C-system of a module over a monad on sets[1]

Vladimir Voevodsky[2,3]

August 2014

Abstract

This is the second paper in a series started in [26] which aims to provide mathematical descriptions of objects and constructions related to the semantical theory of dependent type systems.

We construct for any pair $(\mathbf{R}, \mathbf{LM})$, where R is a monad on sets and LM is a left module over R, a C-system ("contextual category") $CC(\mathbf{R}, \mathbf{LM})$ and describe, using the results of [26] a class of sub-quotients of $CC(\mathbf{R}, \mathbf{LM})$ in terms of objects directly constructed from R and LM. In the special case of the monads of expressions associated with binding signatures this construction gives, for the first time, a mathematically rigorous way of constructing a C-system from a general collection of judgements of the four Martin-Löf kinds that satisfies a well specified set of conditions.

1 Introduction

The first few steps in all approaches to the semantics of dependent type theories remain insufficiently understood. The constructions which have been worked out in detail in the case of a few particular type systems by dedicated authors are being extended to the wide variety of type systems under consideration today by analogy. This is not acceptable in mathematics. Instead we should be able to obtain the required results for new type systems by *specialization* of general theorems formulated and proved for abstract objects the instances of which combine together to produce the objects associated with a given type system.

An approach that follows this general philosophy was outlined in [21]. In this approach the connection between the type theories, which belong to the concrete world of logic and programming, and abstract mathematical concepts such as sets or homotopy types is constructed through the intermediary of C-systems.

C-systems were introduced in [2] (see also [3]) under the name "contextual categories". A modified axiomatics of C-systems and the construction of new C-systems as sub-objects and regular quotients of the existing ones in a way convenient for use in type-theoretic applications are considered in [26].

In the approach of [21], in order to provide a mathematical interpretation (semantics) for a type theory one constructs two C-systems. One C-system is constructed from the formulas of the type theory using as an initial step the construction of the present paper. The second C-system is constructed from the category of abstract mathematical objects using the results of [22]. Both C-systems are then equipped with additional operations corresponding to the "inference rules" of the type theory.

The main component of this approach is the expected result that for a particular class of the inference rules the concrete C-systems built using the constructions of the present paper and equipped

[1] *2000 Mathematical Subject Classification: 18D99, 08C99, 03B15*

[2] School of Mathematics, Institute for Advanced Study, Princeton NJ, USA. e-mail: vladimir@ias.edu

[3] Work on this paper was supported by NSF grant 1100938.

with operations corresponding to these inference rules are initial objects in the category of C-systems with the corresponding operations. This is known as the Initiality Conjecture. In the case of the pure Calculus of Constructions this conjecture was proved in 1988 by Thomas Streicher [19]. The problem of finding an appropriate formulation of the general version of the conjecture and of proving this general version will be the subject of future work.

For such inference rules, then, there are unique homomorphisms from the concrete C-systems to the abstract C-systems that are compatible with the corresponding systems of operations. Since objects and morphisms of concrete C-systems are built from formulas of the type theory and objects and morphisms of abstract C-systems are built from mathematical objects such as sets or homotopy types and the corresponding functions, these homomorphisms provide a mathematical meaning to formulas of type theory.

The existence of such homomorphisms in the particular case of the "standard univalent models" of Martin-Löf type theories and of the Calculus of Inductive Constructions (CIC) provides the only known justification for the use of the proof assistants such as Coq for the formalization of mathematics in the univalent style (see [27], [23]).

It is important to distinguish the concepts of a model of a type theory and the concept of an interpretation of the same type theory. A *model* of type theory can be defined as a C-system that is equipped with the systems of operations corresponding to the inference rules of the type theory. A (categorical) *interpretation* of a type theory with values in a given category $\mathcal{C}$ is a functor from the category underlying the syntactic C-system of the type theory to $\mathcal{C}$.

Only if we know that the initiality result holds for a given type theory can we claim that any its model defines an interpretation by taking the composition of the canonical homomorphism of the C-systems with the functor such as the functor *int* of [22]. A similar problem also arises in the predicate logic but there, since one considers only one fixed system of syntax and inference rules, it can and had been solved once without the development of a general theory.

When one speaks about the univalent model of a Martin-Löf type theory or of the CIC one often fails to distinguish between the model and the interpretation. A construction of a *model* for the version of the Martin-Löf type theory that is used in the UniMath library ([27],[23]) was sketched in [11]. At the time when that paper was written it was unfortunately assumed that a proof of the initiality result can be found in the existing body of work on type theory which is reflected in [11, Theorem 1.2.9] (cf. also [11, Example 1.2.3] that claims as obvious everything that is done in both the present paper and in [26]). Since then it became clear that this is not the case and that a mathematical theory leading to the initiality theorem and providing a proof of such a theorem is lacking and needs to be developed.

As the criteria for what constitutes an acceptable proof were becoming more clear as a result of continuing work on formalization, it also became clear that more detailed and general proofs need to be given to many of the theorems of [11] that are related to the model itself. For the two of the several main groups of inference rules of current type theories it is done in [25] and [24]. Other groups of inference rules will be considered in further papers of the series.

This paper may be considered to be an analog of [22] for the concrete side of the theory in the sense that it provides a very general construction the particular cases of which lead to the concrete (syntactic) C-systems of type theories.

If $R = (R, \eta, \mu)$ is a monad on a category $\mathcal{C}$ (see Definition **??**) then there is defined the Kleisli category $\mathcal{C}_R$ of R whose objects are the same as objects of $\mathcal{C}$ and morphisms from X to Y are

defined as morphisms from X to $R(Y)$ in $\mathcal{C}$. The identity morphisms in $\mathcal{C}_R$ are given by the η operation of R and the composition by the composition in $\mathcal{C}$ and the μ operation of R.

A left R-module LM over R with values in a category $\mathcal{D}$ (see Definition **??**) defines a functor $LM_R : \mathcal{C}_R \to \mathcal{D}$ and this function from left R-modules to functors from the Kleisli category is an equivalence (see **??**).

An important case is the left R-module corresponding to R itself which we will also denote by R.

Monads on the category of sets and left modules over such monads have a number of special ????

Of a particular interest is the case of "syntactic" pairs $(\mathbf{R}, \mathbf{LM})$ where for $X = \{x_1, \ldots, x_n\}$, $R(X)$ and $LM(X)$ are the sets of expressions of some kind with free variables from $\{x_1, \ldots, x_n\}$ modulo an equivalence relation such as α-equivalence. The difference between R and LM is in this case expressed by the fact that one can substitute elements from $R(X)$ for variables both in $R(Y)$ and $LM(Y)$ but elements of $LM(X)$ can not be substituted for variables in either.

The simplest class of syntactic pairs, where $LM = R$, arises from binding signatures (see [7, p.228]). To any such signature Σ one associates a class of expressions with bindings and $R(\{x_1, \ldots, x_n\})$ is the set of such expressions with free variables from the set $\{x_1, \ldots, x_n\}$ modulo α-equivalence.

The more general case when LM is not equal to R arises when one starts to distinguish "type expressions" from "object expressions". The rules of type theories require the possibility to substitute an object expression instead of a variable both in a type expression and in an object expression but do not require to substitute a type expression instead of a variable either in a type or in an object expression. In type theories of proof assistants such as Coq the user may be under the impression that the substitution of type expressions instead of variables occurs (as in substituting $unit$ for T in $iscontr(T)$ in the UniMath to obtain $iscont(unit)$, cf. [23]) this is however due to a "silent" map from object expressions to type expressions that is used in these theories. What actually happens in these substitutions is that an object expression whose type is a universe is substituted instead of a variable in some situations and the same object expression is mapped to the set of type expressions and used as a type expression in others. In our constructions this corresponds to $LM = R$ - an object expression that is an element of $R(X)$ for some set of variables X is considered as an element of the set of type expressions $LM(X)$ using the identity map defined by this equality (more generally one may observe the same illusion when $LM \subset R$).

The question of whether to keep this map silent or to give it a name (usually El) is know in type theory as the difference between the type theories with "Russell universes" (silent map) and "Tarski universes" (explicit map) which is at the center of some of the current controversies about the universe management in proof assistants. It is also the subject of a discussion in the last, unfinished, chapter in [17].

For the purposes of the present paper we fortunately don't need to make a choice between the two approaches since the formalism that we develop is applicable to both. It is however clear from the constructions that the separation between R and LM is a very natural possibility that directly generalizes the case of $LM \subset R$ and creates new examples (e.g. Example **??**).

As was shown in [7] the monad that one associates to a binding signature can be characterized as being an initial object in the category of monads equipped with "left-linear" operations corresponding to the operations of the signature. This provides an abstract mathematical characterization of the concrete objects - expressions modulo α-equivalence or, equivalently, expressions with De Brujin indexes.

An important remark needs to be made here. While monads provide a very convenient way of expressing syntax with bindings in terms familiar to mathematicians the approach based on monads is equivalent to an earlier one pioneered in [4]. For two sets X and Y let $Fun(X,Y)$ be the set of functions from X to Y. In that earlier approach one considers the category F such that $Ob(F) = \mathbf{N}$ and

$$Mor(F) = \amalg_{m,n} Fun(stn(m), stn(n))$$

where $stn(i) = \{0, \ldots, i-1\}$ is the "standard" set with i elements, and functors $Funct(F, Sets)$ from F to $Sets$ (the authors call these functors "presheaves" considering them as presheaves on F^{op}) . This category of functors is equivalent[4] to the category of finitary (co-continuous) functors from $Sets$ to $Sets$. In particular, there is a monoidal structure $(\bullet, V)$ on $Funct(F, Sets)$ corresponding to the composition of functors under this equivalence (cf. [4, Sec. 3]) and finitary monads can be considered as monoids in $Funct(F, Sets)$ with respect to this monoidal structure.

Using this equivalence of concepts (detailed in []) the constructions and results of [7] and [4] can be viewed together as describing different aspects of a fundamental connection between the concrete world of syntax and the abstract world of categorical mathematics.

After this long detour let me clarify that the results and constructions of the present paper do not depend on either [7] or [4], except for the definition of a left module over a monad in [7] and examples. The connection to [7] and [4] will become important only in future papers where we will consider the abstract concept of a system of inference rules and where binding signatures and the corresponding syntactic monads will become essential.

In the present paper, after some general comments about monads on $Sets$ and their modules, we construct for any such monad R and a left module LM over R a C-system (contextual category) $CC(\mathbf{R}, \mathbf{LM})$. We start with a construction of a category $C(R)$ such that $Ob(C(R)) = \mathbf{N}$ is the set of natural numbers whose elements we will denote as $\widehat{m}$, $\widehat{n}$ etc. and

$$Mor(C(R)) = \amalg_{\widehat{m},\widehat{n}} Hom_{Sets_R}(stn(n), stn(m))$$

and the identity and composition is defined such as to make the mapping $\widehat{n} \mapsto stn(n)$ to extend to a fully faithful functor cr_1 from $C(R)^{op}$ to the Kleisli category $Sets_R$ of R. We may sometimes use this functor as a "coercion", in the terminology of proof assistant Coq, i.e., to write $\widehat{n}$ instead of $stn(n)$ and f instead of $\Phi(f)$. We will also use the function $LM \mapsto LM_R$ from left modules to functors on the Kleisli category as a coercion. In agreement with this convention we may write LM for the presheaf of sets on $C(R)$ given by $\widehat{n} \mapsto LM(stn(n))$.

We describe, using the results of [26], all the C-subsystems of $CC(\mathbf{R}, \mathbf{LM})$ in terms of objects directly associated with R and LM.

We then define two additional operations σ and $\widetilde{\sigma}$ on $CC(\mathbf{R}, \mathbf{LM})$ and describe the regular congruence relations (see [26]) on C-subsystems of $CC(\mathbf{R}, \mathbf{LM})$ which are compatible in a certain sense with σ and $\widetilde{\sigma}$.

Such regular congruence relations correspond, in the particular cases of syntactic monads and C-subsystems of $CC(\mathbf{R}, \mathbf{LM})$ generated by systems of inference rules, to the relations that can be described by the two kinds of equality judgements.

[4] In the set-theoretic mathematics this equivalence can not be defined without axiom of choice. The problem lies in the fact that the obvious functor from F to the category of finite sets, while it is fully faithful and essentially surjective, does not have a constructive inverse. In the univalent foundations, while one still can not construct an inverse to the functor from F to finite sets, one can construct an inverse to the corresponding functor from $Funct(FSets, Sets)$ to $Funct(F, Sets)$ using the fact that $Sets$ is a (univalent) category. Cf. [1] and [27, RezkCompletion library].

More precisely, suppose that we are given a type theory that is formulated in terms of the four kinds of judgements originally introduced by Per Martin-Löf in [16, p.161][5]:

$$(x_0 : T_0, \ldots, x_{n-1} : T_{n-1})\, T \ type$$

$$(x_0 : T_0, \ldots, x_{n-1} : T_{n-1})\, t : T$$

$$(x_0 : T_0, \ldots, x_{n-1} : T_{n-1})\, T = T'$$

$$(x_0 : T_0, \ldots, x_{n-1} : T_{n-1})\, t = t' : T$$

to which one adds the judgement

$$(x_0 : T_0, \ldots, x_{n-1} : T_{n-1})\, ok$$

asserting that $(x_0 : T_0, \ldots, x_{n-1} : T_{n-1})$ is a valid context of variable declarations.

Since we are only interested in the α-equivalence classes of judgements we may assume that the variables declared in the context are taken from the set of natural numbers such that the first declared variable is 0, the second is 1 etc. Then, the set of judgements of the form

$$(0 : T_0, \ldots, n-1 : T_{n-1})\, T \ type$$

can be identified with the set of judgements of the form

$$(0 : T_0, \ldots, n-1 : T_{n-1}, n : T)\, ok$$

With this identification the derivable judgements of the type theory whose raw syntax for object expressions is given by a monad R and raw syntax for type expressions by a left R-module LM, can be described as four subsets $\widetilde{B}, B, Beq$ and $\widetilde{Beq}$ where

$$\widetilde{B} \subset \coprod_{n\geq 0} LM(stn(0)) \times \ldots \times LM(stn(n-1))$$

$$B \subset \coprod_{n\geq 0} LM(stn(0)) \times \ldots \times LM(stn(n-1)) \times R(stn(n)) \times LM(stn(n))$$

$$Beq \subset \coprod_{n\geq 0} LM(stn(0)) \times \ldots \times LM(stn(n-1)) \times LM(stn(n))^2$$

$$\widetilde{Beq} \subset \coprod_{n\geq 0} LM(stn(0)) \times \ldots \times LM(stn(n-1)) \times R(stn(n))^2 \times LM(stn(n))$$

The sets on the right hand side of the first two of these inclusions are in the bijective correspondences with the sets $Ob(CC(\mathbf{R}, \mathbf{LM}))$ and $\widetilde{Ob}(CC(\mathbf{R}, \mathbf{LM}))$. It was shown in [26, Proposition 4.3] that for any C-system CC, pairs $(B, \widetilde{B})$ where $B \subset Ob(CC)$ and $\widetilde{B} \subset \widetilde{Ob}(CC)$ that satisfy certain conditions are in a bijective correspondence with C-subsystems of CC. In Proposition 4.1 we give a direct reformulation of these conditions in the case of C-systems of the form $CC(\mathbf{R}, \mathbf{LM})$ in terms of subsets $\widetilde{B}$ and B and in Remark 4.2 we show how these conditions look like in the notation of type theory.

[5] We are not using the notation based on $\vdash$ that became widespread in the modern literature on type theory since it conflicts with other uses of the turnstile symbol in logic.

We then continue our analysis to provide a mathematical meaning to the subsets Beq and $\widetilde{Beq}$ as well. In order to obtain a bijection of Proposition 6.11 between pairs of such subsets that satisfy certain properties and objects that have meaning for general C-systems we introduce operations σ and $\widetilde{\sigma}$.

Proposition 6.2 and subsequent lemmas culminating in Proposition 6.11 form what is probably the most important part of the paper. They provide, for the first time, a rigorous mathematical analysis of the conditions that the derivable definitional equality judgements of a type system have to satisfy in order to define well-behaved equivalence relation on the sets such as the sets of morphisms (context substitutions) of a type theory.

While proving conditions of Proposition 6.2 in the case when $\widetilde{B}$, B, Beq and $\widetilde{Beq}$ are the sets of derivable judgements of a particular type system is something that must be done in order to apply the results of the present paper to this type system, proving these conditions is much less difficult than giving a direct construction of a C-system starting from the syntax and the inference rules.

Providing this explicit set of conditions and proving that they are necessary and sufficient in order to associate a C-system and, therefore, any of the other semantic objects such as a category with families, to a particular type system may be considered to be the main result of this paper.

For morphisms $f : X \to Y$ and $g : Y \to Z$ we denote their composition as $f \circ g$. For functors $F : \mathcal{C} \to \mathcal{C}'$, $G : \mathcal{C}' \to \mathcal{C}''$ we use the standard notation $G \circ F$ for their composition.

We often write y instead of (x, y) for an element of the set $\amalg_{x \in X} Y(x)$ corresponding to $x \in X$ and $y \in Y(x)$. For example, we may write f for the element $(m, (n, f))$ of $Mor(F)$ corresponding to a function $f : \{1, \dots, m\} \to \{1, \dots, n\}$.

Following the notation of the proof assistant Coq we let *unit* denote the distinguished one point set or type and *tt* the only point of *unit*.

This is one the papers extending the material which I started to work on in [20]. I would like to thank the Institute Henri Poincare in Paris and the organizers of the "Proofs" trimester for their hospitality during the preparation of this paper. The work on this paper was facilitated by discussions with Richard Garner and Egbert Rijke.

2 Left modules over monads

Definition 2.1 below is, according to Manes [15, p.30], due to Godement [6]. See also [14, Ch. VI].

Definition 2.1 *[2015.07.30.def1] A monad on a category $\mathcal{C}$ is a collection of data of the form:*

1. *a functor $R : \mathcal{C} \to \mathcal{C}$,*
2. *for any $X \in \mathcal{C}$, a morphism $\eta_X : X \to R(X)$,*
3. *for any $X \in \mathcal{C}$, a morphism $\mu_X : R(R(X)) \to R(X)$*

such that:

1. *for any $f : X \to Y$ one has*
   (a) *$\eta_X \circ R(f) = f \circ \eta_Y$,*

(b) $\mu_X \circ R(f) = R(R(f)) \circ \mu_Y$,

2. *for any* X *one has*

$$R(\mu_X) \circ \mu_X = \mu_{R(X)} \circ \mu_X$$

3. *for any* X *one has*

$$R(\eta_X) \circ \mu_X = Id_{R(X)}$$

and

$$\eta_{R(X)} \circ \mu_X = Id_{R(X)}$$

Remark 2.2 [**2015.07.30.rem1**] It is very easy to construct an equivalence between the type of monads on a given precategory $\mathcal{C}$ (resp. a bijection between the set of monads on $\mathcal{C}$) and the type (resp. set) of monoids in the category $Funct(\mathcal{C},\mathcal{C})$ with respect to the monoidal structure given by the composition of functors. This equivalence, in the case of a type theoretic formalization, is actually an isomorphism where by an isomorphism we mean an isomorphism of the corresponding objects of the syntactic category.

Remark 2.3 [**2015.08.12.rem1**] There are at least two other definitions that specify, over any universe, types of objects that are equivalent to the type of monads. Objects specified by one of these definition are called by Manes "algebraic theories in clone form in $\mathcal{C}$". See [15, Def. 3.2, p.24]. The objects specified by another one, which appears in [15, Exercise 12, p.32] and then more explicitly in [18] are called by Moggi "Kleisli triples". Kleisli triples are more popular than monads in papers related to computer science while monads are more popular in mathematical papers. Formally proving the equivalence between these three definitions as well as constructing examples that show that these equivalences are not isomorphisms in the syntactic category may be the topic for a small project in univalent formalization.

An analogous project in the set-theoretic formalization should be able to construct bijections between the corresponding sets because the underlying function $R_{Ob} : Ob(\mathcal{C}) \to Ob(\mathcal{C})$ remains unchanged by the corresponding equivalences and therefore they can be seen as bijections between the sets of monad, algebraic theory in clone form, and Kleisli triple structures on a given function R_{Ob}.

Problem 2.4 *[**2015.07.30.prob3**] Given a monad (R,η,μ) on a precategory $\mathcal{C}$ to construct a new precategory $\mathcal{C}_R$ and a functor $G_R : \mathcal{C} \to \mathcal{C}_R$.*

The following construction first appeared in [12] who worked with the dual concept that we today would call a co-monad. To obtain the correct correspondence between his construction and Construction 2.5 one needs to replace his $\mathcal{L}$ by our $\mathcal{C}^{op}$, his $\mathcal{K}$ by our $(\mathcal{C}_R)^{op}$ and his G by our $(G_R)^{op}$.

Construction 2.5 [**2015.07.30.constr3**] We set $Ob(\mathcal{C}_R) = Ob(\mathcal{C})$ and

$$Mor(\mathcal{C}_R) = \amalg_{X,Y \in Ob(\mathcal{C})} Hom_{\mathcal{C}}(X, R(Y))$$

For $X \in Ob(\mathcal{C})$ define Id_X in $Mor(\mathcal{C}_R)$ by the formula

$$Id_{X,\mathcal{C}_R} = (X, (X, \eta_X))$$

For $f = (X, (X', f_0 : X \to R(X')))$ and $g = (X', (X'', g_0 : X' \to R(X'')))$ in $Mor(\mathcal{C}_R)$ define their composition $f \circ g$ by the formula

$$f \circ g = (X, (X'', f_0 \circ R(g_0) \circ \mu_{X''}))$$

Define the object component $(G_R)_{Ob}$ of the functor G_R to be the identity function of $Ob(\mathcal{C})$ the morphism component by the formula

$$(G_R)_{Mor}(f : X \to X') = (X, (X', f \circ \eta_{X'}))$$

For the verification of the axioms needed for the proof that these data defines a category and a functor we refer to [12].

The category $\mathcal{C}_R$ specified by Construction 2.5 is called the Kleisli category of R.

Definition 2.6 *[**2015.07.30.def4**] Let $\mathcal{C}$ be a category and (R, η, μ) a monad on $\mathcal{C}$. A left module over R with values in a category $\mathcal{D}$ is a collection of data of the following form*

1. *a functor $LM : \mathcal{C} \to \mathcal{D}$,*
2. *for any $X \in \mathcal{C}$ a morphism $\rho_X : LM(R(X)) \to LM(X)$*

such that:

1. *for any $f : X \to Y$ one has*

$$\rho_X \circ LM(f) = LM(R(f)) \circ \rho_Y$$

2. *for any X one has:*

$$LM(\eta_X) \circ \rho_X = Id_{LM(X)}$$

and

3. *for any X one has:*

$$LM(\mu_X) \circ \rho_X = \rho_{R(X)} \circ \rho_X$$

Example 2.7 [**2015.07.30.ex1**] Taking $\mathcal{D} = \mathcal{C}$, $LM = R$ and $\rho = \mu$ one obtains a left module over R whose underlying functor is R itself.

Remark 2.8 [**2015.07.30.rem1**] It is very easy to construct an equivalence between the type of left modules over a given monad (resp. a bijection between the set of left modules over a given monad) with values in the same category $\mathcal{C}$ and the type (resp. set) of left modules over the corresponding monoid in the monoidal category $(Funct(\mathcal{C}, \mathcal{C}), \circ)$. In the type theoretic case this equivalence is an isomorphism of the corresponding objects of the syntactic category.

The following construction was, to the best of our knowledge, first described in [9, Def. 10, p.550] under the name of a Kleisli extension.

Problem 2.9 *[**2015.07.30.prob5**] For a monad R and a left module LM over R to construct a functor $LM_R : \mathcal{C}_R \to \mathcal{D}$.*

Construction 2.10 [**2015.07.30.constr4**] We define the object component of LM_R to be the object component of LM. For $f = (X, (X', f_0))$ in $Mor(\mathcal{C}_R)$ we define

$$(LM_R)_{Mor}(f) = LM(f_0) \circ \rho_{X'}$$

For the identity morphism axiom we have

$$LM_R(Id_X) = LM_R((X, (X, \eta_X))) = LM(\eta_X) \circ \rho_X = Id_{LM(X)}$$

For the composition axiom we have

$$LM_R((X, (X', f_0)) \circ (X', (X'', g_0))) = LM_R((X, (X'', f_0 \circ R(g_0) \circ \mu_{X''}))) =$$

$$LM(f_0 \circ R(g_0) \circ \mu_{X''}) \circ \rho_{X''} = LM(f_0) \circ LM(R(g_0)) \circ LM(\mu_{X''}) \circ \rho_{X''} =$$

$$LM(f_0) \circ LM(R(g_0)) \circ \rho_{R(X'')} \circ \rho_{X''} = LM(f_0) \circ \rho_{X'} \circ LM(g_0) \circ \rho_{X''}$$

and

$$LM_R((X, (X', f_0))) \circ LM_R((X', (X'', g_0))) = LM(f_0) \circ \rho_{X'} \circ LM(g_0) \circ \rho_{X''}$$

This completes Construction 2.10.

Lemma 2.11 *[**2015.07.30.l1**] For a monad R on $\mathcal{C}$, a left module LM and $f : X \to X'$ in $\mathcal{C}$ one has*

$$LM_R(G_R(f)) = LM(f)$$

Proof: One has

$$LM_R(G_R(f)) = LM_R((X, (X', f \circ \eta_{X'}))) = LM(f \circ \eta_{X'}) \circ \rho_{X'} = LM(f) \circ LM(\eta_{X'}) \circ \rho_{X'} =$$

$$LM(f) \circ Id_{LM(X')} = LM(f)$$

Remark 2.12 A construction of a left R-module from a functor $\mathcal{C}_R \to \mathcal{D}$ is outlined in [9, Prop. 3] and in the same paper there is an outline of a proof that these two constructions are inverse to each other providing a bijection between left R-modules with values in a category $\mathcal{D}$ and functors $\mathcal{C}_R \to \mathcal{D}$. This bijection is identity on the underlying functions from the type (resp. set) of objects of $\mathcal{C}$ to the type (resp. set) of objects of $\mathcal{D}$ so that one can talk about the bijection between the sets of a left R-module structures on a function $Ob(\mathcal{C}) \to Ob(\mathcal{D})$ and the set of functor structures on the same function considered as a function from $Ob(\mathcal{C}_R)$. In type theoretic formalization this equivalence is not an isomorphism in the syntactic category.

Remark 2.13 [**2015.08.12.rem3**] In the univalent foundations the relationship between the type of finitary functors $Sets \to Sets$ (where by $Sets$ we mean the category of h-sets in a fixed universe) and the type of functors $FSets \to Sets$ is not entirely clear at the moment. On the other hand the relationship between the type of functors $FSets \to Sets$ and the type of functors $F \to Sets$ is better in the univalent foundations then in the set-theoretic ones since in the univalent foundations these two types are constructively equivalent (cf. [1, Theorem 8.4]).

Definition 2.14 *[**2015.08.17.def1**] A pre-algebraic theory is a collection of data of the following form*

1. *a function* $R : \mathbf{N} \to Sets$,
2. *for each* n *a function* $\eta_n \in Fun(stn(n), R(n))$,
3. *for each* $f \in Fun(stn(n), R(m))$ *a function* $bind(f) \in Fun(R(n), R(m))$,

such that the following conditions hold:

1. *for all* n, $bind(\eta_n) = Id_{R(n)}$,
2. *for all* $f \in Fun(stn(n), stn(m))$, $\eta_n \circ bind(f) = f$,
3. *for all* $f \in Fun(stn(n), R(n'))$, $g \in Fun(stn(n'), R(n''))$, $bind(f \circ bind(g)) = bind(f) \circ bind(g)$.

Let us introduce the following notation:

$$F(n, m) = Fun(stn(n), stn(m))$$

and, for a pre-algebraic theory R,

$$R(n, m) = Fun(stn(n), R(m))$$

Let $\mathbf{R} = (R, \eta, bind)$ be a pre-algebraic theory. For each pair $n, m \in \mathbf{N}$ define a function

$$\phi_{\mathbf{R}} : F(n, m) \to R(n, m)$$

by the formula

$$\phi_{\mathbf{R}}(f) = f \circ \eta_m$$

For each triple n, m, k and $f \in R(n, m)$, $g \in R(m, k)$ define $f \widehat{\circ} g \in R(n, k)$ by the formula

$$f \widehat{\circ} g = f \circ bind(g)$$

Lemma 2.15 *[**2015.08.18.l1**] Let* $\mathbf{R} = (R, \eta, bind)$ *be a pre-algebraic theory. Then one has:*

1. *for any* n, m, k, l *and* $f \in R(n, m)$, $g \in R(m, k)$, $h \in R(k, l)$ *one has*

$$(f \widehat{\circ} g) \widehat{\circ} h = f \widehat{\circ} (g \widehat{\circ} h)$$

2. *for any* $f \in R(n, m)$ *one has*

$$f \widehat{\circ} \eta_m = f$$
$$\eta_n \widehat{\circ} f = f$$

3. *for any* $f \in F(n, m)$, $g \in R(n, m)$ *one has*

$$\phi_{\mathbf{R}}(f) \widehat{\circ} g = f \circ g$$

4. *for any* $f \in F(n, m)$, $g \in F(m, k)$ *one has*

$$\phi_{\mathbf{R}}(f) \widehat{\circ} \phi_{\mathbf{R}}(g) = \phi_{\mathbf{R}}(f \circ g)$$

Proof:

1. We have

$$(f\widehat{\circ}g)\widehat{\circ}h = (f \circ bind(g)) \circ bind(h) = f \circ (bind(g) \circ bind(h)) = f \circ bind(g \circ bind(h)) = f\widehat{\circ}(g\widehat{\circ}h)$$

2. We have

$$f\widehat{\circ}\eta_m = f \circ bind(\eta_m) = f \circ Id_{R(m)} = f$$
$$\eta_n\widehat{\circ}f = \eta_n \circ bind(f) = f$$

3. We have

$$\phi_{\mathbf{R}}(f)\widehat{\circ}g = \phi_{\mathbf{R}}(f) \circ bind(g) = f \circ \eta_m \circ bind(g) = f \circ (\eta_m \circ bind(g)) = f \circ g$$

4. We have

$$\phi_{\mathbf{R}}(f)\widehat{\circ}\phi_{\mathbf{R}}(g) = f \circ \phi_{\mathbf{R}}(g) = f \circ g \circ \eta_k = (f \circ g) \circ \eta_k = \phi_{\mathbf{R}}(f \circ g)$$

Definition 2.16 *[**2015.08.17.def2**] A pre-algebra over a pre-algebraic theory $(R, \eta, bind)$ is a collection of data of the following form:*

1. *a function $LM : \mathbf{N} \to Sets$,*
2. *for each $f \in R(n, m)$ a function $\rho(f) \in F(LM(n), LM(m))$,*

such that the following conditions hold

1. $\rho(\eta_n) = Id_{LM(n)}$,
2. *for $f \in R(n, m)$, $g \in R(m, k)$, $\rho(f) \circ \rho(g) = \rho(f\widehat{\circ}g)$.*

Remark 2.17 [**2015.08.18.rem1**] Note that for any algebraic pre-theory $\mathbf{R} = (R, \eta, bind)$ the pair $(R, bind)$ is a pre-algebra over $\mathbf{R}$.

3 The C-system $CC(\mathbf{R}, \mathbf{LM})$.

Let $\mathbf{R}$ be a pre-algebraic theory and $\mathbf{LM}$ a pre-algebra over $\mathbf{R}$. Let $CC(\mathbf{R}, \mathbf{LM})$ be the pre-category whose set of objects is

$$Ob(CC(\mathbf{R}, \mathbf{LM})) = \amalg_{n \geq 0} Ob_n(\mathbf{R}, \mathbf{LM})$$

where

$$Ob_n(\mathbf{R}, \mathbf{LM}) = LM(0) \times \ldots \times LM(n-1)$$

Remark 3.1 [**2015.08.14.rem1**] In a univalent formalization based on UniMath one can define $Ob_n(\mathbf{R}, \mathbf{LM})$ as the type $forall(i : stn\, n), (LM\, (i))$.

Define the function l on $Ob(CC(\mathbf{R},\mathbf{LM}))$ setting $l(n,A)=n$.

The set of morphisms of $CC(\mathbf{R},\mathbf{LM})$ is

$$Mor(CC(\mathbf{R},\mathbf{LM})) = \coprod_{\Gamma,\Gamma'\in Ob(CC(\mathbf{R},\mathbf{LM}))} R(l(\Gamma'),l(\Gamma))$$

with the obvious domain and codomain maps.

The composition of morphisms in $CC(\mathbf{R},\mathbf{LM})$ is defined by the formula

$$(\Gamma,(\Gamma',f))\circ(\Gamma',(\Gamma'',g)) = (\Gamma,(\Gamma'',g\widehat{\circ}f))$$

and the identity morphisms by

$$Id_\Gamma = (\Gamma,(\Gamma,\eta_{l(\Gamma)}))$$

The axioms of a category follow from Lemma 2.15(1,2).

Since we will have to deal with elements of the sets of functions $R(stn(m))^{stn(n)}$ and of similar sets such as the sets $Ob_n(CC(\mathbf{R},\mathbf{LM}))$ we need to choose some way to represent them. For the purpose of the present paper we will write such elements as sequences, i.e., to denote the function, which in the notation of λ-calculus is written as $\lambda\, i : stn(n), f_i$ we will write $(f_0,\dots,f_{n-1})$.

To simplify the notation we will sometimes use the mappings $\phi_{\mathbf{R}}$ and ρ as, using the terminology of the proof assistant Coq, coercions. That is, for $f\in R(n,m)$ we may write f instead of $\rho(f)$ and similarly for $f\in F(n,m)$ we may write f instead of $\phi_{\mathbf{R}}(f)$. For example, for $f\in F(n,m)$ and $E\in LM(n)$ we may write $f(E)$ where the complete expression would be $\rho(\phi_{\mathbf{R}}(f))(E)$. Similarly, we will use as coercions the mappings η_n such that if we write $i\in\mathbf{N}$ where an element of $R(stn(n))$ is required it is assumed that i needs to be replaced by $\eta_n(i)$ before the computation can occur.

Let

$$\partial^i_n : stn(n)\to stn(n+1)$$

for $0\le i\le n$ be the increasing inclusion that does not take the value i and

$$\sigma^i_n : stn(n+2)\to stn(n+1)$$

for $0\le i\le n$ be the increasing surjection that takes the value i twice. Taking into account that $stn(n)=[n-1]$ in the notation of [5] these are the standard generators of the simplicial category Δ together with $\partial^0_0 : stn(0)\to stn(1)$. In our sequence notation we have

$$\partial^i_n = (0,\dots,i-1,i+1,\dots,n)$$

and

$$\sigma^i_n = (0,\dots,i-1,i,i,i+1,\dots,n)$$

in particular

$$[\mathbf{2015.07.12.eq5}]\partial^n_n = (0,\dots,n-1) \tag{1}$$

Remark 3.2 [**2015.08.18.rem1**]If we think of $E\in LM(stn(n))$ as of an expression in variables $0,\dots,n-1$ then we have:

$$\partial^i_n(E) = E[0/0,\dots,i-1/i-1,i+1/i,\dots,n/n-1]$$

in particular for $E \in LM(stn(n))$, $\partial^n_n(E)$ is "the same" expression but considered as an expression of $n+1$-variables.

Similarly, for $E \in LM(stn(n+2))$ one has

$$\sigma^i_n(E) = E[0/0, \dots, i/i, i/i+1, \dots, n/n+1]$$

For $\Gamma \in Ob(CC(\mathbf{R}, \mathbf{LM}))$ such that $\Gamma = (n+1, (T_0, \dots, T_n))$ denote by $ft(\Gamma)$ the object $(n, (T_0, \dots, T_{n-1}))$ and by p_Γ the morphism

$$(\Gamma, (ft(\Gamma), \partial^n_n)) : \Gamma \to ft(\Gamma)$$

where we have used our coercion convention to write ∂^n_n instead of $\phi_{\mathbf{R}}(\partial^n_n)$.

Lemma 3.3 *[**2015.07.24.l1**] One has:*

1. *Let $f = (\Gamma, (\Gamma', ff))$ where $ff = (f_0, \dots, f_n)$ be a morphism. Then*

$$f \circ p_{\Gamma'} = (\Gamma, (ft(\Gamma'), (f_0, \dots, f_{n-1})))$$

2. *Let $f = (ft(\Gamma), (\Gamma', ff))$ where $ff = (f_0, \dots, f_n)$ be a morphism. Then*

$$p_\Gamma \circ f = (\partial^m_m(f_0), \dots, \partial^m_m(f_n))$$

where $l(\Gamma) = m+1$.

Remark 3.4 [**2015.08.18.rem2**] In the second assertion of the lemma we use our coercion convention in the right hand side of the equality applied to $\mathbf{R}$ considered as a pre-algebra over itself. The full form of the expressions $\partial^m_m(f_i)$ that we use there are $bind(\phi_{\mathbf{R}}(\partial^m_m))(f_i)$.

Proof: For the first assertion we need to check that for $i = 0, \dots, n-1$ one has:

$$(\partial^n_n \widehat{\circ} ff)(i) = ff(i)$$

We have

$$(\partial^n_n \widehat{\circ} ff)(i) = (\phi_{\mathbf{R}}(\partial^n_n) \widehat{\circ} ff)(i) = (\partial^n_n \circ ff)(i) = ff(\partial^n_n(i)) = ff(i)$$

where the first equality is by our convention that we use $\phi_{\mathbf{R}}$ as a coercion and the second equality is by Lemma 2.15(3).

For the second assertion we need to check that for $i = 0, \dots, n$ one has:

$$(ff \widehat{\circ} \partial^m_m)(i) = bind(\phi_{\mathbf{R}}(\partial^m_m))(ff(i))$$

We have

$$(ff \widehat{\circ} \partial^m_m)(i) = (ff \widehat{\circ} (\phi_{\mathbf{R}}(\partial^m_m)))(i) = (ff \circ bind(\phi_{\mathbf{R}}(\partial^m_m)))(i) = bind(\phi_{\mathbf{R}}(\partial^m_m))(ff(i))$$

Problem 3.5 *[**2015.08.17.prob1**] To construct the structure of a C0-system (cf. [26, Definition 2.1]) on the pre-category $CC(\mathbf{R}, \mathbf{LM})$.*

Construction 3.6 [**2015.08.17.constr1**] The length function l and the function ft have already been defined. The object pt is the only element of $Ob(CC(\mathbf{R}, \mathbf{LM}))$ of length 0. The canonical projections have already been defined for elements of length > 0 and we define p_{pt} as the identity morphism.

Given two objects $\Gamma' = (T'_0, \dots, T'_{m-1})$ and $\Delta = (T_0, \dots, T_n)$ and a morphism

$$f = (\Gamma', (ft(\Delta), (f_0, \dots, f_{n_1}))) : \Gamma' \to ft(\Delta)$$

one defines an object $f^*(\Delta)$ and a morphism $q(f, \Delta) : f^*(\Delta) \to \Delta$ as follows:

$$f^*(\Delta) = (T'_0, \dots, T'_{m-1}, ff(T_n))$$

where $ff = (f_0, \dots, f_{n-1})$ and

$$q(f, \Delta) = (f^*(\Delta), (\Delta, qq(f)))$$

where for $f \in R(m, n)$,

$$qq(f) = (\partial^m_m(f_0), \dots, \partial^m_m(f_{n-1}), m)$$

The first four conditions of [26, Definition 2.1] are obvious. The fifth condition asserts that for all Γ', Δ and f as above the square

$$\begin{array}{ccc} f^*(\Delta) & \xrightarrow{q(f,\Delta)} & \Delta \\ {\scriptstyle p_{f^*(\Delta)}}\downarrow & & \downarrow{\scriptstyle p_\Delta} \\ \Gamma' & \xrightarrow{f} & ft(\Delta) \end{array}$$

commutes. This amounts to proving that for each $i = 0, \dots, n-1$ one has

$$(\partial^n_n \widehat{\circ} qq(f))(i) = (ff \widehat{\circ} \partial^m_m)(i)$$

We have

$$(\partial^n_n \widehat{\circ} qq(f))(i) = (\phi_{\mathbf{R}}(\partial^n_n) \widehat{\circ} qq(f))(i) = (\partial^n_n \circ qq(f))(i) = qq(f)(i) = \partial^m_m(ff(i))$$

where the last equality holds since $i \leq n-1$. On the other hand

$$(ff \widehat{\circ} \partial^m_m)(i) = (ff \widehat{\circ} \phi_{\mathbf{R}}(\partial^m_m))(i) = (ff \circ bind(\phi_{\mathbf{R}}(\partial^m_m)))(i) = bind(\phi_{\mathbf{R}}(\partial^m_m))(ff(i)) = \partial^m_m(ff(i))$$

where the last equality is by our convention on notations.

Proposition 3.7 *[**2015.07.24.prop1**] The data specified above defines a C0-system.*

Proof: The first four conditions of [26, Definition 2.1] are obvious. The fifth condition states that the "canonical squares"

$$\begin{array}{ccc} f^*(\Delta) & \xrightarrow{q(f,\Delta)} & \Delta \\ \downarrow{\scriptstyle p_{f^*(\Delta)}} & & \downarrow{\scriptstyle p_\Delta} \\ \Gamma' & \xrightarrow{f} & \Gamma \end{array}$$

commute. Let $\Gamma = (T_1, \dots, T_n)$ and $\Gamma' = (T'_1, \dots, T'_m)$. We have:

$$q(f, \Delta) \circ p_\Delta = (f_1, \dots, f_n)$$

by (**??**) and Lemma 3.3(1) and

$$p_{f^*(\Delta)} \circ f = p_{f^*(\Delta)} \circ (f_1, \dots, f_n) = (f_1, \dots, f_n)$$

by Lemma **??**(2).

The sixth condition asserts that $Id^*(\Delta) = \Delta$ and $q(Id, \Delta) = Id$. This follows immediately from our definition of f^* and $q(f, \Delta)$ since for Γ such that $l(\Gamma) = n$ we have $Id_\Gamma = (1, \dots, n)$.

The seventh and last condition asserts that for $\Gamma'' = (T''_1, \dots, T''_k)$, $\Gamma' = (T'_1, \dots, T'_m)$, $\Delta = (T_1, \dots, T_{n+1})$, $g = (g_1, \dots, g_m) : \Gamma'' \to \Gamma'$ and $f = (f_1, \dots, f_n) : \Gamma' \to \Gamma$ one has

$$g^*(f^*(\Delta)) = (g \circ f)^*(\Delta)$$

and

$$q(g \circ f, \Delta) = q(g, f^*(\Delta)) \circ q(f, \Delta)$$

For the first equality we have:

$$g^*(f^*(\Delta)) = g^*(T'_1, \dots, T'_m, T_{n+1}(f_1/1, \dots, f_n/n)) = (T''_1, \dots, T''_k, T_{n+1}(f_1/1, \dots, f_n/n)(g_1/1, \dots, g_m/m))$$

and

$$(g \circ f)^*(\Delta) = (T''_1, \dots, T''_k, T_{n+1}((g \circ f)_1/1, \dots, (g \circ f)_n/n))$$

The canonical pull-back square defined by an object $(T_1, \dots, T_{n+1})$ and a morphism

$$(f_1, \dots, f_n) : (R_1, \dots, R_m) \to (T_1, \dots, T_n)$$

is of the form:

$$\begin{array}{ccc} (R_1, \dots, R_m, T_{n+1}(f_1/1, \dots, f_n/n)) & \xrightarrow{(f_1, \dots, f_n, m+1)} & (T_1, \dots, T_{n+1}) \\ \downarrow{\scriptstyle (1, \dots, m)} & & \downarrow{\scriptstyle (1, \dots, n)} \\ (R_1, \dots, R_m) & \xrightarrow{(f_1, \dots, f_n)} & (T_1, \dots, T_n) \end{array} \qquad (2)$$

Proposition 3.8 *[**2009.10.01.prop2**] With the structure defined above $CC(\mathbf{R}, \mathbf{LM})$ is a C-system.*

Proof: Straightforward.

Remark 3.9 [**2014.09.28.rm1**] One can easily construct on the function $(\mathbf{R}, \mathbf{LM}) \mapsto CC(\mathbf{R}, \mathbf{LM})$ the structure of a functor from the "large module category" of [8] to the category of C-systems and their homomorphisms.

Remark 3.10 There is another construction of a pre-category from $(\mathbf{R}, \mathbf{LM})$ which takes as an additional parameter a set Var which is called the set of variables. Let $F_n(Var)$ be the set of sequences of length n of pair-wise distinct elements of Var. Define the pre-category $CC(\mathbf{R}, \mathbf{LM}, Var)$ as follows. The set of objects of $CC(\mathbf{R}, \mathbf{LM}, Var)$ is

$$Ob(CC(\mathbf{R}, \mathbf{LM}, Var)) = \amalg_{n \geq 0} \amalg_{(x_1, \ldots, x_n) \in F_n(Var)} LM(stn(0)) \times \ldots \times LM(\{x_1, \ldots, x_{n-1}\})$$

For compatibility with the traditional type theory we will write the elements of $Ob(CC(\mathbf{R}, \mathbf{LM}, X))$ as sequences of the form $x_1 : E_1, \ldots, x_n : E_n$. The set of morphisms is given by

$$Hom_{CC(\mathbf{R}, \mathbf{LM}, Var)}((x_1 : E_1, \ldots, x_m : E_m), (y_1 : T_1, \ldots, y_n : T_n)) = R(\{x_1, \ldots, x_m\})^n$$

The composition is defined in such a way that the projection

$$(x_1 : E_1, \ldots, x_n : E_n) \mapsto (E_1, E_2(1/x_1), \ldots, E_n(1/x_1, \ldots, n-1/x_{n-1}))$$

is a functor from $CC(\mathbf{R}, \mathbf{LM}, Var)$ to $CC(\mathbf{R}, \mathbf{LM})$.

This functor is clearly an equivalence of categories but not an isomorphism of pre-categories.

There are an obvious final object and the map ft on $CC(\mathbf{R}, \mathbf{LM}, Var)$.

There is however a real problem in making it into a C-system which is due to the following. Consider an object $(y_1 : T_1, \ldots, y_{n+1} : T_{n+1})$ and a morphism $(f_1, \ldots, f_n) : (x_1 : R_1, \ldots, x_m : R_m) \to (y_1 : T_1, \ldots, y_n : T_n)$. In order for the functor to $CC(\mathbf{R}, \mathbf{LM})$ to be a C-system morphism the canonical square build on this pair should have the form

$$\begin{array}{ccc} (x_1 : R_1, \ldots, x_m : R_m, x_{m+1} : T_{n+1}(f_1/1, \ldots, f_n/n)) & \xrightarrow{(f_1, \ldots, f_n, x_{n+1})} & (y_1 : T_1, \ldots, y_{n+1} : T_{n+1}) \\ \downarrow & & \downarrow \\ (x_1 : R_1, \ldots, x_m : R_m) & \xrightarrow{(f_1, \ldots, f_n)} & (y_1 : T_1, \ldots, y_n : T_n) \end{array}$$

where x_{m+1} is an element of Var which is distinct from each of the elements $x_1, \ldots, x_m$. Moreover, we should choose x_{m+1} in such a way the the resulting construction satisfies the C-system axioms for $(f_1, \ldots, f_n) = Id$ and for the compositions $(g_1, \ldots, g_m) \circ (f_1, \ldots, f_n)$. One can easily see that no such choice is possible for a finite set Var. At the moment it is not clear to me whether or not it is possible for an infinite Var.

Recall from [26] that for a C-system CC one defines $\widetilde{Ob}(CC)$ as the subset of $Mor(CC)$ which consists of morphisms s of the form $ft(X) \to X$ such that $l(X) > 0$ and $s \circ p_X = Id_{ft(X)}$.

Lemma 3.11 *[2014.06.30.l2] One has:*

$$\widetilde{Ob}(CC(\mathbf{R}, \mathbf{LM})) \cong \coprod_{n \geq 0} LM(stn(0)) \times \ldots \times LM(stn(n)) \times R(stn(n))$$

Proof: An element of $\widetilde{Ob}(CC(\mathbf{R}, \mathbf{LM}))$ is a section s of the canonical morphism $p_\Gamma : \Gamma \to ft(\Gamma)$. It follows immediately from the definition of $CC(\mathbf{R}, \mathbf{LM})$ that for $\Gamma = (E_1, \ldots, E_{n+1})$, a morphism $(f_1, \ldots, f_{n+1}) \in R(stn(n))^{n+1}$ from $ft(\Gamma)$ to Γ is a section of p_Γ if an only if $f_i = i$ for $i =$

$1, \dots, n$. Therefore, any such section is determined by its last component f_{n+1} and mapping $((E_1, \dots, E_n), (E_1, \dots, E_{n+1}), (f_1, \dots, f_{n+1}))$ to $(E_1, \dots, E_n, E_{n+1}, f_{n+1})$ we get a bijection

$$[\mathbf{2009.10.15.eq2}]\widetilde{Ob}(CC(\mathbf{R}, \mathbf{LM})) \cong \coprod_{n \geq 0} LM(stn(0)) \times \dots \times LM(stn(n)) \times R(stn(n)) \qquad (3)$$

Using the notations of type theory we can write elements of $Ob(CC(\mathbf{R}, \mathbf{LM}))$ as

$$\Gamma = (T_1, \dots, T_n \triangleright)$$

where $T_i \in LM(\widehat{i-1})$ and the elements of $\widetilde{Ob}(CC(\mathbf{R}, \mathbf{LM}))$ as

$$\mathcal{J} = (T_1, \dots, T_n \vdash t : T)$$

where $T_i \in LM(\widehat{i-1})$, $T \in LM(stn(n))$ and $t \in R(stn(n))$.

In this notation the operations $T, \widetilde{T}, S, \widetilde{S}$ and δ which were introduced in [26] take the form:

1. $T((\Gamma, T_{n+1}\triangleright), (\Gamma, \Delta\triangleright)) = (\Gamma, T_{n+1}, t_{n+1}(\Delta)\triangleright)$ when $l(\Gamma) = n$,
2. $\widetilde{T}((\Gamma, T_{n+1}\triangleright), (\Gamma, \Delta \vdash r : R)) = (\Gamma, T_{n+1}, t_{n+1}(\Delta) \vdash t_{n+1}(r : R))$ when $l(\Gamma) = n$,
3. $S((\Gamma \vdash s : S), (\Gamma, S, \Delta\triangleright)) = (\Gamma, s_{n+1}(\Delta[s/n+1])\triangleright)$ when $l(\Gamma) = n$,
4. $\widetilde{S}((\Gamma \vdash s : S), (\Gamma, S, \Delta \vdash r : R)) = (\Gamma, s_{n+1}(\Delta[s/n+1]) \vdash s_{n+1}((r : R)[s/n+1])$ when $l(\Gamma) = n$,
5. $\delta(\Gamma, T\triangleright) = (\Gamma, T \vdash (n+1) : T)$ when $l(\Gamma) = n$.

4 C-subsystems of $CC(\mathbf{R}, \mathbf{LM})$.

Let CC be a C-subsystem of $CC(\mathbf{R}, \mathbf{LM})$. By [26] CC is determined by the subsets $C = Ob(CC)$ and $\widetilde{C} = \widetilde{Ob}(CC)$ in $Ob(CC(\mathbf{R}, \mathbf{LM}))$ and $\widetilde{Ob}(CC(\mathbf{R}, \mathbf{LM}))$.

For $\Gamma = (E_1, \dots, E_n)$ we write $(\Gamma\triangleright_C)$ if $(E_1, \dots, E_n)$ is in C and $(\Gamma \vdash_{\widetilde{C}} t : T)$ if $(E_1, \dots, E_n, T, t)$ is in $\widetilde{C}$.

The following result is an immediate corollary of [26, Proposition 4.3] together with the description of the operations $T, \widetilde{T}, S, \widetilde{S}$ and δ for $CC(\mathbf{R}, \mathbf{LM})$ which is given above.

Proposition 4.1 *[**2009.10.16.prop3**] Let $(\mathbf{R}, \mathbf{LM})$ be a monad on Sets and a left module over it with values in Sets. A pair of subsets*

$$C \subset \coprod_{n \geq 0} \prod_{i=0}^{n-1} LM(stn(i))$$

$$\widetilde{C} \subset \coprod_{n \geq 0} (\prod_{i=0}^{n} LM(stn(i))) \times R(stn(n))$$

corresponds to a C-subsystem CC of $CC(\mathbf{R}, \mathbf{LM})$ if and only if the following conditions hold:

1. $(\rhd_C)$
2. $(\Gamma, T\rhd_C) \Rightarrow (\Gamma\rhd_C)$
3. $(\Gamma \vdash_{\widetilde{C}} r : R) \Rightarrow (\Gamma, R\rhd_C)$
4. $(\Gamma, T\rhd_C) \wedge (\Gamma, \Delta, \vdash_{\widetilde{C}} r : R) \Rightarrow (\Gamma, T, t_{n+1}(\Delta) \vdash_{\widetilde{C}} t_{n+1}(r : R))$ *where* $n = l(\Gamma_1)$
5. $(\Gamma \vdash_{\widetilde{C}} s : S) \wedge (\Gamma, S, \Delta \vdash_{\widetilde{C}} r : R) \Rightarrow (\Gamma, s_{n+1}(\Delta[s/n+1]) \vdash_{\widetilde{C}} s_{n+1}((r : R)[s/n+1]))$ *where* $n = l(\Gamma_1)$,
6. $(\Gamma, T\rhd_C) \Rightarrow (\Gamma, T \vdash_{\widetilde{C}} n+1 : T)$ *where* $n = l(\Gamma)$.

Note that conditions (4) and (5) together with condition (6) and condition (3) imply the following

4a $(\Gamma, T\rhd_C) \wedge (\Gamma, \Delta\rhd_C) \Rightarrow (\Gamma, T, t_{n+1}(\Delta)\rhd_C)$ where $n = l(\Gamma_1)$,

5a $(\Gamma \vdash_{\widetilde{C}} s : S) \wedge (\Gamma, S, \Delta\rhd_C) \Rightarrow (\Gamma, s_{n+1}(\Delta[s/n+1])\rhd_C)$ where $n = l(\Gamma_1)$.

Note also that modulo condition (2), condition (1) is equivalent to the condition that $C \neq \emptyset$.

Remark 4.2 [**2010.08.07.rem1**] If one re-writes the conditions of Proposition 4.1 in the more familiar in type theory form where the variables introduced in the context are named rather than directly numbered one arrives at the following rules:

$$\frac{}{\rhd_C} \qquad \frac{x_1 : T_1, \ldots, x_n : T_n\rhd_C}{x_1 : T_1, \ldots, x_{n-1} : T_{n-1}\rhd_C} \qquad \frac{x_1 : T_1, \ldots, x_n : T_n \vdash_{\widetilde{C}} t : T}{x_1 : T_1, \ldots, x_n : T_n, y : T\rhd_C}$$

$$\frac{x_1 : T_1, \ldots, x_n : T_n, y : T \rhd_C \quad x_1 : T_1, \ldots, x_n : T_n, \ldots, x_m : T_m \vdash_{\widetilde{C}} r : R}{x_1 : T_1, \ldots, x_n : T_n, y : T, x_{n+1} : T_{n+1}, \ldots, x_m : T_m \vdash_{\widetilde{C}} r : R}$$

$$\frac{x_1 : T_1, \ldots, x_n : T_n \vdash_{\widetilde{C}} s : S \quad x_1 : T_1, \ldots, x_n : T_n, y : S, x_{n+1} : T_{n+1}, \ldots, x_m : T_m \vdash_{\widetilde{C}} r : R}{x_1 : T_1, \ldots, x_n : T_n, x_{n+1} : T_{n+1}[s/y], \ldots, x_m : T_m[s/y] \vdash_{\widetilde{C}} (r : R)[s/y]}$$

$$\frac{x_1 : E_1, \ldots, x_n : E_n\rhd_C}{x_1 : E_1, \ldots, x_n : E_n \vdash_{\widetilde{C}} x_n : E_n}$$

which are similar (and probably equivalent) to the "basic rules of DTT" given in [10, p.585]. The advantage of the rules given here is that they are precisely the ones which are necessary and sufficient for a given collection of contexts and judgements to define a C-system.

Lemma 4.3 *[**2009.11.05.l1**] Let CC be as above and let $(E_1, \ldots, E_m), (T_1, \ldots, T_n) \in Ob(CC)$ and $(f_1, \ldots, f_n) \in R(\widehat{m})^n$. Then*

$$(f_1, \ldots, f_n) \in Hom_{CC}((E_1, \ldots, E_m), (T_1, \ldots, T_n))$$

if and only if $(f_1, \ldots, f_{n-1}) \in Hom_{CC}((E_1, \ldots, E_m), (T_1, \ldots, T_{n-1}))$ and

$$E_1, \ldots, E_m \vdash_{\widetilde{C}} f_n : T_n(f_1/1, \ldots, f_{n-1}/n-1)$$

Proof: Straightforward using the fact that the canonical pull-back squares in $CC(\mathbf{R}, \mathbf{LM})$ are given by (2).

Example 4.4 The category $CC(R, R)$ for the identity monad is empty. For the monad of the form $R(X) = pt$ the C-system $CC(R, R)$ has only two subsystems - itself and the trivial one for which $C = pt$.

The first non-trivial example is the monad $R(X) = X \amalg \{*\}$. We conjecture that in this case the set of all C-subsystems of $CC(R, R)$ is *uncountable*.

One can probably show this as follows. Let $\epsilon : \mathbf{N} \to \{0, 1\}$, be a sequence of 0's and 1's. Consider the C-subsystem of CC_ϵ of $CC(R, R)$ which is generated by the set of elements of the form $(*, 1, 2, \dots, n\triangleright) \in Ob(CC(R, R))$ for all $n \geq 0$ and elements $(*, 1, \dots, n+1 \vdash n+2 : *) \in \widetilde{Ob}(CC(R, R))$ for n such that $\epsilon(n) = 1$.

It should be possible to show that $CC_\epsilon \neq CC_{\epsilon'}$ for $\epsilon \neq \epsilon'$ which would imply the conjecture.

5 Operations σ and $\widetilde{\sigma}$ on $CC(\mathbf{R}, \mathbf{LM})$.

C-systems of the form $CC(\mathbf{R}, \mathbf{LM})$ have an important additional structure which will play a role in the next section. This structure is given by two operations:

1. for $\Gamma = (T_1, \dots, T_n, \dots, T_{n+i})$ and $\Gamma' = (T'_1, \dots, T'_n)$ we set

$$\sigma(\Gamma, \Gamma') = (T'_1, \dots, T'_n, T_{n+1}, \dots, T_{n+i})$$

   This gives us an operation with values in Ob defined on the subset of $Ob \times Ob$ which consists of pairs (Γ, Γ') such that $l(\Gamma) > l(\Gamma')$,

2. for $\mathcal{J} = (T_1, \dots, T_{n-1}, \dots, T_{n-1+i} \vdash t : T_{n+i})$, $\Gamma' = (T'_1, \dots, T'_n)$ we set

$$\widetilde{\sigma}(\mathcal{J}, \Gamma') = \begin{cases} (T'_1, \dots, T'_n, T_{n+1}, \dots, T_{n+i-1} \vdash t : T_{n+i}) & \text{for } i > 0 \\ (T'_1, \dots, T'_{n-1} \vdash t : T'_n) & \text{for } i = 0 \end{cases}$$

   This gives us an operation with values in $\widetilde{Ob}$ defined on the subset of $\widetilde{Ob} \times Ob$ which consists of pairs $(\mathcal{J}, \Gamma')$ such that $l(\partial(\mathcal{J})) \leq l(\Gamma')$.

6 Regular sub-quotients of $CC(\mathbf{R}, \mathbf{LM})$.

Let $(\mathbf{R}, \mathbf{LM})$ be as above and

$$Ceq \subset \coprod_{n \geq 0} (\prod_{i=0}^{n-1} LM(stn(i))) \times LM(stn(n))^2$$

$$\widetilde{Ceq} \subset \coprod_{n \geq 0} (\prod_{i=0}^{n} LM(stn(i))) \times R(stn(n))^2$$

be two subsets.

For $\Gamma = (T_1, \ldots, T_n) \in ob(CC(\mathbf{R}, \mathbf{LM}))$ and $S_1, S_2 \in LM(stn(n))$ we write $(\Gamma \vdash_{Ceq} S_1 = S_2)$ to signify that $(T_1, \ldots, T_n, S_1, S_2) \in Ceq$. Similarly for $T \in LM(stn(n))$ and $o, o' \in R(stn(n))$ we write $(\Gamma \vdash_{\widetilde{Ceq}} o = o' : S)$ to signify that $(T_1, \ldots, T_n, S, o, o') \in \widetilde{Ceq}$. When no confusion is possible we will omit the subscripts Ceq and $\widetilde{Ceq}$ at $\vdash$.

Similarly we will write $\rhd$ instead of $\rhd_C$ and $\vdash$ instead of $\vdash_{\widetilde{C}}$ if the subsets C and $\widetilde{C}$ are unambiguously determined by the context.

Definition 6.1 *[**simandsimeq**] Given subsets C, $\widetilde{C}$, Ceq, $\widetilde{Ceq}$ as above define relations $\sim$ on C and $\simeq$ on $\widetilde{C}$ as follows:*

1. *for $\Gamma = (T_1, \ldots, T_n)$, $\Gamma' = (T'_1, \ldots, T'_n)$ in C we set $\Gamma \sim \Gamma'$ iff $ft(\Gamma) \sim ft(\Gamma')$ and*
$$T_1, \ldots, T_{n-1} \vdash T_n = T'_n,$$
2. *for $(\Gamma \vdash o : S)$, $(\Gamma' \vdash o' : S')$ in $\widetilde{C}$ we set $(\Gamma \vdash o : S) \simeq (\Gamma' \vdash o' : S')$ iff $(\Gamma, S) \sim (\Gamma', S')$ and*
$$(\Gamma \vdash o = o' : S).$$

Proposition 6.2 *[**2014.07.10.prop1**] Let C, $\widetilde{C}$, Ceq, $\widetilde{Ceq}$ be as above and suppose in addition that one has:*

1. *C and $\widetilde{C}$ satisfy conditions (1)-(6) of Proposition 4.1 which are referred to below as conditions (1.1)-(1.6) of the present proposition,*
2. (a) $(\Gamma \vdash T = T') \Rightarrow (\Gamma, T\rhd)$
   (b) $(\Gamma, T\rhd) \Rightarrow (\Gamma \vdash T = T)$
   (c) $(\Gamma \vdash T = T') \Rightarrow (\Gamma \vdash T' = T)$
   (d) $(\Gamma \vdash T = T') \wedge (\Gamma \vdash T' = T'') \Rightarrow (\Gamma \vdash T = T'')$
3. (a) $(\Gamma \vdash o = o' : T) \Rightarrow (\Gamma \vdash o : T)$
   (b) $(\Gamma \vdash o : T) \Rightarrow (\Gamma \vdash o = o : T)$
   (c) $(\Gamma \vdash o = o' : T) \Rightarrow (\Gamma \vdash o' = o : T)$
   (d) $(\Gamma \vdash o = o' : T) \wedge (\Gamma \vdash o' = o'' : T) \Rightarrow (\Gamma \vdash o = o'' : T)$
4. (a) $(\Gamma_1 \vdash T = T') \wedge (\Gamma_1, T, \Gamma_2 \vdash S = S') \Rightarrow (\Gamma_1, T', \Gamma_2 \vdash S = S')$
   (b) $(\Gamma_1 \vdash T = T') \wedge (\Gamma_1, T, \Gamma_2 \vdash o = o' : S) \Rightarrow (\Gamma_1, T', \Gamma'_2 \vdash o = o' : S)$
   (c) $(\Gamma \vdash S = S') \wedge (\Gamma \vdash o = o' : S) \Rightarrow (\Gamma \vdash o = o' : S')$
5. (a) $(\Gamma_1, T\rhd) \wedge (\Gamma_1, \Gamma_2 \vdash S = S') \Rightarrow (\Gamma_1, T, t_{i+1}\Gamma_2 \vdash t_{i+1}S = t_{i+1}S')$ $\quad i = l(\Gamma)$
   (b) $(\Gamma_1, T\rhd) \wedge (\Gamma_1, \Gamma_2 \vdash o = o' : S) \Rightarrow (\Gamma_1, T, t_{i+1}\Gamma_2 \vdash t_{i+1}o = t_{i+1}o' : t_{i+1}S)$ $\quad i = l(\Gamma)$
6. (a) $(\Gamma_1, T, \Gamma_2 \vdash S = S') \wedge (\Gamma_1 \vdash r : T) \Rightarrow$
   $(\Gamma_1, s_{i+1}(\Gamma_2[r/i+1]) \vdash s_{i+1}(S[r/i+1]) = s_{i+1}(S'[r/i+1]))$ $\quad i = l(\Gamma_1)$
   (b) $(\Gamma_1, T, \Gamma_2 \vdash o = o' : S) \wedge (\Gamma_1 \vdash r : T) \Rightarrow$
   $(\Gamma_1, s_{i+1}(\Gamma_2[r/i+1]) \vdash s_{i+1}(o[r/i+1]) = s_{i+1}(o'[r/i+1]) : s_{i+1}(S[r/i+1]))$ $\quad i = l(\Gamma_1)$

7.

$$
\begin{array}{ll}
(a) & (\Gamma_1, T, \Gamma_2, S\triangleright) \wedge (\Gamma_1 \vdash r = r' : T) \Rightarrow \\
(\Gamma_1, s_{i+1}(\Gamma_2[r/i+1]) \vdash s_{i+1}(S[r/i+1]) = s_{i+1}(S[r'/i+1])) & i = l(\Gamma_1) \\
(b) & (\Gamma_1, T, \Gamma_2 \vdash o : S) \wedge (\Gamma_1 \vdash r = r' : T) \Rightarrow \\
(\Gamma_1, s_{i+1}(\Gamma_2[r/i+1]) \vdash s_{i+1}(o[r/i+1]) = s_{i+1}(o[r'/i+1]) : s_{i+1}(S[r/i+1])) & i = l(\Gamma_1)
\end{array}
$$

Then the relations $\sim$ and $\simeq$ are equivalence relations on C and $\widetilde{C}$ which satisfy the conditions of [26, Proposition 5.4] and therefore they correspond to a regular congruence relation on the C-system defined by $(C, \widetilde{C})$.

Lemma 6.3 *[**iseqrelsiml1**] One has:*

1. *If conditions (1.2), (4a) of the proposition hold then $(\Gamma \vdash S = S') \wedge (\Gamma \sim \Gamma') \Rightarrow (\Gamma' \vdash S = S')$.*
2. *If conditions (1.2), (1.3), (4a), (4b), (4c) hold then $(\Gamma \vdash o = o' : S) \wedge ((\Gamma, S) \sim (\Gamma', S')) \Rightarrow (\Gamma' \vdash o = o' : S')$.*

Proof: By induction on $n = l(\Gamma) = l(\Gamma')$.

(1) For $n = 0$ the assertion is obvious. Therefore by induction we may assume that $(\Gamma \vdash S = S') \wedge (\Gamma \sim \Gamma') \Rightarrow (\Gamma' \vdash S = S')$ for all $i < n$ and all appropriate Γ,Γ', S and S' and that $(T_1, \dots, T_n \vdash S = S') \wedge (T_1, \dots, T_n \sim T'_1, \dots, T'_n)$ holds and we need to show that $(T'_1, \dots, T'_n \vdash S = S')$ holds. Let us show by induction on j that $(T'_1, \dots, T'_j, T_{j+1}, \dots, T_n \vdash S = S')$ for all $j = 0, \dots, n$. For $j = 0$ it is a part of our assumptions. By induction we may assume that $(T'_1, \dots, T'_j, T_{j+1}, \dots, T_n \vdash S = S')$. By definition of $\sim$ we have $(T_1, \dots, T_j \vdash T_{j+1} = T'_{j+1})$. By the inductive assumption we have $(T'_1, \dots, T'_j \vdash T_{j+1} = T'_{j+1})$. Applying (4a) with $\Gamma_1 = (T'_1, \dots T'_j)$, $T = T_{j+1}$, $T' = T'_{j+1}$ and $\Gamma_2 = (T_{j+2}, \dots, T_n)$ we conclude that $(T'_1, \dots, T'_{j+1}, T_{j+2}, \dots, T_n \vdash S = S')$.

(2) By the first part of the lemma we have $\Gamma' \vdash S = S'$. Therefore by (4c) it is sufficient to show that $(\Gamma \vdash o = o' : S) \wedge (\Gamma \sim \Gamma') \Rightarrow (\Gamma' \vdash o = o' : S)$. The proof of this fact is similar to the proof of the first part of the lemma using (4b) instead of (4a).

Lemma 6.4 *[**iseqrelsim**] One has:*

1. *Assume that conditions (1.2), (2b), (2c), (2d) and (4a) hold. Then $\sim$ is an equivalence relation.*
2. *Assume that conditions of the previous part of the lemma as well as conditions (1.3), (3b), (3c), (3d), (4b) and (4c) hold. Then $\simeq$ is an equivalence relation.*

Proof: By induction on $n = l(\Gamma) = l(\Gamma')$.

(1) Reflexivity follows directly from (1.2) and (2b). For $n = 0$ the symmetry is obvious. Let $(\Gamma, T) \sim (\Gamma', T')$. By induction we may assume that $\Gamma' \sim \Gamma$. By Lemma 6.3(a) we have $(\Gamma' \vdash T = T')$ and by (2c) we have $(\Gamma' \vdash T' = T)$. We conclude that $(\Gamma', T') \sim (\Gamma, T)$. The proof of transitivity is by a similar induction.

(2) Reflexivity follows directly from reflexivity of $\sim$, (1.3) and (3b). Symmetry and transitivity are also easy using Lemma 6.3.

From this point on we assume that all conditions of Proposition 6.2 hold. Let $C' = C/\sim$ and $\widetilde{C}' = \widetilde{C}/\simeq$. It follows immediately from our definitions that the functions $ft : C \to C$ and $\partial : \widetilde{C} \to C$ define functions $ft' : C' \to C'$ and $\partial' : \widetilde{C}' \to C'$.

Lemma 6.5 *[**surjl1**] The conditions (3) and (4) of [26, Proposition 5.4] hold for $\sim$ and $\simeq$.*

Proof: 1. We need to show that for $(\Gamma, T\triangleright)$, and $\Gamma \sim \Gamma'$ there exists $(\Gamma', T'\triangleright)$ such that $(\Gamma, T) \sim (\Gamma', T')$. It is sufficient to take $T = T'$. Indeed by (2b) we have $\Gamma \vdash T = T$, by Lemma 6.3(1) we conclude that $\Gamma' \vdash T = T$ and by (1a) that $\Gamma', T\triangleright$.

2. We need to show that for $(\Gamma \vdash o : S)$ and $(\Gamma, S) \sim (\Gamma', S')$ there exists $(\Gamma' \vdash o' : S')$ such that $(\Gamma' \vdash o' : S') \simeq (\Gamma \vdash o : S)$. It is sufficient to take $o' = o$. Indeed, by (3b) we have $(\Gamma \vdash o = o : S)$, by Lemma 6.3(2) we conclude that $(\Gamma' \vdash o = o : S')$ and by (2a) that $(\Gamma' \vdash o : S')$.

Lemma 6.6 *[**TSetc**] The equivalence relations $\sim$ and $\simeq$ are compatible with the operations $T, \widetilde{T}, S, \widetilde{S}$ and δ.*

Proof: (1) Given $(\Gamma_1, T\triangleright) \sim (\Gamma_1', T'\triangleright)$ and $(\Gamma_1, \Gamma_2\triangleright) \sim (\Gamma_1', \Gamma_2'\triangleright)$ we have to show that

$$(\Gamma_1, T, t_{n+1}\Gamma_2) \sim (\Gamma_1', T', t_{n+1}\Gamma_2').$$

where $n = l(\Gamma_1) = l(\Gamma_1')$.

Proceed by induction on $l(\Gamma_2)$. For $l(\Gamma_2) = 0$ the assertion is obvious. Let $(\Gamma_1, T\triangleright) \sim (\Gamma_1', T'\triangleright)$ and $(\Gamma_1, \Gamma_2, S\triangleright) \sim (\Gamma_1', \Gamma_2', S'\triangleright)$. The later condition is equivalent to $(\Gamma_1, \Gamma_2\triangleright) \sim (\Gamma_1', \Gamma_2'\triangleright)$ and $(\Gamma_1, \Gamma_2 \vdash S = S')$. By the inductive assumption we have $(\Gamma_1, T, t_{n+1}\Gamma_2) \sim (\Gamma_1', T', t_{n+1}\Gamma_2')$. By (5a) we conclude that $(\Gamma_1, T, t_{n+1}\Gamma_2 \vdash t_{n+1}S = t_{n+1}S')$. Therefore by definition of $\sim$ we have $(\Gamma_1, T, t_{n+1}\Gamma_2, t_{n+1}S) \sim (\Gamma_1', T', t_{n+1}\Gamma_2', t_{n+1}S')$.

(2) Given $(\Gamma_1, T\triangleright) \sim (\Gamma_1', T'\triangleright)$ and $(\Gamma_1, \Gamma_2 \vdash o : S) \simeq (\Gamma_1', \Gamma_2' \vdash o' : S')$ we have to show that $(\Gamma_1, T, t_{n+1}\Gamma_2 \vdash t_{n+1}o : t_{n+1}S) \simeq (\Gamma_1', T', t_{n+1}\Gamma_2' \vdash t_{n+1}o' : t_{n+1}S')$ where $n = l(\Gamma_1) = l(\Gamma_1')$. We have $(\Gamma_1, \Gamma_2, S) \sim (\Gamma_1', \Gamma_2', S')$ and $(\Gamma_1, \Gamma_2 \vdash o = o' : S)$. By (5b) we get $(\Gamma_1, T, t_{n+1}\Gamma_2 \vdash t_{n+1}o = t_{n+1}o' : t_{n+1}S)$. By (1) of this lemma we get $(\Gamma_1, T, t_{n+1}\Gamma_2, t_{n+1}S) \sim (\Gamma_1', T', t_{n+1}\Gamma_2', t_{n+1}S')$ and therefore by definition of $\simeq$ we get $(\Gamma_1, T, t_{n+1}\Gamma_2 \vdash t_{n+1}o : t_{n+1}S) \simeq (\Gamma_1', T', t_{n+1}\Gamma_2' \vdash t_{n+1}o' : t_{n+1}S')$.

(3) Given $(\Gamma_1 \vdash r : T) \simeq (\Gamma_1' \vdash r' : T')$ and $(\Gamma_1, T, \Gamma_2\triangleright) \sim (\Gamma_1', T', \Gamma_2'\triangleright)$ we have to show that

$$(\Gamma_1, s_{n+1}(\Gamma_2[r/n+1])) \sim (\Gamma_1', s_{n+1}(\Gamma_2'[r'/n+1])).$$

where $n = l(\Gamma_1) = l(\Gamma_1')$. Proceed by induction on $l(\Gamma_2)$. For $l(\Gamma_2) = 0$ the assertion follows directly from the definitions. Let $(\Gamma_1 \vdash r : T) \simeq (\Gamma_1' \vdash r' : T')$ and $(\Gamma_1, T, \Gamma_2, S\triangleright) \sim (\Gamma_1', T', \Gamma_2', S'\triangleright)$. The later condition is equivalent to $(\Gamma_1, T, \Gamma_2\triangleright) \sim (\Gamma_1', T', \Gamma_2'\triangleright)$ and $(\Gamma_1, T, \Gamma_2 \vdash S = S')$. By the inductive assumption we have $(\Gamma_1, s_{n+1}(\Gamma_2[r/n+1])) \sim (\Gamma_1', s_{n+1}(\Gamma_2'[r'/n+1]))$. It remains to show that $(\Gamma_1, s_{n+1}(\Gamma_2[r/n+1]) \vdash s_{n+1}(S[r/n+1]) = s_{n+1}(S'[r'/n+1]))$. By (2d) it is sufficient to show that $(\Gamma_1, s_{n+1}(\Gamma_2[r/n+1]) \vdash s_{n+1}(S[r/n+1]) = s_{n+1}(S'[r/n+1]))$ and $(\Gamma_1, s_{n+1}(\Gamma_2[r/n+1]) \vdash s_{n+1}(S'[r/n+1]) = s_{n+1}(S'[r'/n+1]))$. The first relation follows directly from (6a). To prove the second one it is sufficient by (7a) to show that $(\Gamma_1, T, \Gamma_2, S'\triangleright)$ which follows from our assumption through (2c) and (2a).

(4) Given $(\Gamma_1 \vdash r : T) \simeq (\Gamma_1' \vdash r' : T')$ and $(\Gamma_1, T, \Gamma_2 \vdash o : S) \simeq (\Gamma_1', T', \Gamma_2' \vdash o' : S')$ we have to show that

$$(\Gamma_1, s_{n+1}(\Gamma_2[r/n+1]) \vdash s_{n+1}(o[r/n+1]) : s_{n+1}(S[r/n+1])) \simeq$$

$$(\Gamma'_1, s_{n+1}(\Gamma'_2[r'/n+1]) \vdash s_{n+1}(o'[r'/n+1]) : s_{n+1}(S'[r'/n+1])).$$

where $n = l(\Gamma_1) = l(\Gamma'_1)$ or equivalently that

$$(\Gamma_1, s_{n+1}(\Gamma_2[r/n+1]), s_{n+1}(S[r/n+1])) \sim (\Gamma'_1, s_{n+1}(\Gamma'_2[r'/n+1]), s_{n+1}(S'[r'/n+1]))$$

and $(\Gamma_1, s_{n+1}(\Gamma_2[r/n+1]) \vdash s_{n+1}(o[r/n+1]) = s_{n+1}(o'[r'/n+1]) : s_{n+1}(S[r/n+1]))$. The first statement follows from part (3) of the lemma. To prove the second statement it is sufficient by (3d) to show that $(\Gamma_1, s_{n+1}(\Gamma_2[r/n+1]) \vdash s_{n+1}(o[r/n+1]) = s_{n+1}(o'[r/n+1]) : s_{n+1}(S[r/n+1]))$ and $(\Gamma_1, s_{n+1}(\Gamma_2[r/n+1]) \vdash s_{n+1}(o'[r/n+1]) = s_{n+1}(o'[r'/n+1]) : s_{n+1}(S[r/n+1]))$. The first assertion follows directly from (6b). To prove the second one it is sufficient in view of (7b) to show that $(\Gamma_1, T, \Gamma_2 \vdash o' : S)$ which follows conditions (3c) and (3a).

(5) Given $(\Gamma, T) \sim (\Gamma', T')$ we need to show that $(\Gamma, T \vdash (n+1) : T) \simeq (\Gamma', T' \vdash (n+1) : T')$ or equivalently that $(\Gamma, T, T) \sim (\Gamma, T', T')$ and $(\Gamma, T \vdash (n+1) = (n+1) : T)$. The second part follows from (3b). To prove the first part we need to show that $(\Gamma, T \vdash T = T')$. This follows from our assumption by (5a).

Lemma 6.7 *[2014.07.12.l1] Let C be a subset of $Ob(CC(\mathbf{R}, \mathbf{LM}))$ which is closed under ft. Let $\leq$ be a transitive relation on C such that:*

1. *$\Gamma \leq \Gamma'$ implies $l(\Gamma) = l(\Gamma')$,*
2. *$\Gamma \in C$ and $ft(\Gamma) \leq F$ implies $\sigma(\Gamma, F) \in C$ and $\Gamma \leq \sigma(\Gamma, F)$.*

Then $\Gamma \in C$ and $ft^i(\Gamma) \leq F$ for some $i \geq 1$, implies that $\Gamma \leq \sigma(\Gamma, F)$.

Proof: Simple induction on i.

Lemma 6.8 *[2014.07.12.l2] Let C and $\leq$ be as in Lemma 6.7. Then one has:*

1. *$(\Gamma, T) \leq (\Gamma, T')$ and $\Gamma \leq \Gamma'$ implies that $(\Gamma, T) \leq (\Gamma', T')$,*
2. *if $\leq$ is ft-monotone (i.e. $\Gamma \leq \Gamma'$ implies $ft(\Gamma) \leq ft(\Gamma')$) and symmetric then $(\Gamma, T) \leq (\Gamma', T')$ implies that $(\Gamma, T) \leq (\Gamma, T')$.*

Proof: The first assertion follows from

$$(\Gamma, T) \leq (\Gamma, T') \leq \sigma((\Gamma, T'), \Gamma') = (\Gamma', T')$$

The second assertion follows from

$$(\Gamma, T) \leq (\Gamma', T') \leq \sigma((\Gamma', T'), \Gamma) = (\Gamma, T')$$

where the second $\leq$ requires $\Gamma' \leq \Gamma$ which follows from ft-monotonicity and symmetry.

Lemma 6.9 *[2014.07.12.l3] Let $C, \leq$ be as in Lemma 6.7, let $\widetilde{C}$ be a subset of $\widetilde{Ob}(CC(\mathbf{R}, \mathbf{LM}))$ and $\leq'$ a transitive relation on $\widetilde{C}$ such that:*

1. *$\mathcal{J} \leq' \mathcal{J}'$ implies $\partial(\mathcal{J}) \leq \partial(\mathcal{J}')$,*

2. $\mathcal{J} \in \widetilde{C}$ *and* $\partial(\mathcal{J}) \leq F$ *implies* $\widetilde{\sigma}(\mathcal{J}, F) \in \widetilde{C}$ *and* $\mathcal{J} \leq' \widetilde{\sigma}(\mathcal{J}, F)$.

Then $\mathcal{J} \in \widetilde{C}$ *and* $ft^i(\partial(\mathcal{J})) \leq F$ *for some* $i \geq 0$ *implies* $\mathcal{J} \leq \widetilde{\sigma}(\mathcal{J}, F)$.

Proof: Simple induction on i.

Lemma 6.10 *[**2014.07.12.l4**] Let* $C, \leq$ *and* $\widetilde{C}, \leq'$ *be as in Lemma 6.9. Then one has:*

1. $(\Gamma \vdash o : T) \leq' (\Gamma \vdash o' : T)$ *and* $(\Gamma, T) \leq (\Gamma', T')$ *implies that* $(\Gamma \vdash o : T) \leq' (\Gamma' \vdash o' : T')$,
2. *if* $(\leq, \leq')$ *is* ∂*-monotone (i.e.* $\mathcal{J} \leq' \mathcal{J}'$ *implies* $\partial(\mathcal{J}) \leq \partial(\mathcal{J}')$*) and* $\leq$ *is symmetric then* $(\Gamma \vdash o : T) \leq' (\Gamma' \vdash o' : T')$ *implies that* $(\Gamma \vdash o : T) \leq' (\Gamma \vdash o' : T)$.

Proof: The first assertion follows from

$$(\Gamma \vdash o : T) \leq' (\Gamma \vdash o' : T) \leq' \widetilde{\sigma}((\Gamma \vdash o' : T), (\Gamma', T')) = (\Gamma' \vdash o' : T')$$

The second assertion follows from

$$\Gamma \vdash o : T) \leq' (\Gamma' \vdash o' : T') \leq' \sigma((\Gamma' \vdash o' : T'), (\Gamma, T)) = (\Gamma \vdash o' : T)$$

where the second $\leq$ requires $\Gamma' \leq \Gamma$ which follows from ∂-monotonicity of $\leq'$ and symmetry of $\leq$.

Proposition 6.11 *[**2014.07.10.prop2**] Let* $(C, \widetilde{C})$ *be subsets in* $Ob(CC(\mathbf{R}, \mathbf{LM}))$ *and* $\widetilde{Ob}(CC(\mathbf{R}, \mathbf{LM}))$ *respectively which correspond to a C-subsystem* CC *of* $CC(\mathbf{R}, \mathbf{LM})$. *Then the constructions presented above establish a bijection between pairs of subsets* $(Ceq, \widetilde{Ceq})$ *which together with* $(C, \widetilde{C})$ *satisfy the conditions of Proposition 6.2 and pairs of equivalence relations* $(\sim, \simeq)$ *on* $(C, \widetilde{C})$ *such that:*

1. $(\sim, \simeq)$ *corresponds to a regular congruence relation on* CC *(i.e., satisfies the conditions of [26, Proposition 5.4]),*
2. $\Gamma \in C$ *and* $ft(\Gamma) \sim F$ *implies* $\Gamma \sim \sigma(\Gamma, F)$,
3. $\mathcal{J} \in \widetilde{C}$ *and* $\partial(\mathcal{J}) \sim F$ *implies* $\mathcal{J} \simeq \widetilde{\sigma}(\mathcal{J}, F)$.

Proof: One constructs a pair $(\sim, \simeq)$ from $(Ceq, \widetilde{Ceq})$ as in Definition 6.1. This pair corresponds to a regular congruence relation by Proposition 6.2. Conditions (2),(3) follow from Lemma 6.3.

Let $(\sim, \simeq)$ be equivalence relations satisfying the conditions of the proposition. Define Ceq as the set of sequences (Γ, T, T') such that $(\Gamma, T), (\Gamma, T') \in C$ and $(\Gamma, T) \sim (\Gamma, T')$. Define $\widetilde{Ceq}$ as the set of sequences (Γ, T, o, o') such that $(\Gamma, T, o), (\Gamma, T, o') \in \widetilde{C}$ and $(\Gamma, T, o) \simeq (\Gamma, T, o')$.

Let us show that these subsets satisfy the conditions of Proposition 6.2. Conditions (2.a-2.d) and (3.a-3d) are obvious.

Condition (4a) follows from (2) by Lemma 6.7. Conditions (4b) and (4c) follow from (3) by Lemma 6.9.

Conditions (5a) and (5b) follow from the compatibility of $(\sim, \simeq)$ with T and $\widetilde{T}$.

Conditions (6a),(6b),(7a),(7b) follow from the compatibility of $(\sim, \simeq)$ with S and $\widetilde{S}$.

References

[1] Benedikt Ahrens, Krzysztof Kapulkin, and Michael Shulman. Univalent categories and the Rezk completion. *Math. Structures Comput. Sci.*, 25(5):1010–1039, 2015.

[2] John Cartmell. Generalised algebraic theories and contextual categories. *Ph.D. Thesis, Oxford University*, 1978. `https://uf-ias-2012.wikispaces.com/Semantics+of+type+theory`.

[3] John Cartmell. Generalised algebraic theories and contextual categories. *Ann. Pure Appl. Logic*, 32(3):209–243, 1986.

[4] Marcelo Fiore, Gordon Plotkin, and Daniele Turi. Abstract syntax and variable binding (extended abstract). In *14th Symposium on Logic in Computer Science (Trento, 1999)*, pages 193–202. IEEE Computer Soc., Los Alamitos, CA, 1999.

[5] P. Gabriel and M. Zisman. *Calculus of fractions and homotopy theory*. Ergebnisse der Mathematik und ihrer Grenzgebiete, Band 35. Springer-Verlag New York, Inc., New York, 1967.

[6] Roger Godement. *Topologie algébrique et théorie des faisceaux*. Actualit'es Sci. Ind. No. 1252. Publ. Math. Univ. Strasbourg. No. 13. Hermann, Paris, 1958.

[7] André Hirschowitz and Marco Maggesi. Modules over monads and linearity. In *Logic, language, information and computation*, volume 4576 of *Lecture Notes in Comput. Sci.*, pages 218–237. Springer, Berlin, 2007.

[8] André Hirschowitz and Marco Maggesi. Higher order theories. *http://arxiv.org/abs/0704.2900*, 2010.

[9] André Hirschowitz and Marco Maggesi. Modules over monads and initial semantics. *Inform. and Comput.*, 208(5):545–564, 2010.

[10] Bart Jacobs. *Categorical logic and type theory*, volume 141 of *Studies in Logic and the Foundations of Mathematics*. North-Holland Publishing Co., Amsterdam, 1999.

[11] Chris Kapulkin, Peter LeFanu Lumsdaine, and Vladimir Voevodsky. The simplicial model of univalent foundations. *Available at http://arxiv.org/abs/1211.2851*, 2012, 2014.

[12] H. Kleisli. Every standard construction is induced by a pair of adjoint functors. *Proc. Amer. Math. Soc.*, 16:544–546, 1965.

[13] F. William Lawvere. Functorial semantics of algebraic theories and some algebraic problems in the context of functorial semantics of algebraic theories. *Repr. Theory Appl. Categ.*, (5):1–121, 2004. Reprinted from Proc. Nat. Acad. Sci. U.S.A. **50** (1963), 869–872 [MR0158921] and ıt Reports of the Midwest Category Seminar. II, 41–61, Springer, Berlin, 1968 [MR0231882].

[14] S. MacLane. *Categories for the working mathematician*, volume 5 of *Graduate texts in Mathematics*. Springer-Verlag, 1971.

[15] Ernest G. Manes. *Algebraic theories*. Springer-Verlag, New York-Heidelberg, 1976. Graduate Texts in Mathematics, No. 26.

[16] Per Martin-Löf. Constructive mathematics and computer programming. In *Logic, methodology and philosophy of science, VI (Hannover, 1979)*, volume 104 of *Stud. Logic Found. Math.*, pages 153–175. North-Holland, Amsterdam, 1982.

[17] Per Martin-Löf. *Intuitionistic type theory*, volume 1 of *Studies in Proof Theory. Lecture Notes*. Bibliopolis, Naples, 1984. Notes by Giovanni Sambin.

[18] Eugenio Moggi. Notions of computation and monads. *Inform. and Comput.*, 93(1):55–92, 1991. Selections from the 1989 IEEE Symposium on Logic in Computer Science.

[19] Thomas Streicher. *Semantics of type theory*. Progress in Theoretical Computer Science. Birkhäuser Boston Inc., Boston, MA, 1991. Correctness, completeness and independence results, With a foreword by Martin Wirsing.

[20] Vladimir Voevodsky. Notes on type systems. *https://github.com/vladimirias/old_notes_on_type_systems*, 2009–2012.

[21] Vladimir Voevodsky. The equivalence axiom and univalent models of type theory. *arXiv 1402.5556*, pages 1–11, 2010.

[22] Vladimir Voevodsky. A C-system defined by a universe category. *arXiv 1409.7925, submitted*, pages 1–33, 2015.

[23] Vladimir Voevodsky. An experimental library of formalized mathematics based on the univalent foundations. *Math. Structures Comput. Sci.*, 25(5):1278–1294, 2015.

[24] Vladimir Voevodsky. Martin-lof identity types in the c-systems defined by a universe category. *arXiv 1505.06446, submitted*, pages 1–51, 2015.

[25] Vladimir Voevodsky. Products of families of types in the C-systems defined by a universe category. *arXiv 1503.07072, submitted*, pages 1–30, 2015.

[26] Vladimir Voevodsky. Subsystems and regular quotients of C-systems. In *Conference on Mathematics and its Applications, (Kuwait City, 2014)*, number to appear, pages 1–11, 2015.

[27] Vladimir Voevodsky, Benedikt Ahrens, Daniel Grayson, et al. *UniMath*: Univalent Mathematics. Available at `https://github.com/UniMath`.