# A combinatorial proof of strong normalisation for the simply typed $\lambda$-calculus

Alexandre Miquel

(DRAFT)

**Abstract**

We present a combinatorial proof of strong normalisation for the simply typed $\lambda$-calculus [1], by exhbiting a measure function from simply-typed $\lambda$-terms to natural numbers that decreases at each reduction step. This proof is a variant of Gandy's proof of normalization [2].

## 1 The simply typed $\lambda$-calculus

Simple types are inductively defined from the following two clauses:

- The symbol $\iota$ is a simple type (the *ground type*);
- If $\tau$ and $\sigma$ are simple types, then so is $\tau \to \sigma$ (an *arrow type*).

Each simple type $\tau$ is equipped with an infinite set of symbols that are called the *variables of type* $\tau$ (notation: $x^\tau$, $y^\tau$, $z^\tau$, etc.) We assume that the sets of variables associated to distinct simple types are disjoint. Simply typed $\lambda$-terms are inductively defined as follows:

- If $x^\tau$ is a variable of type $\tau$, then $x^\tau$ is a simply typed $\lambda$-term of type $\tau$;
- If $x^\tau$ is a variable of type $\tau$ and if $M$ is a simply typed $\lambda$-term of type $\sigma$, then $\lambda x^\tau . M$ is a simply typed $\lambda$-term of type $\tau \to \sigma$;
- If $M$ is a simply typed $\lambda$-term of type $\tau \to \sigma$ and if $N$ is a simply typed $\lambda$-term of type $\tau$, then $MN$ is a simply typed $\lambda$-term of type $\sigma$.

Typed substitution $M\{x^\tau := N\}$ is defined as expected (with the constraint that $N$ is a simply typed $\lambda$-term of type $\tau$). One step $\beta$-reduction (notation: $M \succ_1 M'$) is inductively defined by the following rules:

$$\frac{}{(\lambda x^\tau . M)N \ \succ_1 \ M\{x^\tau := N\}}$$

$$\frac{M \ \succ_1 \ M'}{\lambda x^\tau . M \ \succ_1 \ \lambda x^\tau . M'} \qquad \frac{M \ \succ_1 \ M'}{MN \ \succ_1 \ M'N} \qquad \frac{N \ \succ_1 \ N'}{MN \ \succ_1 \ MN'}$$

Finally, the set of *strongly normalising terms*, written SN, is inductively defined by the unique clause:

- If for all $M'$, $M \succ_1 M'$ implies $M' \in \mathrm{SN}$, then $M \in \mathrm{SN}$

The aim of this paper is to show that

**Theorem 1** — *All the simply typed $\lambda$-terms are strongly normalising.*

## 2 Interpretation of types

To define the measure function, we associate to each simple type $\tau$ a set $|\tau|$ equipped with a well-founded ordering $\ll_\tau \subset |\tau| \times |\tau|$. Formally, the pair $(|\tau|, \ll_\tau)$ is inductively defined on $\tau$ as follows:

- $|\iota| = \mathbb{N}$, and $\ll_\iota$ is the (usual) strict ordering $<_\mathbb{N}$ over natural numbers;
- $|\tau \to \sigma|$ is the set of all increasing functions from $|\tau|$ to $|\sigma|$, that is:

  $$|\tau \to \sigma| \quad = \quad \left\{ f \in |\sigma|^{|\tau|}; \quad \forall v, v' \in |\tau| \;\; (v \ll_\tau v' \;\Rightarrow\; f(v) \ll_\sigma f(v')) \right\}$$

  whereas $\ll_{\tau\to\sigma}$ is the corresponding extensional (strict) ordering:

  $$f \ll_{\tau\to\sigma} g \qquad \text{iff} \qquad \forall v \in |\tau| \quad f(v) \ll_\sigma g(v) \qquad\qquad (f, g \in |\tau \to \sigma|)$$

It is straightforward to check that

**Proposition 1** — *For every simple type $\tau$, the relation $\ll_\tau$ is transitive.*

*Proof.* By induction on $\tau$. □

On the other hand, well-foundedness—and even irreflexivity—of the binary relation $\ll_\tau$ is not that obvious, for it relies on the fact that none of the sets $|\tau|$ is empty.[1] To establish this, we need to define some extra structures.

### 2.1 Translation

Each set $|\tau|$ is equipped with an asymmetric binary operation of translation $+_\tau : |\tau| \times \mathbb{N} \to |\tau|$ which is inductively defined for all $k \in \mathbb{N}$ by

$$\begin{array}{rcll} n +_\iota k & = & n +_\mathbb{N} k & (n \in |\iota|) \\ f +_{\tau\to\sigma} k & = & (v \in |\tau| \mapsto f(v) +_\sigma k) & (f \in |\tau \to \sigma|) \end{array}$$

**Proposition 2** — *For all simple types $\tau$:*

- *The operation $+_\tau : |\tau| \times \mathbb{N} \to |\tau|$ is well defined;*
- *$v \ll_\tau v'$ implies $v +_\tau k \ll v' +_\tau k$ for all $v, v' \in |\tau|$ and $k \in \mathbb{N}$.*

*Proof.* Both items are proved simultaneously, by induction on $\tau$. □

**Proposition 3** — *For all $v \in |\tau|$ and $k, k' \in \mathbb{N}$:*

1. *$v +_\tau 0 = v$*
2. *$(v +_\tau k) +_\tau k' = v +_\tau (k + k')$.*
3. *$k <_\mathbb{N} k'$ implies $v +_\tau k \ll_\tau v +_\tau k'$;*

*Proof.* By induction on $\tau$. □

[1] Would the set $|\tau|$ be empty for some type $\tau$, then the set $|\tau \to \iota|$ would be the singleton formed by the empty function $\varnothing : \varnothing \to \mathbb{N}$ (which is increasing). By definition of the relation $\ll_{\tau\to\iota}$, one would have $\varnothing \ll_{\tau\to\iota} \varnothing$, so that $\ll_{\tau\to\iota}$ would be not well-founded.

## 2.2 The objects $\tau_*$ and $\tau^*$

For each simple type $\tau$, we want to define an element $\tau_* \in |\tau|$ together with an increasing function $\tau^* : |\tau| \to \mathbb{N}$ that we call the *collapse function* associated to the type $\tau$. These structures are defined by mutual induction on $\tau$ by

$$\begin{array}{rclcrcl} \iota_* & = & 0 & \qquad & (\tau \to \sigma)_* & = & (v \in |\tau| \mapsto \sigma_* +_\sigma \tau^*(v)) \\ \iota^*(n) & = & n & & (\tau \to \sigma)^*(f) & = & \sigma^*(f(\tau_*)) \end{array}$$

for all $n \in |\iota|$ and $f \in |\tau \to \sigma|$. We then check that

**Proposition 4** — *For all simple types $\tau$:*

1. *$\tau_* \in |\tau|$ and*
2. *$v \ll_\tau v'$ implies $\tau^*(v) <_{\mathbb{N}} \tau^*(v')$ for all $v, v' \in |\tau|$.*

*Proof.* Both items are proved simultaneously, by induction on $\tau$. □

From the very existence of $\tau_*$ and $\tau^*$ we get:

**Corollary 5** — *For all simple types $\tau$, the set $|\tau|$ is inhabited and the relation $\ll_\tau$ is a well-founded strict ordering on this set.*

Let us also notice that $\tau_*$ and $\tau^*$ enjoy the following algebraic properties:

**Proposition 6** — *For all simple types $\tau$:*

1. *$\tau^*(\tau_*) = 0$;*
2. *$\tau^*(v +_\tau k) = \tau^*(v) + k \quad$ (for all $v \in |\tau|$ and $k \in \mathbb{N}$).*

## 2.3 Large ordering

Similarly to the definition of $\ll_\tau$, we define a partial order $\leq_\tau$ on each set $|\tau|$ by setting:

$$\begin{array}{rcll} n \leq_\iota n' & \equiv & n \leq_{\mathbb{N}} n' & \qquad (n, n' \in |\iota|) \\ f \leq_{\tau\to\sigma} f' & \equiv & \forall v \in |\tau| \;\; f(v) \leq_\sigma f(v') & \qquad (f, f' \in |\tau \to \sigma|) \end{array}$$

By a straightforward induction on $\tau$ we check that:

**Proposition 7** — *The relation $\leq_\tau$ is a partial order on $|\tau|$ that contains the strict ordering $\ll_\tau$, and for all $v, v', v'' \in |\tau|$ one has:*

1. *$v \ll_\tau v'$ and $v' \leq_\tau v''$ imply $v \ll_\tau v''$;*
2. *$v \leq_\tau v'$ and $v' \ll_\tau v''$ imply $v \ll_\tau v''$.*

Moreover, the collapse function $\tau^* : |\tau| \to \mathbb{N}$ is monotonic, as well as the operation of translation $+_\tau : |\tau| \times \mathbb{N} \to |\tau|$:

**Proposition 8** — *For all $v, v' \in |\tau|$ and $k, k' \in \mathbb{N}$:*

1. *If $v \leq_\tau v'$, then $\tau^*(v) \leq_{\mathbb{N}} \tau^*(v')$;*
2. *If $v \leq_\tau v'$ and $k \leq_{\mathbb{N}} k'$, then $v +_\tau k \leq_\tau v' +_\tau k'$.*

*Proof.* We first prove item 2 by induction on $\tau$, and then item 1. □

Actually, we can even characterize $\ll_\tau$ from $\leq_\tau$ and $+_\tau$:

**Proposition 9** — *For all $v, v' \in |\tau|$: $v \ll_\tau v'$ iff $v +_\tau 1 \leq_\tau v'$.*

*Proof.* By induction on $\tau$. □

# 3 Interpretation of simply typed $\lambda$-terms

## 3.1 Valuations

A *valuation* is a function $\phi$ that associates an object $\phi(x^\tau) \in |\tau|$ to each variable $x^\tau$. Given two valuations $\phi$ and $\phi'$, we write

$$\phi \leq \phi' \qquad \text{iff} \qquad \phi(x^\tau) \leq_\tau \phi'(x^\tau) \quad \text{for all variables } x^\tau .$$

Given a valuation, a variable $x^\tau$ and a value $v \in |\tau|$, we write $(\phi, x^\tau \leftarrow v)$ the valuation defined by

$$\begin{array}{rcl} (\phi, x^\tau \leftarrow v)(x^\tau) & = & v \\ (\phi, x^\tau \leftarrow v)(y^\sigma) & = & \phi(y^\sigma) \quad \text{for all variables } y^\sigma \neq x^\tau . \end{array}$$

This operation is monotonic in the sense that $(\phi, x^\tau \leftarrow v) \leq (\phi', x^\tau \leftarrow v')$ as soon as $\phi \leq \phi'$ and $v \leq_\tau v'$.

## 3.2 The interpretation function

To each pair formed by a term $M$ of type $\tau$ and a valuation $\phi$, we associate an object $[M]_\phi \in |\tau|$. Formally, the function $\phi \mapsto [M]_\phi$ is defined by induction on $M$ for all $\phi$ by the equations:

$$\begin{array}{rcl} [x^\tau]_\phi & = & \phi(x^\tau) \\ [\lambda x^\tau . M]_\phi & = & \big(v \in |\tau| \mapsto [M]_{(\phi; x^\tau \leftarrow v)} +_\sigma (\tau^*(v) + 1)\big) \\ [MN]_\phi & = & [M]_\phi\big([N]_\phi\big) \end{array}$$

We check that:

**Proposition 10** — *For all simply typed $\lambda$-terms $M$ of type $\tau$:*

1. *$[M]_\phi \in |\tau|$ for all valuations $\phi$;*
2. *$\phi \leq \phi'$ implies $[M]_\phi \leq_\tau [M]_{\phi'}$ for all valuations $\phi$ and $\phi'$.*

*Proof.* Both items are proved simultaneously, by induction on $M$. □

**Proposition 11** — *Given a term $M$ of type $\sigma$, a variable $x^\tau$, a term $N$ of type $\tau$ and a valuation $\phi$, we have:*

$$[M\{x^\tau := N\}]_\phi = [M]_{(\phi, x^\tau \leftarrow [N]_\phi)} .$$

*Proof.* By induction on $M$. □

**Proposition 12** — *Let $M$ and $M'$ be two terms of type $\tau$. If $M \succ_1 M'$, then $[M']_\phi \ll_\tau [M]_\phi$ for all valuations $\phi$.*

*Proof.* By induction on the derivation of one-step reduction:

- $(\lambda x^\tau . M)N \succ_1 M\{x := N\}$ (Base case).

  For all valuations $\phi$ we have

  $$[(\lambda x^\tau . M)N]_\phi \;=\; [M]_{(\phi; x^\tau \leftarrow [N]_\phi)} +_\sigma (\tau^*([N]_\phi) + 1)\big) \qquad \text{(Def. of } [\_])$$

  whereas

  $$[M\{x := N\}]_\phi \;=\; [M]_{(\phi; x^\tau \leftarrow [N]_\phi)} \qquad \text{(Prop. 11)}$$

  Hence we get $[M\{x := N\}]_\phi \ll_\sigma [(\lambda x^\tau . M)N]_\phi$, since $\tau^*([N]_\phi) + 1 > 0$.

- $\lambda x . M \succ_1 \lambda x . M'$, from $M \succ_1 M'$ ($\xi$-rule).

  Let $\phi$ be a valuation. By IH, we have $[M']_{(\phi; x^\tau \leftarrow v)} \ll_\sigma [M]_{(\phi; x^\tau \leftarrow v)}$ for all $v \in |\tau|$, and thus

  $$[M']_{(\phi; x^\tau \leftarrow v)} +_\sigma (\tau^*(v) + 1) \quad \ll_\sigma \quad [M]_{(\phi; x^\tau \leftarrow v)} +_\sigma (\tau^*(v) + 1)$$

  for all $v \in |\tau|$. Hence $[\lambda x^\tau . M']_\phi \ll_{\tau \to \sigma} [\lambda x^\tau . M]_\phi$.

- $MN \succ_1 M'N$, from $M \succ_1 M'$ (Application, left).

  Let $\phi$ be a valuation. By IH we have $[M']_\phi \ll_{\tau \to \sigma} [M]_\phi$, hence

  $$[M'N]_\phi \;=\; [M']_\phi([N]_\phi) \;\ll_\sigma\; [M]_\phi([N]_\phi) \;=\; [MN]_\phi$$

  by definition of the strict order $\ll_{\tau \to \sigma}$.

- $MN \succ_1 MN'$, from $N \succ_1 N'$ (Application, right).

  Let $\phi$ be a valuation. By IH we have $[N']_\phi \ll_\tau [N]_\phi$, hence

  $$[MN']_\phi \;=\; [M]_\phi([N']_\phi) \;\ll_\sigma\; [M]_\phi([N]_\phi) \;=\; [MN]_\phi$$

  since the function $[M]_\phi$ is increasing. □

### 3.3 The measure function $\varepsilon^\tau$

Fix an arbitrary valuation $\phi_0$—for instance the valuation[2] which is defined by setting $\phi_0(x^\tau) = \tau_*$ for all variables $x^\tau$. To each term $M$ of type $\tau$, we now associate a natural number $\varepsilon^\tau(M)$ by setting:

$$\varepsilon^\tau(M) \;=\; \tau^*([M]_{\phi_0}) \,.$$

From Prop. 4 and Prop. 12 it is now clear that

**Proposition 13** — *If $M \succ_1 M'$, then $\varepsilon(M') <_{\mathbb{N}} \varepsilon(M)$.*

Theorem 1 is then immediate.

[2] As for any normalisation proof, we critically need the fact that the interpretation of every type is inhabited in order to build a valuation and conclude.

## References

[1] H. P. Barendregt. *Lambda Calculi with Types*. Handbook of Logic in Computer Science, Vol. 2, Oxford University Press, 1992.

[2] R. O. Gandy. Proofs of strong normalization. In J. P. Seldin, J. R. Hindley, eds.: *To H. B. Curry: Essays in Combinatory Logic, Lambda Calculus, and Formalism*, p. 457–477. Academic Press, 1980.