Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in Theoretical Computer Science 203 (2008) 123–133

Electronic Notes in Theoretical Computer Science

www.elsevier.com/locate/entcs

Sub-λ-calculi, Classified

François-Régis Sinot

Universidade do Porto (DCC & LIACC)
Rua do Campo Alegre 1021–1055
4169–007 Porto, Portugal

Abstract

When sharing is studied in the λ-calculus, some sub-calculi often pop up, for instance λI or the linear λ-calculus. In this paper, we generalise these to a large class of sub-calculi, parametrised by an arbitrary predicate on the number of occurrences of bound variables. Such a definition only makes sense when the sub-calculi are stable by β-reduction. Surprisingly, we are able to give a complete description and classification of such stable sub-calculi, in a rather algebraic way; and surprisingly again, we discover some unexpected such subcalculi. This could lead to a better understanding of the structure of the λ-calculus.

Keywords: λ-calculus, sharing, linearity

1 Introduction

Sharing is an important but difficult issue, in particular in the λ-calculus. Sometimes, in order to simplify the problem or tackle it in a more focused way, attention is restricted to a particular subsystem of the λ-calculus. Such subsystems include λI, where erasing is forbidden, or the linear λ-calculus, where each bound variable occurs exactly once [1]. These subsystems are defined by imposing some uniform restrictions on the number of occurrences of bound variables.

In this paper, we propose to generalise this idea and study in a systematic way such sub-calculi. Motivation for doing this is to try to understand better the structure of the λ-calculus, which is always useful and may have unexpected applications. At present however, it is essentially a nice exercise in pure λ-calculus.

More precisely, we will define the notion of sub-calculus in a generic setting, and argue that, in order for this definition to make sense, such a system has to verify a certain property, namely stability by β-reduction (Section 3). We will then focus our attention on a concrete class of sub-calculi, parametrised by arbitrary predicates on the usage of resources (number of occurrences of bound variables), and study stability in this setting (Sections 4 and 5). This will lead us to a complete classification of these subcalculi in Section 7.

doi:10.1016/j.entcs.2008.03.038

2 Preliminaries

For self-containment, we recall here some of the few prerequisites of the technical part of this paper. We will be concerned with the λ-calculus, so it makes sense to recall first some general notions on reduction systems (see [4] for a good reference). An *abstract reduction system* (ARS) is just a pair $(A, \to)$ where A is a set (called the set of *objects*) and $\to$ is a binary relation on A (called the *reduction*). We write $t \to u$ if $(t, u) \in \to$. The reflexive transitive closure of $\to$ is $\to^*$, and $\leftarrow$ is the inverse relation of $\to$. A normal form is a $t \in A$ such that there exists no u such that $t \to u$.

The λ-*calculus* is then an ARS $(\Lambda, \to_\beta)$, where the set of λ-*terms* Λ is defined by the following grammar (quotiented by α-*conversion* i.e. renaming of bound variables, avoiding captures):

$$t, u ::= x \mid \lambda x.t \mid t\ u$$

and β-*reduction* is defined by:

$$(\lambda x.t)\ u \to_\beta t\{x := u\}$$

where $t\{x := u\}$ denotes t where all occurrences of x are replaced by u, without name capture. We refer the reader to [1] for more details.

3 General Sub-calculi

We will very soon (in Section 4) turn to the study of a certain concrete family of sub-λ-calculi, but there are also a few general observations that we can make first in a general setting.

In the framework of abstract reduction systems, the most general definition of a sub-system is to restrict both the set of objects and the reduction. But it also makes sense to adopt a more constrained definition and restrict only the set of terms, without touching the reduction (see [4, Def. 1.1.6]). In this paper, focused on the λ-calculus, this last choice is better-suited: we do not want to mess up with β-reduction, and, for instance, have β-redexes which are not redexes in the subsystem. We view the restriction of reduction as an orthogonal aspect, related to strategies, which could be interesting to combine with our approach, but that we do not deal with in this paper for the sake of clarity.

But there is another issue. Let $\Sigma \subseteq \Lambda$ be an arbitrary set of λ-terms. Does it always make sense to say that $(\Sigma, \to_\beta)$ is a sub-calculus ? Imagine there is a $t \in \Sigma$ such that $t \to_\beta u$: clearly, we must require also that $u \in \Sigma$. In other words, a sub-λ-calculus must be closed under β-reduction. Equivalently, we will say in this paper that it must be *stable* (by β-reduction). Be careful: there are many other meanings to stability in the literature.

Definition 3.1 A set of λ-terms Σ is said to be *stable* if it is closed under β-reduction, i.e. if whenever $t \in \Sigma$ and $t \to_\beta u$, then $u \in \Sigma$.

Definition 3.2 Let $\Sigma \subseteq \Lambda$ be a stable set of λ-terms. The λ_Σ-calculus is defined as the ARS $(\Sigma, \rightarrow_\beta)$.

Every λ_Σ-calculus is a sub-λ-calculus, and conversely every sub-λ-calculus is a λ_Σ-calculus for some stable $\Sigma \subseteq \Lambda$. A set of λ-terms thus defines a sub-λ-calculus if and only if it is stable. In other words, the study of sub-λ-calculi is essentially a study of stability of sets of λ-terms.

Starting from Section 4, we will focus our attention on a certain family of sub-λ-calculi, but we may still notice that there are some interesting sub-λ-calculi outside of this family.

Example 3.3 Let $I = \lambda x.x$ and $\Sigma = \{I\} \bigcup_{t,u \in \Sigma} (t\,u)$ the set of terms built from the identity and the application, then Σ is stable.

Proof. Consider $t \in \Sigma$. If there is a redex in t, it is of the form $I\,u \rightarrow_\beta u$ with u in Σ, and it is clear that the reduct is in Σ. □

Example 3.4 The set of simply typable λ-terms [2] is stable. More precisely, let A be a simple type, then the set of λ-terms of type A is stable.

Proof. This is just a paraphrase of subject reduction. □

Example 3.5 Let $\mathrm{SN} \subseteq \Lambda$ be the set of strongly normalising terms, SN is stable.

Proof. If $t \in \mathrm{SN}$ and $t \rightarrow_\beta u$, then clearly $u \in \mathrm{SN}$. □

It is well-known that, with the definition we have chosen, sub-calculi inherit properties from the full system [4], for instance:

Theorem 3.6 *If $\Sigma \subseteq \Lambda$ is stable, the λ_Σ-calculus is confluent.*

Proof. Assume $u_1 \; {}_\beta^*\!\leftarrow t \rightarrow_\beta^* u_2$ in the λ_Σ-calculus. Then there exists a λ-term v such that $u_1 \rightarrow_\beta^* v \; {}_\beta^*\!\leftarrow u_2$, by confluence of the λ-calculus, and v is a λ_Σ-term by stability of Σ. □

In this general setting, we do not have any tool to decide if a given $\Sigma \subseteq \Lambda$ is stable or not, but we can combine stable subsets:

Proposition 3.7 *If Σ_1 and Σ_2 are stable, then $\Sigma_1 \cap \Sigma_2$ and $\Sigma_1 \cup \Sigma_2$ are stable.*

Proof. For the intersection, assume $t \rightarrow_\beta u$ with $t \in \Sigma_1 \cap \Sigma_2$, then $u \in \Sigma_1$ and $u \in \Sigma_2$, hence $u \in \Sigma_1 \cap \Sigma_2$. For the union, assume $t \rightarrow_\beta u$ with $t \in \Sigma_1 \cup \Sigma_2$, then either $t \in \Sigma_1$ and $u \in \Sigma_1 \subseteq \Sigma_1 \cup \Sigma_2$, or $t \in \Sigma_2$ and $u \in \Sigma_2 \subseteq \Sigma_1 \cup \Sigma_2$. □

We also introduce a last way to build a stable set, by "stabilising" an arbitrary set. The notion of stable closure is quite folklore. It will be used mainly in Section 8.

Definition 3.8 Let $\Sigma \subseteq \Lambda$, we define the *stable closure* of Σ, written $\overline{\Sigma}$, as the smallest stable set of terms such that $\Sigma \subseteq \overline{\Sigma}$.

4 Resourced Sub-calculi

Our purpose is to study subsystems of the λ-calculus. The most general way to define them has been mentioned in the previous section, but this representation is too general to be interesting: for instance, there is little hope to obtain a nice characterisation theorem in such a general setting. We thus focus our attention on the following, more restricted class of subsystems, which is parametrised by an arbitrary predicate on the number of occurrences of bound variables. We start with a definition.

Definition 4.1 We define the number of free occurrences of a variable x in a λ-term t, written $|t|_x$, as follows:

$$\begin{aligned} |x|_x &= 1 \\ |y|_x &= 0 \\ |t\,u|_x &= |t|_x + |u|_x \\ |\lambda x.t|_x &= 0 \\ |\lambda y.t|_x &= |t|_x \end{aligned}$$

Remark that, for a given variable x, $|\cdot|_x$ is well-defined on α-equivalence classes (i.e. if $t =_\alpha u$ then $|t|_x = |u|_x$).

We may now define the following, more restricted class of sub-λ-calculi.

Definition 4.2 If $\mathcal{P}$ is a predicate on natural numbers, we define the set of $\lambda_\mathcal{P}$-terms as follows:

$$\begin{aligned} t, u ::= {} & x \mid t\,u \\ & \mid \lambda x.t \qquad\qquad \text{if } \mathcal{P}(|t|_x) \end{aligned}$$

We also define the $\lambda_\mathcal{P}$-calculus as the set of $\lambda_\mathcal{P}$-terms equipped with β-reduction $\to_\beta$ (when this set is stable, we will discuss this further in Section 5). We also simply say that $\mathcal{P}$ is stable whenever the set of $\lambda_\mathcal{P}$-terms is stable. Following usual conventions, $\lambda_\mathcal{P}$ alone means either the set of terms or the calculus (which justifies apparent pleonasms of the form "$\lambda_\mathcal{P}$ is stable").

It makes sense to impose that $\mathcal{P}(|t|_x)$ holds for every free variable x of an open term t (we will call this the *strict convention*). However, this entails some unpleasant syntactic accidents, as will be shown later. These accidents would disappear if we added (unconstrained) constants. However, the pure λ-calculus view of constants is exactly as free variables, which should thus be unconstrained. Consequently, we always assume the *relaxed convention*, unless otherwise stated: we do not impose that $\mathcal{P}(|t|_x)$ holds if x is a free variable of an open term t.

¿From now on, operations on propositions are always implicitly lifted to predicates, which means that, for instance, if $\mathcal{P}$ and $\mathcal{Q}$ are predicates, $\mathcal{P} \wedge \mathcal{Q}$ is the predicate defined by, for all n, $(\mathcal{P} \wedge \mathcal{Q})(n) = \mathcal{P}(n) \wedge \mathcal{Q}(n)$. In the same spirit, $\bot$ and $\top$ denote the predicates which are constantly false and true, respectively. Moreover,

$\mathcal{P}$, $\mathcal{Q}$, implicitly denote predicates on natural numbers. We also allow the definition of predicates by partial application of infix binary predicates, in a Haskell-like style, e.g. (≥ 3) is the predicate defined by, for all n, $(\geq 3)(n) = (n \geq 3)$.

It is natural to wonder how well $\mathcal{P}$ characterises $\lambda_{\mathcal{P}}$.

Proposition 4.3 $(\lambda_{\mathcal{P}} \subseteq \lambda_{\mathcal{Q}}) \iff (\mathcal{P} \Rightarrow \mathcal{Q})$.

Proof.

- $\Rightarrow$ Assume $n \geq 1$ and $\mathcal{P}(n)$. We can build the term $t = \lambda x.\underbrace{x \ldots x}_{n} \in \lambda_{\mathcal{P}} \subseteq \lambda_{\mathcal{Q}}$. Then $t \in \lambda_{\mathcal{Q}}$, hence $\mathcal{Q}(n)$, by definition of $\lambda_{\mathcal{Q}}$. For the case $n = 0$, we use instead $t = \lambda x.z$ where z is a free variable (remember that we assume the relaxed convention).
- $\Leftarrow$ Assume $t \in \lambda_{\mathcal{P}}$ and let $\lambda x.u$ be a subterm of t. By definition of $\lambda_{\mathcal{P}}$, $\mathcal{P}(|u|_x)$ holds, and so does $\mathcal{Q}(|u|_x)$. Since this holds for every sub-abstraction of t, we may conclude $t \in \lambda_{\mathcal{Q}}$.

□

Remark 4.4 The left-to-right implication is false under the strict convention without constants, for instance $\lambda_{\perp} = \lambda_{(=0)} = \varnothing$.

In particular, an immediate corollary of Proposition 4.3 is that $\lambda_{\mathcal{P}} = \lambda_{\mathcal{Q}}$ if and only if $\mathcal{P} \Leftrightarrow \mathcal{Q}$. In other words, $\mathcal{P}$ exactly characterises $\lambda_{\mathcal{P}}$.

In this framework, it is worth stating the following syntactic properties.

Theorem 4.5 *If $\mathcal{P}$ is stable, the $\lambda_{\mathcal{P}}$-calculus is confluent.*

Proof. Consequence of Theorem 3.6. □

Theorem 4.6 *If $\mathcal{P}$ is stable, the $\lambda_{\mathcal{P}}$-calculus is strongly normalising if and only if $\mathcal{P}(n)$ does not hold for any $n \geq 2$.*

Proof. If $\mathcal{P}(n)$ does not hold for any $n \geq 2$, the $\lambda_{\mathcal{P}}$-calculus is a subsystem of $\lambda_{(\leq 1)}$, i.e. the affine λ-calculus, which is strongly normalising. Conversely, assume that $\mathcal{P}(n)$ holds for some $n \geq 2$. Then we can build the non-normalising $\lambda_{\mathcal{P}}$-term $(\lambda x.\underbrace{x \ldots x}_{n})\,(\lambda x.\underbrace{x \ldots x}_{n})$. □

5 Characterising Stability

The notion of $\lambda_{\mathcal{P}}$-calculus only makes sense when the set of $\lambda_{\mathcal{P}}$-terms is stable, hence stability is an important notion. Let us test on an example if it also easy to handle.

Example 5.1 The linear λ-calculus $\lambda_{(=1)}$ is stable.

Proof. Consider an arbitrary β-reduction starting from a linear λ-term:

$$t_0 = C[(\lambda x.t)\,u] \rightarrow_{\beta} C[t\{x := u\}] = t_1$$

Consider a subterm $\lambda y.v_1$ of t_1. It is the residual of a subterm $\lambda y.v_0$ of t_0. There are three cases:

- it is a subterm of u: then $v_1 = v_0$, and $|v_1|_x = 1$;
- it is a subterm of t: then $v_1 = v_0\{x := u\}$, but y is not free in u (otherwise it would be captured), and $|v_1|_y = |v_0\{x := u\}|_y = |v_0|_y = 1$;
- it is a superterm of the whole redex, then there are three subcases: if the unique occurrence of y is outside the redex; if it is in t; or if it is in u. The two first cases are easy; for the last one we use $|t|_x = 1$. In each case, we conclude $|v_1|_y = 1$.

□

The proof is not difficult, but it is tedious. Fortunately, we can characterise stability in a slightly more practical way.

Lemma 5.2 *$\mathcal{P}$ is stable if and only if*

$$\forall m \geq 0, n \geq 0, 0 \leq k \leq n. \left(\mathcal{P}(m) \wedge \mathcal{P}(n) \implies \mathcal{P}(n + k \cdot m - k)\right).$$

Proof. First remark that, with the notations of the lemma, $n + k \cdot m - k \geq 0$, because $n - k \geq 0$ and $k \cdot m \geq 0$. Let's consider an arbitrary β-reduction under an arbitrary binder (if the reduction is not under a binder, this is irrelevant to the kind of conditions we have or easy to deal with):

$$\lambda y.C[(\lambda x.t)\, u] \to_\beta \lambda y.C[t\{x := u\}].$$

Let us write $m = |t|_x$, $n = |C[(\lambda x.t)\, u]|_y$ and $k = |u|_y$. We thus have $|C[t]|_y = n - k$ and $|C[t\{x := u\}]|_y = n + k \cdot m - k$. The reduct thus belongs to $\lambda_{\mathcal{P}}$ if and only if $\mathcal{P}(n + k \cdot m - k)$ holds for all n and k (corresponding to every choice of outer binder λy). Indeed, $\mathcal{P}$ is stable if and only if $\mathcal{P}(n + k \cdot m - k)$ holds whenever $\mathcal{P}(m)$ and $\mathcal{P}(n)$ hold. □

Theorem 5.3 *$\mathcal{P}$ is stable (i.e. the $\lambda_{\mathcal{P}}$-calculus is well-defined) if and only if*

$$\forall m \geq 0, n \geq 1. \left(\mathcal{P}(m) \wedge \mathcal{P}(n) \implies \mathcal{P}(m + n - 1)\right).$$

Proof. Using Lemma 5.2 and the fact that the other implication is trivial, we assume that $\forall m \geq 0, n \geq 1. (\mathcal{P}(m) \wedge \mathcal{P}(n) \implies \mathcal{P}(m + n - 1))$ and we only have to show that $\mathcal{P}(n + k \cdot m - k)$ holds if $m \geq 0$, $n \geq 0$, $0 \leq k \leq n$, $\mathcal{P}(m)$ and $\mathcal{P}(n)$. If $n = 0$, this is trivially true, because $k = 0$, thus $n + k \cdot m - k = 0$, and $\mathcal{P}(0) = \mathcal{P}(n)$ holds by hypothesis. We may thus assume $n \geq 1$, and we prove the statement by induction on k. If $k = 0$, it is true because $n + k \cdot m - k = n$ and $\mathcal{P}(n)$ holds by hypothesis. Let $0 \leq k \leq n - 1$ and assume $\mathcal{P}(n + k \cdot m - k)$ holds. Then, $\mathcal{P}(n + (k + 1) \cdot m - (k + 1)) = \mathcal{P}(m + (n + k \cdot m - k) - 1)$ holds using the assumption, the induction hypothesis, and the fact that $n + k \cdot m - k \geq 1$ because $k \leq n - 1$ and $k \cdot m \geq 0$. We indeed conclude that the statement holds for all k such that $0 \leq k \leq n$. □

6 Examples

Example 6.1 We use Theorem 5.3 to recover some well-known stability results.

- $\lambda_\top$ (aka. the λ-calculus) is stable;
- $\lambda_{(=1)}$ (aka. the linear λ-calculus) is stable;
- $\lambda_{(\geq 1)}$ (aka. λI) is stable;
- $\lambda_{(\leq 1)}$ (aka. the affine λ-calculus) is stable.

Proof. We only show the proof for $\lambda_{(\geq 1)}$. Assume $m \geq 1$ and $n \geq 1$, then $m + n - 1 \geq 1 + 1 - 1 = 1$. Using Theorem 5.3, we conclude that (≥ 1) is stable. The proof shows that 1 plays a special role in this framework. □

Example 6.2 There are also some less usual sub-calculi, with a more questionable computational content.

- $\lambda_\perp$ (where there is no λ-abstraction) is stable (under the strict convention, this calculus is empty);
- $\lambda_{(=0)}$ (where there is no occurrence of bound variables) is stable (under the strict convention, this calculus is empty);
- $\lambda_{(\geq 2)}$ is stable;
- more generally, if $b \geq 1$, $\lambda_{(\geq b)}$ is stable;
- however, if $b \geq 2$, $\lambda_{(\leq b)}$ is not stable.

Proof. The first two sub-calculi are degenerated, which is evidenced by the fact that the condition in Theorem 5.3 is true because the premises of the implication can never be satisfied. Let $b \geq 1$, we verify that $\lambda_{(\geq b)}$ is stable. Let $m \geq b$ and $n \geq b$, then $m + n - 1 \geq 2 \cdot b - 1 \geq b$, since $b - 1 \geq 0$. Thus $\lambda_{(\geq b)}$ is stable. □

Using Theorem 5.3, we give some non-trivial sub-calculi (or non-sub-calculi) of the λ-calculus (of course only those of the form $\lambda_{\mathcal{P}}$ for some $\mathcal{P}$).

Example 6.3 Let $\mathsf{odd}(n) = (\exists k \geq 0. n = 1 + 2 \cdot k)$, then odd is stable. The "odd calculus" λ_{odd} is a simple, non-trivial sub-λ-calculus.

Proof. Assume $\mathsf{odd}(m)$ and $\mathsf{odd}(n)$. Then, there exist $k, k' \geq 0$ such that $m = 1 + 2 \cdot k$ and $n = 1 + 2 \cdot k'$. Then $m + n - 1 = (1 + 2 \cdot k) + (1 + 2 \cdot k') - 1 = 1 + 2 \cdot (k + k')$ with $k + k' \geq 0$, and indeed $\mathsf{odd}(m + n - 1)$ holds. □

Remark 6.4 The "even calculus" λ_{even} defined by $\mathsf{even}(n) = (\exists k. n = 2 \cdot k)$ is not stable (we are therefore reluctant to call it a calculus). This can be seen as a consequence of Theorem 5.3 or directly: $\lambda y.((\lambda x.x\,x)\,y\,y) \to_\beta \lambda y.y\,y\,y$.

In fact, the previous example can be generalised to the following large class of sub-λ-calculi.

Example 6.5 Let $q \geq 1$ and $\mathsf{mult}_q(n) = (\exists k \geq 0. n = 1 + k \cdot q)$, then $\lambda_{\mathsf{mult}_q}$ is stable.

Proof. Similarly, using $(1 + k \cdot q) + (1 + k' \cdot q) - 1 = 1 + (k + k') \cdot q$. □

We will see in Sections 7 and 8 that essentially all $\lambda_{\mathcal{P}}$-calculi can be decomposed in calculi of this form.

7 Classification

With Theorem 5.3 in hand, we characterise further the sub-calculi of the λ-calculus (of the form $\lambda_{\mathcal{P}}$ for some stable $\mathcal{P}$).

Proposition 7.1 *If $\mathcal{P}$ and $\mathcal{Q}$ are stable, then $\mathcal{P} \wedge \mathcal{Q}$ is stable.*

Proof. Consequence of Proposition 3.7 and of the fact that $\lambda_{\mathcal{P}\wedge\mathcal{Q}} = \lambda_{\mathcal{P}} \cap \lambda_{\mathcal{Q}}$, or using Theorem 5.3. □

Remark 7.2 If $\mathcal{P}$ and $\mathcal{Q}$ are stable, $\mathcal{P} \vee \mathcal{Q}$ is not necessarily stable.

Proof. First, we notice that we cannot use Proposition 3.7, because in general $\lambda_{\mathcal{P}} \cup \lambda_{\mathcal{Q}} \subsetneq \lambda_{\mathcal{P}\vee\mathcal{Q}}$. Let $\mathcal{P}(n) = (\exists k \geq 0. n = 1+2\cdot k)$ and $\mathcal{Q}(n) = (\exists k \geq 0. n = 1+3\cdot k)$. According to Example 6.5, $\mathcal{P}$ and $\mathcal{Q}$ are stable. $(\mathcal{P} \vee \mathcal{Q})(3)$ holds since $\mathcal{P}(3)$ holds, $(\mathcal{P} \vee \mathcal{Q})(4)$ holds since $\mathcal{Q}(4)$ holds, but $(\mathcal{P} \vee \mathcal{Q})(3+4-1) = (\mathcal{P} \vee \mathcal{Q})(6)$ does not hold since neither $\mathcal{P}(6)$ or $\mathcal{Q}(6)$ holds. In other words, $\mathcal{P} \vee \mathcal{Q}$ is not stable. A concrete witness illustrating this is $\lambda y.((\lambda x.x\,x\,x)\,y\,y\,y\,y) \to_\beta \lambda y.y\,y\,y\,y\,y\,y$. □

Proposition 7.3 *If $\mathcal{P}$ is stable and $\mathcal{P}(2)$ holds then $\mathcal{P}(n)$ holds for all $n \geq 2$.*

Proof. By induction on $n \geq 2$. $\mathcal{P}(2)$ holds by hypothesis. Assume $\mathcal{P}(n)$ holds, then using Theorem 5.3, $\mathcal{P}(n+1) = \mathcal{P}(n+2-1)$ also holds. □

Proposition 7.4 *If $\mathcal{P}$ is stable and if $\mathcal{P}(0)$ and $\mathcal{P}(n)$ hold for some $n \geq 2$, then $\mathcal{P}(n)$ holds for all $n \geq 0$. In other words, we get the full λ-calculus.*

Proof. We show that $\mathcal{P}(k)$ holds for $0 \leq k \leq n$ by reverse induction on k. $\mathcal{P}(n)$ holds by hypothesis. Let $1 \leq k \leq n$ and assume that $\mathcal{P}(k)$ holds. Then $\mathcal{P}(k-1) = \mathcal{P}(0 + k - 1)$ holds using Theorem 5.3, stability of $\mathcal{P}$, the induction hypothesis, and the facts that $k \geq 1$ and $\mathcal{P}(0)$ holds. In particular, $\mathcal{P}(2)$ holds and we use Proposition 7.3. □

The two previous propositions are somehow analogous to the decomposition in combinators of interaction nets [3], a popular sharing-sensitive framework, well-suited to implement the λ-calculus. Although we do not learn anything new or surprising here, this shows that our framework is adapted to formalise that kind of folklore, but otherwise informal, theorems about sharing issues.

As evidenced in Remark 7.2, disjunction is not a well-behaved operation with respect to stability. However, the following proposition exhibits the particular behaviour of 1, and tends to show that, to some extent, the choice of $\mathcal{P}(1)$ is not relevant for the stability of $\mathcal{P}$.

Proposition 7.5 (i) *If $\mathcal{P}$ is stable, then $\mathcal{P} \vee (=1)$ is stable;*

(ii) *if $\mathcal{P} \vee (=1)$ is stable and not everywhere true, then $\mathcal{P}$ is stable.*

Proof.

(i) Assume $\mathcal{P}$ is stable, $(\mathcal{P}\vee(=1))(m)$ and $(\mathcal{P}\vee(=1))(n)$. If $m = 1$, then $m+n-1 = n$ and $(\mathcal{P} \vee (=1))(m + n - 1)$ holds; and similarly if $n = 1$. Otherwise, both $\mathcal{P}(m)$ and $\mathcal{P}(n)$ hold, and $(\mathcal{P} \vee (=1))(m + n - 1)$ indeed holds.

(ii) Assume $\mathcal{P} \vee (=1)$ is stable, $\mathcal{P}(m)$ and $\mathcal{P}(n)$ hold. Then $(\mathcal{P} \vee (=1))(m + n - 1)$ holds. Either $\mathcal{P}(m + n - 1)$ holds and we are done, or $m + n - 1 = 1$, hence $m = n = 1$ and $\mathcal{P}(1)$ holds, because the case $m = 0$ and $n = 2$ is excluded by Proposition 7.4.

□

Lemma 7.6 *If $\mathcal{P}$ is stable and there exists $n \geq 2$ such that $\mathcal{P}(n)$ holds, then there exists $q \geq 1$ such that $\mathcal{P}(1 + k \cdot q)$ holds for every $k \geq 1$.*

Proof. With the notations of the lemma, let $q = n - 1$. We prove by induction on $k \geq 1$ that $\mathcal{P}(1 + k \cdot q)$ holds. This is true for $k = 1$. Assume $\mathcal{P}(1 + k \cdot q)$ holds, $n + (1 + k \cdot q) - 1 = 1 + (k + 1) \cdot q$ and $\mathcal{P}(1 + (k + 1) \cdot q)$ holds, using Theorem 5.3.□

We now have everything in hand to exhibit a complete classification of the $\lambda_{\mathcal{P}}$-calculi.

Theorem 7.7 *$\mathcal{P}$ is stable if and only if one of the following holds for all n:*

(i) *$\mathcal{P}(n) \Leftrightarrow \bot$;*

(ii) *$\mathcal{P}(n) \Leftrightarrow \top$;*

(iii) *$\mathcal{P}(n) \Leftrightarrow (n = 0)$;*

(iv) *$\mathcal{P}(n) \Leftrightarrow (n = 0 \vee n = 1)$;*

(v) *there exist $0 \leq p \leq \omega$ and $1 \leq q_1 < \ldots < q_p$ pairwise non divisible such that: $\mathcal{P}(n) \Leftrightarrow (\exists k_1, \ldots, k_p \geq 0.n = 1 + \sum_{1\leq i\leq p} k_i \cdot q_i)$;*

(vi) *there exist $1 \leq p \leq \omega$ and $1 \leq q_1 < \ldots < q_p$ pairwise non divisible such that: $\mathcal{P}(n) \Leftrightarrow (\exists k_1, \ldots, k_p \geq 0, 1 \leq j \leq p.k_j \geq 1 \wedge n = 1 + \sum_{1\leq i\leq p} k_i \cdot q_i)$.*

Moreover, this decomposition is unique.

Proof. If one of the cases (i–iv) holds, it has already been noted in Section 6 that $\mathcal{P}$ is stable. If (v) or (vi) holds, this is a consequence of Theorem 5.3, similar to Example 6.5. Conversely, suppose $\mathcal{P}$ is stable. We distinguish cases according to whether or not $\mathcal{P}(0)$ holds.

- If $\mathcal{P}(0)$ holds, does there exist $n \geq 2$ such that $\mathcal{P}(n)$ holds ?
  - · If there is such a n, we are in case (ii), thanks to Proposition 7.4.
  - · Otherwise, we are indeed in case (iii) or (iv).
- If $\mathcal{P}(0)$ does not hold, we look at $\mathcal{P}(1)$.
  - · If $\mathcal{P}(1)$ holds, we prove by induction on $p \geq 0$ that there exist $1 \leq q_1 < \ldots < q_p$ pairwise non divisible such that $(\exists k_1, \ldots, k_p \geq 0.n = 1+\sum_{1\leq i\leq p} k_i \cdot q_i) \Rightarrow \mathcal{P}(n)$. This is true for $p = 0$. Assume this is true for some p, and consider the

smallest n not equal to $(1 + \sum_{1 \leq i \leq p} k_i \cdot q_i)$ for some $k_1, \ldots, k_p \geq 0$ such that $\mathcal{P}(n)$ holds. There are two cases. If there is no such n, that means that the condition is verified and $\mathcal{P}$ is fully described. Otherwise, let $q_{p+1} = n - 1$. Indeed, by construction, $q_{p+1} > q_p$ and none of $q_1, \ldots, q_p$ is a divisor of q_{p+1}. Thanks to Theorem 5.3 and in a similar way to Lemma 7.6, for all $k_{p+1} \geq 0$, $\mathcal{P}(1 + k_{p+1} \cdot q_{p+1})$. Then, using again Theorem 5.3, the statement holds for $p + 1$. If the process stops, the equivalence is clear. If it does not, let's write $\mathcal{P}_p(n) = (\exists k_1, \ldots, k_p \geq 0.n = 1 + \sum_{1 \leq i \leq p} k_i \cdot q_i)$. For all p, $\mathcal{P}_p \Rightarrow \mathcal{P}_{p+1} \Rightarrow \mathcal{P}$ where the first implication is strict. The sequence $(\mathcal{P}_p)_p$ is strictly increasing and bounded, it thus has a limit $\mathcal{P}_\omega$. There is no n such that $\mathcal{P}(n)$ but not $\mathcal{P}_\omega(n)$, because this would contradict the construction. We conclude $\mathcal{P} \Leftrightarrow \mathcal{P}_\omega$.

· If $\mathcal{P}(1)$ does not hold, let's consider the smallest $n \geq 2$ such that $\mathcal{P}(n)$ holds. If there is no such n, we are in case (i). Otherwise, we can proceed as in the previous case, starting at $p = 1$, with $q_1 = n - 1$, and obtain case (vi).

Unicity is clear: the different cases do not overlap, and in cases (v) or (vi), the non-pairwise divisibility of $q_1, \ldots, q_p$ ensures that there is no redundancy. □

Remark 7.8 Theorem 7.7 gives a complete classification of the $\lambda_{\mathcal{P}}$-calculi in terms of equality, but this is not necessarily the "best" description. For instance, we have seen that (≥ 3) is stable, but its description using Theorem 7.7 is case (vi) with $p = \omega$ and q_i is the i-th prime number. In particular, it is not a finite description.

8 Alternative Presentation

Theorem 7.7 gives a complete description of the $\lambda_{\mathcal{P}}$-calculi. It also incites us to look more closely at the apparently more interesting cases, namely cases (v) and (vi). Let us focus on case (v), which means that we restrict our attention to predicates $\mathcal{P}$ such that $\mathcal{P}(1)$ holds but $\mathcal{P}(0)$ does not. All results in this section hold only for this restricted class.

As we have seen, disjunction is not well-behaved with respect to stability. Let us then introduce a well-behaved notion.

Definition 8.1 We define the *sum* of two predicates: $\mathcal{P} \oplus \mathcal{Q} = \overline{\mathcal{P} \vee \mathcal{Q}}$.

Definition 8.2 We say that $\mathcal{P}$ is *irreducible* if there are no $\mathcal{Q}_1$, $\mathcal{Q}_2$ such that $\mathcal{P} = \mathcal{Q}_1 \oplus \mathcal{Q}_2$.

Definition 8.3 We define the q^{th} elementary predicate $\underline{q}$ by for all $n \geq 0$, $\underline{q}(n) = (\exists k \geq 0.n = 1 + k \cdot q)$. It is just a new name for mult_q.

Lemma 8.4 $(\underline{p} \oplus \underline{q})(n) = (\exists k_1, k_2 \geq 0.n = 1 + k_1 \cdot p + k_2 \cdot q)$

Corollary 8.5 *$\underline{k \cdot p} \oplus \underline{p} = \underline{p}$. In particular $\underline{1} \oplus \underline{p} = \underline{1}$ and $\underline{0} \oplus \underline{p} = \underline{p}$.*

Notice that, in general, we do not have $\underline{p} \oplus \underline{q} = \underline{\gcd(p, q)}$, where $\gcd(p, q)$ is the greatest common divisor of p and q. This is because everything is positive, and we cannot "go backwards".

We can then reformulate Theorem 7.7 for the specific class under study in this section, in the following way:

Theorem 8.6 *The irreducibles are* exactly *the* $\underline{q}$*'s for* $q \geq 0$.

Theorem 8.7 *Every* $\mathcal{P}$ *can be decomposed in a sum of irreducibles, and there is a canonical such representation.*

We may also complete the structure with the corresponding *product* $\otimes = \wedge$. Then $\underline{p} \otimes \underline{q} = \underline{\mathrm{lcm}(p,q)}$, where $\mathrm{lcm}(p,q)$ is the least common multiple of p and q, hence there is no duality between $\otimes$ and $\oplus$.

In brief, we have essentially paraphrased previously stated results, in a less general way. Yet, this paraphrase exhibits an intriguing structure, which does not seem to be directly related to well-known algebraic structures. This should probably be taken as a hint that either this structure is interesting and deserves a thorough study, or there is an interesting structure on top of it that should suggest an interesting generalisation of the notion of subcalculus.

9 Conclusion

We have defined and given a complete characterisation of a class of subsystems of the λ-calculus taking into consideration the number of occurrences of variables, which is a crucial issue for sharing. We recover well-known calculi such as λI or the linear λ-calculus, but we also discover unconventional calculi whose interest as a computational model remains to study. Moreover, our characterisation is very algebraic and we hope that it will lead to a better understanding of the λ-calculus and its subsystems.

Acknowledgement

This article has benefited from comments by several anonymous referees.

References

[1] Barendregt, H. P., "The Lambda Calculus: Its Syntax and Semantics," Studies in Logic and the Foundations of Mathematics **103**, North-Holland Publishing Company, 1984, second, revised edition.

[2] Barendregt, H. P., "Lambda Calculi With Types," Handbook of Logic in Computer Science **2**, Oxford University Press, 1992, 117–309 pp.

[3] Lafont, Y., *Interaction combinators*, Information and Computation **137** (1997), pp. 69–101.

[4] Terese, "Term Rewriting Systems," Cambridge Tracts in Theoretical Computer Science **55**, Cambridge University Press, 2003.