C-system of a finitary monad[1]

Vladimir Voevodsky[2,3]

July 2014

Abstract

We construct for any finitary monad M on sets a C-system ("contextual category") $CC(M)$ and describe, using the results of [5] a class of sub-quotients of $CC(M)$ in terms of objects directly constructed from M. In the special case of the monads of expressions associated with dependent signatures these constructions lead to the C-systems of dependent type systems.

1 Introduction

After reminding the definition of a finitary monad on $Sets$ we construct for any such monad M a C-system ("contextual category") $CC(M)$ whose underlying category is equivalent to the opposite category to the category of finite free M-algebras.

We describe, using the results of [5], all the C-subsystems of $CC(M)$ in terms of objects directly associated with M.

We then define two additional operations σ and $\widetilde{\sigma}$ on $CC(M)$ and described in terms of M the regular congruence relations (see [5]) on C-subsystems of $CC(M)$ which are compatible in a certain sense with σ and $\widetilde{\sigma}$.

In the case when M is the finitary monad of expressions under a dependent signature which is described in [4] the results of this paper immediately imply a rigorous construction of a C-system for any system of contexts, typing judgements and definitional equality judgements based on a dependent signature and satisfying the conditions of Remark 4.2 and Propositions 6.2.

This is one of short papers based on the material of [3]. 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 Finitary monads on sets

A finitary monad (on sets) is a monad $M : Sets \to Sets$ that, as a functor, commutes with filtering colimits. Since any set is, canonically, the colimit of the filtering diagram of its finite subsets, a functor $Sets \to Sets$ that commutes with filtering colimits can be equivalently described as a functor $FSets \to Sets$ where $FSets$ is the category of finite sets.

The monad structure consists of two families of maps:

1. for any $X \in Sets$, a function $X \to M(X)$,
2. for any $X \in Sets$, a function $M(M(X)) \to M(X)$

[1] *2000 Mathematical Subject Classification: 03B15, 03B22, 03F50, 03G25*
[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.

which satisfy certain conditions. Given a monad structure on a functor M which commutes with filtering colimits, finite sets X, X' and a function $X' \to M(X)$, the composition $M(X') \to M(M(X)) \to M(X)$ is a function $M(X') \to M(X)$. This allows one to describe finitary monads directly in terms of functors $FSets \to Sets$ as follows:

Lemma 2.1 *The construction outline above defines an equivalence between (the type of) finitary monads on Sets and (the type of) collections of data of the form:*

1. *for every finite set X a set $M(X)$,*
2. *for every finite set X a function $\iota_X : X \to M(X)$,*
3. *for every finite sets X, X' and a function $f : X \to M(X')$, a function $\mu(f) : M(X) \to M(X')$*

which satisfy the following conditions:

1. *for a finite set X, $\mu(\iota_X) = id_{M(X)}$,*
2. *for a function of finite sets $f : X \to X'$, $\iota_X \mu(f) = f$,*
3. *for two functions $f : X \to M(X')$, $g : X' \to M(X'')$, $\mu(f\mu(g)) = \mu(f)\mu(g)$.*

Proof: Straightforward. Cf. [2].

The description of finitary monads given by Lemma 2.1 is much more convenient than the direct definition for formalization.

For $T \in M(\{x_1, \ldots, x_n\})$ and $f : X \to M(X')$ such that $f(x_i) = f_i$ we write $\mu(f)(T)$ as $T(f_1/x_1, \ldots, f_n/x_n)$.

For a finitary monad M we let $M - cor$ ("M-correspondences") to be the full subcategory of the Kleisli category of M whose objects are finite sets. Recall, that the set of morphisms from X to Y in $M - cor$ is the set of maps from X to $M(Y)$ i.e. $M(Y)^X$ (in other words, $M - cor$ is the category of free, finitely generated M-algebras).

We further let $C(M)$ denote the pre-category[4] with

$$Ob(C(M)) = \mathbf{N}$$

$$Mor(C(M)) = \coprod_{m,n \in \mathbf{N}} M(\{1, \ldots, m\})^n$$

which is equivalent, as a category, to $(M - cor)^{op}$.

For $E \in M(\{1, \ldots, m\})$ and $n \geq 1$ we set:

$$t_n(E) = E[n+1/n, n+2/n+1, \ldots, m+1/m]$$

$$s_n(E) = E[n/n+1, n+1/n+2, \ldots, m-1/m]$$

If we were numbering elements of sets with n-elements from 0 then we would have $t_n = M(\partial_{n-1})$ and $t_n = M(\sigma_{n-1})$ where ∂_i and σ_i are the usual generators of the simplicial category.

[4] See the introduction to [5].

Remark 2.2 The correspondence $M \mapsto C(M)$ defines an equivalence between the type of the finitary monads on *Sets* and the type of the pre-category structures on $\mathbf{N}$ which extend the a pre-category structure of finite sets and where the addition remains to be the coproduct.

Remark 2.3 A finitary sub-monad of M is the same as a sub-pre-category in $C(M)$ which contains all objects. Intersection of two sub-monads is a sub-monad which allows one to speak of sub-monads generated by a set of elements.

3 C-system defined by a monad.

Let $CC(M)$ be the pre-category whose set of objects is $Ob(CC(M)) = \amalg_{n\geq 0} Ob_n$ where

$$Ob_n = M(\emptyset) \times \ldots \times M(\{1, \ldots, n-1\})$$

and the set of morphisms is

$$Mor(CC(M)) = \coprod_{m,n\geq 0} Ob_m \times Ob_n \times M(\{1, \ldots, m\})^n$$

with the obvious domain and codomain maps. The composition of morphisms is defined in the same way as in $C(M)$ such that the mapping $Ob(CC(M)) \to \mathbf{N}$ which sends all elements of Ob_n to n, is a functor from $CC(M)$ to $C(M)$. The associativity of compositions follows immediately from the associativity of compositions in $M - cor$.

Note that if $M(\emptyset) = \emptyset$ then $CC(M) = \emptyset$ and otherwise the functor $CC(M) \to C(M)$ is an equivalence, so that in the second case $C(M)$ and $CC(M)$ are indistinguishable as categories. However, as pre-categories they are quite different.

The pre-category $CC(M)$ is given the structure of a C-system as follows. The final object is the only element of Ob_0, the map ft is defined by the rule

$$ft(T_1, \ldots, T_n) = (T_1, \ldots, T_{n-1}).$$

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

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

is of the form:

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

Proposition 3.1 *With the structure defined above $CC(M)$ is a C-system.*

Proof: Straightforward.

Any morphism of monads $M \to M'$ defines a C-system morphism $CC(M) \to CC(M')$.

Remark 3.2 There is another construction of a pre-category from a finitary monad M 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 $CC(M, Var)$ as follows. The set of objects of $CC(M, Var)$ is

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

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

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

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 $CC(M, Var)$ to $CC(M)$.

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

There is an obvious final object and ft map on $CC(M, Var)$.

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 $CC(M)$ 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.

Remark 3.3 The pre-category $C(M)$ also extends to a C-system which is defined as follows. The final object is 0. The map ft is given by

$$ft(n) = \begin{cases} 0 & \text{if } n = 0 \\ n-1 & \text{if } n > 0 \end{cases}$$

The canonical projection $n \to n-1$ is given by the sequence $(1, \ldots, n-1)$. For $f = (f_1, \ldots, f_m) : n \to m$ the canonical square build on f and the canonical projection $m+1 \to m$ is of the form

$$\begin{array}{ccc} n+1 & \xrightarrow{(f_1,\ldots,f_m,n+1)} & m+1 \\ \downarrow & & \downarrow \\ n & \xrightarrow{(f_1,\ldots,f_m)} & m \end{array}$$

Any morphism of monads $M \to M'$ defines a C-system morphism $C(M) \to C(M')$. Non-trivial C-subsystems of $C(M)$ are in one-to-one correspondence with finitary sub-monads of M.

The natural projection $CC(M) \to C(M)$ is a C-system homomorphism. It's C-system sections are in one-to-one correspondence with $M(\emptyset)$ such that $U \in M(\emptyset)$ corresponds to the section which takes the object n of $C(M)$ to the object $(U, \dots, U)$ of $CC(M)$

Recall from [5] that for a C-system CC one defines $\widetilde{Ob}(CC)$ as the subset of $Mot(CC)$ 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 3.4 *One has:*

$$\widetilde{Ob}(CC(M)) \cong \coprod_{n \geq 0} M(\emptyset) \times \dots \times M(\{1, \dots, n-1\}) \times M(\{1, \dots, n\})^2$$

Proof: An element of $\widetilde{Ob}(CC(M))$ is a section s of the canonical morphism $p_\Gamma : \Gamma \to ft(\Gamma)$. It follows immediately from the definition of $CC(M)$ that for $\Gamma = (E_1, \dots, E_{n+1})$, a morphism $(f_1, \dots, f_{n+1}) \in M(\{1, \dots, n\})^{n+1}$ 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

$$\widetilde{Ob}(CC(M)) \cong \coprod_{n \geq 0} M(\emptyset) \times \dots \times M(\{1, \dots, n-1\}) \times M(\{1, \dots, n\})^2 \tag{2}$$

Using the notations of type theory we can write elements of $Ob(CC(M))$ as

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

where $T_i \in M(\{1, \dots, i-1\})$ and the elements of $\widetilde{Ob}(CC(M))$ as

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

where $T_i \in M(\{1, \dots, i-1\})$ and $t, T \in M(\{1, \dots, n\})$.

In this notation the operations $T, \widetilde{T}, S, \widetilde{S}$ and δ which were introduced in [5] take the form:

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

4 C-subsystems of $CC(M)$.

Let CC be a C-subsystem of $CC(M)$. By [5] CC is determined by the subsets $C = Ob(CC)$ and $\widetilde{C} = \widetilde{Ob}(CC)$ in $Ob(CC(M))$ and $\widetilde{Ob}(CC(M))$.

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

The following result is an immediate corollary of [5, Proposition 4.3] together with the description of the operations $T, \widetilde{T}, S, \widetilde{S}$ and δ for $CC(M)$ which is given above.

Proposition 4.1 *Let M be a finitary monad on Sets. A pair of subsets*

$$C \subset \coprod_{n \geq 0} \prod_{i=0}^{n-1} M(\{1, \ldots, i\})$$

$$\widetilde{C} \subset \coprod_{n \geq 0} (\prod_{i=0}^{n-1} M(\{1, \ldots, i\})) \times M(\{1, \ldots, n\})^2$$

corresponds to a C-subsystem CC of $CC(M)$ 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 $B \neq \emptyset$.

Remark 4.2 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 [1, 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 *Let CC be as above and let $(E_1, \ldots, E_m), (T_1, \ldots, T_n) \in Ob(CC)$ and $(f_1, \ldots, f_n) \in M(\{1, \ldots, m\})^n$. 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 $CC(M)$ are given by (1).

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

The first non-trivial example is the monad $M(X) = X \amalg \{*\}$. We conjecture that in this case the set of all subsystems of $CC(M)$ 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 $CC(M)$ which is generated by the set of elements of the form $(*, 1, 2, \ldots, n \rhd) \in Ob(CC(M))$ for all $n \geq 0$ and elements $(*, 1, \ldots, n+1 \vdash n+2 : *) \in \widetilde{Ob}(CC(M))$ 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 $CC(M)$.

C-systems of the form $CC(M)$ have an important additional structure which will play a role in the next section. This structure is given by two operations:

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

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

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

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

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

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

6 Regular sub-quotients of $CC(M)$.

Let M be a finitary monad and

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

$$\widetilde{Ceq} \subset \coprod_{n\geq 0} (\prod_{i=0}^{n-1} M(\{1, \ldots, i\})) \times M(\{1, \ldots, n\})^3$$

be two subsets.

For $\Gamma = (T_1, \ldots, T_n) \in ob(CC(M))$ and $S_1, S_2 \in M(\{1, \ldots, i\})$ 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 $S, o, o' \in S(\{1, \ldots, n\})$ 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 *Given a finitary monad M and 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 *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.

$$\begin{array}{ll}(a) & (\Gamma\vdash T=T')\Rightarrow(\Gamma,T\rhd)\\ (b) & (\Gamma,T\rhd)\Rightarrow(\Gamma\vdash T=T)\\ (c) & (\Gamma\vdash T=T')\Rightarrow(\Gamma\vdash T'=T)\\ (d) & (\Gamma\vdash T=T')\wedge(\Gamma\vdash T'=T'')\Rightarrow(\Gamma\vdash T=T'')\end{array}$$

3.

$$\begin{array}{ll}(a) & (\Gamma\vdash o=o':T)\Rightarrow(\Gamma\vdash o:T)\\ (b) & (\Gamma\vdash o:T)\Rightarrow(\Gamma\vdash o=o:T)\\ (c) & (\Gamma\vdash o=o':T)\Rightarrow(\Gamma\vdash o'=o:T)\\ (d) & (\Gamma\vdash o=o':T)\wedge(\Gamma\vdash o'=o'':T)\Rightarrow(\Gamma\vdash o=o'':T)\end{array}$$

4.

$$\begin{array}{ll}(a) & (\Gamma_1\vdash T=T')\wedge(\Gamma_1,T,\Gamma_2\vdash S=S')\Rightarrow(\Gamma_1,T',\Gamma_2\vdash S=S')\\ (b) & (\Gamma_1\vdash T=T')\wedge(\Gamma_1,T,\Gamma_2\vdash o=o':S)\Rightarrow(\Gamma_1,T',\Gamma_2'\vdash o=o':S)\\ (c) & (\Gamma\vdash S=S')\wedge(\Gamma\vdash o=o':S)\Rightarrow(\Gamma\vdash o=o':S')\end{array}$$

5.

$$\begin{array}{lll}(a) & (\Gamma_1,T\rhd)\wedge(\Gamma_1,\Gamma_2\vdash S=S')\Rightarrow(\Gamma_1,T,t_{i+1}\Gamma_2\vdash t_{i+1}S=t_{i+1}S') & i=l(\Gamma)\\ (b) & (\Gamma_1,T\rhd)\wedge(\Gamma_1,\Gamma_2\vdash o=o':S)\Rightarrow(\Gamma_1,T,t_{i+1}\Gamma_2\vdash t_{i+1}o=t_{i+1}o':t_{i+1}S) & i=l(\Gamma)\end{array}$$

6.

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

7.

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

Then the relations $\sim$ and $\simeq$ are equivalence relations on C and $\widetilde{C}$ which satisfy the conditions of [5, Proposition 5.4] and therefore they correspond to a regular congruence relation on the C-system defined by $(C,\widetilde{C})$.

Lemma 6.3 *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 $(T_1,\ldots,T_n\vdash$

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

(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 *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 *The conditions (3) and (4) of [5, Proposition 5.4] hold for $\sim$ and $\simeq$.*

Proof: 1. We need to show that for $(\Gamma,T\rhd)$, and $\Gamma\sim\Gamma'$ there exists $(\Gamma',T'\rhd)$ 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 $\Gamma',T\rhd$.

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

Proof: (1) Given $(\Gamma_1,T\rhd)\sim(\Gamma_1',T'\rhd)$ and $(\Gamma_1,\Gamma_2\rhd)\sim(\Gamma_1',\Gamma_2'\rhd)$ 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 $(\Gamma_1, T\rhd) \sim (\Gamma'_1, T'\rhd)$ and $(\Gamma_1, \Gamma_2, S\rhd) \sim (\Gamma'_1, \Gamma'_2, S'\rhd)$. The later condition is equivalent to $(\Gamma_1, \Gamma_2\rhd) \sim (\Gamma'_1, \Gamma'_2\rhd)$ 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 $(\Gamma_1, T\rhd) \sim (\Gamma'_1, T'\rhd)$ 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 $(\Gamma_1, T, \Gamma_2\rhd) \sim (\Gamma'_1, T', \Gamma'_2\rhd)$ 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 $(\Gamma_1, T, \Gamma_2, S\rhd) \sim (\Gamma'_1, T', \Gamma'_2, S'\rhd)$. The later condition is equivalent to $(\Gamma_1, T, \Gamma_2\rhd) \sim (\Gamma'_1, T', \Gamma'_2\rhd)$ 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 $(\Gamma_1, T, \Gamma_2, S'\rhd)$ 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 *Let C be a subset of $Ob(CC(M))$ 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 *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 *Let* $C, \leq$ *be as in Lemma 6.7, let* $\widetilde{C}$ *be a subset of* $\widetilde{Ob}(CC(M))$ *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 *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 *Let $(C, \widetilde{C})$ be subsets in $Ob(CC(M))$ and $\widetilde{Ob}(CC(M))$ respectively which correspond to a C-subsystem CC of $CC(M)$. 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 [5, 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] Bart Jacobs. *Categorical logic and type theory*, volume 141 of *Studies in Logic and the Foundations of Mathematics*. North-Holland Publishing Co., Amsterdam, 1999.

[2] 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.

[3] Vladimir Voevodsky. Notes on type systems. Author's website, 2009-2012.

[4] Vladimir Voevodsky. Monads defined by dependent signatures. *In preparation.*, 2014.

[5] Vladimir Voevodsky. Subsystems of C-systems. *http://arxiv.org/abs/1406.7413*, 2014.