}$ and less than $S_{m,n}$ if $m \geq n+1$ or less than or equal to $S_{m,n}$ if $m < n+1$) then the corresponding term is in the form of application $01\widehat{\mathsf{M}}\widehat{\mathsf{N}}$. In order to get strings $\widehat{\mathsf{M}}$ and $\widehat{\mathsf{N}}$, we compute the maximal value $\ell \in \{0, \ldots, n-2\}$ for which

$$k - S_{m+1,n-2} = \sum_{j=0}^{\ell-1} S_{m,j} S_{m,n-2-j} + r \qquad \text{with } r \leq S_{m,\ell} S_{m,n-2-\ell}.$$

  The strings $\widehat{\mathsf{M}}$ and $\widehat{\mathsf{N}}$ are the values $s_{m,\ell}(k')$ and $s_{m,n-2-\ell}(k'')$, respectively, where (k', k'') is the pair of integers encoded by r by the Cantor pairing function.

In Figure 3 the reader may find a Haskell program [12] which computes the values $s_{m,n}(k)$. In this program, the function $s_{m,n}(k)$ is written as `unrankT m n k` and the sequence $S_{m,n}$ is written as `tromp m n`.

8. Number of typable terms

The unranking function allows us to traverse all the closed terms of size n and to filter those that are typable (see [11] and appendix) in order to count them and similarly to traverse all the terms of size n to count those that are typable. Figure 4 left gives the number $T_{0,n}$ of closed typable terms of size n and Figure 4 right gives the number $T_{\infty,n}$ of all typable terms of size n.

Thanks to the unranking function, we can build a *uniform generator of λ-terms* and, using this generator, we can build a *uniform generator of simply typable λ-terms*, which works by sieving the uniformly generated plain terms through a program that checks their typability (see for instance [9]). This way, it is possible to

```
unrankT :: Int -> Int -> Integer -> Term
unrankT m n k
  | m >= n - 1 && k == (tromp m n) = Index $ fromIntegral (n - 1) -- terms 1^{n-1}0
  | k <= (tromp  (m+1) (n-2)) = Abs (unrankT (m+1) (n-2) k) -- terms 00M
  | otherwise = unrankApp (n-2) 0 (k - tromp  (m+1) (n-2)) -- terms 01MN
    where unrankApp n j h
            | h <=  tmjtmnj  = let (dv,rm) = (h-1) `divMod` tmnj
                                  in App (unrankT m j (dv+1)) (unrankT  m (n-j) (rm+1))
            | otherwise = unrankApp n (j + 1) (h -tmjtmnj)
          where tmnj = tromp m (n-j)
                tmjtmnj = (tromp m j) * tmnj
```

FIGURE 3. A Haskell program for computing values of the function $s_{m,n}$

generate uniformly typable closed terms up to size 450 which is rather good since Tromp was able to build a self interpreter[3] for the λ-calculus of size 210.

9. CONCLUSION

We have shown that if we use the size yielded by the binary lambda calculus [18], we get an exponential growth of the number of λ-terms of size n when n goes to infinity. This applies to closed λ-terms, to λ-terms with a bounded number of free variables, and to all λ-terms of size n. Except for the size of all λ-terms, the question of finding the non-exponential factor of the asymptotic approximation of these numbers is still open. Since the generating functions are not standard, we were lead to devise new methods for computing these approximations. Beside, we describe unranking functions (recursive methods) for generating λ-terms from which we derive tools for their uniform generation and for the enumeration of typable λ-terms. The generation of random (typable) terms is limited by the performance of the generators based on the recursive methods aka unranking which needs to handle huge number. Boltzmann samplers [6] should allow us to generate terms of larger size.

REFERENCES

[1] Olivier Bodini, Danièle Gardy, and Bernhard Gittenberger. Lambda-terms of bounded unary height. *2011 Proceedings of the Eighth Workshop on Analytic Algorithmics and Combinatorics (ANALCO)*, 2011.

[2] Olivier Bodini, Danièle Gardy, Bernhard Gittenberger, and Alice Jacquot. Enumeration of generalized *BCI* lambda-terms. *ArXiv e-prints*, May 2013.

[3] Koen Claessen and John Hughes. QuickCheck: a lightweight tool for random testing of Haskell programs. In Martin Odersky and Philip Wadler, editors, *ICFP*, pages 268–279. ACM, 2000.

[4] René David, Katarzyna Grygiel, Jakub Kozik, Christophe Raffalli, Guillaume Theyssier, and Marek Zaionc. Asymptotically almost all λ-terms are strongly normalizing. *Logical Methods in Computer Science*, 9(1:02):1–30, 2013.

[5] Nicolaas G. de Bruijn. Lambda calculus notation with nameless dummies, a tool for automatic formula manipulation, with application to the Church-Rosser theorem. *Indagationes Mathematicae*, 34(5):381–392, 1972.

[6] Philippe Duchon, Philippe Flajolet, Guy Louchard, and Gilles Schaeffer. Boltzmann samplers for the random generation of combinatorial structures. *Combinatorics, Probability & Computing*, 13(4-5):577–625, 2004.

---

[3]Which is not typable by definition!

n	$T_{0,n}$
0	0
1	0
2	0
3	0
4	1
5	0
6	1
7	1
8	1
9	1
10	5
11	4
12	9
13	13
14	23
15	29
16	67
17	94
18	179
19	285
20	503
21	795
22	1503
23	2469
24	4457
25	7624
26	13475
27	23027
28	41437
29	72165
30	128905
31	227510
32	405301
33	715078
34	1280127
35	2279393
36	4086591
37	7316698
38	13139958
39	23551957
40	42383667
41	76278547
42	137609116
43	248447221
44	449201368
45	812315229
46	1470997501

n	$T_{\infty,n}$
0	0
1	0
2	1
3	1
4	2
5	2
6	3
7	5
8	8
9	13
10	22
11	36
12	58
13	103
14	177
15	307
16	535
17	949
18	1645
19	2936
20	5207
21	9330
22	16613
23	29921
24	53588
25	96808
26	174443
27	316267
28	572092
29	1040596
30	1888505
31	3441755
32	6268500
33	11449522
34	20902152
35	38256759
36	70004696
37	128336318
38	235302612
39	432050796
40	793513690
41	1459062947
42	2683714350

FIGURE 4. Number of typable terms

[7] A. Karttunen et al. Ranking and unranking functions. OEIS Wiki. http://oeis.org/wiki/Ranking_and_unranking_function.

[8] Philippe Flajolet and Robert Sedgewick. *Analytic Combinatorics*. Cambridge University Press, 2008.

[9] Katarzyna Grygiel and Pierre Lescanne. Counting and generating lambda terms. *Journal of Functional Programming*, to appear, 2013.

[10] Katarzyna Grygiel and Pierre Lescanne. Counting terms in the binary lambda calculus. Technical report, Arxiv, 2013.

[11] J. Roger Hindley. *Basic Simple Type Theory*. Number 42 in Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 1997.

[12] Simon Peyton Jones, editor. *Haskell 98 language and libraries: the Revised Report*. Cambridge University Press, 2003.

[13] Pierre Lescanne. From $\lambda\sigma$ to $\lambda\upsilon$, a journey through calculi of explicit substitutions. In Hans Boehm, editor, *Proceedings of the 21st Annual ACM Symposium on Principles Of Programming Languages, Portland (Or., USA)*, pages 60–69. ACM, 1994.

[14] Pierre Lescanne. On counting untyped lambda terms. *Theor. Comput. Sci.*, 474:80–97, 2013.

[15] Ming Li and Paul Vitányi. *An introduction to Kolmogorov complexity and its applications (3rd ed.)*. Springer-Verlag New York, Inc., 2008.

[16] John C. Mitchell. *Foundations for Programming Languages*. MIT Press, sep 1996.

[17] Albert Nijenhuis and Herbert S. Wilf. *Combinatorial algorithms, 2nd edition*. Computer science and applied mathematics. Academic Press, New York, 1978.

[18] John Tromp. Binary lambda calculus and combinatory logic. In Marcus Hutter, Wolfgang Merkle, and Paul M. B. Vitányi, editors, *Kolmogorov Complexity and Applications*, volume 06051 of *Dagstuhl Seminar Proceedings*. Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl, Germany, 2006.

[19] Jue Wang. The efficient generation of random programs and their applications. Honors Thesis, Wellesley College, Wellesley, MA, May 2004.

Appendix A. Types and typability

Types determine whether λ-terms actually represent well-defined functions [11]. Here we focus on simple typable terms, because simple typability is decidable. Simple types are of two forms, either variable types α or arrow types $\sigma \to \tau$:

$$\sigma, \tau \ ::= \ \alpha \mid \sigma \to \tau$$

A context $\Gamma = \tau_n, \ldots, \tau_1$ is a finite sequence of types which correspond to declare that index $\underline{1}$ has type τ_1, index $\underline{2}$ has type τ_2 etc. A type judgment $\Gamma \vdash M : \tau$ says that in the context Γ, the λ-term M has type τ. To type a term we use inference rules:

$$\frac{}{\tau_n, \ldots, \tau_i, \ldots \tau_1 \vdash \underline{i} : \tau_i} Var \qquad \frac{\Gamma, \tau \vdash M : \sigma}{\Gamma \vdash \lambda M : \tau \to \sigma} Abs$$

$$\frac{\Gamma \vdash M : \sigma \to \tau \qquad \Gamma \vdash P : \sigma}{\Gamma \vdash MP : \tau} App$$

Definition 1 (Typability). *A term M is* typable *if there exists a context Γ and type σ such that $\Gamma \vdash M : \sigma$.*

Notice that an open term with n free indices require a context of size n to be typable. Therefore a closed term requires an empty context to be typable. Moreover checking typability is solving constraints, mostly constraints generated by rule *App*. For instance, term $\lambda \underline{11}$ cannot be typed since $\underline{1}$ of type say σ cannot be applied to the term $\underline{1}$ of type σ. For that it should be of type $\sigma \to \tau$. Similarly $\lambda\lambda\underline{211}$ of type $(\alpha \to \alpha \to \beta) \to \alpha \to \beta$ cannot be applied to $\lambda\underline{1}$ of type $\gamma \to \gamma$. Therefore $(\lambda\lambda\underline{211})\,\lambda\underline{1}$ is not typable. We also notice that typability can be described neither recursively nor structurally.

E-mail address: GRYGIEL@TCS.UJ.EDU.PL,PIERRE.LESCANNE@ENS-LYON.FR