A Canonical Form for Universe Levels in Impredicative Type Theory

Yoan Géran
Mines Paris PSL, Centre de Recherche en Informatique
Université Paris-Saclay, Laboratoire Méthodes Formelles, ENS Paris-Saclay

Abstract

The imax-successor algebra, where imax is the function defined by imax(n, 0) = 0 and imax(n, s(m)) = max(n, s(m)), is used to represent universe levels in impredicative type theory, in particular with universe polymorphism which introduces level variables, so it is present in proof systems such as Coq and Lean. In particular, we need to know when two elements of this algebra are equivalent, and we may also want to decide the inequality. In this article, we introduce a canonical form for the terms of this algebra, and we provide a canonization algorithm. It permits deciding level equivalence by checking the canonical form equality, and also permits easily checking if a level is smaller than another one.

2012 ACM Subject Classification Theory of computation → Type theory; Theory of computation → Equational logic and rewriting

Keywords and phrases universe levels, canonical form, impredicativity, imax algebra

Digital Object Identifier 10.4230/LIPIcs.CVIT.2016.23

1 Introduction

The formalization of mathematical theorems and the verification of software lead to the development of many logical systems. Predicate Logic is a quite general theory but does not allow for instance to quantify over predicates, preventing the expression of some propositions. Then, more powerful logic have been introduced through the years. This paper being motivated by the universe polymorphism in impredicative type theory, the introduction will briefly remind the history of these theories, to understand what they bring.

Pure Type Systems

A lot of theories are based on extensions of Church's simply-typed λ-calculus which does not permit to express terms over arbitrary types (preventing for instance to talk about all the groups). To address this, Martin-Löf introduces a dependent type theory with a type of all types [24], and later, to avoid paradoxes such as Girard's one, introduced a distinction between small types and large types (which are types containing types, and are also called universes) [26].

In the same years, Girard and Reynolds independently invented System F, an extension of Church's simply-typed λ-calculus with type polymorphism, and even later, Girard presented System F_ω which add type operators *i.e.* the ability to quantify on terms to create types.

The Calculus of Constructions [13] introduced by Coquand in his PhD thesis combined features from both Martin-Löf Type Theory and System F_ω. This system allows quantifying on types or terms to build new types and new terms, and it is the pinnacle of the λ-cube of Barendregt [4], which classifies type systems depending on the quantification possibilities.

The Calculus of Constructions is an elegant system with strong properties such as normalization and logical consistency. However, quantification over Type is not possible since it makes the system incoherent. This lead Coquand to generalize the system with a predicative hierarchy of universes [12], in the same way as predicative Martin-Löf type theory

42nd Conference on Very Important Topics (CVIT 2016).
Editors: John Q. Open and Joan R. Access; Article No. 23; pp. 23:1–23:23
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

[25]. They contain a countable sequence of universes $\mathtt{U}_0\colon \mathtt{U}_1\colon \cdots$, where $\mathtt{U}_0$ is the universe of the propositions, the indices being referred to as universes levels.

These logical systems are generalized under the name of *Pure Type Systems* [5, 6].

▶ **Definition 1.** *A Pure Type System (PTS) is defined by a set of sorts $\mathcal{S}$ (that corresponds to universes), a set of axioms $\mathcal{A} \subseteq \mathcal{S}^2$ and a set of rules $\mathcal{R} \subseteq \mathcal{S}^3$.*

$\mathcal{A}$ describes the sorts typing (s_1 has the type s_2 when $(s_1, s_2) \in \mathcal{A}$), and $\mathcal{R}$ describes the possible quantifications and their typing rules. The terms are the following, where $s \in \mathcal{S}$ and x ranges an infinite set of variables.

$$t ::= s \mid x \mid \Pi x\colon t \cdot t \mid (\lambda x\colon t \cdot t) \mid t\ t$$

and the typing rules are given in Figure 1.

$$(\textsc{Empty})\ \frac{}{[\,]\ \mathrm{WF}} \qquad (\textsc{Decl})\ \frac{\Gamma \vdash A\colon s \qquad x \notin \Gamma}{\Gamma, x\colon A\ \mathrm{WF}} \qquad (\textsc{Var})\ \frac{\Gamma\ \mathrm{WF} \qquad (x\colon A) \in \Gamma}{\Gamma \vdash x\colon A}$$

$$(\textsc{Sort})\ \frac{}{\vdash s_1\colon s_2}\ (s_1, s_2) \in \mathcal{A} \qquad (\textsc{Prod})\ \frac{\Gamma \vdash A\colon s_1 \qquad \Gamma, x\colon A \vdash B\colon s_2}{\Gamma \vdash \Pi x\colon A \cdot B\colon s_3}\ (s_1, s_2, s_3) \in \mathcal{R}$$

$$(\textsc{App})\ \frac{\Gamma \vdash t\colon \Pi x\colon A \cdot B \qquad \Gamma, \vdash u\colon A}{\Gamma \vdash t\ u\colon B[x := u]} \qquad (\textsc{Abs})\ \frac{\Gamma, x\colon A \vdash t\colon B \qquad \Gamma \vdash \Pi x\colon A \cdot B\colon s}{\Gamma \vdash \lambda x \cdot t\colon \Pi x\colon A \cdot B}$$

$$(\textsc{Conv})\ \frac{\Gamma \vdash B\colon s \qquad \Gamma \vdash t\colon A \qquad A \equiv_\beta B}{\Gamma \vdash t\colon B}\ s \in \mathcal{S}$$

Figure 1 Typing rules of PTS.

Both CC^∞ and predicative Martin-Löf type theory have a set of sorts indexed over the natural numbers, with for all $i \in \mathbb{N}$, $\mathtt{U}_i\colon \mathtt{U}_{i+1}$. Their difference reside in their set of rules.

Impredicativity

With the aim of building a consistent system, paradox such as Girard's one should be avoided. When Coquand analysed it, he found that a product from Type to Type could not live in Type: it should live in a greater type (hence the distinction between small and large types).

With an infinite hierarchy of universes, this principle remains: a product from $\mathtt{U}_i$ to $\mathtt{U}_j$ should live in a greater universe. Therefore, in the predicative Martin-Löf type theory, the set of rules is $\{\mathtt{U}_i, \mathtt{U}_j, \mathtt{U}_{\max(i,j)}\}$. The choice of CC^∞ is different (and does not break the consistency either): a product from $\mathtt{U}_i$ to $\mathtt{U}_0$ lives in $\mathtt{U}_0$, so it follows the rules $\{\mathtt{U}_i, \mathtt{U}_j, \mathtt{U}_{\mathrm{imax}(i,j)}\}$ where $\mathrm{imax}\colon \mathbb{N} \to \mathbb{N} \to \mathbb{N}$ is defined for all $i, j \in \mathbb{N}$ by $\mathrm{imax}(i, 0) = 0$ and $\mathrm{imax}(i, j+1) = \max(i, j+1)$.

This corresponds to the so-called impredicativity of Prop (hence the name imax for impredicative max) which notably permits to say that we can quantify over all the propositions and still get a new proposition. It is a philosophical questioning: should $\Pi P\colon \mathrm{Prop}, P \to P$ be considered as a proposition since it is created by quantifying over all the propositions?

Universe Polymorphism

A PTS can be enriched with universe polymorphism which allows the user to quantify over universes [27, 22, 14]. For instance, it permits to declare simultaneously the identity for all the types of any universes with $\lambda s\colon \mathcal{S} \cdot \lambda A\colon s \cdot \lambda x\colon A \cdot x$. This feature adds universe variables to the language of a PTS. In the case of CC^∞, it is equivalent to extend the syntax of the levels with level variables.

▶ **Definition 2** (Levels)**.** *A level is a term of the grammar*

$$\ell \coloneqq 0 \mid S(\ell) \mid \max(\ell, \ell) \mid \mathrm{imax}(\ell, \ell) \mid x$$

where x is an element of a countable set of variables $\mathcal{X}$. We denote by $\mathfrak{L}$ the set of the levels, and we say that a level is concrete if it does not contain any variable.

▶ **Definition 3.** *We call $CC^\infty_\forall$ the extension of CC^∞ with universe polymorphism.*

The universe polymorphic identity of $CC^\infty_\forall$ is the term

$$\mathrm{id} ::= \lambda i \colon \mathfrak{L}, \lambda A \colon \mathtt{U}_i, \lambda x \colon A, x.$$

We can use it by instantiating the level variable. For instance, id 1 Prop is the identity of Prop while id 2 $\mathtt{U}_1$ is $\mathtt{U}_1$'s one. This instantiation is done throughout substitution functions, which replace a level variable by a level, and valuation functions which replace level variables by integers.

▶ **Definition 4** (Valuation)**.** *A function $\sigma \colon \mathcal{X} \to \mathbb{N}$ is called a valuation. For all valuations σ, we define inductively the value of a level ℓ over σ, denoted as $[\![\ell]\!]_\sigma$, with*

$$[\![0]\!]_\sigma = 0 \qquad [\![S(\ell)]\!]_\sigma = S([\![\ell]\!]_\sigma) \qquad [\![x]\!]_\sigma = \sigma(x)$$
$$[\![\max(\ell_1, \ell_2)]\!]_\sigma = \max\big([\![\ell_1]\!]_\sigma, [\![\ell_2]\!]_\sigma\big) \qquad [\![\mathrm{imax}(\ell_1, \ell_2)]\!]_\sigma = \mathrm{imax}\big([\![\ell_1]\!]_\sigma, [\![\ell_2]\!]_\sigma\big)$$

This interpretation through the valuations explains why, even if levels are abstract terms, we defined them with the same symbols 0, s, max and imax that are used for the natural numbers. Indeed, the concrete levels can clearly be identified as the natural numbers and the levels' semantic, through the valuations, justifies to use the same symbol and permits to see the valuations as functions that *realise* levels, turning them into concrete ones.

Besides, two levels can also be compared using these valuations. They are equivalent if they give the same concrete levels through any valuation.

▶ **Definition 5** (Level comparison)**.** *Let $\ell_1, \ell_2 \in \mathfrak{L}$. We say that $\ell_1 \leqslant \ell_2$ if for all valuations σ, $[\![\ell_1]\!]_\sigma \leqslant [\![\ell_2]\!]_\sigma$. In the same way, we say that $\ell_1 \equiv \ell_2$ if for all valuations σ, $[\![\ell_1]\!]_\sigma = [\![\ell_2]\!]_\sigma$. Hence, $\ell_1 \equiv \ell_2$ if and only if $\ell_1 \leqslant \ell_2$ and $\ell_2 \leqslant \ell_1$.*

This equivalence shows that universe such as $\mathtt{U}_x$ and $\mathtt{U}_{\max(x,x)}$ should be identified. However, it is not obvious to check. It is not syntactic, like it was without universe polymorphism, and the imax function makes it complicated.

The aim of this paper is to address this problem. To do so, we study the imax-successor algebra, and we provide a canonical form for its terms, hence a way to decide level equivalence by syntactic comparison of the canonical form.

Motivation

Our main motivation lies in the interoperability between proof systems Indeed, it became a big challenge in the research on proof-checking, which aims to avoid the redevelopment of the same proof. Instead of developing translators from each system to another one, *logical frameworks* propose to define theories in a common language, which makes translation easier.

The $\lambda\Pi$-calculus modulo rewriting ($\lambda\Pi/\equiv$) [10] is a logical framework that extends $\lambda\Pi$ (the simply-typed λ-calculus with dependent types) with higher-order rewrite rules [16, 28] that can be used to define functions, but also types; terms are then identified modulo β and these rewrite rules. The computational part of the type theories can then be represented using the expressiveness of rewrite systems.

The Calculus of Constructions and its subtheories can be expressed in $\lambda\Pi/\equiv$ [8], and, in [15], Cousineau and Dowek showed how to express some PTS. Therefore, several systems have been encoded in $\lambda\Pi/\equiv$: HOL-Light [29, 1], Agda [19], Matita [1], but also parts of Coq [18, 9]. Besides, since there exist multiple implementations of $\lambda\Pi/\equiv$ such as Dedukti [2], Lambdapi [23], or Kontroli [17], these embeddings have been implemented, leading to effective translations [29, 20, 21].

To define $\mathrm{CC}^{\infty}_{\forall}$ in $\lambda\Pi/\equiv$, we need to define its levels. It can be done with a type `nat` together with functions `max`, and `imax`, and rewrite rules to define them. This permits to express CC^{∞} in $\lambda\Pi/\equiv$, but the equivalence relation that comes with level variables adds some difficulties.

Indeed, for all term u of $\mathrm{CC}^{\infty}_{\forall}$, let us note $|u|$ its translation in $\lambda\Pi/\equiv$, and let us consider a function $f\colon \mathtt{U}_i \to \mathtt{U}_j$ and a term $t\colon \mathtt{U}_k$ where $k \equiv i$. Since $f\ t$ is well-typed, then $|f\ t|$ should be well-typed in $\lambda\Pi/\equiv$. Therefore, $|t|$ should have the type $|\mathtt{U}_i|$, whereas it has the type $|\mathtt{U}_k|$. We deduce that $|\mathtt{U}_i|$ and $|\mathtt{U}_k|$ should be convertible types, and then that equivalent levels should be convertible terms. With a canonical form and rewrite rules to compute it, this statement becomes decidable!

Related Work

The max-successor algebra is well-studied, and so, some solutions exist in the predicative case. In [30], Voevodsky represented each level as $\max(n, n_1 + x_1, \ldots, n_k + x_k)$ where $n \geqslant \max(n_1, \ldots, n_k)$. Then, if there exists $i \neq j$ such that $x_j = x_i$, we simplify the term and keep only $\max(n_i, n_j) + x_i$. Therefore, we obtain a minimal representation for the max-successor algebra.

In [19], Genestier encoded the universe polymorphism of Agda in $\lambda\Pi/\equiv$ using a similar idea and a representation modulo associativity and commutativity (for the max symbol), and Blanqui gave another presentation of this algebra in [7], with an encoding without matching modulo associativity and commutativity.

The imax-successor algebra is less studied. [3] proposed an encoding, but it does not fully reflect the equalities; for instance, the levels $\max(\mathrm{imax}(x, y), x)$ and $\max(x, y)$ are not convertible. Besides, Férey also worked on the encoding of universe polymorphism [18].

Finally, an algorithm to check level inequality, and so level equivalence, is presented in [11], but it does not rely on a canonical form.

Outline

In Section 2, we study the imax-successor algebra and extend it in order to propose a representation. Then, Section 3 shows that this leads to a canonical form which is generalized to the level extension in Section 4. A canonization algorithm is given in Section 5.

2 Level Representation

To begin with, let us present our procedure. We use the same idea presented above in the predicative case: find a subset E of levels such that any level can be represented as $\max U$

with $U \subset E$, and such that $\max U$ has a minimal representation that ensures this uniqueness property:

$$\max U \equiv \max V \iff U = V.$$

In the predicative case, $E = \mathbb{N} \cup \{n + x, n \in \mathbb{N}, x \in \mathcal{X}\}$; and the minimal representation consists in having one term n (the maximum of two integers can be simplified) and for all $x \in \mathcal{X}$ at most one term $n + x$ since $\max(n + x, m + x) = \max(n, m) + x$. To obtain the canonical representation, we push the successor symbols inside the max, and we obtain $\max U$ with $U \subset E$. Then, U can be simplified by removing u if there exists $v \in U$ such that $v \neq u$ and $u \leqslant v$, leading to the minimal representation.

This gives us the intuition that we need. An element of E should be very basic and simple in the sense that it is not equivalent to a maximum of other levels.

In this first section, we study the imax-successor algebra and its equivalences in order to find such a subset of levels.

2.1 Levels as Maximum

The very first step to simplify the terms is to pull the max symbol out and then to express any level as a maximum of levels that do not contain any max, that is the principle of our idea. The successor can be distributed over max since for all $u, v \in \mathfrak{L}$, $S(\max(u, v)) \equiv \max(S(u), S(v))$, and the two next propositions show how to distribute imax over max.

▶ **Proposition 6.** *For all $u, v, w \in \mathfrak{L}$,*

$$\operatorname{imax}(u, \max(v, w)) \equiv \max(\operatorname{imax}(u, v), \operatorname{imax}(u, w)).$$

Proof. Let $t = \operatorname{imax}(u, \max(v, w))$, $t_1 = \operatorname{imax}(u, v)$ and $t_2 = \operatorname{imax}(u, w)$, and let σ be a valuation.

- If $[\![v]\!]_\sigma = [\![w]\!]_\sigma = 0$, then $[\![\max(t_1, t_2)]\!]_\sigma = 0 = [\![t]\!]_\sigma$.
- If $[\![v]\!]_\sigma \neq 0$ and $[\![w]\!]_\sigma = 0$, then $[\![\max(t_1, t_2)]\!]_\sigma = \max([\![u]\!]_\sigma, [\![v]\!]_\sigma) = [\![t]\!]_\sigma$.
- If $[\![v]\!]_\sigma = 0$ and $[\![w]\!]_\sigma \neq 0$, then $[\![\max(t_1, t_2)]\!]_\sigma = \max([\![u]\!]_\sigma, [\![w]\!]_\sigma) = [\![t]\!]_\sigma$.
- Else, $[\![\max(t_1, t_2)]\!]_\sigma = \max([\![u]\!]_\sigma, [\![v]\!]_\sigma, [\![w]\!]_\sigma) = [\![t]\!]_\sigma$.

◀

▶ **Proposition 7.** *For all $u, v, w \in \mathfrak{L}$,*

$$\operatorname{imax}(\max(u, v), w) \equiv \max(\operatorname{imax}(u, w), \operatorname{imax}(v, w)).$$

Proof. Let σ be a valuation. If $[\![w]\!]_\sigma = 0$, then both terms are evaluated to 0. Else, they are evaluated to $\max([\![u]\!]_\sigma, [\![v]\!]_\sigma, [\![w]\!]_\sigma)$. ◀

Then, any level can be expressed as a maximum of levels without max. Note that for this, we consider that max takes a set of levels as argument. We obtain this theorem.

▶ **Theorem 8.** *For all $t \in \mathfrak{L}$, there exists $u_1, \ldots, u_n$ in the grammar*

$$\ell \coloneqq 0 \mid S(\ell) \mid \operatorname{imax}(\ell, \ell) \mid x$$

such that $t \equiv \max(u_1, \ldots, u_n)$.

2.2 Simplification of the Levels

We can now focus on levels without maximum. The uniqueness property sought for the representation requires the levels to be very basic, and then search to simplify them.

The main issue is imax: its asymmetry complicates its interaction with other symbols. The previous equivalences show how to remove the interaction between imax and max, now, we will study how imax interacts with the other symbols. We aim to restrict the localisation of the imax symbol to specific parts of the levels in order to understand and control their influence on the levels semantic.

Firstly, we recall these equivalences that are direct consequences of the semantic of imax. They permit to deal with 0 and the successor.

▶ **Proposition 9.** *For all $u, v \in \mathfrak{L}$,*

$$\text{imax}(u, 0) \equiv 0 \qquad \text{imax}(0, v) \equiv v \qquad \text{imax}(u, S(v)) \equiv \max(u, S(v))$$

And we show how to remove imax symbol in second argument of imax.

▶ **Proposition 10.** *For all $u, v, w \in \mathfrak{L}$,*

$$\text{imax}(u, \text{imax}(v, w)) \equiv \max(\text{imax}(u, w), \text{imax}(v, w)).$$

Proof. Let σ be a valuation. If $[\![w]\!]_\sigma = 0$, then both terms are evaluated to 0. Else, they are evaluated to $\max([\![u]\!]_\sigma, [\![v]\!]_\sigma, [\![w]\!]_\sigma)$. ◀

Thus, we can consider that the second argument of imax is always a variable. It is more complicated to directly enforce the form of its first argument, but we can obtain one restriction by distributing S over imax. However, we cannot do it as directly as we distribute the S over max, as shown in the next example.

▶ **Example 11.** We consider the levels $t_1 = S(\text{imax}(y, x))$ and $t_2 = \text{imax}(S(y), S(x))$. By considering a valuation σ such that $\sigma(x) = 0$ and $\sigma(y) = 1$, $t_1 \not\equiv t_2$.

▶ **Proposition 12.** *For all $u, v \in \mathfrak{L}$,*

$$S(\text{imax}(u, v)) \equiv \max(S(v), \text{imax}(S(u), v)).$$

Proof. Let σ be a valuation. If $[\![v]\!]_\sigma = 0$, then both terms are evaluated to 1. Else they are evaluated to $S(\max([\![u]\!]_\sigma, [\![v]\!]_\sigma))$. ◀

Finally, all of these propositions lead to this grammar restriction.

▶ **Theorem 13.** *For all $t \in \mathfrak{L}$, there exists $u_1, \ldots, u_n$ in the grammar*

$$\ell := S^{k+1}(x) \mid S^k(0) \mid \text{imax}(\ell, x)$$

such that $t \equiv \max(u_1, \ldots, u_n)$.

▶ Remark 14. For all t in the grammar of Theorem 13, there exists $x_1, \ldots, x_n \in \mathcal{X}$, and $v = S^k(0)$ or $v = S^{k+1}(x)$ such that $t = \text{imax}(\text{imax}(\text{imax}(\cdots \text{imax}(v, x_1), x_2) \cdots)), x_{n-1}), x_n)$. We will note such a term t by $[v, x_1, \ldots, x_n]$.

2.3 Introducing New Levels

Here, we continue the simplification process in order to find simple enough terms to reach the uniqueness property. Indeed, the terms of the grammar of Theorem 13 are still not simpler enough.

▶ **Example 15.** Let us consider $t = \max(\text{imax}(x, y), x)$. Then, $t \equiv \max(x, y)$.

The problem is the following: if $\text{imax}(x, y)$ permits to consider x if y is not zero, it also consider y in all cases. Then, it is redundant with y and lead to the equivalence $\text{imax}(x, y) \equiv \max(y, \text{imax}(x, y))$. We would like to obtain $\text{imax}(x, y) = \max(y, t)$ with some level t, but $\text{imax}(x, y)$ cannot be simplified more.

In fact, the second argument of imax has too many responsibilities since it should be taken into account, but it is also a condition to take into account the first argument.

This leads us to think that these responsibilities should be separated by introducing a term $f(x, y)$ such that $[\![f(x, y)]\!]_\sigma$ is 0 if $[\![y]\!]_\sigma = 0$ and $[\![x]\!]_\sigma$ otherwise. This permits us to simplify $\text{imax}(x, y)$ into $\max(y, f(x, y))$, and since $f(x, y) \leq x$, $\max(y, f(x, y), x)$ can be turned into $\max(y, x)$.

Since, the imax are nested in the grammar of Theorem 13, we may need to have multiple variables as conditions; we generalize this idea of new terms and extend the level's grammar with two symbols $\mathcal{V}$ and $\mathcal{C}$.

▶ **Definition 16** (Extended levels). *An extended level is a term of the grammar*

$$\ell := 0 \mid S(\ell) \mid \max(\ell, \ell) \mid \text{imax}(\ell, \ell) \mid x \mid \mathcal{V}(\{\ell, \ldots, \ell\}, \ell, k) \mid \mathcal{C}(\{\ell, \ldots, \ell\}, k)$$

where $k \in \mathbb{N}$. We extend $[\![\cdot]\!]_\sigma$ and the level comparison to the extended levels with

$$[\![\mathcal{V}(E, u, k)]\!]_\sigma = \begin{cases} 0 & \text{if } \exists v \in E, [\![v]\!]_\sigma = 0 \\ [\![u]\!]_\sigma + k & \text{else} \end{cases}$$

$$[\![\mathcal{C}(E, k)]\!]_\sigma = \begin{cases} 0 & \text{if } \exists u \in E, [\![u]\!]_\sigma = 0 \\ k & \text{else} \end{cases}$$

We denote by $\mathfrak{L}^+$ the set of extended levels.

The symbols $\mathcal{V}$ and $\mathcal{C}$ stand for 'variable sublevel' and 'constant sublevel' in the sense that their semantic consists in taking into account a non-constant or a constant extended level u when a set of extended levels E does not contain a null one.

▶ **Definition 17.** *We denote by $\mathfrak{S}$ the set of sublevels. Let $u \in \mathfrak{S}$, $u = \mathcal{V}(E, v, k)$ or $u = \mathcal{C}(E, k)$. We call E the* verification conditions *of u denoted by $\text{VC}(u)$, and k is its* constant part *denoted by $\omega(u)$. We also define the* variable part *of u denoted by $\nu(u)$ which is 0 in the case of a constant sublevel and v in the case of a variable sublevel.*

Besides, we said that a verification condition u is checked *(by a valuation σ) if $[\![u]\!]_\sigma \neq 0$ and we said that a sublevel u is active if $[\![u]\!]_\sigma \neq 0$.*

For all $u \in \mathfrak{S}$ and for all valuations σ, we then have $[\![u]\!]_\sigma = \omega(u) + [\![\nu(u)]\!]_\sigma$ if u is active and 0 otherwise. Moreover, u is active if its verifications conditions are checked and $\omega(u) + [\![\nu(u)]\!]_\sigma \neq 0$.

If $\mathfrak{L}$ is semantically, and even syntactically, a subset of $\mathfrak{L}^+$, one could note that reverse is not true. Indeed, some extended levels are not equivalent to any level. For instance, let

$x \in \mathcal{X}$ and $u = \mathcal{C}(\{x\}, 1)$. Then, there is no $v \in \mathfrak{L}$ such that $u \equiv v$. So, even some sublevels are not equivalent to any level.

In fact, since for all $u \in \mathfrak{L}^+$, $u \equiv \mathcal{V}(\emptyset, u, 0)$, the sublevels are as powerful as the levels. However, this is not a problem. Keeping with our idea, we just want to show that any level is equivalent to a maximum of sublevels, so we do it for the grammar presented in Theorem 13. For that, we show how sublevels permit to replace nested imax.

► **Proposition 18.** *Let $E = \{x_1, \ldots, x_n\} \subset \mathcal{X}$, $k \in \mathbb{N}$, $y \in \mathcal{X}$, and for all $i \in \{1, \ldots, n\}$, $u_i = \mathcal{V}(\{x_{i+1}, \ldots, x_n\}, x_i, 0)$. Then,*

$$[S^k(y), x_1, \ldots, x_n] \equiv \max(\mathcal{V}(E, y, k), u_1, \ldots, u_n),$$
$$[S^{k+1}(0), x_1, x_n] \equiv \max(\mathcal{C}(E, k+1), u_1, \ldots, u_n).$$

Proof. Let σ be a valuation, t be the left-hand side term and $u = k + \sigma(y)$ and $u = k + 1$ in the second case. If for all $1 \leqslant i \leqslant n$, $\sigma(x_i) \neq 0$, then

$$[\![t]\!]_\sigma = \max(\sigma(x_n), \ldots, \sigma(x_1), u)$$
$$\forall u \in \{1, \ldots, n\}, [\![u_i]\!]_\sigma = \sigma(x_i)$$
$$[\![\mathcal{V}(E, y, k)]\!]_\sigma = k + \sigma(y) \qquad [\![\mathcal{C}(E, k)]\!]_\sigma = k + 1$$

and else, we take the largest $i \in \{1, \ldots, n\}$ such that $\sigma(x_i) = 0$, then

$$[\![t]\!]_\sigma = \max(\sigma(x_n), \ldots, \sigma(x_{i+1}))$$
$$\forall j \in \{1, \ldots, i\}, [\![u_j]\!]_\sigma = 0$$
$$[\![\mathcal{V}(E, y, k)]\!]_\sigma = 0 \qquad [\![\mathcal{C}(E, k)]\!]_\sigma = 0$$
$$\forall j \in \{i+1, \ldots, n\}, [\![u_j]\!]_\sigma = \sigma(x_j)$$

hence the equality. ◄

These equivalences only differ in the first sublevel of the max which is a variable sublevel in the first case (to consider $S^k(y)$) and a constant one in the second case (to consider k).

► Remark 19. There is no syntactic restriction on the verification conditions; they are not necessarily variables but can be any type of levels. In the same way, the variable part of a variable sublevel can be any type of level. Proposition 18 states that we only need them to be variables, but we made this choice of presentation to facilitate the level instantiation (developed in Section 5). Indeed, a variable will then be replaced by any level, and we want to make this substitution transparent in our level representation.

2.4 An Appropriate Set of Sublevels

We have restrained our study to the sublevels. Now, we show that some of them are not necessary, in the sense that they can be obtained as a maximum of other ones. The first restriction is related to the representation of 0. Indeed, for all $E \subset \mathcal{X}$, $\mathcal{C}(E, 0) \equiv 0$. Since we already have $0 \equiv \max(\emptyset)$, we can remove all these sublevels. The second restriction is a little more subtle and is illustrated with this example.

► **Example 20.** With $t_1 = \mathcal{V}(\emptyset, x, 0)$ and $t_2 = \mathcal{V}(\{x\}, x, 0)$, we have $t_1 \equiv t_2$ since for all valuation σ, $[\![t_1]\!]_\sigma = \sigma(x) = [\![t_2]\!]_\sigma$.

The issue here is the fact that the variable part of a variable sublevel does not necessarily appear in its first argument. This is the key of the following equivalence.

▶ **Proposition 21.** *Let $x \in \mathcal{X}$, $E \subset \mathcal{X} \setminus \{x\}$ and $k \in \mathbb{N}$. Then*

$$\mathcal{V}(E, x, k) \equiv \max(\mathcal{V}(E \cup \{x\}, x, k), \mathcal{C}(E, k)).$$

Proof. Let σ be a valuation, $t = \mathcal{V}(E, x, k)$, $u = \mathcal{C}(E, k)$, and $v = \mathcal{V}(E \cup \{x\}, x, k)$.
- If there exists $y \in E$ such that $\sigma(y) = 0$, then $[\![t]\!]_\sigma = [\![u]\!]_\sigma = [\![v]\!]_\sigma = 0$.
- Else, if $\sigma(x) = 0$, then $[\![t]\!]_\sigma = k$, $[\![u]\!]_\sigma = k$ and $[\![v]\!]_\sigma = 0$.
- Else, $\sigma(x) \neq 0$, and then $[\![t]\!]_\sigma = \sigma(x) + k$, $[\![u]\!]_\sigma = k$ and $[\![v]\!]_\sigma = \sigma(x) + k$.

Hence the result. ◀

We end up with this set of sublevels which permits to express any level.

▶ **Definition 22** (Canonical sublevels). *A canonical sublevel is an element of the set*

$$\mathbf{S} = \{\mathcal{V}(E, x, k), E \subset \mathcal{X}, x \in E\} \cup \{\mathcal{C}(E, k), E \subset \mathcal{X}, k > 0\}$$

▶ **Theorem 23.** *Let $t \in \mathfrak{L}$. Then there exists a finite $U \subset \mathbf{S}$ such that $t \equiv \max U$.*

▶ **Definition 24.** *Let U be a finite subset of $\mathbf{S}$. We say that $\max U$ is a representation and we denote by $\mathfrak{R}$ the set of representation.*

Besides, for all $t \in \mathfrak{L}^+$, we say that $\max U$ is a representation of t if $t \equiv \max U$, and we say that the elements u of U are the elements of the representation (denoted as $u \in \max U$ by convenience).

▶ Remark 25. In the rest of the paper, representations will often be denoted by U or V, as if they were subsets of canonical sublevels. However, keep in mind that they are special cases of levels and not subsets. In particular, they can be compared using $\equiv$ and $\leqslant$

The canonical sublevels correspond to the set of sublevels that we search for, and, in the next section, we will show how to ensure the uniqueness property.

We could try to merge the two types of sublevels by introducing a special variable $\mathbb{1}$ such that for all valuation σ, $\sigma(\mathbb{1}) = 1$. We will then see $\mathcal{C}(E, K+1)$ as $\mathcal{V}(E \cup \{\mathbb{1}\}, \mathbb{1}, K)$. This simplifies some results but makes the presentation less clear, and the distinction should still be done in a lot of cases.

▶ Remark 26. Let $u \in \mathbf{S}$ and let σ be a *valuation*. Then u is active if and only if all its verification conditions are checked.

Proof. If u is a constant sublevel $\mathcal{C}(E, K)$, then $[\![u]\!]_\sigma = 0$ if some VC is not checked, and $[\![u]\!]_\sigma = K > 0$ else.

Else, $u = \mathcal{V}(E, x, K)$ with $x \in E$, then $[\![u]\!]_\sigma = 0$ if some VC is not checked, and $[\![u]\!]_\sigma = K + \sigma(x) \geq \sigma(x) = 1$ else. ◀

3 A Canonical Form for Levels

The previous section defined $\mathfrak{R}$, the set of representations, and showed that any level is equivalent to one of its elements. The goal of this one is to show that any level has a minimal representation and that it is unique. This will be the canonical form.

▶ **Definition 27** (Minimal representation). *Let $U \in \mathfrak{R}$. We say that U is minimal if and only if for all $u, v \in U$ such that $u \neq v$, u and v are incomparable. We denote by $\mathbf{R}$ the set of the minimal representations.*

By Theorem 23, any level has a representation, so a minimal one since the set of representation is well-founded. The challenging part is the uniqueness. To show it, we study the core of the definition of a minimal representation: the sublevel comparison.

3.1 Sublevel Comparison

The sublevels can be easily compared. It is quite normal since we choose them to be very basic.

▶ **Theorem 28** (Sublevels comparison). *Elements of* $\mathbf{S}$ *are compared as follows.*

$$\mathcal{V}(E,x,L) \nleqslant \mathcal{C}(F,K) \tag{1}$$
$$\mathcal{C}(E,L) \leqslant \mathcal{C}(F,K) \iff F \subset E \wedge L \leqslant K \tag{2}$$
$$\mathcal{C}(E,L) \leqslant \mathcal{V}(F,x,K) \iff (F \subset E \wedge L \leqslant K+1) \tag{3}$$
$$\mathcal{V}(E,x,L) \leqslant \mathcal{V}(F,y,K) \iff F \subset E \wedge x = y \wedge L \leqslant K \tag{4}$$

Proof. With σ such that $\sigma(x) = K+1$ and $\sigma(y) = 1$ if $y \neq x$, we show the first case. Indeed, $[\![\mathcal{V}(E,x,L)]\!]_\sigma = K+1+L$, and $K = [\![\mathcal{C}(F,K)]\!]_\sigma$ hence $\mathcal{V}(E,x,L) \nleqslant \mathcal{C}(F,K)$. The cases 2, 3 and 4 correspond to Propositions 29–31 proved below. ◀

▶ **Proposition 29.** *Let* $E, F \subset \mathcal{X}$ *and* $L, K \in \mathbb{N}$. *Then*

$$\mathcal{C}(E,L) \leqslant \mathcal{C}(F,K) \iff F \subset E \wedge L \leqslant K.$$

Proof. We note $t_1 = \mathcal{C}(E,L)$ and $t_2 = \mathcal{C}(F,K)$. Let us suppose $F \subset E$ and $L \leqslant K$. Let σ be a valuation.

- If there exists $y \in F$ such that $\sigma(y) = 0$, then $[\![t_2]\!]_\sigma = 0$ and since $F \subset E$, $[\![t_1]\!]_\sigma = 0$.
- Else, $[\![t_1]\!]_\sigma \leqslant K \leqslant L = [\![t_2]\!]_\sigma$.

In both cases, $[\![t_1]\!]_\sigma \leqslant [\![t_2]\!]_\sigma$ hence $t_1 \leqslant t_2$.

Now, we show the other implication by contraposition.

- If there exists $y \in F$ such that $y \notin E$, we take σ such that $\sigma(y) = 0$ and for all $z \neq y$, $\sigma(z) = 1$. Then, $[\![t_1]\!]_\sigma = L > 0 = [\![t_2]\!]_\sigma$.
- If $L < K$ we take σ such that for all y, $\sigma(y) = 1$. Then, $[\![t_2]\!]_\sigma = K > L = [\![t_1]\!]_\sigma$.

◀

▶ **Proposition 30.** *Let* $E, F \subset \mathcal{X}$, $x \in E$ *and* $K, L \in \mathbb{N}$. *Then*

$$\mathcal{C}(E,L) \leqslant \mathcal{V}(F,x,K) \iff (F \subset E \wedge L \leqslant K+1).$$

Proof. We note $t_1 = \mathcal{C}(E,L)$ and $t_2 = \mathcal{V}(F,x,K)$. Let us suppose $F \subset E$ and $L \leqslant K+1$. Let σ be a valuation.

- If there exists $y \in F$ such that $\sigma(y) = 0$, then $[\![t_2]\!]_\sigma = 0$ and since $F \subset E$, $[\![t_1]\!]_\sigma = 0$.
- Else, $\sigma(x) \geq 1$ (because $x \in F$) and then $[\![t_2]\!]_\sigma = \sigma(x) + K \geq 1 + K \geq L \geq [\![t_1]\!]_\sigma$.

In both cases, $[\![t_1]\!]_\sigma \leqslant [\![t_2]\!]_\sigma$ hence $t_1 \leqslant t_2$.

Now, we show the other implication by contraposition. First, we note that $L > 0$.

- If there exists $y \in F$ such that $y \notin E$, we take σ such that $\sigma(y) = 0$ and for all $z \neq y$, $\sigma(z) = 1$. Then, $[\![t_1]\!]_\sigma = K > 0 = [\![t_2]\!]_\sigma$.
- If $L > K+1$ we take σ such that for all y, $\sigma(y) = 1$. Then, $[\![t_1]\!]_\sigma = L > K+1 = [\![t_1]\!]_\sigma$.

◀

▶ **Proposition 31.** *Let* $E, F \subset \mathcal{X}$, $x \in E$, $y \in F$ *and* $L, K \in \mathbb{N}$. *Then*

$$\mathcal{V}(E,x,L) \leqslant \mathcal{V}(F,y,K) \iff F \subset E \wedge x = y \wedge L \leqslant K.$$

Proof. We note $t_1 = \mathcal{V}(E,x,L)$ and $t_2 = \mathcal{V}(F,y,K)$. Let us suppose $F \subset E$, $x = y$ and $L \leqslant K$. Let σ be a valuation.

- If there exists $y \in F$ such that $\sigma(y) = 0$, then $[\![t_2]\!]_\sigma = 0$ and since $F \subset E$, $[\![t_1]\!]_\sigma = 0$.
- Else, $[\![t_1]\!]_\sigma \leqslant \sigma(x) + L \leqslant \sigma(x) + K = [\![t_2]\!]_\sigma$.

In both cases, $[\![t_1]\!]_\sigma \leqslant [\![t_2]\!]_\sigma$ hence $t_1 \leqslant t_2$.

Now, we show the other implication by contraposition.

- If there exists $z \in F$ such that $z \notin E$, we take σ such that $\sigma(z) = 0$ and for all $j \neq z$, $\sigma(j) = 1$. We note that $z \neq x$ (since $z \notin E$ and $x \in E$) hence $\sigma(x) = 1$. Then, $[\![t_1]\!]_\sigma = L + 1 > 0 = [\![t_2]\!]_\sigma$.
- If $x \neq y$ we take σ such that $\sigma(x) = K+2$, $\sigma(y) = 1$ and for all $z \neq x$ and $z \neq y$, $\sigma(z) = 1$. Then, $[\![t_1]\!]_\sigma = K + L + 2 > K + 1 = [\![t_2]\!]_\sigma$.
- If $L > K$ we take σ such that for all z, $\sigma(z) = 1$. Then, $[\![t_1]\!]_\sigma = L + 1 > K + 1 = [\![t_2]\!]_\sigma$.

◀

As a corollary, we get that the sublevel equivalence is a syntactic equality, which is quite natural; the uniqueness property would be impossible otherwise.

▶ **Corollary 32.** *Let $t_1, t_2 \in \mathbf{S}$. Then $t_1 \equiv t_2 \iff t_1 = t_2$.*

Proof. We have $t_1 \equiv t_2 \iff t_1 \leqslant t_2 \wedge t_2 \leqslant t_1$, we conclude with Theorem 28. ◀

3.2 The Uniqueness Property

Now, we can show the uniqueness property. First, we show that two equivalent minimal representations have the same variable sublevels.

▶ **Proposition 33.** *Let $U, V \in \mathbf{R}$ such that $U \equiv V$. Then*

$$\mathcal{V}(E, x, k) \in U \iff \mathcal{V}(E, x, k) \in V.$$

Proof. Let $\mathcal{V}(E, x, k)$ be a sublevel of U. We consider σ such that

$$\sigma(y) = \begin{cases} 2 + \max\{\omega(u), u \in U \text{ or } u \in V\} & \text{if } y = x \\ 1 & \text{if } y \in E \setminus \{x\} \\ 0 & \text{else} \end{cases}$$

We have $[\![U]\!]_\sigma = [\![\mathcal{V}(E, x, k)]\!]_\sigma = k + \sigma(x)$ and then $[\![V]\!]_\sigma = k + \sigma(x)$. Then,

- either there exists $\mathcal{V}(F, y, l)$ in V such that $\sigma(y) + l = \sigma(x) + k$ and $F \subset E \cup \{x\} = E$ (else F contains a variable z such that $\sigma(z) = 0$),
- or there exists $\mathcal{C}(F, l)$ in V such that $l = \sigma(x) + k$.

Since $\sigma(x) > \max\{\omega(u), u \in U \text{ or } u \in V\}$, we deduce that it is the first case (we cannot have $\sigma(x) + k = l$) and $y = x$ (otherwise we will have $\sigma(y) = 1$ and $\sigma(x) > 1 + l$). Then, there exists $\mathcal{V}(F, x, k) \in V$ with $F \subset E$.

If $F \subsetneq E$, then by the same reasoning, we show that there exists $\mathcal{V}(G, x, k) \in U$ with $G \subset F \subsetneq E$. But, by minimality, it is impossible to have $\mathcal{V}(E, x, k)$ and $\mathcal{V}(G, x, k)$ in U with $G \subset E$ since they are comparable.

Then $E = F$ and $\mathcal{V}(E, x, k)$ is also an element of V. ◀

And we show the same for the constant sublevels.

▶ **Proposition 34.** *Let $U, V \in \mathbf{R}$ such that $U \equiv V$. Then*

$$\mathcal{C}(E, k) \in U \iff \mathcal{C}(E, k) \in V.$$

Proof. Let $\mathcal{C}(E,k)$ be a sublevel of U. We show the result by induction on E. If $E=\emptyset$, we consider σ the zero function. Then, $[\![U]\!]_\sigma = k$, hence $[\![V]\!]_\sigma = k$. Since $k>0$, it follows that $\mathcal{C}(\emptyset,k)$ is a sublevel of V.

In the induction case, we consider σ such that $\sigma(x)=1$ if $x\in E$ and $\sigma(x)=0$ otherwise, hence $[\![U]\!]_\sigma = k$. Then, $[\![V]\!]_\sigma = k$ and since $k>0$,

- either there exists $\mathcal{V}(F,x,l)\in V$ such that $F\subset E$ and $\sigma(x)+l=k$,
- or there exists $\mathcal{C}(F,k)\in V$ such that $F\subset E$.

In the first case, we have $x\in F\subset E$, then $\sigma(x)=1$ and $l=k-1$. Then, by Proposition 33, $\mathcal{V}(F,x,k-1)\in U$ which is impossible by Definition 27 since it would be comparable with $\mathcal{C}(E,k)\in U$.

Then, there exists $\mathcal{C}(F,k)\in V$ such that $F\subset E$. If $F\subsetneq E$, we apply the induction hypothesis and obtain $\mathcal{C}(F,k)\in U$, impossible because it would be comparable with $\mathcal{C}(E,k)$.

Then $E=F$ and $\mathcal{C}(E,k)$ is also an element of V. ◀

We immediately obtain that equivalence of minimal representations is set equality.

▶ **Proposition 35.** *For all $U,V\in\mathbf{R}$, $U\equiv V \iff U=V$.*

Proof. The reverse implication is trivial and the direct one is a consequence of Propositions 33 and 34. ◀

Finally, we obtain the main theorem: the existence and uniqueness of a minimal representation for each level, that is to say a canonical form. First, we show the intuitive property that the minimal representation of a maximum of sublevels is formed with some of them.

▶ **Proposition 36.** *For all $U\in\mathfrak{R}$, there exists a unique $V\in\mathbf{R}$ such that $U\equiv V$. Besides, for all $v\in V$, $v\in U$.*

Proof. We apply the following procedure. Let E be the elements of U. While there exists $u,v\in E$ such that $u\leqslant v$, we remove u from E, and we obtain a minimal representation such that $V\subset U$. Proposition 35 permits to obtain its uniqueness. ◀

▶ **Theorem 37** (Representation)**.** *For all $t\in\mathfrak{L}$, there exists a unique $U\in\mathbf{R}$ such that $t\equiv U$. We say that U is the minimal representation of t.*

Proof. By Theorem 23, there exists $U\in\mathbf{S}$ such that $t\equiv U$, and by Proposition 36, there exists a unique minimal representation of U. ◀

This theorem states the existence of a canonical form c for $\mathfrak{L}$, c being the function that associates any level to its minimal representation.

3.3 Level Comparison

The canonical form gives us a simple decision procedure for the equivalence, but also more generally for the comparison problems. Indeed, a sublevel can be compared to a level using its representation.

▶ **Lemma 38.** *Let $u\in\mathbf{S}$ and $V\subset\mathfrak{R}$. Then $u\leqslant V$ if and only if there exists $v\in V$ such that $u\leqslant v$.*

Proof. The reverse implication is trivial. We show the direct one by contraposition. We suppose that for all $v\in V$, $u\nleqslant v$.

If $u=\mathcal{C}(E,k)$, we consider σ such that $\sigma(x)=1$ if $x\in E$ and 0 otherwise. Then, for all $v\in V$, we have either

- $v = \mathcal{V}(F, x, l)$ or $v = \mathcal{C}(F, l)$ with $F \not\subseteq E$ hence $[\![v]\!]_\sigma = 0 < k = [\![u]\!]_\sigma$,
- or $v = \mathcal{V}(F, x, l)$ with $F \subseteq E$ and $l < k - 1$ hence $[\![v]\!]_\sigma = l + 1 < k = [\![u]\!]_\sigma$,
- or $v = \mathcal{C}(F, l)$ with $l < k$ hence $[\![v]\!]_\sigma = l < k = [\![t]\!]_\sigma$.

Then $u \not\leqslant V$.

Else, $u = \mathcal{V}(E, x, k)$. We consider $M = \max\{\omega(v), v \in V\}$ and σ such that $\sigma(x) = M + 2$, $\sigma(y) = 1$ if $y \in E \setminus \{x\}$ and 0 otherwise. Then, for all $v \in V$, we have either

- $v = \mathcal{C}(F, l)$ hence $[\![v]\!]_\sigma \leqslant l < k + M + 2 = [\![u]\!]_\sigma$,
- or $v = \mathcal{V}(F, y, l)$ and $F \not\subseteq E$ hence $[\![v]\!]_\sigma = 0 < [\![u]\!]_\sigma$,
- or $v = \mathcal{V}(F, y, l)$ with $F \subseteq E$ and $x \neq y$, hence $[\![v]\!]_\sigma = l + 1 < k + M + 2 = [\![u]\!]_\sigma$,
- or $v = \mathcal{V}(F, x, l)$ with $F \subseteq E$ and $l < k$, hence $[\![v]\!]_\sigma = l + M + 2 < k + M + 2 = [\![u]\!]_\sigma$.

Then $u \not\leqslant V$. ◀

Therefore, we can compare two levels, for instance by comparing each sublevel of the minimal representation of the first one to the second one. More generally, two representations are compared in the following way.

▶ **Theorem 39.** *Let $U, V \in \mathfrak{R}$. Then, $U \leqslant V$ if and only if for all $u \in U$, there exists $v \in V$ such that $u \leqslant v$.*

Proof. If $U \leqslant V$, then for all $u \in U$, $u \leqslant V$ and by Lemma 38, there exists $v \in V$ such that $u \leqslant v$. The reverse implication is trivial. ◀

One can note that Lemma 38 gives us a new proof of the uniqueness property stated in Proposition 35.

Proof. Let $U, V \in \mathbf{R}$ such that $U \equiv V$. We want to show that for all $u \in U$, $u \in V$.

We have $u \leqslant U \leqslant V$, hence by Lemma 38, there exists $v \in V$ such that $u \leqslant v$. In the same way, there exists $u' \in U$ such that $v \leqslant u'$. Then, by Definition 27, $u' = u$ (because the elements of U are incomparable), and then $u \equiv v$ hence $u = v$ by Corollary 32. ◀

This shows that there is a link between Lemma 38 and the uniqueness property. In fact, this lemma should be understood as an *independence* lemma. Indeed, if we consider $\max(u_1, \ldots, u_n)$ as a linear combination of $u_1, \ldots, u_n$, then this lemma states that the only way to be smaller than a linear combination is to depend on and be smaller than one of the elements of this combination.

This analogy provides a new point of view on our work: $\mathbf{S}$ is a 'linearly independent' family (uniqueness of the minimal representation) which generates all the levels through 'linear combinations'.

4 A Canonical Form for Extended Levels

In this section we are interested in extending the representation theorem to the whole extended levels. This is motivated by the level instantiation. Indeed, if $u = \max(\mathcal{V}(\{x\}, x, 0))$, then instantiate x with $\max(y, z)$ gives the term $\max(\mathcal{V}(\{\max(y, z)\}, \max(y, z), 0))$, which is not canonical. To obtain its canonical form, we simplify this term in $\max(\mathcal{V}(\{y\}, y, 0), \mathcal{V}(\{z\}, z, 0))$.

But, for now, we do not have an algorithm to perform this simplification, and in fact, we do not even know if all the extended levels have representations. The different types of level are sorted by expressiveness in Figure 2, and our goal is to collapse its last floor, showing that $\mathfrak{L}^+ \equiv \mathbf{R}$.

One could note that for all $u, v \in \mathfrak{L}$ and $x \in \mathcal{X}$, $u[x/v] \in \mathfrak{L}$, and so $c(u)[x/v]$ always has a canonical form. But extending the representation theorem and having a general canonization

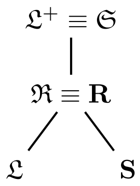

Figure 2 Comparison of the different types of level.

algorithm is convenient since it results in computing the canonical form of the extended level $c(u)[x/v]$ to obtain a canonical form after substitution.

We show by induction that any extended level has a representation. The result is already shown for the basis case of 0 and the variables, and the case of the maximum is easy since for all $U, V \in \mathfrak{R}$, $\max(U \cup V)$ is a representation of $\max(U, V)$.

The Successor

We define $\mathrm{inc}\colon \mathbf{S} \to \mathbf{S}$ such that $\mathrm{inc}(\mathcal{C}(E, K)) = \mathcal{C}(E, K+1)$ and $\mathrm{inc}(\mathcal{V}(E, x, K)) = \mathcal{V}(E, x, K+1)$, in order to define the successor of a canonical sublevel in terms of representation.

▶ **Proposition 40.** *For all $u \in \mathbf{S}$,*

$$S(u) \equiv \max(\mathrm{inc}(u), \mathcal{C}(\emptyset, 1)).$$

Proof. Let σ be a valuation. If there exists $x \in \mathrm{VC}(u)$ such that $\sigma(x) = 0$, then both terms are evaluated to 0. Else, both terms are evaluated to $[\![u]\!]_\sigma + 1$. ◀

We immediately deduce the result.

▶ **Proposition 41.** *Let $U \in \mathfrak{R}$. Then,*

$$S(U) \equiv \max\{\mathrm{inc}(u), u \in U\} \cup \{\mathcal{C}(\emptyset, 1)\}.$$

The Impredicative Maximum

Let $U, V \in \mathbf{R}$. Following the equivalences $\mathrm{imax}(0, u) \equiv u$ and $\mathrm{imax}(u, 0) \equiv 0$, and Propositions 6 and 7,

$$\mathrm{imax}(U, V) \equiv \begin{cases} \max(\emptyset) & \text{if } V = \max(\emptyset) \\ V & \text{if } U = \max(\emptyset) \\ \max\limits_{\substack{u \in U \\ v \in V}} \mathrm{imax}(u, v) & \text{else} \end{cases} \tag{5}$$

Then, it is sufficient to show that for all $u, v \in \mathbf{S}$, $\mathrm{imax}(u, v)$ has a representation. We could then obtain a representation of $\mathrm{imax}(U, V)$ by taking the elements of the ones of $\mathrm{imax}(u, v)$ for all $u \in U$ and $v \in V$.

▶ **Proposition 42.** *Let $v \in \mathbf{S}$, $E \subset \mathcal{X}$, $x \in E$ and $K \in \mathbb{N}$. Then,*

$$\mathrm{imax}(\boldsymbol{\mathcal{C}(E, K+1)}, v) \equiv \max(\mathcal{C}(E \cup \mathrm{VC}(v), K+1), v)$$
$$\mathrm{imax}(\boldsymbol{\mathcal{V}(E, x, K)}, v) \equiv \max(\mathcal{V}(E \cup \mathrm{VC}(v), x, K), v)$$

Proof. Let t be the bold term, and let σ be a valuation.

- If there exists $y \in \mathrm{VC}(v)$ such that $\sigma(y) = 0$, then both terms are evaluated to 0.
- Else if there exists $y \in E$ such that $\sigma(y) = 0$, then they are both evaluated to $[\![v]\!]_\sigma$.
- Else, for all $y \in E \cup \mathrm{VC}(v)$, $\sigma(y) = 0$ and they are evaluated to $\max([\![v]\!]_\sigma, \omega(v) + [\![\nu(v)]\!]_\sigma)$.

◀

The Sublevels

We consider that each of the VCs of a sublevel are representations. First, we show how to remove a max as head-symbol of such a VC U. For the sublevel to be active, its VC should be checked and in the case of U, it means that one of its elements u is active. So it leads to split it into a maximum.

▶ **Proposition 43.** *Let $V \subset \mathfrak{L}^+$, $U \subset \mathfrak{R}$, $K \in \mathbb{N}$, and $w \in \mathfrak{L}^+$.*

$$\mathcal{C}(V \cup \{U\}, K) \equiv \max\{\mathcal{C}(V \cup \{\mathrm{VC}(u)\}, K), u \in U\}$$
$$\mathcal{V}(V \cup \{U\}, w, K) \equiv \max\{\mathcal{V}(V \cup \{u\}, w, K), u \in U\}$$

Proof. Let σ be a valuation, t be the left-hand side term and t' be the right-hand side term. We note $w = 0$ in the case of the constant sublevel. We note that $[\![t']\!]_\sigma$ and $[\![t]\!]_\sigma$ are either K or 0, and if $K = 0$ and $[\![w]\!]_\sigma = 0$, then both terms are evaluated to 0. Else, since the VC of a canonical sublevel u are checked with σ if and only if $[\![u]\!]_\sigma \neq 0$,

$$\begin{aligned} [\![t]\!]_\sigma = K + [\![w]\!]_\sigma &\iff \exists u \in U, [\![u]\!]_\sigma \neq 0 \\ &\iff \exists u \in U, \forall x \in \mathrm{VC}(u), \sigma(x) \neq 0 \\ &\iff [\![t']\!]_\sigma = K + [\![w]\!]_\sigma. \end{aligned}$$

Hence the result. ◀

Note that this result is true when $U = \max(\emptyset)$, since we obtain $\max(\emptyset)$.

An induction on the number of VC of the constant sublevel t permits to remove max as head-symbol of all the verification conditions. We have to consider all the combinations of VC of its guards. The result is similar in both constant and variable sublevel cases, but we prefer to split it into two propositions, even if we only have one proof.

▶ **Proposition 44.** *Let $U_1, \ldots, U_n \in \mathfrak{R}$, $K \in \mathbb{N}$. We note $t = \mathcal{C}(\{U_1, \ldots, U_n\}, K)$ and for all $u_1, \ldots, u_n \in \mathbf{S}$ we define*

$$P(u_1, \ldots, u_n) = \mathcal{C}\Big(\bigcup_{1 \le i \le n} \mathrm{VC}(u_i), K\Big)$$

Then $t \equiv \max\{P(u_1, \ldots, u_n), u_1 \in U_1, \ldots, u_n \in U_n\}$.

▶ **Proposition 45.** *Let $U_1, \ldots, U_n \in \mathfrak{R}$, $K \in \mathbb{N}$, $w \in \mathfrak{L}^+$, and $t = \mathcal{V}(\{U_1, \ldots, U_n\}, w, K)$. For all $u_1, \ldots, u_n \in \mathbf{S}$, we define*

$$P(u_1, \ldots, u_n) = \mathcal{V}\Big(\bigcup_{1 \le i \le n} \mathrm{VC}(u_i), w, K\Big)$$

Then $t \equiv \max\{P(u_1, \ldots, u_n), u_1 \in U_1, \ldots, u_n \in U_n\}$.

Proof. Let us note t' the right-hand side term, and let σ be a valuation. We note $w = 0$ in the case of the constant sublevel. We note that $[\![t']\!]_\sigma$ and $[\![t]\!]_\sigma$ are either K or 0. Of course, if

$K = 0$ and $[\![w]\!]_\sigma = 0$, both terms are evaluated to 0. Else, since the verifications conditions of a canonical sublevel u are checked with σ if and only if $[\![u_i]\!]_\sigma \neq 0$, hence

$$[\![t']\!]_\sigma = K + [\![w]\!]_\sigma \iff \forall i \in \{1, \dots, n\}, \exists u_i \in U_i, [\![u_i]\!]_\sigma \neq 0$$
$$\iff \forall i \in \{1, \dots, n\}, [\![U_i]\!]_\sigma \neq 0$$
$$\iff [\![t]\!]_\sigma = K + [\![w]\!]_\sigma.$$

Checkmate! ◀

The term $P(u_0, \dots, u_n)$ means that if $u_0 \in U_0, \dots, u_n \in U_n$ are checked, then $U_0, \dots, U_n$ are checked and the sublevel that we simplify is active as well as $P(u_0, \dots, u_n)$.

Here again, we note that the result still holds if there exists i such that $U_i = \emptyset$, since both terms are equivalent to 0.

The Constant Sublevels

The induction case of the constant sublevel is solved by Proposition 44. Indeed, in the constant sublevel case, for all $u_1, \dots, u_n \in \mathbf{S}$, $P(u_0, \dots, u_n) \in \mathbf{S}$ if $K > 0$ (hence we obtain a representation of t). Besides, if $K = 0$, a representation of t is $\max \emptyset$.

The Variable Sublevels

However, in the variable sublevels, it is not the case; Proposition 45 only permits us to obtain variable sublevels where the verification conditions are variables. Besides, the variable part of $P(u_0, \dots, u_n)$ is not necessarily a variable. That is why we now take a look at the variable part of variable sublevels.

▶ **Proposition 46.** *Let $U \in \mathfrak{R}, V \subset \mathfrak{L}^+$, and $K \in \mathbb{N}$. Then,*

$$\mathcal{V}(V, U, K) \equiv \max\{\mathcal{V}(V, u, K), u \in U\}.$$

Proof. Let σ be a valuation. If there exists $v \in V$ such that $[\![v]\!]_\sigma = 0$, then both terms are evaluated to 0. Else, for all $u \in U$, $[\![\mathcal{V}(V, u, K)]\!]_\sigma = K + [\![u]\!]_\sigma$ and $[\![\mathcal{V}(V, U, K)]\!]_\sigma = K + \max_{u \in U} [\![u]\!]_\sigma$ hence the result. ◀

So, it results in sublevels as variable part of variable sublevels, and we watch these in the next proposition.

▶ **Proposition 47.** *Let $V \subset \mathfrak{L}^+$, $u \in \mathfrak{S}$, and $K \in \mathbb{N}$. We note*

$$f(u) = \begin{cases} \mathcal{C}(V \cup \mathrm{VC}(u), K + \omega(u)) & \text{if } \nu(u) = 0 \\ \mathcal{V}(V \cup \mathrm{VC}(u) \cup \{u\}, \nu(u), K + \omega(u)) & \text{else} \end{cases}$$

Then $\mathcal{V}(V, u, K) \equiv \max(\mathcal{C}(V, K), f(u))$.

Proof. Let σ be a valuation.

- If there exists $v \in V$ such that $[\![v]\!]_\sigma = 0$, then both terms are evaluated to 0.
- Else if there exists $w \in \mathrm{VC}(u)$ such that $[\![w]\!]_\sigma = 0$, they are evaluated to K.
- Else, they are evaluated to $K + [\![u]\!]_\sigma$.

◀

Note that when $u \in \mathbf{S}$, having $\mathrm{VC}(u)$ as VC is equivalent to having u as VC which simplifies $f(u)$ in the case where u is a variable sublevel.

We can apply these two propositions to $P(u_0, \ldots, u_n)$ in Proposition 45, and we get the following.

▶ **Proposition 48.** *Let $K \in \mathbb{N}$, $U_1, \ldots, U_n \in \mathfrak{R}$, $V \in \mathfrak{R}$, and $t = \mathcal{V}(\{U_1, \ldots, U_n\}, V, K)$. For all $v, u_0, \ldots, u_n \in \mathbf{S}$, we define*

$$f(u_1, \ldots, u_n) = \mathcal{C}\Big(\bigcup_{1 \leq i \leq n} \mathrm{VC}(u_i), K\Big),$$

$g(v, u_1, \ldots, u_n)$ to be (by noting $u_0 = v$),

$$\begin{cases} \mathcal{C}\Big(\bigcup_{0 \leq i \leq n} \mathrm{VC}(u_i), K + \omega(v)\Big)] & \text{if } \nu(v) = 0 \\ \mathcal{V}\Big(\bigcup_{0 \leq i \leq n} \mathrm{VC}(u_i), \nu(v), K + \omega(v)\Big)] & \text{else} \end{cases}$$

and

$$Q(v, u_1, \ldots, u_n) = \max(f(u_1, \ldots, u_n), g(v, u_1, \ldots, u_n)).$$

Then,

$$t \equiv \max\{Q(v, u_1, \ldots, u_n), u_1 \in U_1, \ldots, u_n \in U_n, v \in V\}$$

Proof. By Proposition 45, it is sufficient to shat for all $u_1 \in U_1, \ldots, u_n \in U_n$,

$$P(u_0, \ldots, u_n) \equiv \max\{Q(v, u_1, \ldots, u_n), v \in V\}.$$

It is the case by Propositions 46 and 47. ◀

Here, $g(v, u_1, \ldots, u_n)$, corresponds to the case where all the VC $u_1, \ldots, u_n$ are checked, and v is active, and $f(v, u_1, \ldots, u_n)$ corresponds to the case where $u_1, \ldots, u_n$ are checked but v is not active (therefore it is the same as having a constant sublevel, which explains that f corresponds to the function P in the constant sublevel case), hence they form $Q(v, u_0, \ldots, v_n)$.

▶ Remark 49. As for the constant sublevels case, we should take care to consider the sublevels $f(u_1, \ldots, u_n)$ only if its constant part is not 0. Otherwise, it is equivalent to 0, and it can be removed from the max.

Besides, one could think that we should have the same consideration with $g(v, u_1, \ldots, u_n)$ when v is a constant sublevel, but since $v \in \mathbf{S}$ (because it is an element of a representation), then $\omega(v) > 0$.

After that, the case of the variable sublevel is solved.

General Representation Theorem

All the induction cases are done. We obtain the main result.

▶ **Theorem 50.** *For all $u \in \mathfrak{L}^+$, there exists an unique $v \in \mathbf{R}$ such that $u \equiv v$.*

Proof. By induction, u has a representation, and therefore a minimal one by Proposition 36. ◀

Algorithm 1 Canonization algorithm

```
Data: u ∈ 𝔏⁺
Result: c(u), the canonical form of u
Function normalize(u)
    if u = 0 then
        return max(∅)
    else if u = x then
        return max(𝒱({x}, x, 0))
    else if u = max(u, v)) then
        // Algorithm 3
    else if u = S(u) then
        // Algorithm 4
    else if u = imax(u, v) then
        // Algorithm 5
    else if u = 𝒞({U_1, ..., U_n}, K) then
        // Algorithm 6
    else if u = 𝒱({U_1, ..., U_n}, V, K) then
        // Algorithm 6
end
```

5 Computation Algorithm

We design a recursive algorithm suited to the inductive structure of $\mathfrak{L}^+$. It is presented in Algorithm 1 which already contains the code for the basis cases 0 and x for which the canonical form are respectively $\max(\emptyset)$ and $\max(\mathcal{V}(\{x\}, x, 0))$.

We are now interested in the code to compute the canonical form in the other cases. We will generally use the results developed in Section 4, since they give us a representation for some type a level. Then, it is sufficient to minimize this representation. For that, we write Algorithm 2 which inserts a sublevel in an independent set of sublevels.

Algorithm 2 Insertion algorithm

```
Data: U ⊂ S independent, v ∈ S
Result: W such that c(max(U ∪ {v})) = max(W)
Function insert(U, v)
    W ← ∅
    for u ∈ U do
        if v ⩽ u then
            return U
        else if v ⩽̸ u then
            W ← W ∪ {u}
        end
    end
    return W ∪ {v}
end
```

The Maximum

To compute the canonical form of $\max(u,v)$, we use insert to add the sublevels of v to the one of u in Algorithm 2.

Algorithm 3 Case of the maximum

Data: $u, v \in \mathfrak{L}^+$
Result: $c(\max(u,v))$
$s \leftarrow \text{sublevels}(\text{normalize}(u))$
for $v_i \in \text{normalize}(v)$ **do**
  | $s \leftarrow \text{insert}(s, v_i)$
end
return $\max(s)$

The Successor

Thanks to Proposition 40, for all $U \in \mathbf{R}$, we know a representation of $S(U)$ for $U \in \mathbf{R}$. To obtain its canonical form, we could use insert to add its sublevels to an initially empty set. Besides, we have a simpler operation.

▶ **Proposition 51.** *Let* $U \in \mathbf{R}$ *and* $E = \{\text{inc}(u), u \in U\}$*.*

$$c(S(U)) = \begin{cases} \max E & \text{if } \exists u \in U, \text{VC}(u) = \emptyset \\ \max E \cup \{\mathcal{C}(\emptyset, 1)\} & \text{else} \end{cases}$$

Proof. First, we note that for all $u, v \in \mathbf{S}$, if u and v are incomparable, then $\text{inc}(u)$ and $\text{inc}(v)$ are also incomparable. By Proposition 40, $\max\{\text{inc}(u), u \in U\} \cup \{\mathcal{C}(\emptyset, 1)\}$ is equivalent to $S(U)$. We distinguish two cases.

- If there exists $u \in U$ such that $\text{VC}(u) = \emptyset$, then $u = \mathcal{C}(\emptyset, N_i)$ with $N_i > 0$ hence $\mathcal{C}(\emptyset, 1) \leqslant \text{inc}(u)$ and the result holds since the other elements are incomparable.
- Else, for all $u \in U$, $\mathcal{C}(\emptyset, 1)$ and u are incomparable. Indeed, let σ be a valuation such that for all $x \in \mathcal{X}$, $\sigma(x) = 0$. Then $[\![\text{inc}(u)]\!]_\sigma = 0$ (since $\text{VC}(u) \neq \emptyset$), hence $\mathcal{C}(\emptyset, 1) \not\leqslant \text{inc}(u)$. Conversely, let σ be a valuation such that for all $x \in \mathcal{X}$, $\sigma(x) = 2$. Then, $[\![\text{inc}(u)]\!]_\sigma = [\![\nu(\text{inc}(u))]\!]_\sigma + \omega(\text{inc}(u)) > 1$ (because for a constant sublevel $\omega(\text{inc}(u)) > 1$, and for a variable one $[\![\nu(\text{inc}(u))]\!]_\sigma = 2$), hence $\text{inc}(u) \not\leqslant \mathcal{C}(\emptyset, 1)$.

◀

We implement this strategy in Algorithm 4.

The Impredicative Maximum

For all $u, v \in \mathbf{S}$, Proposition 42 expresses $\text{imax}(u,v)$ as a maximum of canonical sublevels, and for all $U, V \in \mathbf{R}$, Equation (5) expresses $\text{imax}(U,V)$ as a maximum of $\text{imax}(u,v)$ with $u \in U$ and $v \in V$ (hence $u, v \in \mathbf{S}$). Using these two results, we design Algorithm 5.

The Constant Sublevels

The computation of the canonical form of a constant sublevel relies on Proposition 44. Here, we immediately returns $\max(\emptyset)$ if some VC is 0, and we do not forget the case $K = 0$ which results in 0.

Algorithm 4 Case of the successor

```
Data: u ∈ 𝔏⁺
Result: The canonical form of S(u)
U ← normalize(u)
s ← {inc(u), u ∈ U}
for u ∈ U do
    if VC(u) = ∅ then
        return max(s)
    end
end
return max(s ∪ 𝒞(∅, 1))
```

Algorithm 5 Case of the impredicative maximum

```
Data: u and v levels
Result: The canonical form of imax(u, v)
U ← normalize(u)
V ← normalize(v)
s ← sublevels(V)
for u ∈ U, v ∈ V do
    if u = 𝒱(E, x, K) then
        s ← insert(s, 𝒱(E ∪ VC(v)), x, K)
    else if u = 𝒞(E, K) then
        s ← insert(s, 𝒞(E ∪ VC(v)), K)
    end
end
return max(s)
```

Algorithm 6 Case of the constant sublevels

```
Data: U_1, ..., U_n ∈ 𝔏⁺, K ∈ ℕ
Result: c(𝒞({U_1, ..., U_n}, K))
if K = 0 then
    return max(∅)
for 1 ≤ i ≤ n do
    U_i ← normalize(U_i)
    if U_i = max(∅) then
        return max(∅)
    end
end
s ← ∅
for u_1 ∈ U_1, ..., u_n ∈ U_n do
    s ← insert(𝒞(⋃_{1≤i≤n} VC(u_i), K), s)
end
return max(s)
```

The Variable Sublevels

The case of the variable sublevel is very similar and relies on Proposition 48.

Algorithm 7 Case of the variable sublevels

```
Data: U_1, ..., U_n, V ∈ 𝔏^+, K ∈ ℕ
Result: c(𝒱({U_1, ..., U_n}, V, K))
for 1 ≤ i ≤ n do
    U_i ← normalize(U_i)
    if U_i = max(∅) then
        return max(∅)
    end
end
V ← normalize(V)
s ← ∅
for u_1 ∈ U_1, ..., u_n ∈ U_n do
    if K ≠ 0 then
        s ← insert(𝒞(∪_{1≤i≤n} VC(u_i), K), s)
    end
    for u_0 ∈ V do
        if u_0 = 𝒞(E, L) then
            s ← insert(𝒞(∪_{0≤i≤n} VC(u_i), K + L), s)
        else if u_0 = 𝒱(E, x, L) then
            s ← insert(𝒱(∪_{0≤i≤n} VC(u_i), x, K + L), s)
        end
    end
end
return max(s)
```

▶ **Theorem 52** (Correction). *Let* $u \in \mathfrak{L}^+$*. Then,* $\text{normalize}(u)$ *computes* $c(u)$*, the canonical form of* u*.*

Proof. First, we note that the insert function terminates and is correct. Then, we show that normalize terminates since each recursive call is on smaller terms. The correction of the algorithm follows from the explanation of each cases. ◀

6 Conclusion

We study the imax-successor and introduced a canonical form for its terms, which gives us an easy procedure decision for the equivalence problem. For that, we extended the grammar with new terms called sublevels, and we expressed any term as a maximum of sublevels, what we have called a representation. Since not all representations are actually terms of the algebra, a next step could be to characterize the representations that are. This could lead to an even better understanding of the imax-successor algebra.

In this article, we only provide a naive canonization algorithm that can be improved. However, one could note that the size of a canonical form can be exponential in the size of the initial term (take for instance nested imax with the term $u = [x_1, \ldots, x_n]$ where the variables x_i are all different).

Finally, this representation can be expressed in $\lambda\Pi/\equiv$ with rewrite rules, which is our initial motivation, and it is used in a Work In Progress translator from Lean to Dedukti[1]

[1] `https://github.com/Deducteam/Lean2dk`

showing that it can indeed be used to express $CC^{\infty}_{\forall}$ in $\lambda\Pi/\equiv$. The next step here, is to study how the expression of universe polymorphism, thanks to this level representation, behaves well together with other features such as inductive types or cumulativity.

References

1 Ali Assaf. *A framework for defining computational higher-order logics.* Theses, École polytechnique, 09 2015. URL: `https://pastel.archives-ouvertes.fr/tel-01235303`.

2 Ali Assaf, Guillaume Burel, Raphaël Cauderlier, David Delahaye, Gilles Dowek, Catherine Dubois, Frédéric Gilbert, Pierre Halmagrand, Olivier Hermant, and Ronan Saillard. Dedukti : a logical framework based on the $\lambda\pi$-calculus modulo theory. 2016.

3 Ali Assaf, Gilles Dowek, Jean-Pierre Jouannaud, and Jiaxiang Liu. Encoding Proofs in Dedukti: the case of Coq proofs. In *Proceedings Hammers for Type Theories*, Proc. Higher-Order rewriting Workshop, Coimbra, Portugal, July 2016. Easy Chair. URL: `https://inria.hal.science/hal-01330980`.

4 Henk Barendregt. Introduction to generalized type systems. *Journal of Functional Programming*, 1(2):125–154, 1991. `doi:10.1017/S0956796800020025`.

5 Henk Barendregt, S. Abramsky, D. Gabbay, T. Maibaum, and Henk (Hendrik) Barendregt. Lambda calculi with types. 10 2000.

6 Stefano Berardi. *Type dependence and constructive mathematics.* PhD thesis, PhD thesis, Dipartimento di Informatica, Torino, Italy, 1990.

7 Frédéric Blanqui. Encoding Type Universes Without Using Matching Modulo Associativity and Commutativity. In Amy P. Felty, editor, *7th International Conference on Formal Structures for Computation and Deduction (FSCD 2022)*, volume 228 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 24:1–24:14, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. URL: `https://drops.dagstuhl.de/opus/volltexte/2022/16305`, `doi:10.4230/LIPIcs.FSCD.2022.24`.

8 Frédéric Blanqui, Gilles Dowek, Émilie Grienenberger, Gabriel Hondet, and François Thiré. Some Axioms for Mathematics. In Naoki Kobayashi, editor, *6th International Conference on Formal Structures for Computation and Deduction (FSCD 2021)*, volume 195 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 20:1–20:19, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. URL: `https://drops.dagstuhl.de/opus/volltexte/2021/14258`, `doi:10.4230/LIPIcs.FSCD.2021.20`.

9 Mathieu Boespflug and Guillaume Burel. Coqine: Translating the calculus of inductive constructions into the $\lambda\pi$-calculus modulo. In *in "Second International Workshop on Proof Exchange for Theorem Proving*, 2012.

10 Mathieu Boespflug, Quentin Carbonneaux, and Olivier Hermant. The λΠ-calculus Modulo as a Universal Proof Language. *CEUR Workshop Proceedings*, 878, 06 2012.

11 Mario Carneiro. The Type Theory of Lean. Master's thesis, Carnegie Mellon University, 2019. URL: `https://github.com/digama0/lean-type-theory/releases`.

12 Thierry Coquand. An analysis of girard's paradox. In *Proceedings of the First Annual IEEE Symposium on Logic in Computer Science (LICS 1986)*, pages 227–236. IEEE Computer Society Press, 06 1986.

13 Thierry Coquand and Gérard Huet. The calculus of constructions. *Information and Computation*, 76(2):95–120, 1988. URL: `https://www.sciencedirect.com/science/article/pii/0890540188900053`, `doi:10.1016/0890-5401(88)90005-3`.

14 Judicaël Courant. Explicit universes for the calculus of constructions. In Victor A. Carreño, César A. Muñoz, and Sofiène Tahar, editors, *Theorem Proving in Higher Order Logics*, pages 115–130, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

15 Denis Cousineau and Gilles Dowek. Embedding pure type systems in the lambda-pi-calculus modulo. pages 102–117, 06 2007. `doi:10.1007/978-3-540-73228-0_9`.

16 Nachum Dershowitz and Jean-Pierre Jouannaud. Rewrite systems. In Jan van Leeuwen, editor, *Handbook of Theoretical Computer Science, Volume B: Formal Models and Semantics*, pages 243–320. Elsevier and MIT Press, 1990. doi:10.1016/b978-0-444-88074-1.50011-1.

17 Michael Färber. Safe, fast, concurrent proof checking for the lambda-pi calculus modulo rewriting. In *Proceedings of the 11th ACM SIGPLAN International Conference on Certified Programs and Proofs*, CPP 2022, page 225–238, New York, NY, USA, 2022. Association for Computing Machinery. doi:10.1145/3497775.3503683.

18 Gaspard Férey. *Higher-Order Confluence and Universe Embedding in the Logical Framework . (Confluence d'ordre supérieur et encodage d'univers dans le Logical Framework.* PhD thesis, École normale supérieure Paris-Saclay, France, 2021. URL: https://lmf.cnrs.fr/downloads/Perso/Ferey-thesis.pdf.

19 Guillaume Genestier. Encoding Agda Programs Using Rewriting. In Zena M. Ariola, editor, *5th International Conference on Formal Structures for Computation and Deduction (FSCD 2020)*, volume 167 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 31:1–31:17, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. URL: https://drops.dagstuhl.de/opus/volltexte/2020/12353, doi:10.4230/LIPIcs.FSCD.2020.31.

20 Yoan Géran. Mathématiques inversées de Coq. Master's thesis, ENS Paris-Saclay, September 2021. URL: https://inria.hal.science/hal-04319183.

21 Yoan Géran. Stt∀ geocoq, 2021. URL: https://github.com/Karnaj/sttfa_geocoq_euclid.

22 Robert Harper and Robert Pollack. Type checking with universes. In *2nd International Joint Conference on Theory and Practice of Software Development*, TAPSOFT '89, page 107–136, NLD, 1991. Elsevier Science Publishers B. V.

23 Gabriel Hondet and Frédéric Blanqui. The New Rewriting Engine of Dedukti (System Description). In Zena M. Ariola, editor, *5th International Conference on Formal Structures for Computation and Deduction (FSCD 2020)*, volume 167 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 35:1–35:16, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. URL: https://drops.dagstuhl.de/opus/volltexte/2020/12357, doi:10.4230/LIPIcs.FSCD.2020.35.

24 Per Martin-Löf. A theory of types, 1971. Preprint, Stockholm University.

25 Per Martin-Löf. An intuitionistic theory of types: Predicative part. *Studies in logic and the foundations of mathematics*, 80:73–118, 1975.

26 Per Martin-Löf. An intuitionistic theory of types. In Giovanni Sambin and Jan M. Smith, editors, *Twenty-five years of constructive type theory (Venice, 1995)*, volume 36 of *Oxford Logic Guides*, pages 127–172. Oxford University Press, 1998.

27 Matthieu Sozeau and Nicolas Tabareau. Universe polymorphism in coq. In Gerwin Klein and Ruben Gamboa, editors, *Interactive Theorem Proving*, pages 499–514, Cham, 2014. Springer International Publishing.

28 Terese. *Term rewriting systems.*, volume 55 of *Cambridge tracts in theoretical computer science*. Cambridge University Press, 2003.

29 François Thiré. Sharing a library between proof assistants: Reaching out to the HOL family. In Frédéric Blanqui and Giselle Reis, editors, *Proceedings of the 13th International Workshop on Logical Frameworks and Meta-Languages: Theory and Practice, LFMTP@FSCD 2018, Oxford, UK, 7th July 2018*, volume 274 of *EPTCS*, pages 57–71, 2018. doi:10.4204/EPTCS.274.5.

30 Vladimir Voevodsky. A universe polymorphic type system, 10 2014. An unfinished unreleased manuscript. URL: https://www.math.ias.edu/Voevodsky/files/files-annotated/Dropbox/Unfinished_papers/Type_systems/UPTS_current/Universe_polymorphic_type_sytem.pdf.