Counting proofs in propositional logic

René DAVID[*] Marek ZAIONC[†]

October 13, 2008

Abstract

We give a procedure for counting the number of different proofs of a formula in various sorts of propositional logic. This number is either an integer (that may be 0 if the formula is not provable) or infinite.

1 Introduction

The aim of the paper is to give a procedure for counting the number of different normal proofs of a formula in propositional logic. By the well known Curry Howard correspondence, this is similar to count the number of different normal closed terms of some fixed type in an extension of the $\lambda\mu$ calculus.

We show that this number is the least fix-point of a system of polynomial equations in some natural complete lattice and we give an algorithm for finding such a least fix-point.

The similar problem of counting closed typed lambda terms was studied (see [1]) but never published by Ben- Yelles. Some description of the Ben-Yelles solution can be found in Hindley's book [4]. Similarly Hirokawa in [5] proved that the complexity of the question whether a given simple type (implicational formula) possess an infinite number of normal terms (or infinite number of proofs) is polynomial space complete. Recently similar research about counting λ-calculus objects for program synthesis was done by Wells and Yakobowski in [9].

2 The logic

2.1 Formulae and proofs

Definition 1 *Let $\mathcal{A}$ be a set of atomic constants. The set $\mathcal{F}$ of formulae is defined by the following grammar*

$$\mathcal{F} ::= \mathcal{A} \cup \{\bot\} \mid \mathcal{F} \to \mathcal{F} \mid \mathcal{F} \wedge \mathcal{F} \mid \mathcal{F} \vee \mathcal{F}$$

As usual $\neg F$ will be an abbreviation for $F \to \bot$.

Definition 2 *The rules for proofs in classical logic are the following.*

$$\frac{}{\Gamma, A \vdash A}\, ax$$

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \to B} \to_i \qquad \frac{\Gamma_1 \vdash A \to B \quad \Gamma_2 \vdash A}{\Gamma_1, \Gamma_2 \vdash B} \to_e$$

[*]Lama, Université de Savoie, Campus Scientifique. 73376 Le Bourget du lac. Email : rene.david@univ-savoie.fr

[†]Theoretical Computer Science, Jagiellonian University, Gronostajowa 3, 30-387 Kraków, Poland. Email : zaionc@tcs.uj.edu.pl

$$\frac{\Gamma_1 \vdash A_1 \quad \Gamma_2 \vdash A_2}{\Gamma_1, \Gamma_2 \vdash A_1 \wedge A_2} \wedge_i \qquad \frac{\Gamma \vdash A_1 \wedge A_2}{\Gamma \vdash A_i} \wedge_e$$

$$\frac{\Gamma \vdash A_j}{\Gamma \vdash A_1 \vee A_2} \vee_i \qquad \frac{\Gamma \vdash A_1 \vee A_2 \quad \Gamma_1, A_1 \vdash C \quad \Gamma_2, A_2 \vdash C}{\Gamma, \Gamma_1, \Gamma_2 \vdash C} \vee_e$$

$$\frac{\Gamma, \neg A \vdash \bot}{\Gamma \vdash A} \bot_e \qquad \frac{\Gamma, \neg A \vdash A}{\Gamma, \neg A \vdash \bot} \bot_i$$

2.2 Terms coding proofs

It is well known that a proof, in intuitionistic implicational logic, can be coded by a simply typed λ-term. The same thing can, in fact, be done for proofs, in classical logic, of any kind of formulae. The extension from intuitionistic logic to classical logic is the $\lambda\mu$-calculus introduced by Parigot in [6]. The extension to formulae using all the usual connectors has been introduced by de Groote in [3]. The next definition is a presentation of this calculus.

Definition 3 *Let $\mathcal{V}$ and $\mathcal{W}$ be disjoint sets of variables. The set of $\lambda\mu^{\rightarrow\wedge\vee}$-terms is defined by the following grammar*

$$\mathcal{T} ::= \mathcal{V} \mid \lambda\mathcal{V}.\mathcal{T} \mid (\mathcal{T}\ \mathcal{E}) \mid \langle \mathcal{T}, \mathcal{T} \rangle \mid \omega_1\mathcal{T} \mid \omega_2\mathcal{T} \mid \mu\mathcal{W}.\mathcal{T} \mid (\mathcal{W}\ \mathcal{T})$$

$$\mathcal{E} ::= \mathcal{T} \mid \pi_1 \mid \pi_2 \mid [\mathcal{V}.\mathcal{T}, \mathcal{V}.\mathcal{T}]$$

The next definition shows how the terms introduced in definition 3 code the proofs.

Definition 4 *The typing rules for the $\lambda\mu^{\rightarrow\wedge\vee}$-terms are as follows*

$$\frac{}{\Gamma, x : A \vdash x : A} ax \qquad \frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x.M : A \rightarrow B} \rightarrow_i$$

$$\frac{\Gamma_1 \vdash M : A \rightarrow B \quad \Gamma_2 \vdash N : A}{\Gamma_1, \Gamma_2 \vdash (M\ N) : B} \rightarrow_e$$

$$\frac{\Gamma, \alpha : \neg A \vdash M : A}{\Gamma, \alpha : \neg A \vdash (\alpha\ M) : \bot} \bot_i \qquad \frac{\Gamma, \alpha : \neg A \vdash M : \bot}{\Gamma \vdash \mu\alpha.M : A} \bot_e$$

$$\frac{\Gamma_1 \vdash M : A_1 \quad \Gamma_2 \vdash N : A_2}{\Gamma_1, \Gamma_2 \vdash \langle M, N \rangle : A_1 \wedge A_2} \wedge_i \qquad \frac{\Gamma \vdash M : A_1 \wedge A_2}{\Gamma \vdash (M\ \pi_i) : A_i} \wedge_e$$

$$\frac{\Gamma \vdash M : A_j}{\Gamma \vdash \omega_j M : A_1 \vee A_2} \vee_i$$

$$\frac{\Gamma \vdash M : A_1 \vee A_2 \quad \Gamma_1, x_1 : A_1 \vdash N_1 : C \quad \Gamma_2, x_2 : A_2 \vdash N_2 : C}{\Gamma, \Gamma_1, \Gamma_2 \vdash (M\ [x_1.N_1, x_2.N_2]) : C} \vee_e$$

Remark

Note that, in definition 2, the letter Γ represents a finite multi-set of formulae whereas, in definition 4, it represents a finite multi-set of *indexed* formulae i.e. a finite set of pairs denoted as $x : A$ or $\alpha : \neg A$ where $x \in \mathcal{V}$, $\alpha \in \mathcal{W}$ and $A \in \mathcal{F}$ (where each variable occurs only once).

In the rest of the paper, we will continue to use the same notation for these two formally distinct notions. Such a multi-set will be called a *context*. In a particular sentence which of the two notions is meant will usually be clear ... from the context.

Definition 5 *The set G of goals is the set of ordered pairs denoted as $\Gamma \vdash A$ where $A \in \mathcal{F}$ and Γ is a context.*

2.3 Normal terms and proofs

To avoid to have, for each formula, either zero or infinitely many proofs, we will only consider normal proofs. A proof is normal if the term that represents it is normal. This corresponds of course to the usual notion of cut elimination in natural deduction. The precise definition is given below.

Since every term is normalizing i.e. can be reduced to a normal term (cf. theorem 8, item 1), if a formula has a proof then it also have a normal proof. Thus the restriction does not change the problem.

Some other restrictions will be done on the kind of proofs we are counting. The first ones are necessary (to avoid to have, for each formula, either zero or infinitely many proofs) and will be given and justified in the comments of section 3.4. Other ones will be given in section 3.6.

Definition 6 *The reduction rules for the $\lambda\mu^{\rightarrow\wedge\vee}$-calculus are given below. Variables M, N, L are in $\mathcal{T}$ and ε is in $\mathcal{E}$. A variable x belongs to $\mathcal{V}$ while α is taken from $\mathcal{W}$.*

$$(\lambda x.M\ N) \triangleright_\beta M[x := N]$$
$$(\mu\alpha.M\ N) \triangleright_\mu \mu\alpha.M[(\alpha\ L) := (\alpha\ (L\ N))]$$
$$(\langle M_1, M_2\rangle\ \pi_i) \triangleright M_i$$
$$(\omega_i M\ [x_1.N_1, x_2.N_2]) \triangleright N_i[x_i := M]$$
$$(M\ [x_1.N_1, x_2.N_2]\ \varepsilon) \triangleright (M\ [x_1.(N_1\ \varepsilon), x_2.(N_2\ \varepsilon)])$$
$$(\mu\alpha.M\ \varepsilon) \triangleright \mu\alpha.M[(\alpha\ L) := (\alpha\ (L\ \varepsilon))]$$

Note that we have also considered the so-called *permutative conversions* (the rule $(M\ [x_1.N_1, x_2.N_2]\ \varepsilon) \triangleright (M\ [x_1.(N_1\ \varepsilon), x_2.(N_2\ \varepsilon)])$) and the so-called *classical cuts* (the rule $(\mu\alpha.M\ \varepsilon) \triangleright \mu\alpha.M[(\alpha\ N) := (\alpha\ (N\ \varepsilon))]$) which are necessary to ensure that a normal proof have the sub-formula property (cf. theorem 8, item 2).

Definition 7 *Let t be a $\lambda\mu^{\rightarrow\wedge\vee}$-term and $g = \Gamma \vdash A$ be a goal.*

- *We say that t is a proof of g if $\Gamma \vdash t : A$.*
- *We say that t is normal if it contains no redex i.e. if it cannot be reduced by the rules of definition 6.*

Theorem 8 *Let t be a proof of $g = \Gamma \vdash A$. Then,*

1. *t can be reduced into a normal proof of g.*
2. *If t is normal and B is a formula that occurs in the typing tree of t then, there is a sub-formula C of a formula in $\{A\} \cup \Gamma$ such that $B = C$ or $B = \neg C$.*

Proof Standard. See for example [7] or [8]. $\square$

Theorem 9 *There is an algorithm that, given a formula F, computes the number (i.e. either an integer or ∞) of distinct normal proofs of F.*

Proof This is an immediate corollary of theorem 27 below whose statement and proof is given in the next section. $\square$

3 Proof of the main result

3.1 The idea of the proof

The idea of the proof is quite simple. To each goal g of the form $\Gamma \vdash A$ we associate a variable n_g that, intuitively, gives the number of normal proofs of g. By looking at the possible ways of proving g (either use an introduction rule or an elimination rule or a proof by contradiction) we get equations relating the n_g. We will show that the numbers we are looking for is the minimal solution of this set of equations. The two main technical difficulties are the following.

- We have to be able to compute the solution of these equations. This follows from the fact that they only use integers, addition an multiplication. An addition corresponds to the possibility of proving a goal in different ways. A multiplication means that, to prove the goal, we have to prove two different things. Thus the equations are polynomial and we will show that, for this kind of equations, we can always compute the minimal solution.

- The other point is a bit more difficult. To be able to compute its solution, the set of equations must be finite but, without sufficient care, it is not ! Since, by the sub-formula property (theorem 8 above), we know that the formulae that appear in a normal proof are sub-formulae of the initial formula, the set of goals must, intuitively, be finite (which would imply that the set of equations also is finite) but since, in Γ, a formula can be repeated many times it is not true that the set of goals is finite. To solve this problem, we proceed as follows. When, in a proof of some goal we introduce a new goal, say h, which is the same as a goal h' that has already been introduced except that it adds some hypothesis that were already present in h', we do not consider it as a new one i.e. we do not build an equation for it. This is because we can show that h, h' have the same number of proofs. But, to do that, we need some book keeping because to show that h, h' have the same number of proofs, we need the fact that h and h' are, somehow, in the same part of a proof. This will be ensured by the order we put on the variables n_g. Doing in this way, Konig's lemma ensures that the set of equations is finite.

3.2 Polynomials

Definition 10

- *The set $\mathbb{N} \cup \{\omega\}$ will be denoted as $\overline{\mathbb{N}}$*
- *The usual order and operations on $\mathbb{N}$ are extended to $\overline{\mathbb{N}}$ by*
  - *$i \leq \omega$ and $i + \omega = \omega + i = \omega$ for every $i \in \overline{\mathbb{N}}$,*
  - *$0 \cdot \omega = \omega \cdot 0 = 0$,*
  - *$i \cdot \omega = \omega \cdot i = \omega$ for every $i \neq 0$.*
- *The set $\overline{\mathbb{N}}^k$ is naturally ordered by $(a_1, ..., a_k) \leq (b_1, ..., b_k)$ if $a_i \leq b_i$ for all i.*

Lemma 11 *$\overline{\mathbb{N}}^k$ is a complete lattice.*

Proof Obvious. □

Definition 12

- *The set of polynomials is the least set of functions (of several variables) from $\overline{\mathbb{N}}$ to $\overline{\mathbb{N}}$ that contains the constant functions and is closed by addition and multiplication.*
- *The order on polynomials is the point-wise order, i.e. if $f(x_1, ..., x_n), g(x_1, ..., x_n)$ are polynomials, $f \leq g$ iff $\forall x_1, ..., x_n, f(x_1, ..., x_n) \leq g(x_1, ..., x_n)$.*

Definition 13 • *A polynomial system of equations (PSE for short) is a set $\{E_1, ..., E_n\}$ where E_i is the equation $x_i = f_i(x_1, ..., x_n)$ and f_i is a polynomial in the variables $x_1, ..., x_n$. Such a system will be abbreviated as $\vec{x} = F(\vec{x})$.*

- *Let $\vec{x} = F(\vec{x})$ by a PSE. We say that $\vec{a}$ is a minimal solution of the system if $\vec{a} = F(\vec{a})$ and, for every $\vec{b}$ such that $\vec{b} = F(\vec{b})$, we have $\vec{a} \leq \vec{b}$.*
- *We denote by F^i the i-iteration of F, i.e. $F^0(\vec{x}) = \vec{x}$ and $F^{i+1}(\vec{x}) = F(F^i(\vec{x}))$.*

Proposition 14 *Let $\vec{x} = F(\vec{x})$ be a PSE. Then, this system has a (unique) minimal solution $\vec{a}$ (that we will denote by $min(F)$). Moreover we have $min(F) = \bigsqcup_{i=0}^{\infty} F^i(\vec{0}) = \bigcap\{\vec{x} \mid F(\vec{x}) \leq \vec{x}\}$.*

Proof Since it is easy to check that F is increasing, this is a special case of the Knaster-Tarski lemma. $\square$

Lemma 15 *Let $f(x, \vec{y}) = f_0(\vec{y}) + \sum_{i \geq 1} f_i(\vec{y}) x^i$ be a polynomial (where $\vec{y}$ is possibly empty) and let $h(\vec{y}) = \sum_{i \geq 1} f_i(\vec{y})$. Then, $g(\vec{y}) = f_0(\vec{y}) + f_0(\vec{y}) \cdot h(\vec{y}) \cdot \omega$ is the minimal solution of the equation $x = f(x, \vec{y})$.*

Proof If $f_0(\vec{y}) = 0$ then the minimal solution is 0. If $h(\vec{y}) = 0$, then for all $i \geq 1$, $f_i(\vec{y}) = 0$ and the minimal solution is $f_0(\vec{y})$. Otherwise, it is easy to check that the minimal solution is ω. In any cases the minimal solution is $g(\vec{y})$. $\square$

Lemma 16 *Let $\vec{x} = F(\vec{x})$ by a PSE. The minimal solution of this system can be computed from F.*

Proof The algorithm to compute this solution is the following. Choose one variable, call it x and call $\vec{y}$ the remaining variables. The system then looks like: $x = f(x, \vec{y})$ and $\vec{y} = G(x, \vec{y})$. Use lemma 15 to find the polynomial $g(\vec{y})$ which is the minimal solution of the equation $x = f(x, \vec{y})$. Repeat the process with the system $\vec{y} = G(g(\vec{y}), \vec{y})$. It is clear that, in this way, we find a solution of the system. Denote by $(a, \vec{b})$ this solution. By proposition 14, let $(x_0, \vec{y_0}) = min(F)$. Since $(a, \vec{b})$ is a solution of the system we have $(x_0, \vec{y_0}) \leq (a, \vec{b})$. Thus it remains to show that $(a, \vec{b}) \leq (x_0, \vec{y_0})$. Since x_0 is a solution of the equation $x = f(x, \vec{y_0})$ we have $g(\vec{y_0}) \leq x_0$. Define F' by $F'(\vec{y}) = G(g(\vec{y}), \vec{y})$. By the monotonicity of G, $F'(\vec{y_0}) = G(g(\vec{y_0}), \vec{y_0}) \leq G(x_0, \vec{y_0}) = \vec{y_0}$. But since the minimal solution of F' is $\bigcap\{\vec{y} \mid F'(\vec{y}) \leq \vec{y}\}$ we have $\vec{b} \leq \vec{y_0}$. By the monotonicity of g, $a = g(b_0) \leq g(\vec{y_0}) \leq x_0$. $\square$

3.3 Some preliminary results

Definition 17 • *We will denote by $\mathcal{F}'$ the set of formulae to which we have added a special element denoted as $*$.*

- *Let E be a set of lists of elements of $\mathcal{F}'$ and A be a formula. We will denote by $[A :: E]$ the set $\{[A :: L] \mid L \in E\}$ where $[A :: L]$ denotes the list L on the beginning of which we have added A.*

Remark
Note that the definition implies that, if E is empty, then so is $[A :: E]$.

Definition 18 *Let A, B be formulae. The set $Elim(A, B)$ of lists of elements of $\mathcal{F}'$ is defined, by induction on the size of A, in the following way.*

- *If $A = B$, then $Elim(A, B) = [*]$.*

- *If $A \neq B$ then,*

  - If A is atomic, $Elim(A, B) = \emptyset$

  - If $A = C \to D$, $Elim(A, B) = [C :: Elim(D, B)]$

  - If $A = A_1 \wedge A_2$, $Elim(A, B) = Elim(A_1, B) \cup Elim(A_2, B)$

  - If $A = A_1 \vee A_2$, $Elim(A, B) = \{[A]\}$

Lemma 19 *Let A, B be formulae and let $L \in Elim(A, B)$. Then the last element of L is either $*$ or a disjunction.*

Proof By induction on A. $\square$

Comments and examples

- The role of the particular symbol $*$ and the set $Elim(A, B)$ will become clear in item 3 of the next lemma. The intuition is the following. $Elim(A, B)$ is the set of lists satisfying the following properties.

  - If $L = [A_1 :: ... :: A_n :: *]$ then, to be able to prove B in some context Γ *by using a sequence of elimination rules starting with* A, it is enough to prove $A_1, ..., A_n$ in the context Γ.

  - If $L = [A_1 :: ... :: A_{n-1} :: D_1 \vee D_2]$ then, to be able to prove B in some context Γ *by using a sequence of elimination rules starting with* A, it is enough to prove $A_1, ..., A_n$ in the context Γ and to prove B both in the contexts $\Gamma \cup \{D_1\}$ and $\Gamma \cup \{D_2\}$.

- Assume B, B' are distinct atomic formulae and $A = (A_1 \to D_1 \vee D_2) \wedge (A_2 \to A_3 \to B) \wedge (A_4 \to B')$. Then $Elim(A, B) = \{L_1, L_2\}$ where $L_1 = [A_1 :: D_1 \vee D_2]$ and $L_2 = [A_2 :: A_3 :: *]$

Lemma 20 *Let t be a normal proof of $\Gamma \vdash B$. Then, t is in one of the following form (where the t_i are normal)*

1. *Either*

   - $t = \lambda x.t_1$, $B = B_1 \to B_2$ and $\Gamma, x : B_1 \vdash t_1 : B_2$

   - $t = \mu\alpha.t_1$ and $\Gamma, \alpha : \neg B \vdash t_1 : \bot$

   - $t = \langle t_1, t_2 \rangle$, $B = B_1 \wedge B_2$ and $\Gamma \vdash t_i : B_i$

   - $t = \omega_i t_1$, $B = B_1 \vee B_2$ and $\Gamma \vdash t_1 : B_i$.

2. *Or $t = (\alpha\ t_1)$ and $\Gamma \vdash t_1 : A$ where $\Gamma \vdash \alpha : \neg A$*

3. *Or $t = (x\ t_1\ ...\ t_n)$ and, for some A such that $\Gamma \vdash x : A$ and some $L \in Elim(A, B)$, we have*

   - either $L = [A_1 :: ... :: A_n :: *]$ and the t_i are proofs of $\Gamma \vdash A_i$

   - or $L = [A_1 :: ... : A_{n-1} :: D_1 \vee D_2]$ and, for $i < n$, the t_i are proofs of A_i and $t_n = [x_1.u_1, x_2.u_2]$ and the u_i are proofs of $\Gamma, x_i : D_i \vdash B$.

Proof By induction on the size of the proof. The only non immediate point is that we cannot use an elimination rule when the type is a disjunction. This is because, otherwise, we will get a proof of the form $(x\ t_1\ ...\ t_k\ [x_1.N_1, x_2.N_2]\ \varepsilon)$ which is not normal. $\square$

Definition 21 *Let t be normal proof. The size of t (denoted as $size(t)$) is defined as follows.*

- $size(\lambda x.t_1) = size(\mu\alpha.t_1) = size(\omega_i t_1) = size(t_1) + 1$
- $size(\langle t_1, t_2\rangle) = max(size(t_1), size(t_2)) + 1$
- $size((x\ t_1\ ...\ t_n) = max(size(t_1), ..., size(t_n)) + 1$

Definition 22 1. *The set $\mathcal{P}$ of partial (normal) terms is defined by the following grammar*

$$\mathcal{P} := \mathcal{V} \mid G \mid \lambda x.\mathcal{P} \mid \mu\alpha.\mathcal{P} \mid \langle \mathcal{P}, \mathcal{P}\rangle \mid \omega_i\mathcal{P} \mid (x\ \mathcal{P}\ ...\ \mathcal{P})$$

2. *The typing rules for $\mathcal{P}$ are the ones of $\mathcal{T}$ plus the additional rule*

$$\frac{}{\Gamma \vdash g : A} \qquad \textit{if } g = \Gamma \vdash A$$

Remark
A normal proof is partial term that contains no goal.

Definition 23 *Let g be a goal. We denote by $\#(g)$ the number (considered as an element of $\overline{\mathbb{N}}$) of distinct normal proofs of g.*

Definition 24
- *Let Γ, Γ' be two contexts. We say that Γ is equivalent to Γ' (denoted as $\Gamma \sim \Gamma'$) if, for any $A \in \mathcal{F}$, Γ contains a declaration $x : A$ iff Γ' contains a declaration $y : A$.*
- *Let $g = \Gamma \vdash B$ and $g' = \Gamma' \vdash B'$. We say that g is equivalent to g' (denoted as $g \sim g'$) if $B = B'$ and $\Gamma \sim \Gamma'$.*

Thus two goals g, g' are equivalent iff their conclusions are the same and they have same set of hypothesis but each hypothesis may appear a different number of times in g and g'.

Lemma 25 *Let t be a partial proof of goal g. Assume $t \neq g$ and contains some goal $g' \sim g$. Then $\#(g) = \#(g')$.*

Proof It is clear that g has no proof iff g' has no proof. Assume then that $\#(g) \geq 1$. Let $g'' = \Gamma'' \vdash A \sim g$ be such that, for any formula B, the number of occurrences of B in Γ or in Γ' is less or equal to the number of occurrences of B in Γ''.

We first show that $\#(g'') = \omega$. It is clear that the term t' obtained from t by replacing g' by g'' also is a partial proof of g'' and, if u is a proof of g, it also is a proof of g''. Then, the u_n defined by $u_0 = u$ and $u_{n+1} = t'[g'' := u_n]$ are distinct normal proofs of g.

We then show that $\#(g) = \omega$ (and, by symmetry, $\#(g') = \omega$). Assume, toward a contradiction, that $\#(g)$ is finite. To each proof of g'' associate the proof of g obtained by replacing the occurrences of a variable in $\Gamma'' - \Gamma$ by one with the same type in Γ. Since $\#(g)$ is finite and $\#(g'')$ is infinite, there are infinitely many proofs of g'' that have the same image by this transformation. But this is impossible since, in a proof, each variable occurs only finitely many times. $\square$

3.4 The equations

To every goal $g = \Gamma \vdash A$ we associate a polynomial system of equations (denoted as $PSE(g)$) of the form $\vec{n} = P(\vec{n})$ where a goal g_i is associated to each variable n_i and f_i is a polynomial that, intuitively, computes the number of normals proofs of g_i of a given size from the number of proofs (of smaller size) of the other goals needed to prove g_i.

$PSE(g)$ is defined by the following algorithm. This algorithm builds, step by step, a partially ordered set V of variables (denoted as n with some index), a function F that associates goals to the variables and a set E of equations of the form $n_i = f_i(\vec{n})$. We will show (see lemma 26 below) that it terminates. $PSE(g)$ will be the set of equations we have built when the algorithm terminates.

It is important to note that the function F is not necessarily injective i.e. to different variables may correspond to the same goal. The reason will be given in the comments after the description of the algorithm.

- *Initial step*

Set $V = \{n_0\}$, $F(n_0) = g$ and $E = \emptyset$.

- *General step*

If, for all $n_i \in V$, there is an equation $n_i = f_i$ in E, then stop. Otherwise, choose some n_i for which E has no equation. We introduce new variables and build the polynomial f_i as the sum of three polynomials in the following way. The first one corresponds to a proof of $g = F(n_i)$ beginning by an introduction rule, the second corresponds to a proof of g by contradiction and the last corresponds to a proof of g by using some hypothesis and several elimination rules.

In the definition of these polynomials we will adopt the following convention. If h is a goal, when we say " let n be a variable for h " this will mean that either $F(n_j) \sim h$ for some $n_j < n_i$ and then n is such an n_j (if there are several choose one) or, if no such variable exists, choose a fresh index j and set $F(n_j) = h$. For each variable n_j introduced in this way, we set $n_j > n_k$ for each k such that $n_i \geq n_k$.

1. The first polynomial P depends on the main connector of B.
   - If B is an atomic formula, then $P = 0$
   - If $B = C \to D$ then let $h = \Gamma, y : C \vdash D$, then let $P = n_j$ where n_j is a variable for h.
   - If $B = B_1 \wedge B_2$. Let h_i be the goal $\Gamma \vdash B_i$. Then $P = n_{i_1}.n_{i_2}$ where n_{i_1}, n_{i_2} are variables for h_1, h_2.
   - If $B = B_1 \vee B_2$. Let h_i be the goal $\Gamma \vdash B_i$. Then $P = n_{i_1} + n_{i_2}$ where n_{i_1}, n_{i_2} are variables for h_1, h_2..
2. The second polynomial Q is as follows.
   - If $B = \bot$ or $B = \neg C$ or if there is already in Γ an hypothesis of the form $\alpha : \neg B$, then $Q = 0$.
   - Otherwise, let $h = \Gamma, \alpha : \neg B \vdash \bot$ and $Q = n_j$ where n_j is a variable for h.
3. The last polynomial is the sum of (over all the hypothesis H in Γ) of the polynomials R_H defined as follows.

- If H is $x : A$, R_H is the sum (over $L \in Elim(A, B)$) of the polynomials $R_{H,L}$ defined below.

  - Assume $L = [A_1 :: ... :: A_p :: *]$. Then $R_{H,L} = n_{i_1}. \ ... \ .n_{i_p}$ where $g_i = \Gamma \vdash A_i$ and $n_{i_1}, ..., n_{i_p}$ are variables for $g_1, ..., g_p$. In particular, if $p = 0$, this means $R_{H,L} = 1$.

  - Assume $L = [A_1 :: ... :: A_p :: D_1 \vee D_2]$. Then, let $g_i = \Gamma \vdash A_i$, $h_i = \Gamma', y : D_i \vdash B$ where Γ' is Γ from which we have deleted the hypothesis $x : A$. Let $n_{i_1}, ..., n_{i_p}$ be variables for $g_1, ..., g_p$, let n_{j_1}, n_{j_2} be variables for h_1, h_2. Then $R_{H,L} = n_{i_1}. \ ... \ .n_{i_p}.n_{j_1}.n_{j_2}$

- If H is $\alpha : \neg A$ then $R_H = n_j$ where $h = \Gamma \vdash A$ and n_j is a name for h.

Comments

- The fact that $Q = 0$ in the first case of a proof by contradiction means that (1) we forbid to prove $\perp$ or $\neg C$ by contradiction and (2) if we are in a part of the proof in which we already have assumed $\neg B$, toward a contradiction, we are no more allowed to prove B by contradiction. We made these restrictions because, otherwise, for any formula, the number of its proofs would be either 0 or ω.

- The reason of eliminating the hypothesis $x : A$ in the case of an elimination of the disjunction is the following. If we already are in one of the branch of a proof by case, we are no more allowed to, again, distinguish the same two cases. Otherwise, the number of proofs would always be either 0 or ω.

It is easy to check that, with the restrictions, the proof system remains complete.

- The fact that a goal may have different names i.e. we may have $F(n_i) = F(n_j)$ for $i \neq j$ comes from the following reason. A goal h may appear in different proofs of g or in different parts of a proof of g. Of course $\#(h)$ does not depend on the place where h appears but the condition that let us decide to give it a new name or not depends of this place. We know, by lemma 25, that $\#(h) = \#(h')$ if $h \sim h'$ and h is below h' in some part of a proof but there is no reason to have $\#(h) = \#(h')$ if they appear in different proofs or in independent part of a proof.

Lemma 26 *The algorithm given above terminates.*

Proof Since the goals are made of sub-formulae of the formulae in g, there are only finitely many possible non equivalent goals. Also note that, when we try to find a proof for a goal h and we have to consider some goal h_1, we give a new name to h_1 (i.e. we introduce a new variable n_i such that $F(n_i) = h_1$ for which, later, we will have to find an equation) only when there is no $h_2 \sim h_1$ below h in the branch of the proof of g that the algorithm, intuitively, constructs. Thus, all the branches are finite. Since there are only finitely many rules, by Konig's lemma, only finitely many variables can be introduced and thus the algorithm terminates. $\square$

3.5 Proof of theorem 9

It is an immediate consequence of lemma 25 and theorem 27 below.

Theorem 27 *Let g be a goal and let $\vec{a}$ be the minimal solution of $PSE(g)$. Then, for each variable n_i occurring in $PSE(g)$ we have $a_i = \#(F(a_i))$.*

Proof Let $PSE(g)$ be the set $\vec{n} = P(\vec{n})$ of equations and $\vec{b}$ be defined by $b_i = \#(F(n_i))$. It follows from lemma 25 that $\vec{b}$ is a solution of $PSE(g)$. Thus, we have $\vec{a} \leq \vec{b}$. Let $u_k = P^k(\vec{0})$. Since $\vec{a}$ is the minimal solution of the system $\vec{n} = P(\vec{n})$ we have $\vec{a} = \bigsqcup_{k=0}^{\infty} u_k$. Denote by $d_i(k)$ the number of normal proofs of $F(n_i)$ of size k and $\overrightarrow{d(k)}$ the vector whose components are the $d_i(k)$. Then $\vec{b} = \bigsqcup_{k=0}^{\infty} \overrightarrow{d(k)}$. Note that the equations are done so that $\overrightarrow{d(k+1)} \leq P(\overrightarrow{d(k)})$. An immediate induction shows that, for each k, $\overrightarrow{d(k)} \leq u_k$. It follows then that $\vec{b} \leq \vec{a}$. $\square$

Remark

If, instead of interpreting the variables and coefficients in $\overline{\mathbb{N}}$, we interpret them in the set $\{0,1\}$ where the operations and the order are the ones of $\mathbb{N}$ except that $1+1=1$, the conclusion of the theorem is then that $a_h = 1$ iff the goal h is provable.

3.6 Some other restrictions on proofs

Definition 28 *We say that a normal term t is in η-long normal form if the following holds for every sub-term u of t.*

- *If u has type $A \to B$ then either $u = \lambda x.u'$ or u is applied to some other term.*
- *If u has type $A \wedge B$ then $u = \langle u_1, u_2 \rangle$ for some terms u_1, u_2.*

The algorithm we have given in the previous sections has been designed to get the number of normal proofs in classical logic. It can be easily transformed if we want to only count proofs satisfying some constraints.

1. If we want to have proofs in minimal logic i.e. the logic where the rules $\perp_i$ and $\perp_e$ are deleted, we just forget the second step (which corresponds to proof by contradiction) in the definition of the set of equations

2. If we want to have proofs in intuitionistic logic, i.e. the logic where the rules $\perp_i$, and $\perp_e$ are deleted and replaced by the rule

$$\frac{\Gamma \vdash \perp}{\Gamma \vdash A}$$

   we replace the polynomial given in the second step of the definition of the set of equations by the following one. If g is $\Gamma \vdash B$ and $B \neq \perp$ then $Q = n_h$ where h is $\Gamma \vdash \perp$ and $Q = 0$ otherwise.

3. Instead of changing the logic, we may also want to restrict the form of the proofs we are looking for. The main usual restriction is to ask to have proofs in η-long normal form. It is well known that, with this restriction, the system remains complete. If we want such proofs it is enough, in the definition of the equations to ask that, if the goal is $\Gamma \vdash B$ and the main connector of B is either an arrow or a conjunction, then we cannot use a proof by contradiction or use an elimination rule.

4. Our algorithm gives two normal proofs for the formula $A \to A$. These proofs are $\lambda x.x$ and $\lambda x.\mu\alpha.(\alpha\ x)$. We could consider that these two proofs are the same and, actually, there is a reduction rule in the $\lambda\mu$-calculus that ensures that the second term reduces to the first one. This rule, that looks like the η-rule of the λ-calculus, is the following $\mu\alpha.(\alpha\ M) \triangleright M$ if α does not occur in M. It intuitively means that if, in a proof of A by contradiction, in fact you have a proof M of A that does not use $\neg A$, you can eliminate the use of the rule for proof by contradiction.

It would be more difficult to consider this rule in the definition of normal proof. This is because it is non local and our algorithm, by essence, can only consider local configurations.

3.7 From polynomials to formulae

In the previous sections we have associated to each formula F a set of polynomial equations whose minimal solution gives the number of normal proofs of F. The opposite construction is also possible as the next proposition shows.

Definition 29 *Let F be a formula of implicational propositional logic i.e. F is built from atomic formulae by using only the arrow as connectors. The rank of F (denoted as $r(F)$) is defined by the following rules.*

- *If F is atomic, then $r(F) = 0$*
- *If $F = A \to B$, then $r(F) = max(r(A) + 1, r(B))$*

Proposition 30 *Let E be a polynomial system of equations with n variables. We can compute n formulae $A_1, ..., A_n$ of implicational logic such that, if $(a_1, ..., a_n)$ is the minimal solution of E then, for all $i \leq n$, a_i is the number of proofs of A_i in η-long normal form. Moreover we may assume that $r(A_i) \leq 2$ for all $i \leq n$.*

Proof Let $\vec{x} = F(\vec{x})$ be the system and $F = (f_1, ..., f_n)$. We take n fresh ground types $O_1, \dots, O_n$. For each polynomial f_p we construct a formula B_p in the following way. For each monomial $M_i = x_1^{\alpha_1} \cdot \dots \cdot x_n^{\alpha_n}$ which appears in f_p let T_i be the formula $O_1^{\alpha_1}, \dots, O_n^{\alpha_n} \to O_p$. Remember that constant 1 can be obtain as the monomial $x_1^{\alpha_1} \cdot \dots \cdot x_n^{\alpha_n}$ when all $\alpha_i = 0$. The formula associated to f_p is $T_1, \dots, T_m \to O_p$. The fact that these formulae satisfy the desired conclusion is straightforward. □

4 Examples

Example 1

We want to compute the number of normal proofs of the formula F below

$$F = F_1 \to F_2 \to F_3 \to F_4 \to F_5 \to F_6 \to A$$

where

$F_1 = B \to C \to C$	$F_2 = F_3 = C$	$F_4 = B \to C \to B$
$F_5 = C \to C \to A$	$F_6 = A \to B \to A$	

To avoid too many equations we will restrict ourselves to proofs in η-long normal form and in minimal logic and, to simplify notations, we will use the same name for a goal and the variable attached to it and, if a goal has several names, the corresponding variables will be the same with, possibly, some index. Also note that, since we will not write the terms representing the proofs, there is no need to give names to the hypothesis and thus we will write contexts simply as multi-sets of formulae.

Let $\Gamma = F_1, F_2, F_3, F_4, F_5, F_6$. The goals are:

$$x \text{ is } \Gamma \vdash A,$$
$$y, y_1 \text{ are } \Gamma \vdash B$$
$$z, z_1 \text{ are } \Gamma \vdash C.$$

The order on these variables is given by: $x < y, z$; $y < z_1$ and $z < y_1$.

The set of equations is

$$\begin{array}{cc} \multicolumn{2}{c}{x = xy + z^2} \\ y = yz_1 & z_1 = 2 + yz_1 \\ z = 2 + y_1 z & y_1 = y_1 z \end{array}$$

The minimal solution is $x = 4, y = y_1 = 0, z = z_1 = 2$ and, therefore, there are exactly 4 proofs of F in η-long normal form.

Example 2

We want to compute the number of normal proofs of the formula F below where $\neg_c B$ is the abbreviation of $B \to C$. This formula is a kind of translation (provable in minimal logic) of Pierce law.

$$F = ((A \to \neg_c\neg_c B) \to \neg_c\neg_c A) \to \neg_c\neg_c A$$

Again, we adopt the same restrictions and conventions of notations as in the previous example.
Let $F_1 = (A \to \neg_c\neg_c B) \to \neg_c\neg_c A$, $F_2 = \neg_c A$ and $\Gamma = \alpha_1 : F_1, \alpha_2 : F_2$.
The goals are

$$\begin{array}{c} x \text{ is } \Gamma \vdash C, \\ y, y_1 \text{ are } \Gamma, A, \neg_c B \vdash C, \\ z, z_1 \text{ are } \Gamma, A \vdash C, \\ u \text{ is } \Gamma \vdash A, \\ v, v_1 \text{ are } \Gamma, A, \neg_c B \vdash A, \\ w, w_1 \text{ are } \Gamma, A, \neg_c B \vdash B \\ r, r_1 \text{ are } \Gamma, A \vdash A. \end{array}$$

The order on these variables is given by: $x < y, z, u$; $y < z_1, v, w$; $z_1 < r_1$; $z < y_1, r$; $y_1 < v_1, w_1$
The set of equations is

$$\begin{array}{cccc} & x = yz + u & & \\ & y = yz_1 + v + w & z_1 = yz_1 + r_1 & \\ & z = y_1 z + r & y_1 = y_1 z + v_1 + w_1 & \\ v = v_1 = 1 & w = w_1 = 0 & r = r_1 = 1 & u = 0 \end{array}$$

The minimal solution is $x = y = z = y_1 = z_1 = \omega$ and, therefore, there are infinitely many proofs of F in η-long normal forms.

Example 3

Let F be the formula $\neg A \vee A$. It is known that F is not provable in intuitionistic logic. We will show that, in classical logic, the are infinitely many distinct proofs in η-long normal form. Since the number of equations to be written is quite big we will only write some of those that imply that the number is infinite. To simplify we will also omit some intermediate goals and/or equations when the relations between the corresponding variables are easy to show.

The useful goals are the following

$$\begin{array}{c} x \text{ is } \vdash F \\ x_1 \text{ is } \vdash A,\ x_2 \text{ is } \vdash \neg A \text{ and } x_3 \text{ is } \alpha : \neg F \vdash \bot \\ a \text{ is } \alpha : \neg F \vdash A \text{ and } b \text{ is } \alpha : \neg F \vdash \neg A \\ a_1 \text{ is } \alpha : \neg F, \beta : \neg A \vdash \bot \text{ and } a_2 \text{ is } \alpha : \neg F \vdash \neg A \\ c \text{ is } \alpha : \neg F, \beta : \neg A, y : A \vdash \bot \\ c_1 \text{ is } \alpha : \neg F, \beta : \neg A, y : A \vdash A \\ d \text{ is } \alpha : \neg F, \beta : \neg A, y : A, z : A \vdash \bot \end{array}$$

Some equations are

$$\begin{gathered}
x = x_1 + x_2 + x_3 \\
x_1 = 0,\ x_2 = 0 \\
x_3 = a + b \\
a = a_1 + a_2 \\
a_1 = c \quad (\star) \\
c = 2.c_1 + d \\
c_1 = 1 \\
\text{The use of lemma 25 gives } d = c.
\end{gathered}$$

$(\star)$ a_1 actually is the sum of c and some other variables that are easily shown to be 0.

References

[1] C.B Ben-Yelles. *Type assignment in the lambda calculus. Syntax and semantics.* Thesis, Mathematics Department, University of Wales Swansea, Swansea, UK (1979).

[2] W. Dekkers. *Reducibility of types in Typed Lambda Calculus.* Information and Computation vol 77, No 2 pp 131– 137 (1988).

[3] P. de Groote. *Strong Normalization of Classical Natural Deduction with Disjunction.* Lecture Notes in Computer Science 2044 pp 182-196 (2001).

[4] J.R. Hindley. *Basic Simple Type Theory.* Cambridge Tracts in Theoretical Computer Science 42. Cambridge University Press 1997.

[5] S. Hirokawa. *Infiniteness of Proof(α) is P-Space Complete.* Theoret. Comput. Sci. 206 no. 1-2, pp 331–339 (1998).

[6] M. Parigot. *$\lambda\mu$-Calculus: An Algorithmic Interpretation of Classical Natural Deduction.* Lecture Notes in Computer Science 624 pp 190-201 (1992).

[7] A.S. Troelstra, H. Schwichtenberg. *Basic proof theory.* Cambridge University Press 1996.

[8] D. Van Dalen. *Logic and structure.* Springer 1997.

[9] J. B. Wells, B. Yakobowski. *GraphBased Proof Counting and Enumeration with Applications for Program Fragment Synthesis.* Lecture Notes in Computer Science 3573, pp 262-277 (2005).