Contextual category of a finitary monad

Vladimir Voevodsky[1,2]

July 2014

Examples:

1. If $M = Id$ i.e. $M(X) = X$ the $M - cor = FSets$ is the category of finite sets. It is easy to see that the category of finite sets is the free category with finite coproducts generated by one object. Therefore, $(FSets)^{op}$ can be thought of the free category with finite products generated by one object.

2. Let M be given by $M(X) = X \amalg A$ where A is a set. This corresponds to the system of expressions where all expressions are either variables or constants and the set of constants is A. The category $(M - cor)^{op}$ can be though of as the free category with finite products generated by an object U and the set A of morphisms $pt \to U$.

1 Systems of expressions

Note: [**?**], [**?**].

Free systems of expressions. Let M be a set and let $T(M)$ be the set of finite rooted trees whose vertices (including the root) are labeled by elements of M and such that for any vertex the set of edges leaving this vertex is ordered. Note that such ordered trees have no symmetries. We will use the following notations. For $T \in T(M)$ let $Vrtx(T)$ be the set of vertices of T and for $v \in Vrtx(T)$ let $lbl(v) = lbl(v)_T \in M$ be the label on v. We will sometimes write $v \in T$ instead of $v \in Vrtx(T)$. For $v \in Vrtx(T)$ let $[v] = [v]_T \in T(M)$ be the subtree in T which consists of v and all the vertices under v. Let $val(v)$ be the valency of v i.e. the number of edges leaving v and $ch_1(v), \ldots, ch_{val(v)}(v) \in Vrtx(T)$ be the "children" of v i.e. the end points of these edges. Let further $br_i(v) = [ch_i(v)]$ be the branches of $[v]$. We write $v \leq w$ (resp. $v < w$) if $v \in [w]$ (resp. $v \in [w] - w$). We say that two vertices v and w are independent if $v \notin [w]$ and $w \notin [v]$.

For three sets A, B and $Cont$ let

$$AllExp(A, B; Con) = T(A \amalg B \amalg (Con \times (\amalg_{n \geq 0} B^n)))$$

Elements of $AllExp(A, B; Con)$ are called expressions over the alphabet Con (or with a set of constructors Con), free variables from A and bound variables from B.

An expression is called unambiguous if it satisfies the following conditions:

1. if $lbl(v) \in A \amalg B$ then $val(v) = 0$,

2. (a) if $v < v'$, $lbl(v) = (c; x_1, \ldots, x_n)$ and $lbl(v') = (c'; x'_1, \ldots, x'_{n'})$ then $\{x_1, \ldots, x_n\} \cap \{x'_1, \ldots, x'_{n'}\} = \emptyset$,

[1]School of Mathematics, Institute for Advanced Study, Princeton NJ, USA. e-mail: vladimir@ias.edu
[2]Work on this paper was supported by NSF grant 1100938.

(b) if $lbl(v) = (c; x_1, \ldots, x_n)$ then $x_i \neq x_j$ for $i \neq j$,

3. if $lbl(v) = (c; x_1, \ldots, x_n)$ and $lbl(v') \in \{x_1, \ldots, x_n\}$ then $v' \in [v]$.

The first conditions says that a vertex labeled by a variable is a leaf. The second one is equivalent to saying that if the same variable is bound at two different vertices v, v' then these vertices are independent i.e. $[v] \cap [v'] = \emptyset$ and that a vertex can not bind the same variable twice. The third one says that all the leaves labeled by a bound variable lie under the vertex where it is boud. We let $UAExp(A, B; Con)$ denote the subset of unambiguous expressions in $AllExp(A, B; Con)$. Note that for for any $T \in UAExp(A, B; Con)$ and $v \in Vrtx(T)$ there is a subset $Ext(v) \subset B$ such that

$$[v] \in UAExp(A \amalg Ext(v), B \backslash Ext(v); Con)$$

Any triple of maps $f_{Con} : A \to A'$, $f_B : B \to B'$, $f_{Con} : Con \to Con'$ define a map

$$f_* = (f_A, f_B, f_{Con})_* : AllExp(A, B; Con) \to AllExp(A', B'; Con')$$

which changes labels in the obvious way. If f_B is injective then f_* maps unambiguous expressions to unambiguous ones.

An element T of $UAExp(A, B; Con)$ is said to be strictly unambiguous if for any $v \neq v'$ in $Vrtx(T)$ such that $lbl(v) = (c; x_1, \ldots, x_n)$ and $lbl(v') = (c'; x'_1, \ldots, x'_{n'})$ one has $\{x_1, , \ldots, x_n\} \cap \{x'_1, \ldots, x'_{n'}\} = \emptyset$ i.e. if the names of all bound variables are different. We let $SUAExp(A, B; Con)$ denote the subset of strictly unambiguous expressions in $UAExp(A, B; Con)$.

An element T of $UAExp(A, B; Con)$ is said to be α-equivalent to an element T' of $UAExp(A, B'; Con)$ if there is a set B'' , an element $T'' \in UAExp(A, B''; Con)$ and two maps $f : B'' \to B$, $f' : B'' \to B'$ such that $T = (Id, f, Id)_*(T'')$ and $T' = (Id, f', Id)_*(T'')$. The following lemma is straightforward:

Lemma 1.1 *[***2009.09.08.l1***] For any two sets A and Con one has:*

1. *α-equivalence is an equivalence relation,*

2. *for any set B and any element $T \in UAExp(A, B; Con)$ there exists an element $T' \in UAExp(A, \mathbf{N}; Con)$ such that $T \overset{\alpha}{\sim} T'$ and T' is strictly unambiguous,*

3. *fwo strictly unambiguous elements $T, T' \in UAExp(A, B; Con)$ are α-equivalent if and only if there exists a permutation $f : B \to B$ such that $(Id, f, Id)_*(T) = T'$ (cf. swapping).*

We let $Exp_\alpha(A; Con)$ denote the set of α-equivalence classes in $\amalg_B UAExp(A, B; Con)$. In view of Lemma 1.1 this set is well defined and can be also defined as the set of equivalence classes in $SUAExp(A, \mathbf{N}; Con)$ modulo the equivalence relation generated by the permutations on $\mathbf{N}$.

Note that for two α-equivalent expressions T_1, T_2 and a vertex $v \in V(T_1) = V(T_2)$ the expressions $[v]_{T_1}$ and $[v]_{T_2}$ need not be α-equivalent since some of the variables which are bound in T_1 may be free in $[v]$.

The maps $(f_A, f_B, f_{Con})_*$ respect α-equivalence. Therefore for any $f_A : A \to A'$ and $f_{Con} : Con \to Con'$ there is a well defined map

$$(f_A, f_{Con})_* : Exp_\alpha(A; Con) \to Exp(A'; Con')$$

which make $Exp_\alpha(-;-)$ into a covariant functors from pairs of sets to sets. In addition there is a well defined notion of substitution on $Exp_\alpha(-;Con)$ which may be considered as a collection of maps of the form:

$$Exp_\alpha(A;Con) \times (\prod_{a\in A} Exp_\alpha(X_a;Con)) \to Exp_\alpha(\amalg_{a\in A} X_a;Con)$$

given for all pairs $(A;\{X_a\}_{a\in A})$ where A is a set and $\{X_a\}_{a\in A}$ a family of sets parametrized by A. Alternatively, the substitution structure can be seen as a collection of maps

$$Exp_\alpha(Exp_\alpha(A;Con);Con) \to Exp_\alpha(A;Con)$$

given for all A and Con. These maps make the functor $Exp_\alpha(-;Con)$ into a monad (triple) on the category of sets which functorially depends on the set Con.

Example 1.2 [**lambda**] The mapping which sends a set X to the set of α-equivalence classes of terms of the untyped λ-calculus with free variables from X is a sub-triple of $Exp_\alpha(-;Con)$ where $Con = \{\lambda, ev\}$. Elements T of $UAExp(X, \mathbf{N}; \{\lambda, ev\})$ which belong to this sub-triple are characterized by the following "local" conditions:

1. for each $v \in T$, $lbl(v) \in X \amalg \mathbf{N} \amalg \{ev\} \amalg \{\lambda\} \times \mathbf{N}$
2. if $lbl(v) \in \{\lambda\} \times \mathbf{N}$ then $val(v) = 1$
3. if $lbl(v) = ev$ then $val(v) = 2$.

Example 1.3 [**propositional**]The mapping which sends a set X to the set of terms of the propositional calculus with free variables from X is a sub-triple of $Exp_\alpha(-;C_0)$ where $C_0 = \{\vee, \wedge, \neg, \Rightarrow\}$. Elements T of $UAExp(X, \mathbf{N}; C_0)$ which belong to this sub-triple are characterized by the following "local" conditions:

1. for all $v \in T$, $lbl(v) \in X \amalg C_0$
2. if $lbl(v) \in \{\vee, \wedge, \Rightarrow\}$ then $val(v) = 2$
3. if $lbl(v) = \neg$ then $val(v) = 1$.

Example 1.4 [**multisorted**] Consider first order logic with several sorts $GS = \{S_1, \ldots, S_n\}$. Let GP be the set of generating predicates and GF the set of generating functions. Let $C_1 = C_0 \amalg \{\forall, \exists\}$ and $C_2 = C_1 \amalg GP \amalg GF \amalg GS$. We can identify the α-equivalence classes of formulas of the first order language defined by GS and GF with free variables from a set X with a subset in $Exp_\alpha(X, \mathbf{N}; C_2)$. Vertices which are labeled by $(\forall; x)$ and $(\exists; x)$ have valency two. For such a vertex v, the first branch of $[v]$ is one vertex labeled by an element of GS giving the sort over which the quantification occurs and the second branch is the expression which is quantified. Now however, these subsets do not form a sub-triple of Exp_α since not all substitutions are allowed. By allowing all substitutions irrespectively of the sort we get (for each X) a subset in $Exp_\alpha(X; C_2)$ whose elements will be called pseudo-formulas.

The following operations on expressions are well defined up to the α-equivalence:

1. If $T_1, \ldots, T_m \in Exp_\alpha(A; Con)$, $a_1, \ldots, a_n$ are pair-wise different elements of A and $M \in Con$ we will write $(M, a_1, \ldots, a_n)(T_1, \ldots, T_m)$ for the expression whose root v is labeled by $(M, a_1, \ldots, a_n)$, $val(v) = n$ and $br_i(v) = T_i$.

2. For $T_1, T_2 \in Exp_\alpha(A; Con)$ and $v \in T_1$ we let $T_1(T_2/[v])$ be the expression obtained by replacing $[v]$ in T_1 with T_2' where T_2' is obtained from T_2 by the change of bound variables such that the bound variables of T_2' do not conflict with the variables of T_1.

3. For $T, R_1, \ldots, R_n \in Exp_\alpha(A; Con)$ and $y_1, \ldots, y_n \in A$ we let $T(R_1/y_1, \ldots, R_n/y_n)$ denote the expression obtained by changing R_i's by α-equivalent R_i' such that $bnd(R_i') \cap bnd(R_j)' = \emptyset$ for $i \neq j$, changing T to an α-equivalent T' such that $bnd(T') \cap (var(R_1') \cup \ldots \cup var(R_n')) = \emptyset$ and then replacing all the leaves of T' marked by y_i by R_i'.

In all the examples considered above, these operations correspond to the usual operations on formulas. The first operation can be used to directly associate expressions in our sense with the formulas. For example, the expression associated with the formula $\forall x : S.P(x, y)$ in a multi-sorted predicate calculus is $(\forall, x)(S, P(x, y))$ where as was mentioned above we use the same notation for an element of $A \amalg B \amalg (Con \times (\amalg_{n \geq 0} B^n))$ and the one vertex tree with the corresponding label.

Note: about representing elements of $AllExp(A, B; Con)$ by linear sequences of elements of $A \amalg B \amalg$??.

Reduction structures. Another component of the structure present in systems of expressions used in formal systems is the reduction relation. It is very important for our approach to type systems that the reduction relation is defined on all pseudo-formulas and is compatible with the substitution structure even when not all psedu-formulas are well formed formulas. In what follows we will consider, instead of a particular syntactic system, a pair $(S, \triangleright)$ where S is a continuous triple on the category of sets and $\triangleright$ is a reduction structure on S i.e. a collection of relations $\triangleright_X$ on $S(X)$ given for all finite sets X satisfying the following two conditions:

1. if $E \in S(\{x_1, \ldots, x_n\})$, $f_1, \ldots, f_n, f_i' \in S(\{y_1, \ldots, y_m\})$ and $f_i \triangleright_{\{y_1, \ldots, y_m\}} f_i'$ then
$$E(f_1/x_1, \ldots, f_i/x_i, \ldots f_n/x_n) \triangleright_{\{x_1, \ldots, x_n\}} E(f_1/x_1, \ldots, f_i'/x_i, \ldots f_n/x_n),$$

2. if $E, E' \in S(\{x_1, \ldots, x_n\})$, $f_1, \ldots, f_n \in S(\{y_1, \ldots, y_m\})$ and $E \triangleright_{\{x_1, \ldots, x_n\}} E'$ then
$$E(f_1/x_1, \ldots, f_n/x_n) \triangleright_{\{x_1, \ldots, x_n\}} E'(f_1/x_1, \ldots, f_n/x_n).$$

The following two results are obvious but important.

Proposition 1.5 *[2009.10.17.prop1] Let S be a continuous triple on Sets and $\triangleright_\alpha$ be a family of reduction structures on S. Then the intersection $\cap_\alpha \triangleright_\alpha : X \mapsto \cap_\alpha \triangleright_{\alpha,X}$ is a reduction structure on S.*

Corollary 1.6 *[2009.10.17.cor1] For any family (X_α, pre_α) of pairs of the form (X, pre) where X is a set and pre is a relation on $S(X)$ (i.e. a subset of $S(X) \times S(X)$) there exists the smallest reduction structure $\triangleright = \triangleright(X_\alpha, pre_\alpha)$ on S such that for each α and each $(f, g) \in pre_\alpha$ one has $f \triangleright g$.*

2 C-systems defined by a triple.

Let S be a continuous triple on $Sets$. Let $S-cor$ be the full subcategory of the Kleisli category of S whose objects are finite sets. Recall, that the set of morphisms from X to Y in $S-cor$ is the set of maps from X to $S(Y)$ i.e. $S(Y)^X$ (in other words, $S-cor$ is the category of free, finitely generated S-algebras). We will construct two C-systems $C(S)$ and $CC(S)$which are based on $(S-cor)^{op}$.

Examples:

1. If $S = Id$ i.e. $S(X) = X$ the $S - cor = FSets$ is the category of finite sets. It is easy to see that the category of finite sets is the free category with finite coproducts generated by one object. Therefore, $(FSets)^{op}$ can be thought of the free category with finite products generated by one object.

2. Let S be given by $S(X) = X \amalg A$ where A is a set. This corresponds to the system of expressions where all expressions are either variables or constants and the set of constants is A. The category $(S - cor)^{op}$ can be though of as the free category with finite products generated by an object U and the set A of morphisms $pt \to U$.

The categories of sets, finite sets or even the category of finite linearly ordered sets and their isomorphisms are all level 1 categories and so is the category $S-cor$. We can get a set-level model $C(S)$ for $(S-cor)^{op}$ by setting $Ob(C(S)) = \mathbf{N}$ and $Hom_{C(S)}(n,m) = S(\{1,\dots,n\})^m$.

The category $C(S)$ 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 $(1,\dots,n-1)$. For $f = (f_1,\dots,f_m) : n \to m$ the canonical square build on f and the canonical projection $m+1 \to m$ is of the form

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

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

Note: add notes that a continuous sub-triple of S is exactly the same as a subcategory in $S-cor$ which contains all (isomorphism classes of) objects. Intersection of two sub-triples is a sub-triple which allows us to speak of sub-triples (systems of expressions etc.) generated by a set of expressions. For the construction of type systems the category $S-cor$ is replaced by the C-system $CC(S,X)$.

Note: that continuous triples on $Sets$ are the same as category structures on $\mathbf{N}$ which extend the a category structure of finite sets and where the addition remains to be coproduct.

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

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

and the set of morphisms is

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

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

Note that if $S(\emptyset)=\emptyset$ then $CC(S)=\emptyset$ and otherwise the functor $CC(S)\to(S-cor)^{op}$ is an equivalence, so that in the second case $C(S)$ and $CC(S)$ are indistinguishable as level 1 categories. However, as set level categories they are quite different.

The category $CC(S)$ is given 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 $(T_1,\ldots,T_{m+1})$ and a morphism $(f_1,\ldots,f_m)\in S(\{1,\ldots,n\})^m$ from $(R_1,\ldots,R_n)$ to $(T_1,\ldots,T_m)$ is of the form

[**2009.11.05.oldeq1**]

$$\begin{array}{ccc} (R_1,\ldots,R_n,T_{m+1}(f_1/1,\ldots,f_m/m)) & \xrightarrow{(f_1,\ldots,f_m,n+1)} & (T_1,\ldots,T_{m+1}) \\ \downarrow & & \downarrow \\ (R_1,\ldots,R_n) & \xrightarrow{(f_1,\ldots,f_m)} & (T_1,\ldots,T_m) \end{array} \qquad (1)$$

Proposition 2.1 *[**2009.10.01.prop2**] With the maps defined above $CC(S)$ is a C-system.*

Proof: Straightforward.

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

Any morphism of triples $S\to S'$ defines a C-system morphism $CC(S)\to CC(S')$. C-subsystems of $CC(S)$, which are discussed in more detail below, provide an important class of type systems over S.

There is another construction of a category from a continuous triple S 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 category $CC(S,Var)$ as follows. The set of objects of $CC(S,Var)$ is

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

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

$$Hom_{CC(S,,Var)}((x_1:E_1,\ldots,x_n:E_n),(y_1:T_1,\ldots,y_m:T_m)) = S(\{x_1,\ldots,x_n\})^m$$

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 $CC(S, X)$ to $CC(S)$. This functor is clearly an equivalence. There is an obvious final object and ft map on $CC(S, X)$. There is however a real problem in making it into a C-system which is due to the following. Consider an object $(y_1 : T_1, \ldots, y_{m+1} : T_{m+1})$ and a morphism $(f_1, \ldots, f_m) : (x_1 : R_1, \ldots, x_n : R_n) \to (y_1 : T_1, \ldots, y_m : T_m)$. In order for the functor to $CC(S)$ to be a C-system morphism the canonical square build on this pair should have the form

$$\begin{array}{ccc} (x_1 : R_1, \ldots, x_n : R_n, x_{n+1} : T_{m+1}(f_1/1, \ldots, f_m/m)) & \xrightarrow{(f_1,\ldots,f_m,n+1)} & (y_1 : T_1, \ldots, y_{m+1} : T_{m+1}) \\ \downarrow & & \downarrow \\ (x_1 : R_1, \ldots, x_n : R_n) & \xrightarrow{(f_1,\ldots,f_m)} & (y_1 : T_1, \ldots, y_m : T_m) \end{array}$$

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

3 C-subsystems of $CC(S)$.

Let TS be a C-subsystem of $CC(S)$. By Lemma **??**, TS is determined by the subsets $B = Ob(TS)$ and $\widetilde{B} = \widetilde{Ob}(TS)$ in $Ob(CC(S))$ and $\widetilde{Ob}(CC(S))$. By definition we have

$$Ob(CC(S)) = \coprod_{n\geq 0} \prod_{i=0}^{n-1} S(\{1, \ldots, i\})$$

An element of $\widetilde{Ob}(CC(S))$ is given by a pair (Γ, s) where $\Gamma \in Ob(CC(S))$ is an object and $s : ft(\Gamma) \to \Gamma$ is a section of the canonical morphism $p_\Gamma : \Gamma \to ft(\Gamma)$. It follows immediately from the definition of $CC(S)$ that for $\Gamma = (E_1, \ldots, E_{n+1})$, a morphism $(f_1, \ldots, f_{n+1}) \in S(\{1, \ldots, n\})^{n+1}$ from $ft(\Gamma)$ to Γ is a section of p_Γ if an only if $f_i = i$ for $i = 1, \ldots, n$. Therefore, any such section is determined by its last component f_{n+1} and mapping $((E_1, \ldots, E_{n+1}), (f_1, \ldots, f_{n+1}))$ to $(E_1, \ldots, E_n, E_{n+1}, f_{n+1})$ we get a bijection

$$[\mathbf{2009.10.15.eq2}]\widetilde{Ob}(CC(S)) \cong \coprod_{n\geq 0} (\prod_{i=0}^{n-1} S(\{1, \ldots, i\})) \times S(\{1, \ldots, n\})^2 \quad (2)$$

For $\Gamma = (E_1, \ldots, E_n)$ we write $(\Gamma \rhd_{TS})$ if $(E_1, \ldots, E_n)$ is in B and $(\Gamma \vdash_{TS} t : T)$ if $(E_1, \ldots, E_n, T, t)$ is in $\widetilde{B}$. When no confusion is possible we will write $\vdash$ instead of $\vdash_{TS}$. We also write $l(\Gamma) = n$ and $ft(\Gamma) = (E_1, \ldots, E_{n-1})$.

The following result is an immediate corollary of Proposition **??**.

Proposition 3.1 *[***2009.10.16.prop3***] Let S be a continuous triple on Sets. A pair of subsets*

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

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

defines a C-subsystem of $CC(S)$ if and only if the following conditions hold:

1. $(\rhd)$
2. $(\Gamma\rhd) \Rightarrow (ft(\Gamma)\rhd)$
3. $(\Gamma \vdash t : T) \Rightarrow (\Gamma, T\rhd)$
4. $(\Gamma_1, \Gamma_2, \vdash o : S) \wedge (\Gamma_1, T\rhd) \Rightarrow (\Gamma_1, T, s_{i+1}\Gamma' \vdash s_{i+1}o : s_{i+1}S)$ *where* $i = l(\Gamma_1)$
5. $(\Gamma_1, T, \Gamma_2 \vdash o : S) \wedge (\Gamma_1 \vdash r : T) \Rightarrow (\Gamma_1, d_{i+1}(\Gamma_2[r/i+1]) \vdash d_{i+1}(t[r/i+1]) : d_{i+1}(T[r/i+1]))$ *where* $i = l(\Gamma_1)$
6. $(\Gamma, T\rhd) \Rightarrow (\Gamma, T \vdash n+1 : T)$ *where* $n = l(\Gamma)$.

where for $E \in S(\{1,\ldots,k\})$, $s_iE = E[i+1/i,\ldots,k+1/k] \in S(\{1,\ldots,k+1\}$ *and* $d_iE = E[i/i+1,\ldots,k-1/k] \in S(\{1,\ldots,k-1\}$

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

4a $(\Gamma_1, \Gamma_2\rhd) \wedge (\Gamma_1, T\rhd) \Rightarrow (\Gamma_1, T, s_{i+1}\Gamma_2\rhd)$ where $i = l(\Gamma_1)$

5a $(\Gamma_1, T, \Gamma_2\rhd) \wedge (\Gamma_1 \vdash r : T) \Rightarrow (\Gamma_1, d_{i+1}(\Gamma_2[r/i+1])\rhd)$ where $i = l(\Gamma_1)$.

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

Remark 3.2 [**2010.08.07.rem1**] If one re-writes the conditions of Proposition 3.1 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:

$$\frac{}{\rhd} \qquad \frac{x_1 : E_1, \ldots, x_n : E_n\rhd}{x_1 : E_1, \ldots, x_{n-1} : E_{n-1} \vdash} \qquad \frac{x_1 : E_1, \ldots, x_n : E_n \rhd t : T}{x_1 : E_1, \ldots, x_n : E_n \vdash}$$

$$\frac{x_1 : E_1, \ldots, x_n : E_n \vdash t : T \quad x_1 : E_1, \ldots, x_i : E_i, y : F\rhd}{x_1 : E_1, \ldots, x_i : E_i, y : F, x_{i+1} : E_{i+1}, \ldots, x_n : E_n \vdash t : T}, \quad i = 0, \ldots, n$$

$$\frac{x_1 : E_1, \ldots, x_n : E_n \vdash t : T \quad x_1 : E_1, \ldots, x_i : E_i \vdash r : E_{i+1}}{x_1 : E_1, \ldots, x_i : E_i, x_{i+2} : E_{i+2}[r/x_{i+1}], \ldots, x_n : E_n[r/x_{i+1}] \vdash t[r/x_{i+1}] : T[r/x_{i+1}]}, \quad i = 0, \ldots, n-1$$

$$\frac{x_1 : E_1, \ldots, x_n : E_n\rhd}{x_1 : E_1, \ldots, x_n : E_n \vdash x_n : E_n}$$

which are similar to (and probably equivalent) the "basic rules of DTT" given in [**?**, 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 3.3 *[**2009.11.05.l1**] Let S, B, $\widetilde{B}$ be as above and let $(E_1, \ldots, E_n), (T_1, \ldots, T_m) \in B$ and $(f_1, \ldots, f_m) \in S(\{1, \ldots, n\})^m$. Then*

$$(f_1, \ldots, f_m) \in Hom_{TS}((E_1, \ldots, E_n), (T_1, \ldots, T_m))$$

if and only if $(f_1, \ldots, f_{m-1}) \in Hom_{TS}((E_1, \ldots, E_n), (T_1, \ldots, T_{m-1}))$ and

$$(E_1, \ldots, E_n, T_m(f_1/1, \ldots, f_{m-1}/m-1), f_m) \in \widetilde{B}$$

Proof: Straightforward using the fact that the canonical pull-back squares in $CC(S)$ are given by (1).