A universe polymorphic type system

Vladimir Voevodsky

Started January 11, 2012. Work in progress.

Notes from discarded attempts:

1. Jan. 21, 2012 Without distinguishing types and terms whose type is a universe (i.e. without El constructor) and without type cumulativity necessarily leads to a system where

a. one has to introduce explicit functions going both ways between the types represented by a term of U_n and its image in U_{n+1} together with additional reduction rules for making these functions strict isomorphisms,

b. one will have a proliferation of non-equal but strictly isomorphic types sometimes having the same universe level as for example $X \to Y$ and $X \to j\,Y$ for $X : U_{n+1}$, $Y : U_n$.

2. Jan. 27, 2012. In a system with T- and o-expressions one can not restrict oneself to $forall$ operators acting only between terms of the same universe level since it leads to the loss of confluence for the reduction related to the interaction of $forall$ and j and the reduction related to the interaction of two j's.

3. The introduction of resizing rules rr0 and rr1 leads naturally to a situation where definitional equality is no more determined as combination of $\equiv_{\mathcal{A}}$ with the equivalence relation generated by reductions. While reductions and associated notion of non-reducible terms (which exist conjecturally) remain, two non-reducible terms may be definitionally equal in a "non-trivial" way.

4. An example of a derivable term E with a sub-term f such that replacing f be f' which is connected to f by an object of $Id f f'$ makes the resulting term $E[f'/f]$ non-derivable. One takes $f : Bool \to \sum t : \mathcal{U}, [El](T)$ (where $\mathcal{U}$ is a universe) given by $f(true) = (T_0, a)$, $f(false) = (T_0, b)$. Then $E = Id\, T_0, [pr2](f\, true), [pr2](f\, false)$. If f' is a function connected by an object of Id to f and such that $[pr1](f'\, true)$ is not definitionally equal to T_0 then $E[f'/f]$ is not derivable. There should be even more elementary examples.

5. Aug. 17, 2012. Started to re-write the specification using the definition of a type system based on four classes of sentences: "contexts", T-terms equalities in a context, judgements and o-terms equalities in a type in a context.

Contents

1 Introduction

We will work on the semi-syntactic level with named variables. We have three classes of variables - o-variables which can be substituted by o-expressions (see below), T-variables which can be substituted by T-expressions (see below) and u-level variables. Semi-syntactic means here that we are going to define expressions or pre-terms as planar trees with vertices marked either by a name of a variable or by a u-level expression (see below) or by a sequence of the form $(S; x_1, \ldots, x_n)$ where S is a special symbol (from the given set of allowed special symbols), $n \geq 0$ and $x_1, \ldots, x_n$ are names of o-variables. The occurrences of variables named in the node label in the nodes below are considered to be bound. All expressions are considered up to α-equivalence i.e. the renaming of bound variables and since the problems related to the possible conflict of variable names are the usual ones we ignore them in this exposition.

This level of formalization is an intermediate one between the syntactic (linear) presentation of expressions on the one hand and nameless presentation (using some version of de Brujin indexes) on the other. The choice of a particular syntactic representation which is necessary at the level of the user interface and of a particular nameless representation which is necessary for the internal manipulation of expressions by the system can vary and are left for the next stage of concretization.

We describe several systems here by starting with the simplest one and later adding more and more features. Common to all the systems defined in this paper is that they are fully universe polymorphic. In particular there is a class of expressions which are called u-level expressions which are used to specify the level of universes and which can contain u-variables. They also come with their own concept of equality $\equiv_{\mathcal{A}}$ modulo a given set of universe constraints $\mathcal{A}$. Note that what we describe at each step is not an individual type system but a family of type systems parametrized by "universe contexts" - the lists of permitted universe variables and required constraints between these variables and by lists of T-variables. To each universe context and a list of T-variables there corresponds a type system in the sense of **??** and in particular a contextual category.

We will also discuss transformations between universe contexts and lists of type variables which lead to functors between the corresponding contextual categories or, on the more practical level, to automatic translations of definitions, theorems and proofs built in one universe context and with one list of type variables into similar objects built in another universe context and/or with another list of type variables.

In each of the systems we define first terms, T-terms and o-terms. These are expressions which satisfy certain simple local conditions. For a given universe context and a sequence of T-variables, the derivable terms of the corresponding type system which "denote" types will always be T-terms and the derivable terms which "denote" objects of types will always be o-terms. The concepts of contexts as sequences of the form $(x_1 : E_1, \ldots, x_n : E_n)$ where E_i is a T-term with free o-variables from the set $\{x_1, \ldots, x_{i-1}\}$ and judgements as sequences of the form $(x_1 : E_1, \ldots, x_n : E_n; o : T)$ where $(E_1, \ldots, E_i)$ is a context, T is a T-expression with free o-variables from $\{x_1, \ldots, x_n\}$ and o is an o-expression with free o-variables from $\{x_1, \ldots, x_n\}$ are standard.

The terms are introduced in such a way that for any context $\Gamma = (x_1 : E_1, \ldots, x_n : E_n)$ and any o-term o with free variables from $\{x_1, \ldots, x_n\}$ there is an easily computable T-term $\tau_\Gamma(o)$ which in the case when both the Γ and o are derivable gives a canonical form for the type of o. To have this typing function defined on all o-terms we had to introduce some unusual features into the syntax. For example evaluation node in our system is a quantifier of the form $[ev; x](o_1, o_2, T)$ where T is the target type of the function o_1 and x is the "dependent variable" in T. For derivable terms this extra arguments can be reconstructed from the other data. Whether it makes sense to keep those arguments in the implementations or not is a question which should be considered separately.

We further consider the equivalence relation $\sim_{\mathcal{A}}$ on terms. It is based on another important concept of essential and non-essential nodes and sub-terms of a term. Two terms E_1 and E_2 are $\sim_{\mathcal{A}}$ equivalent if the expressions $Ess(E_1)$ and $Ess(E_2)$ obtained from E_1 and E_2 by the removal of all non-essential sub-expressions are $\equiv_{\mathcal{A}}$ equivalent.

Both $\equiv_{\mathcal{A}}$ and $\sim_{\mathcal{A}}$ are shown to be decidable and in practice are expected to be very easily decidable i.e. to have low complexity of the decision procedure.

The next structure which we define for each type system is the set of reduction rules which are also defined on all terms. The reduction rules depend on the universe context and especially on the set of universe constraints of this context $\mathcal{A}$ and we denote the transitive version of reducibility relative to $\mathcal{A}$ by $\succ_{\mathcal{A}}$ and the transitive reflexive version by $\succeq_{\mathcal{A}}$. We only consider reductions which occur at the essential nodes of a term. While our reduction rules can be equally applied to non-essential nodes it is not clear whether that such an extended notion has any practical uses.

Definitional equality of a given type system is the third member of a sequence of equivalence relation $\equiv_{\mathcal{A}}$, $\sim_{\mathcal{A}}$ and finally $\stackrel{d}{=}_{\mathcal{A}}$. As mentioned above, the relation $\equiv_{\mathcal{A}}$ is the relation of equality between u-level subexpressions which was mentioned above. The relation $\sim_{\mathcal{A}}$ is $\equiv_{\mathcal{A}}$ between the essential sub-expressions of the corresponding terms. Finally, definitional equality is defined as the equivalence relation generated by $\succeq_{\mathcal{A}}$ and $\sim_{\mathcal{A}}$.

To verify that our reduction rules are consistent we show local confluence (i.e. confluence of two one step reductions) in each system modulo $\sim_{\mathcal{A}}$ i.e. for $E \succ_{\mathcal{A}} E_1$ and $E \succ_{\mathcal{A}} E_2$ where both reductions are one-step ones we show the existence of E_1' and E_2' such that $E_1 \succeq_{\mathcal{A}} E_1'$, $E_2 \succeq_{\mathcal{A}} E_2'$ and $E_1 \sim_{\mathcal{A}} E_1'$. The confluence holds in most cases for general terms but in some exceptional cases it is expected to hold only for derivable terms.

We then describe the derivation rules in the context/judgement form. A slight difference between our form for derivation rules and the usual ones is that the rules for the formation of new type expressions are written directly as rules for adding a new component to a context i.e. we write $\Gamma, x : T \triangleright$ where one would usually write $\Gamma \vdash T : Type$.

For practical implementation of a type system in the style in which Coq is written one has to be able to deal with a situation when a user-written proof tactic produces a term whose correctness needs to be verified by an independent type-checker. Since the derivation steps of such a term are not supplied by the tactic there must be an algorithm which, given a context Γ or a judgement $\Gamma \vdash t : T$ is able to determined whether or not the context or the judgement is derivable.

(???) There is a more or less obvious algorithm of this sort for each of our type systems. The correctness and termination properties of these algorithms depend on a number of properties of each system which are usually summed up under the name of meta-theory. We prove some of these theorems and formulate others as conjectures for the future work. This choice is a consequences of three circumstances. Firstly, one can start to work on the implementation of a proof assistant based on the systems introduced here having just the syntax and the type checking algorithms. Secondly, the consistency of the proposed type systems can be verified by model construction which does not require most of the meta-theory results. Thirdly, due to the relative complexity of even the first of our type systems it appears to be unreasonable to try to prove more complex meta-theorems "by hand" and one should probably start by formalizing the definitions we give in one of the accepted proof assistants such as Coq and then proceed to giving formal proof of the meta-theorems. (???)

The last system (called simply TS) described here represents the first layer of the system which I hope will eventually be implemented in the new proof assistant. The second layer is formed by "quasi-constructive" features such as functional extensionality, ctf-terms and univalence axiom. The difference here lies in the nature of the algorithms which are required for computations in the fully constructive first layer and quasi-constructive second. We expect that an appropriate version of strong normalization holds for TS and that any object of type $\mathbf{N}$ (natural numbers) which is in the normal form is a numeral. Therefore, computation on terms of this layer can be done by a normalization algorithm.

The property that any object of type $\mathbf{N}$ normalizes to a numeral is known to become false when one adds quasi-constructive features. However, it is expected that a weaker property which is still sufficient for computation holds. Namely, it is expected that there is a terminating algorithm which, given a derivable object o of type $\mathbf{N}$ (in any context) which uses quasi-constructive features, produces a derivable object o' of the same type which is build without the use of these features and an object of the identity type between o and o'. This property should generalize to other "datatypes".

On notations and conventions: If E is a labelled tree, S is a branch of E and S' is another labelled tree we write $E[S'/S]$ for the labelled tree obtained by the direct substitution of the branch S' instead of S. If E, S are expressions with free variables from a set Fv and $x \in Fv$ we write $E[S/x]$ for the expression obtained by the substitution of all the occurrences of x by S with a possible renaming of bound variables such as to avoid name conflicts.

We will write $\vec{a}$ for a sequence $a_1, \ldots, a_n$ and let $dim(\vec{a})$ denote the number of items in the sequence.

We assume three alphabets for variables one for u-level variables, one for T-variables and one for o-variables. Unless otherwise indicated variables named x, y, z (with or without diacritics) are o-variables, variables named X, Y, Z (with or without diacritics) and the ones named p, q, r are u-level variables (see below). The letters i, j, k, l, m, n are reserved for numerals (e.g. in indexes).

We write $[L](B_1, \ldots, B_n)$ for a labelled (planar i.e. with ordered branches) tree with the root node labelled by L and branches $B_1, \ldots, B_n$. We also abbreviate $[L]([L'](B_1, \ldots, B_n))$ as $[L][L'](B_1, \ldots, B_n)$.

We will use the following lemma which holds for systems of expressions with any choice of labels:

Lemma 1.0.1 *[**sublemma**] Let E, o_1, o_2 be expressions with free variables from a set F. Let $x, y \in F$ where $x \neq y$. Assume that o_1 does not depend on y. Then one has $(E[o_2/y])[o_1/x] = (E[o_1/x])[o_2[o_1/x]/y]$.*

Proof: Easy induction by the depth of E with the cases when $E = [x]$, $E = [y]$ and $E = [L](B_1, \ldots, B_n)$ where $L \neq x, y$ to be considered separately.

2 Universe contexts

Definition 2.0.1 *[**d21a**] A u-level expression is either a numeral or a u-level variable or an expression of the form $M + n$ where M is a u-level expression and n is a numeral or an expression of the form $max(M_1, M_2)$ where M_1, M_2 are two u-level expressions.*

Note that u-level expressions are exactly "linear" function of u-level variables in the tropical (max-plus) semi-ring.

Definition 2.0.2 *[**d21b**] Let Fu be a finite set. A subset $\mathcal{A}$ of $\mathbf{N}^{Fu}$ is called admissible if it is defined by a system of equations of the form $M_i = N_i$ where M_i, N_i are u-level expressions in variables from Fu.*

Lemma 2.0.3 *[**uld**] Let $\mathcal{A}$ be an admissible subset of Fu and M, N are u-level expressions with variables from Fu. Then the condition "for all $\vec{n} \in \mathcal{A}$ one has $M(\vec{n}) = N(\vec{n})$" is decidable.*

Proof: The condition considered in the lemma can be expressed in Presburger arithmetic which is decidable. For a further discussion see [**?**].

Since $M(\vec{n}) \geq N(\vec{n})$ is equivalent to $M(\vec{n}) = max(M(\vec{n}), N(\vec{n}))$, Lemma 2.0.3 implies that conditions of the form "for all $\vec{n} \in \mathcal{A}$ one has $M(\vec{n}) \geq N(\vec{n})$" is also decidable.

Remark 2.0.4 Note also that any u-level expression can be re-written in the form $M = max(n_0, u_1 + n_1, \ldots, u_l + n_l)$ where u_i are u-level variables and $n_i \in \mathbf{N}$. By choosing a linear ordering on Fu and replacing $max(u + n, u + n')$ by $u + max(n, n')$ we can further assume that in our representation $u_i < u_{i+1}$ according to our ordering. This effectively provides us with a normal form of a u-level expression.

However as will be seen below we will be mostly interested in equality of u-level expressions not on the whole of $\mathbf{N}^{Fu}$ but on a given admissible subset of $\mathbf{N}^{Fu}$.

Definition 2.0.5 *[***uc***] A universe context UC is a pair $(Fu, \mathcal{A})$ where Fu is a finite sequence of u-level variables and $\mathcal{A}$ an admissible domain in $\mathbf{N}^{Fu}$.*

3 System TS0

We first describe the system $TS0$ which has only elements related to universes and dependent products and later add the elements related to other constructions.

3.1 TS0-terms and the typing function

Definition 3.1.1 *[***d01***] The following labels are permitted in the expressions of $TS0$: names of o-variables, names of T-variables, u-level expressions, $\mathcal{U}$, El, $(\prod; x)$, u, j, $(ev; x)$, $(\lambda; x)$, $(forall; x)$.*

Definition 3.1.2 *[***d02***] We distinguish three classes of expressions:*

1. *u-level expressions,*

2. *T-expressions are the ones with the root node $[X]$ where X is the name of a T-variable, $[\mathcal{U}]$, $[El]$ or $[\prod; x]$,*

3. *o-expressions are the ones with the root $[x]$ where x is the name of an o-variable, $[u]$, $[j]$, $[ev; x]$, $[\lambda; x]$ or $[forall; x]$.*

Definition 3.1.3 *[***d03***] A TS0-term is a T-expression or an o-expression (with labels from the set described in Definition 3.1.1) which satisfies the following conditions:*

1. *any node of the form $[M]$ where M is a u-level expression has valency 0 (i.e. is a leaf),*

2. *any node of the form $[X]$ where X is the name of a T-variable has valency 0 (i.e. is a leaf),*

3. *any node of the form $[\mathcal{U}]$ has valency 1 and its only branch is a node labelled by a u-level expression,*

4. *any node of the form* $[El]$ *has valency* 1 *with its only branch being an o-expression,*

5. *any node of the form* $[\prod; x]$ *has valency* 2*, both of its branches are T-expressions and the first branch does not contain* $[x]$*,*

6. *any node of the form* $[x]$ *where* x *the name of an o-variable has valency* 0 *(i.e. is a leaf),*

7. *any node of the form* $[u]$ *has valency* 1 *and its only branch is a node labelled by a u-level expression,*

8. *any node of the form* $[j]$ *has valency* 2 *and both of its branches are nodes labelled by u-level expressions,*

9. *any node of the form* $[ev; x]$ *has valency* 3 *its first two branches are o-expressions which do not contain* $[x]$ *and the third branch is a T-expression,*

10. *any node of the form* $[\lambda; x]$ *has valency* 2*, its first branch is a T-expression which does not contain* $[x]$ *and its second branch is an o-expression,*

11. *any node of the form* $[forall; x]$ *has valency* 4*, its first two branches are nodes labelled by u-level expressions and last two branches are o-expressions and the third branch does not contain* $[x]$*.*

We define o-terms as terms which are o-expression and T-terms as terms which are T-expressions. Intuitively, T-terms correspond to types and o-terms correspond to objects of types.

For $x \in Fv$ (resp. $X \in FV$) we will often write x instead of $[x]$ (resp. X instead of $[X]$). We will abbreviate $[\mathcal{U}][M]$ as $\mathcal{U}_M$, $[u][M]$ as u_M, $[j]([M_1],[M_2])$ as j_{M_1,M_2} and $[forall; x]([M_1],[M_2],o_1,o_2)$ as $[forall_{M_1,M_2}; x](o_1,o_2)$.

Definition 3.1.4 *[**d04**] Let* E *be a TS0-term. A node is called non-essential if it belongs to the subexpression* T *of a subexpression of the form* $[ex; x](o_1, o_2, T)$*. Otherwise a node is called essential. A subexpression is called non-essential (resp. essential) if its root node is non-essential (resp. essential).*

For a term E *we let* $Ess(E)$ *denote the subexpression of* E *(which is not a term anymore) which is obtained by removing all non-essential subexpressions from* E

Note that the root node of any term is essential, therefore $Ess(E)$ is always non-empty and therefore an expression.

Remark 3.1.5 There is a similarity in our description of TS0 to the description of the Calculus of Constructions which is given in [**?**]. In [**?**], T-terms are the ones which start with $[\prod; x]$, $[Prop]$ (analog of our $\mathcal{U}_0$) or $[Proof]$ (analog of our $[El]$) and o-terms are the ones which start with $[x]$, $[\lambda; x]$, $[App]$ (the analog of our $[ev; x]$ - see remark below) or $[\forall; x]$. The later is an analog of our $[forall_{0,0}; x]$ and $[forall_{1,0}; x]$ which in the case of CC coincide due to the impredicativity of *Prop*. For the same reason the first node of $[\forall; x]$ is a T-term which is analogous to $[El](o_1)$ in $[forall_{M,0}](o_1, o_2)$ with $M = 0, 1$.

It should be easy to describe a subset of TS0-terms which can be directly mapped to CC-terms while preserving all the main constructions.

Remark 3.1.6 In our description the evaluation node $[ev; x]$ is a quantifier i.e. it bounds a variable. This is due to the fact that we want to include the target type of function being evaluated as the third branch of the evaluation expression and this type may include, in dependent cases, an additional variable. This allows us to define the typing function on all terms before we proceed to the notions of definitional equality and derivable.

It might also be convenient to have our third argument of evaluation present on the level of implementations as an implicit argument. While, due to the expected properties of normalization in the system, it can always be inferred up do definitional equality from other arguments, this inference may require computation which can be avoided by providing an explicit value of this argument.

We let $TS0$ denote the set of TS0-terms and $TT0$ and $oT0$ denote the subsets of T-terms and o-terms.

According to the general rule stated above, variable x appearing under the nodes $[\prod, x]$, $[ev; x]$, $[\lambda, x]$ and $[forall; x]$ is called bound. Variables which are not bound are called free. We will write $TS0(Fu, FV, Fv)$, $TT0(Fu, FV, Fv)$ and $oT0(Fu, FV, Fv)$ for the sets of terms, T-terms and o-terms respectively with u-level variables from the set Fu, free T-variables from the set FV and free o-variables from the set Fv.

Lemma 3.1.7 *[**l02**] One has:*

1. *Any branch of a TS0-term is a TS0-term or a u-level expression,*
2. *For $E \in TS0(Fu, FV, Fv)$ (resp. $TT0(Fu, FV, Fv)$, $oT0(Fu, FV, Fv)$), $X \in FV$ and $o \in oT0(Fu, FV, Fv)$ one has $E[o/x] \in TS0(Fu, FV, Fv)$ (resp. $E[o/x] \in TT0(Fu, FV, Fv)$, $E[o/x] \in oT0(Fu, FV, Fv)$),*
3. *For $E \in TS0(Fu, FV, Fv)$ (resp. $TT0(Fu, FV, Fv)$, $oT0(Fu, FV, Fv)$), $x \in Fv$ and $o \in oT0(Fu, FV, Fv)$ one has $E[o/x] \in TS0(Fu, FV, Fv)$ (resp. $E[o/x] \in TT0(Fu, FV, Fv)$, $E[o/x] \in oT0(Fu, FV, Fv)$).*

Proof: Straightforward.

The correctness of the following definition follows from Lemma 3.1.7(3).

Definition 3.1.8 *[**dtau0**] For any finite set Fv and any function $\Gamma : Fv \to TT0(Fu, FV, Fv)$ we define the typing function $\tau = \tau_\Gamma : oT0(Fu, FV, Fv) \to TT0(Fu, FV, Fv)$ relative to Γ inductively (by induction on the number of nodes in the term) as follows:*

1. *For* $x \in Fv$, $\tau([x]) = \Gamma(x)$,

2. $\tau([u_M]) = \mathcal{U}_{M+1}$,

3. $\tau([j_{M_1,M_2}]) = [\prod; x](\mathcal{U}_{M_1}, \mathcal{U}_{M_2})$,

4. $\tau([ev; x](o_1, o_2, T)) = T[o_2/x]$,

5. $\tau([\lambda; x](T, o)) = [\prod; x](T, \tau_\Delta(o))$ *where* Δ *is the function on on* $Fv \amalg \{x\}$ *which equals* Γ *on* Fv *and* T *on* x.

6. $\tau([forall_{M_1,M_2}; x](o_1, o_2)) = \mathcal{U}_{max(M_1,M_2)}$.

Proposition 3.1.9 *[***tausub0***] Let* $\Gamma : Fv \to TT0(Fu, FV, Fv)$ *be a function,* $o, s \in oT0(Fu, FV, Fv)$ *and* $x \in Fv$. *Assume that the following conditions hold:*

1. s *does not depend on* x,

2. $\Gamma(x)$ *does not depend on* x,

3. $\tau_\Gamma(s) = \Gamma(x)$.

Then one has

$$(\tau_\Gamma(o))[s/x] = \tau_{\Gamma[s/x]}(o[s/x])$$

where $\Gamma[s/x]$ *is the function* $x' \mapsto \Gamma(x')[s/x]$.

Proof: Let us proceed by induction on the depth of o. We have to consider the various options for the structure of o listed in Definition 3.1.8.

1. $o = [x']$ for $x' \in Fv$. We have two sub-cases

   (a) if $x' \neq x$ we have

   $$(\tau_\Gamma(o))[s/x] = (\tau_\Gamma([x']))[s/x] = \tau_{\Gamma[s/x]}([x']) = \tau_{\Gamma[s/x]}(o[s/x])$$

   where the middle equality holds by definition of $\Gamma[s/x]$,

   (b) if $x' = x$ we have

   $$(\tau_\Gamma(o))[s/x] = (\tau_\Gamma([x])[s/x] = (\Gamma(x))[s/x] = \Gamma(x)$$

   where the last equality holds by condition (2) of the proposition and

   $$\tau_{\Gamma[s/x]}(o[s/x]) = \tau_{\Gamma[s/x]}(s) = \tau_\Gamma(s)$$

   where the last equality holds by condition (1) of the proposition. Together with condition (3) these equalities imply the assertion of the proposition for $o = [x]$.

2. $o = [u_M]$. Then $\tau_\Gamma(o)[s/x] = \mathcal{U}_{M+1} = \tau_{\Gamma[s/x]}(o[s/x])$.

3. $o = [j_{M_1,M_2}]$. Then $\tau_\Gamma(o)[s/x] = [\prod; y](\mathcal{U}_{M_1}, \mathcal{U}_{M_2}) = \tau_{\Gamma[s/x]}(o[s/x])$.

4. $o = [ev; y](o_1, o_2, T)$. Then

$$\tau_\Gamma(o)[s/x] = (T[o_2/y])[s/x]$$

and

$$\tau_{\Gamma[s/x]}(o[s/x]) = (T[s/x])[(o_2[s/x])/y]$$

Since s does not depend on y these two expressions are equal by Lemma 1.0.1.

5. $o = [\lambda; y](T, o')$. We have

$$(\tau_\Gamma(o))[s/x] = ([\prod; y](T, \tau_\Delta(o')))[s/x] = [\prod; y](T[s/x], \tau_\Delta(o')[s/x])$$

and

$$\tau_{\Gamma[s/x]}(o[s/x]) = \tau_{\Gamma[s/x]}([\lambda; y](T[s/x], o'[s/x])) = [\prod; y](T[s/x], \tau_{\Delta'}(o'[s/x]))$$

where Δ' equals $\Gamma[s/x]$ on Fv and $T[s/x]$ on y. Therefore, $\Delta' = \Delta[s/x]$. The conditions (1)-(3) still hold for Δ and the depth of o' is strictly less than the depth of o. Therefore by the inductive assumption $\tau_\Delta(o')[s/x] = \tau_{\Delta[s/x]}(o'[s/x])$.

6. $o = [forall_{M_1,M_2}; y](o_1, o_2)$. We have $\tau_\Gamma(o)[s/x] = \mathcal{U}_{max(M_1,M_2)}$ and

$$\tau_{\Gamma[s/x]}(o[s/x]) = \tau_{\Gamma[s/x]}([forall_{M_1,M_2}; y](o_1[s/x], o_2[s/x])) = \mathcal{U}_{max(M_1,M_2)}.$$

3.2 Equivalence relations $\equiv_\mathcal{A}$ and $\sim_\mathcal{A}$ on TS0-terms

[**sim0**] Let $UC = (Fu, \mathcal{A})$ be a universe context and FV a sequence of T-variables. In this section we define two simple equivalence relations on TS0-terms with u-variables from Fu and T-variables from FV which depend on $\mathcal{A}$. Recall that we always consider expressions up to the α-equivalence (renaming of bound variables).

For $E \in TS0(Fu, FV, Fv)$ and an element $\vec{n} \in \mathbf{N}^{Fu}$ we write $E[\vec{n}]$ for the TS0-term obtained by evaluating all of the u-level expressions in E on $\vec{n}$.

Definition 3.2.1 *[**dequivA0**] Define an equivalence relation $\equiv_\mathcal{A}$ on $TS0(Fu, FV, Fv)$ for all Fv by the condition that $E_1 \equiv_\mathcal{A} E_2$ iff for all $\vec{n} \in \mathcal{A}$ one has $E_1[\vec{n}] = E_2[\vec{n}]$.*

Lemma 3.2.2 *[**lequivA0**] The equivalence relation $\equiv_\mathcal{A}$ is decidable.*

Proof: It follows immediately from Lemma 2.0.3.

Note that $E_1 \equiv_\emptyset E_2$ iff E_1 and E_2 differ only in their u-level subexpressions.

Lemma 3.2.3 *[**equivsub01**] In the notations used above let $E, E' \in TS0(Fu, FV, Fv)$ and $E \equiv_{\mathcal{A}} E'$ then one has:*

1. *for $x \in Fv$ and $s \in oT0(Fu, FV, Fv)$ one has $E[s/x] \equiv_{\mathcal{A}} E'[s/x]$,*
2. *for $X \in FV$ and $T \in TT0(Fu, FV, Fv)$ one has $E[T/X] \equiv_{\mathcal{A}} E'[T/X]$.*

Proof: Straightforward by induction on the number of nodes in E and E'.

Lemma 3.2.4 *[**equivsub02**] Let E be an term of the form $E = [L](B_1, \dots, B_n)$. Suppose further that for each $i = 1, \dots, n$ one has $B_i \equiv_{\mathcal{A}} B'_i$. Then $E \equiv_{\mathcal{A}} [L](B'_1, \dots, B'_n)$.*

Proof: Straightforward.

Lemma 3.2.5 *[**redsub03**] In the notations used above let $E \in TS0(Fu, FV, Fv)$ then one has:*

1. *Let $s, s' \in oT0(Fu, FV, Fv)$ such that $s \equiv_{\mathcal{A}} s'$ and $x \in Fv$, then $E[s/x] \equiv_{\mathcal{A}} E[s'/x]$.*
2. *Let $T, T' \in TT0(Fu, FV, Fv)$ such that $T \equiv_{\mathcal{A}} T'$ and $X \in FV$, then $E[T/X] \equiv_{\mathcal{A}} E[T'/x]$.*

Proof: Straightforward.

Definition 3.2.6 *[**dsimA0**] Define an equivalence relation $\sim_{\mathcal{A}}$ on $TS0(Fu, FV, Fv)$ for all Fv by the condition that $E \sim_{\mathcal{A}} E'$ iff $Ess(E) \equiv_{\mathcal{A}} (')$*

Lemma 3.2.7 *[**Beqdec0**] The equivalence relation $\sim_{\mathcal{A}}$ is decidable.*

Proof: Follows easily from Lemma 3.2.2.

Lemma 3.2.8 *[**simsub01**] In the notations used above let $E, E' \in TS0(Fu, FV, Fv)$ and $E \sim_{\mathcal{A}} E'$ then one has:*

1. *for $x \in Fv$ and $s \in oT0(Fu, FV, Fv)$ one has $E[s/x] \sim_{\mathcal{A}} E'[s/x]$,*
2. *for $X \in FV$ and $T \in TT0(Fu, FV, Fv)$ one has $E[T/X] \sim_{\mathcal{A}} E'[T/X]$.*

Proof: Straightforward.

Lemma 3.2.9 *[**simsub02**] Let E be an term of the form $E = [L](B_1, \dots, B_n)$. Suppose further that for each $i = 1, \dots, n$ one has $B_i \sim_{\mathcal{A}} B'_i$. Then $E \sim_{\mathcal{A}} [L](B'_1, \dots, B'_n)$.*

Proof: Straightforward.

Lemma 3.2.10 *[***simsub03***] In the notations used above let $E \in TS0(Fu, FV, Fv)$ then one has:*

1. *Let $s, s' \in oT0(Fu, FV, Fv)$ such that $s \sim_{\mathcal{A}} s'$ and $x \in Fv$, then $E[s/x] \sim_{\mathcal{A}} E[s'/x]$.*
2. *Let $T, T' \in TT0(Fu, FV, Fv)$ such that $T \sim_{\mathcal{A}} T'$ and $X \in FV$, then $E[T/X] \sim_{\mathcal{A}} E[T'/x]$.*

Proof: Straightforward.

3.3 Derivation trees

Notes:

Definition 3.3.1 *[***cuttingsrface***] For a rooted tree E a "cutting surface" S is a set of vertices such that the path from each leave of the tree to the root passes through exactly one vertex in S.*

For example the sets of all leaves or the set consisting only of the root are cutting surfaces.

Definition 3.3.2 *[***csdepth***] A depth of a rooted tree E relative to a cutting surface S is the maximal distance (number of edges one has to cross) from elements of S to the root of the tree.*

For example the depth of a tree relative to $S = \{root\}$ is 0 and the depth relative to the set of all leaves is the depth of the tree.

Lemma 3.3.3 *[***surface***] Let S be any subset of vertices of a rooted tree E such that the path from any leaf to the root passes through at least one element of S. Let further S_0 be the subset of S which is defined by the condition that $v \in S_0$ if an only if $v \in S$ and the path from v to the root does not contain any other elements of S. Then S_0 is a cutting surface.*

Proof: For any leaf l of E let $S(l)$ be the set of elements of S which lie on the path from l to the root. Then $S(l) \cap S_0$ consists of exactly one element, namely the element in $S(l)$ which is closest to the root.

Given two cutting surfaces S_1, S_2 we say that $S_1 \geq S_2$ if the path from any element of S_1 to the root contains an element of S_2. We will write $inf(S_1, S_2)$ for the cutting surface constructed according to Lemma 3.3.3 from the set of vertices $S_1 \cup S_2$. Note that $inf(S_1, S_2)$ is indeed the greatest lower bound of the set $\{S_1, S_2\}$.

1. Each derivation tree is rooted and each branch of a derivation tree is a derivation tree.
2. Each derivation tree defines a derivable sentence. In particular there are five kinds of derivation trees - the ones which define four kinds of sentences and the ones which define u-level expressions.
3. Each vertex of a derivation tree is labelled by the (number or name of) the corresponding derivation rule. The kind of the branch corresponding to a given vertex is completely determined by the label of the vertex.
4. Each derivation rule defines the following data structure:
   (a) the number of premises (each premise being a sentence),
   (b) the kind of each of the premises,
   (c) a "pattern matching" condition which the premises should satisfy.
5. The correctness of a derivation tree is determined by the condition that at every vertex the number and kinds of branches are the ones which are prescribed by the label and that the corresponding pattern matching condition holds.

3.4 Derivation rules of TS0

As was mentioned in the introduction the systems which we described are not type systems but families of type systems parametrized by universe contexts and sequences of T-variables.

We are going to describe now the derivation rules for sentences of the type system $TS0_{UC,FV}$ associated with a given universe context $UC = (Fu, \mathcal{A})$ and a sequence of T-variables FV. The description is given in the usual type-theoretic notation. It is assumed that new names of variables which appear in the derivation rules do not conflict with the previously existing ones. Everywhere below we use the letter M with or without diacritics to denote a u-level expression in variables from Fu. The letter Γ with or without diacritics is used to denote a derivable context. By $l(\Gamma)$ we denote the "length" of Γ i.e. $l(x_1 : T_1, \ldots, x_n : T_n) = n$.

According to [?] a type system over a system of expressions is given by sets of *derivable* sentences of four kinds. In the sequent notation the sentences of each of the four kinds have the form

$$\Gamma \rhd$$

$$\Gamma \vdash o : T$$

$$\Gamma \vdash T = T'$$

$$\Gamma \vdash o = o' : T$$

Derivable sentences are those which can be obtained by the derivation rules given below. Since precise derivation trees for sentences are going to be important to us it is convenient to have relatively small number of different derivation rules. Normally, one would include

$$\frac{\Gamma \rhd}{\Gamma, x : \mathcal{U}_M \rhd}$$

as a family of derivation rules one for each well-formed u-level expression M. This is however inconvenient in particular because different u-level expressions can be equivalent and we do not want to deal with equivalence relations on the set of the derivation rules. Because of that we include in our consideration a fifth class of sentences which we write simply as M with the meaning of such a sentence being that M is a well-formed u-level expression in variables from Fu. From the abstract point of view it means that our four classes of derivable sentences are actually defined by an inductive procedure which also involves a fifth class but later the fifth class is not considered as a part of the resulting type system.

Note that even with this addition we actually have infinitely many derivation rules due to the issue with the "rule"

$$\frac{\Gamma, x : T, \Gamma' \rhd}{\Gamma, x : T, \Gamma' \vdash x : T}$$

which is actually a family of infinitely many derivation rules parametrized by natural numbers i with $i = l(\Gamma)$ being a condition of the rule.

1. for each $u \in Fu$
$$\frac{}{u}$$

2. $$\frac{M}{M+1}$$

3. $$\frac{M_1 \quad M_2}{max(M_1, M_2)}$$

4. $$\frac{}{\rhd}$$

5. for each $X \in FV$,
$$\frac{\Gamma \rhd}{\Gamma, x : X \rhd}$$

6. for each $i \in \mathbf{N}$
$$\frac{\Gamma, x : T, \Gamma' \rhd \quad \textit{where } l(\Gamma) = i}{\Gamma, x : T, \Gamma' \vdash x : T}$$

7. $$\frac{\Gamma, x : T \rhd \quad \Gamma, x : T' \rhd \quad T \sim_A T'}{\Gamma \vdash T \stackrel{d}{=} T'}$$

8. $$\frac{\Gamma \vdash T_1 \stackrel{d}{=} T_2}{\Gamma \vdash T_2 \stackrel{d}{=} T_1}$$

9. $$\frac{\Gamma \vdash T_1 \stackrel{d}{=} T_2 \quad \Gamma \vdash T_2 \stackrel{d}{=} T_3}{\Gamma \vdash T_1 \stackrel{d}{=} T_3}$$

10.

$$\frac{\Gamma \vdash o : T \quad \Gamma \vdash o' : T \quad o \sim_A o'}{\Gamma \vdash o \stackrel{d}{=} o' : T}$$

11.

$$\frac{\Gamma \vdash o_1 \stackrel{d}{=} o_2 : T}{\Gamma \vdash o_2 \stackrel{d}{=} o_1 : T}$$

12.

$$\frac{\Gamma \vdash o_1 \stackrel{d}{=} o_2 : T \quad \Gamma \vdash o_2 \stackrel{d}{=} o_3 : T}{\Gamma \vdash o_1 \stackrel{d}{=} o_3 : T}$$

13.

$$\frac{\Gamma \vdash o : T \quad \Gamma \vdash T \stackrel{d}{=} T'}{\Gamma \vdash o : T'}$$

14.

$$\frac{\Gamma \vdash o \stackrel{d}{=} o' : T \quad \Gamma \vdash T \stackrel{d}{=} T'}{\Gamma \vdash o \stackrel{d}{=} o' : T'}$$

15.

$$\frac{\Gamma \rhd \quad M}{\Gamma, x : \mathcal{U}_M \rhd}$$

16.

$$\frac{\Gamma \rhd \quad M}{\Gamma \vdash u_M : \mathcal{U}_{M+1}}$$

17.

$$\frac{\Gamma \vdash o : \mathcal{U}_M}{\Gamma, x : [El](o) \rhd}$$

18.

$$\frac{\Gamma \rhd \quad M}{\Gamma \vdash [El](u_M) \stackrel{d}{=} \mathcal{U}_M}$$

19.

$$\frac{\Gamma \vdash o \stackrel{d}{=} o' : \mathcal{U}_M}{\Gamma \vdash [El](o) \stackrel{d}{=} [El](o')}$$

20.

$$\frac{\Gamma \vdash o : \mathcal{U}_M \quad \Gamma \vdash o' : \mathcal{U}_M \quad \Gamma \vdash [El](o) \stackrel{d}{=} [El](o')}{\Gamma \vdash o \stackrel{d}{=} o' : \mathcal{U}_M}$$

21.

$$\frac{\Gamma, x : T_1, y : T_2 \rhd}{\Gamma, y : [\prod; x](T_1, T_2) \rhd}$$

22.

$$\frac{\Gamma \vdash T_1 \stackrel{d}{=} T_1' \quad \Gamma, x : T_1 \vdash T_2 \stackrel{d}{=} T_2'}{\Gamma \vdash [\prod; x](T_1, T_2) \stackrel{d}{=} [\prod; x](T_1', T_2')}$$

23.

$$\frac{\Gamma, x : T_1 \vdash o : T_2}{\Gamma \vdash [\lambda; x](T_1, o) : [\prod; x](T_1, T_2)}$$

24.

$$\frac{\Gamma \vdash T_1 \stackrel{d}{=} T_1' \quad \Gamma, x : T_1 \vdash o \stackrel{d}{=} o' : T_2}{\Gamma \vdash [\lambda; x](T_1, o) \stackrel{d}{=} [\lambda; x](T_1', o') : [\prod; x](T_1, T_2)}$$

25.

$$\frac{\Gamma \vdash f : [\prod; x](T_1, T_2) \quad \Gamma \vdash o : T_1}{\Gamma \vdash [ev; x](f, o, T_2) : T_2[o/x]}$$

26.

$$\frac{\begin{array}{ll} \Gamma \vdash T_1 \stackrel{d}{=} T_1' & \Gamma \vdash f \stackrel{d}{=} f' : [\prod; x](T_1, T_2) \\ \Gamma, x : T_1 \vdash T_2 \stackrel{d}{=} T_2' & \Gamma \vdash o \stackrel{d}{=} o' : T_1 \end{array}}{\Gamma \vdash [ev; x](f, o, T_2) \stackrel{d}{=} [ev; x](f', o', T_2') : T_2[o/x]}$$

27.

$$\frac{\Gamma \vdash o_1 : T_1 \quad \Gamma, x : T_1 \vdash o_2 : T_2}{\Gamma \vdash [ev; y]([\lambda; x](T_1, o_2), o_1, T_2) \stackrel{d}{=} o_2[o_1/x] : T_2[o_1/x]}$$

28.

$$\frac{\Gamma \vdash f : [\prod; x](T_1, T_2)}{\Gamma \vdash [\lambda; x](T_1, [ev; y](f, x, T_2[y/x])) \stackrel{d}{=} f : [\prod; x](T_1, T_2)}$$

29.

$$\frac{\Gamma \rhd \quad M_1 \quad M_2 \quad M_1 \leq_{\mathcal{A}} M_2}{\Gamma \vdash j_{M_1,M_2} : [\prod; x](\mathcal{U}_{M_1}, \mathcal{U}_{M_2})}$$

30.

$$\frac{\Gamma \vdash o : \mathcal{U}_{M_1} \quad M_2 \quad M_1 \leq_{\mathcal{A}} M_2}{\Gamma \vdash [El][ev; x](j_{M_1,M_2}, o, \mathcal{U}_{M_2}) \stackrel{d}{=} [El](o)}$$

31.

$$\frac{\Gamma \vdash o_1 : \mathcal{U}_{M_1} \quad \Gamma, x : [El](o_1) \vdash o_2 : \mathcal{U}_{M_2}}{\Gamma \vdash [forall_{M_1,M_2}; x](o_1, o_2) : \mathcal{U}_{max(M_1,M_2)}}$$

32.

$$\frac{\Gamma \vdash o_1 : \mathcal{U}_{M_1} \quad \Gamma, x : [El](o_1) \vdash o_2 : \mathcal{U}_{M_2}}{\Gamma \vdash [El][forall_{M_1,M_2}; x](o_1, o_2) \stackrel{d}{=} [\prod; x]([El](o_1), [El](o_2))}$$

3.5 TS0(UC,FV) are type systems

[**ts0aretypesystems**] Given a system of expressions such as the one given by TS0 terms relative to a given universe context $UC = (Fu, \mathcal{A})$ and a sequence of T-variables FV one defines a type system over this system of expressions as four classes of sentences of the forms $\Gamma\rhd$, $\Gamma \vdash o : T$, $\Gamma \vdash T = T'$ and $\Gamma \vdash o = o' : T$ which satisfy a number of technical conditions listed in [**?**]. In this section we will show that these conditions hold for the classes of derivable sentences of TS0.

In what follows we fix a universe context UC and a sequence of T-variables FV and consider all the notions relative to this context and this sequence.

Lemma 3.5.1 *[***dertree***] Any derivation tree for sequent of one of the following forms:*

$$\Gamma, \Gamma'\rhd$$

$$\Gamma, \Gamma' \vdash o : T$$

$$\Gamma, \Gamma' \vdash T = T'$$

$$\Gamma, \Gamma' \vdash o = o' : T$$

has (a unique) smallest cutting surface whose elements represent contexts of the form $\Gamma\rhd$ *and u-level expressions. There is at least one element representing* $\Gamma\rhd$ *in this cutting surface.*

Proof: It is easy to see from the shape of the derivation rules that there can be no cutting surface for any of the four main kinds of sentences which contains only u-level expressions. Therefore the second assertion of the lemma is automatically satisfied.

Since the the greatest lower bound of any two cutting surfaces is defined and is contained (as a subset) in the union of these surfaces, it is sufficient to show that that in each of the four cases for any derivation tree there exists at least one cutting surface satisfying the conditions of the lemma.

We proceed by induction on the depth of the derivation tree.

Looking at the derivation rules we see that each of the premises for any derivation rule for a context of the form $\Gamma, \Gamma'\rhd$ where Γ' is non-empty has either the same form or of the form $\Gamma, \Gamma' \vdash o : T$ or equals $\Gamma\rhd$.

Each of the premises for any derivation rule for a judgement of the form $\Gamma, \Gamma' \vdash o : T$ is either of the same form or of the form $\Gamma, \Gamma' \vdash T = T'$, or of the form $\Gamma, \Gamma'\rhd$ where Γ' is non-empty or equals $\Gamma\rhd$.

Each of the premises for any derivation rule for a judgement of the form $\Gamma, \Gamma' \vdash T = T'$ is either of the same form or of the form $\Gamma, \Gamma' \vdash o : T$, or of the form $\Gamma, \Gamma' \vdash o = o' : T$, or of the form $\Gamma, \Gamma'\rhd$ where Γ' is non-empty or equals $\Gamma\rhd$.

Each of the premises for any derivation rule for a judgement of the form $\Gamma, \Gamma' \vdash o : T$ is either of the same form or of the form $\Gamma, \Gamma' \vdash T = T'$, or of the form $\Gamma, \Gamma' \vdash o = o' : T$, or of the form $\Gamma, \Gamma'\rhd$ where Γ' is non-empty or equals $\Gamma\rhd$.

Combining these properties of our derivation rules with the induction on the depth of the derivation tree we obtain the assertion of the lemma.

Remark 3.5.2 Note that the assertion of Lemma 3.5.1 is not tautological and really depends on the form of the derivation rules which one chooses in the definition of a type system. For example, if we included the rule

$$\frac{\Gamma \rhd}{\Gamma, x : X \vdash x : X}$$

for $X \in FV$ into our list of the generating derivation rules then Lemma 3.5.1 would become false. Indeed then one would have a derivation tree for $x : X \vdash x : X$ which has only one edge terminating in the empty context $\rhd$ and in particular no vertices corresponding to the context $x : X \rhd$.

Definition 3.5.3 *[**derdepthgamma**] For a sentence of one of the forms listed in Lemma 3.5.1 we define its derivation depth relative to* Γ *as the minimum over all its derivation trees of the depth of this tree relative to the cutting surface defined in Lemma 3.5.1.*

Note that the derivation depth relative to Γ is 0 only for $\Gamma \rhd$. Note also that for any sentence of the form considered in Lemma 3.5.1 whose derivation depth relative to Γ is greater than 0 there exists a derivation rule which generates this sentence such that all its premises are again of the same form and their derivation depth relative to Γ is strictly less than of the original sentence.

As an immediate corollary of Lemma 3.5.1 we get the following result.

Lemma 3.5.4 *[**dertree0**] For any derivable sequent of one of the following forms*

$$\Gamma, \Gamma' \rhd$$

$$\Gamma, \Gamma' \vdash o : T$$

$$\Gamma, \Gamma' \vdash T = T'$$

$$\Gamma, \Gamma' \vdash o = o' : T$$

the sequent $\Gamma \rhd$ *is derivable.*

Lemma 3.5.5 *[**dertree1**] One has the following derivation rules:*

1.

$$\frac{\Gamma, x_1 : T_1 \rhd \qquad \Gamma, \Gamma' \rhd}{\Gamma, x_1 : T_1, \Gamma' \rhd}$$

2.

$$\frac{\Gamma, x_1 : T_1 \rhd \qquad \Gamma, \Gamma' \vdash o : T}{\Gamma, x_1 : T_1, \Gamma' \vdash o : T}$$

3.

$$\frac{\Gamma, x_1 : T_1 \rhd \qquad \Gamma, \Gamma' \vdash T = T'}{\Gamma, x_1 : T_1, \Gamma' \vdash T = T'}$$

4.

$$\frac{\Gamma, x_1 : T_1 \rhd \qquad \Gamma, \Gamma' \vdash o = o' : T}{\Gamma, x_1 : T_1, \Gamma' \vdash o = o' : T}$$

Proof: We prove that for any derivable sentence St of one of the four main kinds which starts with Γ the sentence obtained from St by replacing Γ with $\Gamma, x : T$ is derivable if $\Gamma, x : T \rhd$ itself is derivable. We proceed by induction on the derivation depth of S relative to Γ. We now have to consider case by case each of the generating derivation rule families which produce a sentence of a main kind. Since $\Gamma, x : T \rhd$ is derivable we may assume that Γ' is nonempty or that S of the three later kinds. The verification is straightforward.

Remark 3.5.6 The key to the validity of the proof of Lemma 3.5.5 is that for any of the derivation rules one of the following possibilities holds:

1. the product sentence of the derivation rule is of the three later kinds and changing its context part Γ with $\Gamma, x : T$ both in the product and in the premises again produces a derivation rule,

2. the product sentence is of the first kind i.e. of the form $\Gamma \rhd$ where $\Gamma = \Gamma_1, \Gamma_2$ with $l(\Gamma_2) \leq 1$ and replacing Γ_1 with $\Gamma_1, x : T$ both in the product and in the premises again produces a derivation rule.

These conditions would not hold if we had a generating derivation rule with the product of the form $\Gamma_1, \Gamma_2 \rhd$ where $l(\Gamma_2) > 1$ and Γ_2 does not directly appear in the premise e.g. a rule such as

$$\frac{\Gamma \rhd \qquad M}{\Gamma, x : \mathcal{U}_M, o : [El](x) \rhd}$$

or if we had a generating derivation rule with the product of one of the three later kinds of the form $\Gamma_0, \Gamma_1 \vdash \mathcal{J}$ where Γ_1 is nonempty and does not directly appear in the premises e.g. a rule such as

$$\frac{\Gamma \vdash f : [\prod; x](T_1, T_2)}{\Gamma, y : T_1 \vdash [ev; x](f, y, T_2) : T_2[y/x]}$$

Lemma 3.5.7 *[***dertree2***] One has the following derivation rules:*

1.

$$\frac{\Gamma, \Gamma'' \rhd \qquad \Gamma, \Gamma' \rhd}{\Gamma, \Gamma', \Gamma'' \rhd}$$

2.

$$\frac{\Gamma, \Gamma'' \rhd \qquad \Gamma, \Gamma' \vdash a : T}{\Gamma, \Gamma', \Gamma'' \vdash a : T}$$

3.

$$\frac{\Gamma, \Gamma'' \rhd \quad \Gamma, \Gamma' \vdash T = T'}{\Gamma, \Gamma', \Gamma'' \vdash T = T'}$$

4.

$$\frac{\Gamma, \Gamma'' \rhd \quad \Gamma, \Gamma' \vdash o = o' : T}{\Gamma, \Gamma', \Gamma'' \vdash o = o' : T}$$

Proof: By induction on the length of Γ'' using Lemma 3.5.5.

Lemma 3.5.8 *[***dertree3***] One has the following derivation rules:*

$$\frac{\Gamma \vdash a : S \quad \Gamma, x : S, \Gamma' \rhd}{\Gamma, \Gamma'[a/x] \rhd}$$

$$\frac{\Gamma \vdash a : S \quad \Gamma, x : S, \Gamma' \vdash o : T}{\Gamma, \Gamma'[a/x] \vdash o[a/x] : T[a/x]}$$

$$\frac{\Gamma \vdash a : S \quad \Gamma, x : S, \Gamma' \vdash T = T'}{\Gamma, \Gamma'[a/x] \vdash T[a/x] = T'[a/x]}$$

$$\frac{\Gamma \vdash a : S \quad \Gamma, x : S, \Gamma' \vdash o = o' : T}{\Gamma, \Gamma'[a/x] \vdash o[a/x] = o'[a/x] : T[a/x]}$$

Proof: ???

Lemma 3.5.9 *[***dertree4***] One has the following derivation rules:*

$$\frac{\Gamma \vdash o : T}{\Gamma, x : T \rhd}$$

$$\frac{\Gamma \vdash T = T'}{\Gamma, x : T \rhd}$$

$$\frac{\Gamma \vdash o = o' : T}{\Gamma \vdash o : T}$$

Proof: ???

Lemma 3.5.10 *[***dertree5***] One has the following derivation rules:*

1.

$$\frac{\Gamma \vdash T = T' \quad \Gamma, x : T, \Gamma' \rhd}{\Gamma, x : T', \Gamma' \rhd}$$

2.

$$\frac{\Gamma \vdash T = T' \quad \Gamma, x : T, \Gamma' \vdash o : S}{\Gamma, x : T', \Gamma' \vdash o : S}$$

3.

$$\frac{\Gamma\vdash T=T' \quad \Gamma,x:T,\Gamma'\vdash S=S'}{\Gamma,x:T',\Gamma'\vdash S=S'}$$

4.

$$\frac{\Gamma\vdash T=T' \quad \Gamma,x:T,\Gamma'\vdash o=o':S}{\Gamma,x:T',\Gamma'\vdash o=o':S}$$

Proof: ??? By the induction on the derivation depth of $\Gamma,x:T,\Gamma'\rhd$ and $\Gamma,x:T,\Gamma'\vdash a:T''$ relative to $\Gamma,x:T\rhd$. If the derivation depth is 0 then we are in the first case and the assertion is obvious. For the inductive step consider the derivation rules one by one. There are three cases to be considered for the first reduction rule. The other reduction rules are straightforward.

Lemma 3.5.11 *[***dertree6***] One has the following derivation rule:*

$$\frac{\Gamma\vdash T=T' \quad \Gamma\vdash o=o':T}{\Gamma\vdash o=o':T'}$$

Proof: ???

Lemma 3.5.12 *[***dertree7***] One has the following derivation rules:*

1.

$$\frac{\Gamma\vdash a=a':S \quad \Gamma,x:S,\Gamma',y:T\rhd}{\Gamma,\Gamma'[a/x]\vdash T[a/x]=T[a'/x]}$$

2.

$$\frac{\Gamma\vdash a=a':S \quad \Gamma,x:S,\Gamma'\vdash o:T}{\Gamma,\Gamma'[a/x]\vdash o[a/x]=o[a'/x]:T[a/x]}$$

Proof: ???

Recall from [**?**] that subsets in the sets of all contexts and judgements in a given system of expressions form a type system i.e. define a contextual subcategory in the contextual category defined by the underlying system of expressions iff they satisfy the following conditions:

$$\frac{}{\rhd}$$

$$\frac{x_1:E_1,\ldots,x_n:E_n\rhd}{x_1:E_1,\ldots,x_{n-i}:E_{n-i}\rhd}\ (i\le n) \qquad \frac{x_1:E_1,\ldots,x_n:E_n\vdash t:T}{x_1:E_1,\ldots,x_n:E_n,x:T\rhd}\ (n\ge 0)$$

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

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

$$i = 0, \ldots, n-1$$

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

Combining the lemmas proved above we get the following result.

Theorem 3.5.13 *[***mainth1***] For any given universe context UC and a sequence of T-variables FV the derivable sentences form a type system.*

Lemma 3.5.14 *[***lm1***] Let $\Gamma, x : [El](o) \rhd$ be a derivable context. Then there exists a u-level expression M such that $\Gamma \vdash o : \mathcal{U}_M$ is derivable.*

Proof: There is a unique derivation rule which produces contexts of the form $\Gamma, x : [El](o) \rhd$ and the premise of this rule is of the form $\Gamma \vdash o : \mathcal{U}_M$.

Theorem 3.5.15 *[***th1***] Let $\Gamma, x : T \rhd$ be a derivable context. Then there is a derivable context of the form $\Gamma, x : [El](o) \rhd$ such that $[El](o) \succeq_{\mathcal{A}} T$ and all the reductions involved are from the following list: Elforall, Eltotal, Elcoprod, Elpaths.*

Proof: By induction on the derivation depth of $\Gamma, x : T \rhd$ relative to Γ. Since T is a T-term it has one of the following forms $[El](o)$, $[\prod; x](T_1, T_2)$, $[\sum; x](T_1, T_2)$, $[\amalg](T_1, T_2)$, $[Id](T, o_1, o_2)$. If $T = [El](o)$ we take $T' = T$.

If $T = [\prod; x](T_1, T_2)$ then $\Gamma, x : T_1, y : T_2 \rhd$ is derivable as the premise of the only derivation rule which generates $[\prod; x]$-terms. By Lemma 12.6.3(1), $\Gamma, x : T_1 \rhd$ is also derivable. By induction there are derivable contexts $\Gamma, x : T_1, y : [El](o_2) \rhd$ and $\Gamma, x : [El](o_1) \rhd$ such that $[El](o_1) \succ_{\mathcal{A}} T_1$ and $[El](o_2) \succ_{\mathcal{A}} T_2$ through reductions from the list in the conditions of the theorem. We then take $o = [forall_{M_1,M_2}; x](o_1, o_2)$ and apply Lemma **??**.

The cases of $[\sum; x]$ and $[\amalg](T_1, T_2)$ are similar to the case of $[\prod; x]$. Finally the case of $[Id]$ is similar with the use of Lemma 12.6.3(2) instead of Lemma 12.6.3(1).

4 Abbreviations, conventions and definitions

[**abb0**] In order to make our definitions more readable we introduce the following conventions. We write them here in the context of TS0 terms but they will be applied in the same way to the terms of the other systems considered in this paper.

1. Let $E \in TS0(Fu, FV, Fv)$ where $Fu(u_1, \ldots, u_k)$, $FV = (X_1, \ldots, X_m)$. Let $Fu' = (u'_1, \ldots, u'_{k'})$, $FV' = (X'_1, \ldots, X'_{m'})$ and $Fv' = (x'_1, \ldots, x'_n)$. Let $M_1, \ldots, M_k$ be u-level expressions in variables from Fu' and $T_1, \ldots, T_m \in TT(Fu', FV', Fv)$. We will then write
$$E\, M_1 \ldots M_k\, T_1 \ldots T_m$$
for
$$E' = E[M_1/u_1, \ldots, M_k/u_k, T_1/X_1, \ldots, T_m/X_m]$$
considered as an element of $TS0(Fu', FV', Fv)$.

2. We will write $\prod x : T_1, T_2$ for $[\prod; x](T_1, T_2)$ and $\lambda x : T_1, o_1$ for $[\lambda; x](T_1, o_1)$. When Y does not depend on x we will write $X \to Y$ for $\prod x : X, Y$. We will also write $\prod(x_1 : T_1) \ldots (x_n : T_n), T$ for
$$\prod x_1 : T_1, \prod x_2 : T_2, \ldots, \prod x_n : T_n, T$$
and $X_1 \to \ldots \to X_n$ for $X_1 \to (X_2 \to \ldots (X_{n-1} \to X_n) \ldots)$.

3. We will write $*t$ for $[El](t)$

4. Let Γ be a context. We will write $f a$ for $[ev; x](f, a, T_2)$ if $\tau_\Gamma(f) = \prod x : T_1, T_2$. Note that $=$ here means the equality of expressions (up to the α-equivalence). In particular T_2 is well defined (again up to the α-equivalence) by this condition as the second branch of $\tau_\Gamma(f)$.

5. We will write:
$$Def. \quad D(u_1, \ldots, u_k; \mathcal{A})(X_1, \ldots, X_m : Type)(x_1 : T_1) \ldots (x_n : T_n) := T$$
where D is a name (identifier) and $T \in TT0(Fu, \{X_1, \ldots, X_m\}, \{x_1, \ldots, x_n\})$ to signify the assertion that the context
$$x_1 : T_1, \ldots, x_n : T_n, x : T \triangleright$$
is derivable in TS0 relative to the universe context $(u_1, \ldots, u_k; \mathcal{A})$ and the sequence of T-variables $(X_1, \ldots, X_m)$ and that we will use D as a notation for the term T.

   ???Proofs of such assertions should be provided where necessary wince they will form the basis for our annotation language.

6. We will write:
$$Def. \quad D(u_1, \ldots, u_k; \mathcal{A})(X_1, \ldots, X_m : Type)(x_1 : T_1) \ldots (x_n : T_n) := o : T$$
where D is a name (identifier) and
$$T \in TT0(\{u_1, \ldots, u_k\}, \{X_1, \ldots, X_m\}, \{x_1, \ldots, x_n\})$$
$$o \in TT0(\{u_1, \ldots, u_k\}, \{X_1, \ldots, X_m\}, \{x_1, \ldots, x_n\})$$

to signify the assertion that the judgement

$$x_1 : T_1, \ldots, x_n : T_n \vdash o : T$$

is derivable in TS0 relative to the universe context $(u_1, \ldots, u_k; \mathcal{A})$ and the sequence of T-variables $(X_1, \ldots, X_m)$ and that we will use D as the notation for the term $\lambda(x_1 : T_1) \ldots (x_n : T_n)o$.

??? Note on proofs of such assertions in the annotation language.

Similarly we will write

$$Def. \quad D(u_1, \ldots, u_k; \mathcal{A})(X_1, \ldots, X_m : Type)(x_1 : T_1) \ldots (x_n : T_n) := o$$

for

$$Def. \quad D(u_1, \ldots, u_k; \mathcal{A})(X_1, \ldots, X_m : Type)(x_1 : T_1) \ldots (x_n : T_n) := o : \tau_\Gamma(o)$$

where $\Gamma = (x_1 : T_1, \ldots, x_n : T_n)$.

7. We will write

$$Th. \quad D(u_1, \ldots, u_k; \mathcal{A})(X_1, \ldots, X_m : Type)(x_1 : T_1) \ldots (x_n : T_n) : T$$

where D is a name (identifier) and

$$T \in TT0(Fu, \{X_1, \ldots, X_m\}, \{x_1, \ldots, x_n\})$$

$$o \in TT0(Fu, \{X_1, \ldots, X_m\}, \{x_1, \ldots, x_n\})$$

To signify that there is a derivable judgement of the form

$$x_1 : T_1, \ldots, x_n : T_n \vdash o : T$$

in TS0 relative to to the universe context $(u_1, \ldots, u_k; \mathcal{A})$ and the sequence of T-variables $(X_1, \ldots, X_m)$ and that we will use D as the notation for o.

???? Unlike $Def.$ which is verified mechanically $Th.$ requires a proof i.e. an actual construction of o. We will nevertheless use this notation below without providing an explicit form of o in some cases when we know that such an o can be constructed from existing Coq proofs but writing it explicitly would take to much space.

8. We will use $\{\ldots\}$ instead of $(\ldots)$ for arguments of definitions which can be uniquely reconstructed under the assumption that the definition produces a derivable context or judgement. Such arguments need not be specified in definition "applications". We will also use obvious abbreviations of the notations introduced above in cases when k, m or n equals 0 or when $\mathcal{A}$ is an empty set of conditions.

4.1 Reducibility relation on TS0 terms

Definition 4.1.1 *[**drd0**] Let $UC = (Fu, \mathcal{A})$ be universe context and FV a sequence of T-variables. Define a relation $\succ_{\mathcal{A}}$ on $TS0(Fu, FV, Fv)$ for each Fv as the transitive closure of the union of the following reduction relations:*

1. *Elu-reduction. An essential subterm of the form $[El][u_M]$ reduces to $\mathcal{U}_M$.*
2. *Elj-reduction. An essential subterm of the form $[El][ev;y]([j_{M_1,M_2}], o, T)$ reduces to $[El](o)$,*
3. *Elforall-reduction. An essential subterm of the form $[El][forall_{M_1,M_2};x](o_1,o_2)$ reduces to $[\prod;x]([El](o_1),[El](o_2))$,*
4. *jMM-reduction. An essential subterm of the form $[j_{M_1,M_2}]$ such that $M_1 \equiv_{\mathcal{A}} M_2$, reduces to $[\lambda;x](\mathcal{U}_{M_1},[x])$ where x is a new, relative to the whole term, name of a variable,*
5. *beta-reduction. An essential subterm of the form $[ev;z]([\lambda;x](T_1,o_1),o_2,T_2)$ reduces to $o_1[o_2/x]$,*
6. *jj-reduction. An essential subterm of the form $[ev;z]([j_{M_2',M_3}],[ev;y]([j_{M_1,M_2}],o_1,T_1),T_2)$ such that $M_2' \equiv_{\mathcal{A}} M_2$ reduces to $[ev;z]([j_{M_1,M_3}],o_1,T_2)$,*
7. *eta-reduction. An essential subterm of the form $[\lambda;x](X,[ev;y](f,[x],Y))$ where f does not depend on x reduces to f.*
8. *forallj-reductions:*
   1. *(a) forallja-reduction. An essential subterm of the form*

      $$[forall_{M_1,M_2};x]([ev;z]([j_{M_0,M_1'}],o_1,T),o_2)$$

      such that $M_1' \equiv_{\mathcal{A}} M_1$, reduces to

      $$[ev;z]([j_{max(M_0,M_2),max(M_1,M_2)}],[forall_{M_0,M_2};x](o_1,o_2),\mathcal{U}_{max(M_1,M_2)})$$

   2. *(b) foralljb-reduction. An essential subterm of the form*

      $$[forall_{M_1,M_3};x](o_1,[ev;z]([j_{M_2,M_3'}],o_2,T))$$

      such that $M_3' \equiv_{\mathcal{A}} M_3$ reduces to

      $$[ev;z]([j_{max(M_1,M_2),max(M_1,M_3)}],[forall_{M_1,M_2};x](o_1,o_2),\mathcal{U}_{max(M_1,M_3)})$$

Remark 4.1.2 One verifies easily that a reduction of a TS0-term is again a TS0-term and that reductions preserve the subsets of T-terms and o-terms.

Let $\succeq_{\mathcal{A}}$ be the transitive and reflexive closure of $\succ_{\mathcal{A}}$.

Lemma 4.1.3 *[**redsub01**] In the notations used above let $E, E' \in TS0(Fu, FV, Fv)$ and $E \succ_{\mathcal{A}} E'$ then one has:*

1. *for $x \in Fv$ and $s \in oT0(Fu, FV, Fv)$ one has $E[s/x] \succ_{\mathcal{A}} E'[s/x]$,*
2. *for $X \in FV$ and $T \in TT0(Fu, FV, Fv)$ one has $E[T/X] \succ_{\mathcal{A}} E'[T/X]$.*

Proof: It is sufficient to consider one step reductions. Of all reductions of Definition 4.1.1 the only non-trivial case is that of the beta-reduction. It follows from Lemma 1.0.1.

Lemma 4.1.4 *[**redsub02**] Let E be an term of the form $E = [L](B_1, \dots, B_n)$ and suppose that B_i are essential if and only if $i \in I$. Suppose further that for each $i \in I$ one has $B_i \succeq_{\mathcal{A}} B'_i$ and let $B'_i = B_i$ for $i \notin I$. Then $E \succeq_{\mathcal{A}} [L](B'_1, \dots, B'_n)$.*

Proof: Straightforward.

In the following lemma we let $E[s'/x]_{ess}[s/x]_{ness}$ denote the expression obtained by the substitution of s' for all essential occurrences of x and s for all non-essential occurrences of x and use the notation for the substitutions of T-variables.

Lemma 4.1.5 *[**redsub03**] In the notations used above let $E \in TS0(Fu, FV, Fv)$ then one has:*

1. *Let $s, s' \in oT0(Fu, FV, Fv)$ such that $s \succ_{\mathcal{A}} s'$ and $x \in Fv$, then*
$$E[s/x] \succeq_{\mathcal{A}} E[s'/x]_{ess}[s/x]_{ness}$$
2. *Let $T, T' \in TT0(Fu, FV, Fv)$ such that $T \succ_{\mathcal{A}} T'$ and $X \in FV$, then*
$$E[T/X] \succeq_{\mathcal{A}} E[T'/x]_{ess}[T/x]_{ness}$$

Proof: Straightforward by induction on the number of nodes in E using Lemma 4.1.4.

Lemma 4.1.6 *[**lequivsimred0**] In the notations used above let $E \succeq_{\mathcal{A}} E'$ and $F \equiv_{\mathcal{A}} E$ (resp. $F \sim_{\mathcal{A}} E$). Then there is an term F' such that $F \succeq_{\mathcal{A}} F'$ and $F' \equiv_{\mathcal{A}} E'$ (resp. $F' \sim_{\mathcal{A}} E'$).*

Proof: It follows easily from the fact that all our reducibility conditions are invariant under both $\equiv_{\mathcal{A}}$ and $\sim_{\mathcal{A}}$.

Remark 4.1.7 Note that the analog of Lemma 4.1.6 with $F' \equiv_{\mathcal{A}} E'$ does not hold. Indeed, in the case of β-reduction the substitution may produce a term with many copies of the same u-level expression and replacing some but not all of these copies by an equivalent u-level expression may produce F' which can not be obtained by reduction of any F.

4.2 Local confluence for general terms of TS0

The following theorem asserts "local" confluence for one step reductions except for one special case. The confluence in this special case is expected to hold for derivable terms (see below).

Theorem 4.2.1 *[**lc0**] Let $UC = (Fu, \mathcal{A})$ be a universe context and FV a sequence of T-variables. Let $E \in TS0(Fu, FV, Fv)$. Let further $E \succ_{\mathcal{A}} E_1$ and $E \succ_{\mathcal{A}} E_2$ be two one-step reductions. Then there exists a TS0-terms E_1', E_2' from $TS0(Fu, FV, Fv)$ such that $E_1 \succeq_{\mathcal{A}} E_1'$, $E_2 \succeq_{\mathcal{A}} E_2'$ and $E_1' \sim_{\mathcal{A}} E_2'$ unless E_1 is obtained by eta-reduction of a subterm of the form $[\lambda; x](T_1, [ev; y]([\lambda; z](T_2, o), [x], T_3))$ to $[\lambda; z](T_2, o)$ and E_2 is obtained by beta-reduction of this subterm at the node $[ev; y]$ to $[\lambda; x](T_1, o[x/z])$. In this cases confluence is expected for derivable terms (see below).*

Proof: It is sufficient to consider the case when the first reduction is relative to subterm $S = E$ i.e. occurs at the root.

1. Elu-reduction at the root. $E = [El][u_M]$. Then no other reductions may occur in E and therefore there are no confluence issues.

2. Elj-reduction at the root. $E = [El][ev; y]([j_{M_1,M_2}], o, T)$ and

$$E_1 = [El](o).$$

   If the second reduction occurs inside o or T then the statement is obvious. It remains to consider possible second reductions occurring at the "exposed" nodes $[El]$, $[ev; y]$ and $[j_{M_1,M_2}]$. This leads to the following actual cases:

   (a) jMM-reduction at $[j_{M_1,M_2}]$. Then $M_1 \equiv_{\mathcal{A}} M_2$,

$$E_1 = [El](o) = E_1'$$

$$E_2 = [El][ev; y]([\lambda; x](\mathcal{U}_{M_1}, [x]), o, T) \succ [El](o) = E_2'$$

   (b) jj-reduction at $[ev; y]$. Then $o = [ev; z]([j_{M_0,M_1'}], o', T')$ where $M_1' \equiv_{\mathcal{A}} M_1$ and

$$E_1 = [El][ev; z]([j_{M_0,M_1'}], o', T') \succ [El](o') = E_1'$$

$$E_2 = [El][ev; y]([j_{M_0,M_2}], o', T) \succ [El](o') = E_2'$$

3. Elforall-reduction at the root. $E = [El][forall_{M_1,M_2}; x](o_1, o_2)$ and

$$E_1 = [\prod; x]([El](o_1), [El](o_2)).$$

   Exposed nodes where second reduction may occur are $[El]$ and $[forall_{M_1,M_2}; x]$ with the following actual cases:

(a) forallja-reduction at $[forall_{M_1,M_2};x]$. Then $o_1 = [ev;z]([j_{M_0,M'_1}], o'_1, T')$ where $M'_1 \equiv_{\mathcal{A}} M_1$ and

$$E_1 = [\prod;x]([El][ev;z]([j_{M_0,M'_1}], o'_1, T'), [El](o_2)) \succ$$

$$[\prod;x]([El](o'_1), [El](o_2)) = E'_1$$

$$E_2 = [El][ev;z]([j_{max(M_0,M_2),max(M_1,M_2)}], [forall_{M_0,M_2};x](o'_1, o_2), \mathcal{U}_{max(M_1,M_2)}) \succ$$

$$[El][forall_{M_0,M_2};x](o'_1, o_2) \succ [\prod;x]([El](o'_1), [El](o_2)) = E'_2$$

(b) foralljb-reduction at $[forall_{M_1,M_2};x]$. Then $o_2 = [ev;z](j_{M_3,M'_2}, o'_2, T')$ where $M'_2 \equiv_{\mathcal{A}} M_2$ and

$$E_1 = [\prod;x]([El](o_1), [El][ev;z](j_{M_3,M'_2}, o'_2, T')) \succ$$

$$[\prod;x]([El](o_1), [El](o'_2)) = E'_1$$

$$E_2 = [El][ev;z]([j_{max(M_1,M_3),max(M_1,M_2)}], [forall_{M_1,M_3};x](o_1, o'_2), \mathcal{U}_{max(M_1,M_2)}) \succ$$

$$[El][forall_{M_1,M_3};x](o_1, o'_2) \succ [\prod;x]([El](o_1), [El](o'_2)) = E'_2$$

4. jMM-reduction at the root. $E = [j_{M_1,M_2}]$ such that $M_1 \equiv_{\mathcal{A}} M_2$ and

$$E_1 = [\lambda;x](\mathcal{U}_{M_1}, [x])$$

There are exposed nodes where second reduction may occur.

5. beta-reduction at the root. $E = [ev;z]([\lambda;x](T_1, o_1), o_2, T_2)$ and

$$E_1 = o_1[o_2/x].$$

Exposed nodes where second reduction may occur are $[ev;z]$ and $[\lambda;x]$ with the only actual case of the eta-reduction at $[\lambda;x](X,Y)$. Then $Y = [ev;y](o,[x],T)$ where o does not depend on x and

$$E_1 = [ev;y](o, o_2, T[o_2/x]) = E'_1$$

$$E_2 = [ev;z](o, o_2, T_2) = E'_2$$

6. jj-reduction at the root. $E = [ev;z]([j_{M'_2,M_3}], [ev;y]([j_{M_1,M_2}], o, T), T')$ such that $M'_2 \equiv_{\mathcal{A}} M_2$ and

$$E_1 = [ev;z]([j_{M_1,M_3}], o, T')$$

Exposed nodes where second reduction may occur are $[ev;z]$, $[j_{M'_2,M_3}]$, $[ev;y]$ and $[j_{M_1,M_2}]$ with the following actual cases:

(a) jMM-reduction at $[j_{M'_2,M_3}]$. Then $M'_2 \equiv_{\mathcal{A}} M_3$ and

$$E_1 = [ev;z]([j_{M_1,M_3}], o, T') = E'_1$$

$$E_2 = [ev;z]([\lambda;x](\mathcal{U}_{M'_2}, [x]), [ev;y]([j_{M_1,M_2}], o, T), T') \succ$$

$$[ev;y]([j_{M_1,M_2}], o, T) = E'_2$$

(b) jj-reduction at $[ev;y]$. Then $o = [ev;t]([j_{M_0,M_1'}], o', T)$ where $M_1' \equiv_{\mathcal{A}} M_1$ and

$$E_1 = [ev;z]([j_{M_1,M_3}], [ev;t]([j_{M_0,M_1'}], o', T), T') \succ$$
$$[ev;z]([j_{M_0,M_3}], o', T') = E_1'$$
$$E_2 = [ev;z]([j_{M_2',M_3}], [ev;y]([j_{M_0,M_2}], o', T), T') \succ$$
$$[ev;z]([j_{M_0,M_3}], o', T') = E_2'$$

(c) jMM-reduction at $[j_{M_1,M_2}]$. Then $M_1 \equiv_{\mathcal{A}} M_2$ and

$$E_1 = [ev;z]([j_{M_1,M_3}], o, T) = E_1'$$
$$E_2 = [ev;z]([j_{M_2',M_3}], [ev;y]([\lambda;x](\mathcal{U}_{M_1}, [x]), o, T'), T) \succ$$
$$[ev;z]([j_{M_2',M_3}], o, T) = E_2'$$

7. eta-reduction at the root. $E = [\lambda;x](T_1, [ev;y](o, [x], T_2))$ such that o does not depend on x and

$$E_1 = o$$

Exposed nodes where second reduction may occur are $[\lambda;x]$ and $[ev;y]$ with the only actual case being beta-reduction at $[ev;y]$. This is the exceptional case from the condition of the theorem.

8. forallj-reductions at the root:

(a) forallja-reduction at the root. $E = [forall_{M_1,M_2};x]([ev;z]([j_{M_0,M_1'}], o_1, T), o_2)$ such that $M_1' \equiv_{\mathcal{A}} M_1$ and

$$E_1 = [ev;z]([j_{max(M_0,M_2),max(M_1,M_2)}], [forall_{M_0,M_2};x](o_1, o_2), \mathcal{U}_{max(M_1,M_2)})$$

Exposed nodes where second reduction may occur are $[forall_{M_1,M_2};x]$, $[ev;z]$ and $[j_{M_0,M_1'}]$ with the following actual cases:

i. foralljb-reduction at $[forall_{M_1,M_2};x]$. Then $o_2 = [ev;z']([j_{M_3,M_2'}], o_2', T')$ where $M_2' \equiv_{\mathcal{A}} M_2$ and

$$E_1 = [ev;z]([j_{max(M_0,M_2),max(M_1,M_2)}],$$
$$[forall_{M_0,M_2};x](o_1, [ev;z']([j_{M_3,M_2'}], o_2', T')), \mathcal{U}_{max(M_1,M_2)}) \succ$$
$$[ev;z]([j_{max(M_0,M_2),max(M_1,M_2)}],$$
$$[ev;z']([j_{max(M_0,M_3),max(M_0,M_2')}], [forall_{M_0,M_3};x](o_1, o_2'), \mathcal{U}_{max(M_0,M_2')}),$$
$$\mathcal{U}_{max(M_1,M_2)}) \succ$$
$$[ev;z]([j_{max(M_0,M_3),max(M_1,M_2)}], [forall_{M_0,M_3};x](o_1, o_2'), \mathcal{U}_{max(M_1,M_2)}) = E_1'$$
$$E_2 = [ev;z']([j_{max(M_1,M_3),max(M_1,M_2)}],$$
$$[forall_{M_1,M_3};x]([ev;z]([j_{M_0,M_1'}], o_1, T), o_2'), \mathcal{U}_{max(M_1,M_2)}) \succ$$
$$[ev;z']([j_{max(M_1,M_3),max(M_1,M_2)}],$$
$$[ev;z]([j_{max(M_0,M_3),max(M_1,M_3)}], [forall_{M_0,M_3};x](o_1, o_2'), \mathcal{U}_{max(M_1,M_3)}),$$
$$\mathcal{U}_{max(M_1,M_2)}) \succ$$
$$[ev;z']([j_{max(M_0,M_3),max(M_1,M_2)}], [forall_{M_0,M_3};x](o_1, o_2'), \mathcal{U}_{max(M_1,M_2)}) = E_2'$$

ii. jj-reduction at $[ev;z]$. Then $o_1 = [ev;z']([j_{M_3,M_0'}], o_1', T')$ where $M_0' \equiv_{\mathcal{A}} M_0$ and

$$E_1 = [ev;z]([j_{max(M_0,M_2),max(M_1,M_2)}],$$
$$[forall_{M_0,M_2};x]([ev;z']([j_{M_3,M_0'}], o_1', T'), o_2), \mathcal{U}_{max(M_1,M_2)}) \succ$$
$$[ev;z]([j_{max(M_0,M_2),max(M_1,M_2)}],$$
$$[ev;z']([j_{max(M_3,M_2),max(M_0,M_2)}], [forall_{M_3,M_2};x](o_1', o_2), \mathcal{U}_{max(M_0,M_2)}),$$
$$\mathcal{U}_{max(M_1,M_2)}) \succ$$
$$[ev;z]([j_{max(M_3,M_2),max(M_1,M_2)}], [forall_{M_3,M_2};x](o_1', o_2), \mathcal{U}_{max(M_1,M_2)}) = E_1'$$
$$E_2 = [forall_{M_1,M_2};x]([ev;z]([j_{M_3,M_1'}], o_1', T), o_2) \succ$$
$$[ev;z]([j_{max(M_3,M_2),max(M_1,M_2)}], [forall_{M_3,M_2};x](o_1', o_2), \mathcal{U}_{max(M_1,M_2)}) = E_2'$$

iii. jMM-reduction at $[j_{M_0,M_1'}]$. Then $M_0 \equiv_{\mathcal{A}} M_1'$ and

$$E_1 = [ev;z]([j_{max(M_0,M_2),max(M_1,M_2)}], [forall_{M_0,M_2};x](o_1, o_2), \mathcal{U}_{max(M_1,M_2)}) \succ$$
$$[ev;z]([\lambda;x](\mathcal{U}_{max(M_0,M_2)}, [x]), [forall_{M_0,M_2};x](o_1, o_2), \mathcal{U}_{max(M_1,M_2)}) \succ$$
$$[forall_{M_0,M_2};x](o_1, o_2) = E_1'$$
$$E_2 = [forall_{M_1,M_2};x]([ev;x]([\lambda;x](\mathcal{U}_{M_0}, [x]), o_1, T), o_2) \succ$$
$$[forall_{M_1,M_2};x](o_1, o_2) = E_2'$$

(b) foralljb-reduction at the root. $E = [forall_{M_1,M_3};x](o_1, [ev;z]([j_{M_2,M_3'}], o_2, T))$ such that $M_3' \equiv_{\mathcal{A}} M_3$ and

$$E_1 = [ev;z]([j_{max(M_1,M_2),max(M_1,M_3)}], [forall_{M_1,M_2};x](o_1, o_2), \mathcal{U}_{max(M_1,M_3)}),$$

Exposed nodes where second reduction may occur are $[forall_{M_1,M_3};x]$, $[ev;z]$ and $[j_{M_2,M_3''}]$ with the following actual cases:

i. forallja-reduction at $[forall_{M_1,M_3};x]$. Then $o_1 = [ev;z']([j_{M_0,M_1'}], o_1', T')$ where $M_1' \equiv_{\mathcal{A}} M_1$ and

$$E_1 = [ev;z]([j_{max(M_1,M_2),max(M_1,M_3)}],$$
$$[forall_{M_1,M_2};x]([ev;z']([j_{M_0,M_1'}], o_1', T'), o_2), \mathcal{U}_{max(M_1,M_3)}) \succ$$
$$[ev;z]([j_{max(M_1,M_2),max(M_1,M_3)}],$$
$$[ev;z']([j_{max(M_0,M_2),max(M_1,M_2)}], [forall_{M_0,M_2};x](o_1', o_2), \mathcal{U}_{max(M_0,M_2)}),$$
$$\mathcal{U}_{max(M_1,M_3)}) \succ$$
$$[ev;z]([j_{max(M_0,M_2),max(M_1,M_3)}], [forall_{M_0,M_2};x](o_1', o_2), \mathcal{U}_{max(M_1,M_3)}) = E_1'$$
$$E_2 = [ev;z']([j_{max(M_0,M_3),max(M_1,M_3)}],$$
$$[forall_{M_0,M_3};x](o_1', [ev;z]([j_{M_2,M_3'}], o_2, T), \mathcal{U}_{max(M_1,M_3)}) \succ$$
$$[ev;z']([j_{max(M_0,M_3),max(M_1,M_3)}],$$
$$[ev;z]([j_{max(M_0,M_2),max(M_0,M_3)}], [forall_{M_0,M_2};x](o_1', o_2), \mathcal{U}_{max(M_0,M_3)}),$$
$$\mathcal{U}_{max(M_1,M_3)}) \succ$$
$$[ev;z']([j_{max(M_0,M_2),max(M_1,M_3)}], [forall_{M_0,M_2};x](o_1', o_2), \mathcal{U}_{max(M_1,M_3)}) = E_2'$$

ii. jj-reduction at $[ev; z]$. Then $o_2 = [ev; z']([j_{M_0,M_2'}], o_2', T')$ where $M_2' \equiv_{\mathcal{A}} M_2$ and

$$E_1 = [ev; z]([j_{max(M_1,M_2),max(M_1,M_3)}],$$
$$[forall_{M_1,M_2}; x](o_1, [ev; z']([j_{M_0,M_2'}], o_2', T')), \mathcal{U}_{max(M_1,M_3)}) \succ$$
$$[ev; z]([j_{max(M_1,M_2),max(M_1,M_3)}],$$
$$[ev; z']([j_{max(M_1,M_0),max(M_1,M_2)}], [forall_{M_1,M_0}; x](o_1, o_2'), \mathcal{U}_{max(M_1,M_2)}),$$
$$\mathcal{U}_{max(M_1,M_3)}) \succ$$
$$[ev; z]([j_{max(M_1,M_0),max(M_1,M_3)}], [forall_{M_1,M_0}; x](o_1, o_2'), \mathcal{U}_{max(M_1,M_3)}) = E_1'$$
$$E_2 = [forall_{M_1,M_3}; x](o_1, [ev; z]([j_{M_0,M_3'}], o_1', T)) \succ$$
$$[ev; z]([j_{max(M_1,M_0),max(M_1,M_3)}], [forall_{M_1,M_0}; x](o_1, o_2'), \mathcal{U}_{max(M_1,M_3)}) = E_2'$$

iii. jMM-reduction at $[j_{M_2,M_3'}]$. Then $M_2 \equiv_{\mathcal{A}} M_3'$ and

$$E_1 = [ev; z]([j_{max(M_1,M_2),max(M_1,M_3)}], [forall_{M_1,M_2}; x](o_1, o_2), \mathcal{U}_{max(M_1,M_3)}) \succ$$
$$[ev; z]([\lambda; x](\mathcal{U}_{max(M_1,M_2)}, [x]), [forall_{M_1,M_2}; x](o_1, o_2), \mathcal{U}_{max(M_1,M_3)}) \succ$$
$$[forall_{M_1,M_2}; x](o_1, o_2) = E_1'$$
$$E_2 = [forall_{M_1,M_3}; x](o_1, [ev; x]([\lambda; x](\mathcal{U}_{M_2}, [x]), o_2, T)) \succ$$
$$[forall_{M_1,M_3}; x](o_1, o_2) = E_2'$$

5 Adding the unit pt - system TS2

5.1 TS2 terms and typing function

Definition 5.1.1 *[**d21**] The following labels are permitted in the expressions of TS2 - the labels permitted in TS1, Pt, pt and tt.*

The notions of u-level expressions, T- and o- terms in TS2 are defined as follows:

Definition 5.1.2 *[**d22**]*

1. *expressions with the root node caring a TS1-label is an u-level expression, o-expression or a T-expression according to the rules of Definition 6.1.3,*
2. *expressions with the root of the form $[Pt]$ are T-expressions,*
3. *expressions with the root of the form $[pt]$ or $[tt]$ are o-expressions.*

Definition 5.1.3 *[**d23**] A TS2-term is a TS2-expression such that:*

1. *any node caring one of the TS1-labels satisfies the conditions of Definition 6.1.3,*
2. *any node of the form $[Pt]$ has valency 0,*
3. *any node of the form $[pt]$ has valency 0,*
4. *any node of the form $[tt]$ has valency 0.*

Definition 5.1.4 *[**d24**] A node in TS2 term is called non-essential if it satisfies the conditions of Definition 6.1.4. The same applies to the definition of essential nodes, essential and non-essential subexpressions and of $Ess(E)$ is extended to TS2-terms in the obvious way.*

We let $TS2$ denote the set of TS2-terms and $TT2$ and $oT2$ denote the subsets of T-terms and o-terms. The obvious analog of Lemma 3.1.7 holds for TS1-expressions. We extend to TS2 the abbreviations introduced for TS1-expressions.

Definition 5.1.5 *[**dtau2**] Under the assumptions of Definition 3.1.8 we extend the typing function τ_Γ to o-terms of TS2 as follows:*

1. *the value of τ_Γ on an o-term whose root node carries a TS1-label is computed according to the rules of Definition 6.1.5.*
2. $\tau_\Gamma([pt]) = \mathcal{U}_0$,
3. $\tau_\Gamma([tt]) = Pt$.

We have the obvious analog of Proposition 3.1.9 for TS2.

5.2 Equivalence relations $\equiv_{\mathcal{A}}$ and $\sim_{\mathcal{A}}$ on TS2-terms

The relations $\equiv_{\mathcal{A}}$ and $\sim_{\mathcal{A}}$ are extended to TS2-terms in the obvious way. We also have obvious analogs of all the statements of Section 3.2.

5.3 Derivation rules of TS2

The derivation rules for contexts and judgements of TS2 are the derivation rules for TS1 together with the following additional ones:

$$\frac{\Gamma\rhd}{\Gamma, x : Pt\rhd}$$

$$\frac{\Gamma\rhd}{\Gamma\vdash tt : Pt}$$

$$\frac{\Gamma\vdash o : Pt}{\Gamma\vdash o = tt : Pt}$$

$$\frac{\Gamma\rhd}{\Gamma\vdash pt : \mathcal{U}_0}$$

$$\frac{\Gamma\rhd}{\Gamma\vdash [El](pt) = Pt}$$

5.4 Construction of the eliminator for Pt

$$\frac{\Gamma, x : Pt, y : T\rhd \qquad \Gamma\vdash o : T[tt/x]}{\Gamma\vdash [pt_r; x](o, T) : [\prod; x](Pt, T)}$$

Remark 5.4.1 [**unitpos**] The obvious idea to introduce pt not through the eliminators $[pt_{r,M}]$ but by the condition that Pt has one distinguished term tt and any other term $a \neq tt$ reduces to tt leads to a system without confluence since

$$[pair; x]([pr1; y](Pt, T, c), [pr2; y](Pt, c, T), T[x/y])$$

reduces both to c and to $[pair; x](tt, [pr2; y](Pt, c, T), T[x/y])$ and these two terms can not be reduced to a common term (cf. [**?**][Exercise p.88]). Of the other reductions which are supported by the univalent model namely:

$$[\prod; x](Pt, T) \succ_{\mathcal{A}} T \quad [\prod; x](T, Pt) \succ_{\mathcal{A}} Pt$$

$$[\sum; x](Pt, T) \succ_{\mathcal{A}} T \quad [\sum; x](T, Pt) \succ_{\mathcal{A}} T$$

the first two lead to derivable terms whose types are changed by β-reduction:

1. $o = [ev; x]([\lambda; y](Pt, o_1), o_2, T_2) \succ o_1[o_2/y] = o'$ where $o_1 : [\prod; x](T_1, T_2)$, $o_2 : T_1$ and $\tau_\Gamma(o) = T_2[o_2/x]$, $\tau_\Gamma(o') = [\prod; x](T_1, T_2)$.

2. $o = [ev;x]([\lambda;y](T_1,o_1),o_2,Pt) \succ o_1[o_2/y] = o'$ where $o_1 : T_2$, $o_2 : [\prod;y](T_1,T_2)$ and $\tau_\Gamma(o) = Pt$, $\tau_\Gamma(o') = T_2[o_2/y]$.

and the other two seem to lead to similar problems with sum-related reductions.

5.5 Reducibility relation on TS2 terms

Definition 5.5.1 *[***drd2***] Let $UC = (Fu, \mathcal{A})$ be universe context and FV a sequence of T-variables. Define a relation $\succ_{\mathcal{A}}$ on $TS2(Fu, FV, Fv)$ for any Fv as the transitive closure of the union of the following reduction relations:*

1. *Reductions of Definition 6.4.1.*

2. *iotapt-reduction. An essential subterm of the form $[ev;x]([pt_r;y](o,T),tt,T')$ where $T[x/y] \sim_{\mathcal{A}} T'$ reduces to o.*

The obvious analogs of Lemmas 4.1.3-4.1.6 hold for TS2.

5.6 Local confluence for general terms of TS2

The local confluence theorem for TS2 has exactly the same form as Theorem 6.5.1 and the same proof since there are no non-trivial confluence dependencies between the new reduction rule of Definition 5.5.1 and the reduction rules of Definition 6.4.1.

6 Adding dependent sums - system TS1

6.1 TS1 terms and typing function

Definition 6.1.1 *[***d11***] The following labels are permitted in the expressions of TS1 - the labels permitted in TS0, $(\sum;x)$, $(pair;x)$, $(pr1;x)$, $(pr2;x)$, $(total;x)$.*

The notions of u-level expressions, T- and o- terms in TS1 are defined as follows:

Definition 6.1.2 *[***d12***]*

1. *expressions with the root node caring a TS0-label is an u-level expression, o-expression or a T-expression according to the rules of Definition 3.1.2,*

2. *expressions with the root of the form $[\sum;x]$ are T-expressions,*

3. *expressions with the root node of the form $[pair;x]$, $[pr1;x]$, $[pr2;x]$ and $[total;x]$ are o-expressions.*

Definition 6.1.3 *[**d13**] A TS1-term is a TS1-expression such that:*

1. *any node caring one of the TS0-labels satisfies the conditions of Definition 3.1.3,*
2. *any node of the form $[\sum; x]$ has valency 2, both its branches are T-expressions and the first one does not contain $[x]$,*
3. *any node of the form $[pair; x]$ has valency 3, its first two branches are o-expressions which do not contain $[x]$ and the third branch a T-expression,*
4. *any node of the form $[pr1; x]$ has valency 3, its first branch is a T-expression which does not contain on $[x]$, its second branch a T-expression and the third branch an o-expression,*
5. *any node of the form $[pr2; x]$ has valency 3, its first branch is a T-expression which does not contain on $[x]$, its second branch a T-expression and the third branch an o-expression,*
6. *any node of the form $[total; x]$ has valency 4 and its first two branches are u-level expressions and the last two are o-expressions of which the first one does not contain $[x]$.*

Definition 6.1.4 *[**d14**] A node in TS1 term is called non-essential if it satisfies the conditions of Definition 3.1.4 or if it belongs to the subexpression T_1 of a subexpression of the form $[pr1; x](T_1, T_2, o)$ or if it belongs to subexpression T_1 of a subexpression of the form $[pr2; x](T_1, T_2, o)$. The definition of essential nodes, essential and non-essential subexpressions and of $Ess(E)$ is extended to TS1-terms in the obvious way.*

We let $TS1$ denote the set of TS1-terms and $TT1$ and $oT1$ denote the subsets of T-terms and o-terms. The obvious analog of Lemma 3.1.7 holds for TS1-expressions. We extend to TS1 the abbreviations introduced for TS0-expressions and abbreviate $[total; x](M_1, M_2, o_1, o_2)$ to $[total; x]_{M_1,M_2}(o_1, o_2)$ and $[\sum; x](T_1, T_2)$ to $\sum x : T_1, T_2$. We also write $T_1 \times T_2$ for $\sum x : T_1, T_2$ when T_2 does not depend on x and use the notation $\sum(x_1 : T_1) \ldots (x_n : T_n), T$ for $\sum x_1 : T_1, \sum x_2 : T_2, \ldots, \sum x : T_n, T$. We will write $[pair; x](T_2, o_1, o_2)$ for $[pair; x](\tau_\Gamma(o_1), T_2, o_1, o_2)$.

Definition 6.1.5 *[**dtau1**] Under the assumptions of Definition 3.1.8 we extend the typing function τ_Γ to o-terms of TS1 as follows:*

1. *the value of τ_Γ on an o-term whose root node carries a TS0-label is computed according to the rules of Definition 3.1.8.*
2. $\tau_\Gamma([pair; x](o_1, o_2, T)) = [\sum; x](\tau_\Gamma(o_1), T)$,
3. $\tau_\Gamma([pr1; x](T_1, T_2, o)) = T_1$,
4. $\tau_\Gamma([pr2; x](T_1, T_2, o)) = T_2[[pr1; x](T_1, T_2, o)/x]$,

5. $\tau_\Gamma([total;x]_{M_1,M_2}(o_1,o_2)) = \mathcal{U}_{max(M_1,M_2)}$.

Remark 6.1.6 As is seen from Definition 6.1.5 it is not necessary to include T_2 into the $pr1$ expressions and, therefore, not necessary to make $pr1$ into a quantifier in order to define τ_Γ.The reason for the inclusion of T_2 into $pr1$ lies in the proof of Theorem 6.5.1 below. Without this information it is impossible to define reductions on TS1-terms in a way which preserves (local) confluence. A particular case of the problem can be seen in the case of iotatotal2-reduction at the root combined with etatotal-reduction at *pair*. See proof of Theorem 6.5.1.

We have the obvious analog of Proposition 3.1.9 for TS1.

6.2 Equivalence relations $\equiv_\mathcal{A}$ and $\sim_\mathcal{A}$ on TS1-terms

[**sim1**] The relations $\equiv_\mathcal{A}$ and $\sim_\mathcal{A}$ are extended to TS1-terms in the obvious way. We also have obvious analogs of all the statements of Section 3.2.

6.3 Derivation rules of TS1

The derivation rules for contexts and judgements of TS1 are the derivation rules for TS0 together with the following additional ones:

$$\frac{\Gamma, x : T_1, y : T_2 \rhd}{\Gamma, y : [\sum; x](T_1, T_2)}$$

$$\frac{\Gamma, x : T_1, y : T_2 \rhd \qquad \Gamma \vdash o_1 : T_1 \quad \Gamma \vdash o_2 : T_2[o_1/x]}{\Gamma \vdash [pair; x](o_1, o_2, T_2) : [\sum; x](T_1, T_2)}$$

$$\frac{\Gamma \vdash a : [\sum; x](T_1, T_2)}{\Gamma \vdash [pr1; x](T_1, T_2, a) : T_1}$$

$$\frac{\Gamma \vdash a : [\sum; x](T_1, T_2)}{\Gamma \vdash [pr2; x](T_1, T_2, a) : T_2[[pr1; x](T_1, T_2, a)/x]}$$

$$\frac{\Gamma, x : T_1, y : T_2 \rhd \qquad \Gamma \vdash o_1 : T_1 \quad \Gamma \vdash o_2 : T_2[o_1/x]}{\Gamma \vdash [pr1; x'](T_1, T_2[x'/x], [pair; x](o_1, o_2, T_2)) = o_1 : T_1}$$

$$\frac{\Gamma, x : T_1, y : T_2 \rhd \qquad \Gamma \vdash o_1 : T_1 \quad \Gamma \vdash o_2 : T_2[o_1/x]}{\Gamma \vdash [pr2; x'](T_1, T_2[x'/x], [pair; x](o_1, o_2, T_2)) = o_2 : T_2[o_1/x]}$$

$$\frac{\Gamma \vdash a : [\sum; x](T_1, T_2)}{\Gamma \vdash [pair; x]([pr1; x'](T_1, T_2[x'/x], a), [pr2; x'](T_1, T_2[x'/x], a), T_2) = a : [\sum; x](T_1, T_2)}$$

$$\frac{\Gamma \vdash o_1 : \mathcal{U}_{M_1} \quad \Gamma, x : [El](o_1) \vdash o_2 : \mathcal{U}_{M_2}}{\Gamma \vdash [total_{M_1,M_2}; x](o_1, o_2) : \mathcal{U}_{max(M_1,M_2)}}$$

$$\frac{\Gamma \vdash o_1 : \mathcal{U}_{M_1} \quad \Gamma, x : [El](o_1) \vdash o_2 : \mathcal{U}_{M_2}}{\Gamma \vdash [El][total_{M_1,M_2}; x](o_1, o_2) = [\sum; x]([El](o_1), [El](o_2))}$$

$$\frac{\Gamma, x : Pt, y : T \rhd}{\Gamma \vdash [\sum; x](Pt, T) = T[tt/x]}$$

$$\frac{\Gamma, x : Pt \vdash o : \mathcal{U}_M}{\Gamma \vdash [total_{0,M}; x](pt, o) = o[tt/x] : \mathcal{U}_M}$$

$$\frac{\Gamma, x : T \rhd}{\Gamma \vdash [\sum; x](T, Pt) = T}$$

$$\frac{\Gamma \vdash o : \mathcal{U}_M}{\Gamma \vdash [total_{M,0}; x](o, pt) = o : \mathcal{U}_M}$$

$$\frac{\Gamma, x : Pt, y : T \rhd}{\Gamma \vdash [\prod; x](Pt, T) = T[tt/x]}$$

$$\frac{\Gamma, x : Pt \vdash o : \mathcal{U}_M}{\Gamma \vdash [forall_{0,M}; x](pt, o) = o[tt/x] : \mathcal{U}_M}$$

$$\frac{\Gamma, x : T \rhd}{\Gamma \vdash [\prod; x](T, Pt) = Pt}$$

$$\frac{\Gamma \vdash o : \mathcal{U}_M}{\Gamma \vdash [prod_{M,0}; x](o, pt) = pt : \mathcal{U}_M}$$

Remark 6.3.1 [**sumelim**] Note that our derivation and reduction rules for the dependent sum allow one to define "eliminators" which one would have if one introduced dependent sums as particular cases of inductive definitions as it is done in Coq. For $\Gamma, x : T_1, y : T_2 \rhd$ the eliminator for the corresponding dependent sum is a family of terms I_M of type (we use the standard abbreviation $\to$ for non-dependent product)

$$[\prod; P](([\sum; x](T_1, T_2) \to \mathcal{U}_M),$$

$$[\prod; s0]([\prod; s1](T_1, [\prod; s2](T_2, [ev; a](P, [pair; x](s1, s2, T_2), \mathcal{U}_M))),$$

$$[\prod; c]([\sum; x](T_1, T_2), [ev; a](P, c, \mathcal{U}_M))))$$

which one defines as

$$[\lambda; P](([\sum; x](T_1, T_2) \to \mathcal{U}_M),$$

$$[\lambda; s0]([\prod; s1](T_1, [\prod; s2](T_2, [ev; a](P, [pair; x](s1, s2, T_2), \mathcal{U}_M))),$$

$$[\lambda; c]([\sum; x](T_1, T_2), [ev; s2]([ev; s1](P, [pr1](c, T_1),_{)}, [pr2; x](T_1, c, T_2)), \_)))$$

which is well-typed since $[pair; x]([pr1](T_1, c), [pr2; x](T_1, c, T_2)) =_{\mathcal{A}} c$.

6.4 Reducibility relation on TS1 terms

[**red1**]

Definition 6.4.1 *[**drd1**] Let $UC = (Fu, \mathcal{A})$ be universe context and FV a sequence of T-variables. Define a relation $\succ_{\mathcal{A}}$ on $TS1(Fu, FV, Fv)$ for each Fv as the union of the transitive closures of the following reduction relations:*

1. *Reductions of Definition 4.1.1.*

2. *Eltotal-reduction. An essential subterm of the form $[El][total_{M_1,M_2}; x](o_1, o_2)$ reduces to $[\sum; x]([El](o_1), [El](o_2))$.*

3. *iotatotal1-reduction. An essential subterm of the form $[pr1; x](T_1, T_2, [pair; y](o_1, o_2, T_2'))$ such that $T_2[y/x] \sim_{\mathcal{A}} T_2'$ reduces to o_1.*

4. *iotatotal2-reduction. An essential subterm of the form $[pr2; x](T_1, T_2, [pair; y](o_1, o_2, T_2'))$ such that $T_2[y/x] \sim_{\mathcal{A}} T_2'$ reduces to o_2.*

5. *etatotal-reduction. An essential subterm of the form*

   $$[pair; x]([pr1; y](T_1', T_2', a'), [pr2; y](T_1'', T_2'', a''), T_2)$$

   such that $a' \sim_{\mathcal{A}} a''$ and $T_2'[x/y] \sim_{\mathcal{A}} T_2 \sim_{\mathcal{A}} T_2''[x/y]$ reduces to a'.

6. *totalj-reductions:*

   (a) *totalja-reduction. An essential subterm of the form*

   $$[total_{M_1,M_2}; x]([ev; z]([j_{M_0,M_1'}], o_1, T), o_2)$$

   such that $M_1' \equiv_{\mathcal{A}} M_1$, reduces to

   $$[ev; z]([j_{max(M_0,M_2),max(M_1,M_2)}], [total_{M_0,M_2}; x](o_1, o_2), \mathcal{U}_{max(M_1,M_2)})$$

   (b) *totaljb-reduction. An essential subterm of the form*

   $$[total_{M_1,M_3}; x](o_1, [ev; z]([j_{M_2,M_3'}], o_2, T))$$

   such that $M_3' \equiv_{\mathcal{A}} M_3$ reduces to

   $$[ev; z]([j_{max(M_1,M_2),max(M_1,M_3)}], [total_{M_1,M_2}; x](o_1, o_2), \mathcal{U}_{max(M_1,M_3)})$$

Remark 6.4.2 [**etatotal1**] The condition that $T_2'[x/y] \equiv_{\mathcal{A}} T_2$ is does not follow automatically for derivable terms. Indeed, if a term is derivable then one would expect that $T_2'[o_1/y]$ is convertible to $T_2[o_1/x]$ which does not imply that $T_2'[x/y]$ is convertible to T_2.

Let $\succeq_{\mathcal{A}}$ be the transitive and reflexive closure of $\succ_{\mathcal{A}}$. The obvious analogs of Lemmas 4.1.3-4.1.6 hold for TS1.

6.5 Local confluence for general terms of TS1

Theorem 6.5.1 *[***lc1***] Let $UC = (Fu, \mathcal{A})$ be a universe context and FV a sequence of T-variables. Let $E \in TS1(Fu, FV, Fv)$. Let further $E \succ_{\mathcal{A}} E_1$ and $E \succ_{\mathcal{A}} E_2$ be two one-step reductions. Then there exists a TS1-terms E_1', E_2' from $TS1(Fu, FV, Fv)$ such that $E_1 \succeq_{\mathcal{A}} E_1'$, $E_2 \succeq_{\mathcal{A}} E_2'$ and $E_1' \sim_{\mathcal{A}} E_2'$ unless the exceptional case of Theorem 4.2.1 occurs.*

Proof: It is sufficient to consider the case when the first reduction is relative to subterm $S = E$ i.e. occurs at the root. It is further sufficient to consider cases when one of the two reductions belongs to the list of new items in Definition 6.4.1.

1. Elj-reduction at the root. $E = [El][ev; y]([j_{M_1,M_2}], o, T)$ and
$$E_1 = [El](o).$$
No new exposed nodes where second reduction may occur compared to TS0.

2. Elforall-reduction at the root. $E = [El][forall_{M_1,M_2}; x](o_1, o_2)$ and
$$E_1 = [\prod; x]([El](o_1), [El](o_2)).$$
No new exposed nodes where second reduction may occur compared to TS0.

3. jMM-reduction at the root. $E = [j_{M_1,M_2}]$ such that $M_1 \equiv_{\mathcal{A}} M_2$ and
$$E_1 = [\lambda; x](\mathcal{U}_{M_1}, [x])$$
No new exposed nodes where second reduction may occur compared to TS0.

4. beta-reduction at the root. $E = [ev; z]([\lambda; x](T_1, o_1), o_2, T_2)$ and
$$E_1 = o_1[o_2/x].$$
No new exposed nodes where second reduction may occur compared to TS0.

5. jj-reduction at the root. $E = [ev; z]([j_{M_2',M_3}], [ev; y]([j_{M_1,M_2}], o, T), T')$ such that $M_2' \equiv_{\mathcal{A}} M_2$ and
$$E_1 = [ev; z]([j_{M_1,M_3}], o, \mathcal{U}_{M_3})$$
No new exposed nodes where second reduction may occur compared to TS0.

6. eta-reduction at the root. $E = [\lambda; x](T_1, [ev; y](o, [x], T_2))$ such that o does not depend on x and
$$E_1 = o$$
No new exposed nodes where second reduction may occur compared to TS0.

7. forallj-reductions at the root:
   (a) forallja-reduction at the root. $E = [forall_{M_1,M_2}; x]([ev; z]([j_{M_0,M_1'}], o_1, T), o_2)$ such that $M_1' \equiv_{\mathcal{A}} M_1$ and
$$E_1 = [ev; z]([j_{max(M_0,M_2),max(M_1,M_2)}], [forall_{M_0,M_2}; x](o_1, o_2), \mathcal{U}_{max(M_1,M_2)})$$
   No new exposed nodes where second reduction may occur compared to TS0.
   (b) foralljb-reduction at the root. $E = [forall_{M_1,M_3}; x](o_1, [ev; z]([j_{M_2,M_3'}], o_2, T))$ such that $M_3' \equiv_{\mathcal{A}} M_3$ and
$$E_1 = [ev; z]([j_{max(M_1,M_2),max(M_1,M_3)}], [forall_{M_1,M_2}; x](o_1, o_2), \mathcal{U}_{max(M_1,M_3)}),$$
   No new exposed nodes where second reduction may occur compared to TS0.

8. Eltotal-reduction at the root. $E = [El][total_{M_1,M_2}; x](o_1, o_2)$ and
$$E_1 = [\sum; x]([El](o_1), [El](o_2)).$$
The node where second reduction may occur is $[total_{M_1,M_2}; x]$ with the following actual cases:
   (a) totalja-reduction at $[total_{M_1,M_2}; x]$. Then $o_1 = [ev; z]([j_{M_0,M_1'}], o_1', T)$ where $M_1' \equiv_{\mathcal{A}} M_1$ and
$$E_1 = [\sum; x]([El][ev; z]([j_{M_0,M_1'}], o_1', T), [El](o_2)) \succ$$
$$[\sum; x]([El](o_1'), [El](o_2)) = E_1'$$
$$E_2 = [El][ev; z]([j_{max(M_0,M_2),max(M_1,M_2)}], [total_{M_0,M_2}; x](o_1', o_2), \mathcal{U}_{max(M_1,M_2)}) \succ$$
$$[El][total_{M_0,M_2}; x](o_1', o_2) \succ [\sum; x]([El](o_1'), [El](o_2)) = E_2'$$
   (b) foralljb-reduction at $[total_{M_1,M_2}; x]$. Then $o_2 = [ev; z](j_{M_3,M_2'}, o_2', T)$ where $M_2' \equiv_{\mathcal{A}} M_2$ and
$$E_1 = [\sum; x]([El](o_1), [El][ev; z](j_{M_3,M_2'}, o_2', T)) \succ$$
$$[\sum; x]([El](o_1), [El](o_2')) = E_1'$$
$$E_2 = [El][ev; z]([j_{max(M_1,M_3),max(M_1,M_2)}], [total_{M_1,M_3}; x](o_1, o_2'), \mathcal{U}_{max(M_1,M_2)}) \succ$$
$$[El][forall_{M_1,M_3}; x](o_1, o_2') \succ [\sum; x]([El](o_1), [El](o_2')) = E_2'$$

9. iotatotal1-reduction at the root. $E = [pr1; x](T_1, T_2, [pair; y](o_1, o_2, T_2'''))$ where $T_2[y/x] \sim_{\mathcal{A}} T_2'''$ and

$$E_1 = o_1.$$

The exposed node where second reduction may occur is $[pair; y]$ with the only actual case being

(a) etatotal-reduction at $[pair; y]$. Then

$$E = [pr1; x](T_1, T_2, [pair; y]([pr1; z'](T_1', T_2', a'), [pr2; z''](T_1'', T_2'', a'')), T_2'''))$$

where $a' \sim_{\mathcal{A}} a''$ and $T_2'[y/z'] \sim_{\mathcal{A}} T_2''' \sim_{\mathcal{A}} T_2''[y/z'']$. Then

$$E_1 = [pr1; z'](T_1', T_2', a')$$

$$E_2 = [pr1; x](T_1, T_2, a')$$

10. iotatotal2-reduction at the root. $E = [pr2; x](T_1, T_2, [pair; y](o_1, o_2, T_2'''))$ where $T_2[y/x] \sim_{\mathcal{A}} T_2'''$ and

$$E_1 = o_2.$$

The exposed node where second reduction may occur is $[pair; y]$ with the only actual case being

(a) etatotal-reduction at $[pair; y]$. Then

$$E = [pr2; x](T_1, T_2, [pair; y]([pr1; z'](T_1', T_2', a'), [pr2; z''](T_1'', T_2'', a'')))$$

where $a \equiv_{\mathcal{A}} a'$ and $T_2'[y/z'] \sim_{\mathcal{A}} T_2''' \sim_{\mathcal{A}} T_2''[y/z'']$. Then

$$E_1 = [pr2; z''](T_1'', T_2'', a'')$$

$$E_2 = [pr2; z](T_1', T_2', a')$$

11. etatotal-reduction at the root. $E = [pair; x]([pr1; y](T_1', T_2', a'), [pr2; y](T_1'', T_2'', a''), T_2)$ where $a \sim_{\mathcal{A}} a'$, $T_2'[x/y] \sim_{\mathcal{A}} T_2 \sim_{\mathcal{A}} T_2''[x/y]$ and

$$E_1 = a.$$

The exposed nodes where second reduction may occur are $[pr1; y]$ and $[pr2; y]$ the actual cases being

(a) iotatotal1-reduction at $[pr1; y]$. Then

$$E = [pair; x]([pr1; y](T_1', T_2', [pair; z'](o_1', o_2', T_2''')), [pr2; y](T_1'', T_2'', [pair; z''](o_1'', o_2'', T_2''''')), T_2)$$

where $o_1' \sim_{\mathcal{A}} o_1''$, $o_2' \sim_{\mathcal{A}} o_2''$, $T_2'''[z''/z'] \sim_{\mathcal{A}} T_2''''$, $T_2'[z'/y] \sim_{\mathcal{A}} T_2'''$ and

$$E_1 = [pair; z'](o_1', o_2', T_2''') = E_1'$$

$$E_2 = [pair; x](o_1', [pr2; y](T_1'', T_2'', [pair; z''](o_1'', o_2'', T_2'''')), T_2) \succ_{\mathcal{A}}$$
$$\succ_{\mathcal{A}} [pair; x](o_1', o_2'', T_2) = E_2'$$

(b) iotatotal2-reduction at $[pr2;y]$. Then

$$E = [pair;x]([pr1;y](T_1',T_2',[pair;z'](o_1',o_2',T_2''')),[pr2;y](T_1'',T_2'',[pair;z''](o_1'',o_2'',T_2'''')),T_2)$$

where $o_1' \sim_{\mathcal{A}} o_1''$, $o_2' \sim_{\mathcal{A}} o_2''$, $T_2'''[z''/z'] \sim_{\mathcal{A}} T_2''''$, $T_2''[z''/y] \sim_{\mathcal{A}} T_2''''$ and

$$E_1 = [pair;z'](o_1',o_2',T_2''') = E_1'$$

$$E_2 = [pair;x]([pr1;y](T_1',T_2',[pair;z'](o_1',o_2',T_2''')),o_2'',T_2) \succ_{\mathcal{A}}$$
$$\succ_{\mathcal{A}} [pair;x](o_1',o_2'',T_2) = E_2'$$

12. totalj-reductions at the root:

(a) totalja-reduction at the root. $E = [total_{M_1,M_2};x]([ev;z]([j_{M_0,M_1'}],o_1,T),o_2)$ such that $M_1' \equiv_{\mathcal{A}} M_1$ and

$$E_1 = [ev;z]([j_{max(M_0,M_2),max(M_1,M_2)}],[total_{M_0,M_2};x](o_1,o_2),\mathcal{U}_{max(M_1,M_2)})$$

Exposed nodes where second reduction may occur are $[total_{M_1,M_2};x]$, $[ev;z]$ and $[j_{M_0,M_1'}]$ with the following actual cases:

i. totaljb-reduction at $[total_{M_1,M_2};x]$. Then $o_2 = [ev;z']([j_{M_3,M_2'}],o_2',T')$ where $M_2' \equiv_{\mathcal{A}} M_2$ and

$$E_1 = [ev;z]([j_{max(M_0,M_2),max(M_1,M_2)}],$$
$$[total_{M_0,M_2};x](o_1,[ev;z']([j_{M_3,M_2'}],o_2',T')),\mathcal{U}_{max(M_1,M_2)}) \succ$$
$$[ev;z]([j_{max(M_0,M_2),max(M_1,M_2)}],$$
$$[ev;z']([j_{max(M_0,M_3),max(M_0,M_2')}],[total_{M_0,M_3};x](o_1,o_2'),\mathcal{U}_{max(M_0,M_2')}),$$
$$\mathcal{U}_{max(M_1,M_2)}) \succ$$
$$[ev;z]([j_{max(M_0,M_3),max(M_1,M_2)}],[total_{M_0,M_3};x](o_1,o_2'),\mathcal{U}_{max(M_1,M_2)}) = E_1'$$
$$E_2 = [ev;z']([j_{max(M_1,M_3),max(M_1,M_2)}],$$
$$[total_{M_1,M_3};x]([ev;z]([j_{M_0,M_1'}],o_1,T),o_2'),\mathcal{U}_{max(M_1,M_2)}) \succ$$
$$[ev;z']([j_{max(M_1,M_3),max(M_1,M_2)}],$$
$$[ev;z]([j_{max(M_0,M_3),max(M_1,M_3)}],[total_{M_0,M_3};x](o_1,o_2'),\mathcal{U}_{max(M_1,M_3)}),$$
$$\mathcal{U}_{max(M_1,M_2)}) \succ$$
$$[ev;z']([j_{max(M_0,M_3),max(M_1,M_2)}],[total_{M_0,M_3};x](o_1,o_2'),\mathcal{U}_{max(M_1,M_2)}) = E_2'$$

ii. jj-reduction at $[ev;z]$. Then $o_1 = [ev;z']([j_{M_3,M_0'}],o_1',T')$ where $M_0' \equiv_{\mathcal{A}} M_0$ and

$$E_1 = [ev;z]([j_{max(M_0,M_2),max(M_1,M_2)}],$$
$$[total_{M_0,M_2};x]([ev;z']([j_{M_3,M_0'}],o_1',T'),o_2),\mathcal{U}_{max(M_1,M_2)}) \succ$$
$$[ev;z]([j_{max(M_0,M_2),max(M_1,M_2)}],$$
$$[ev;z']([j_{max(M_3,M_2),max(M_0,M_2)}],[total_{M_3,M_2};x](o_1',o_2),\mathcal{U}_{max(M_0,M_2)}),$$

$$\mathcal{U}_{max(M_1,M_2)}) \succ$$

$$[ev;z]([j_{max(M_3,M_2),max(M_1,M_2)}],[total_{M_3,M_2};x](o'_1,o_2),\mathcal{U}_{max(M_1,M_2)}) = E'_1$$

$$E_2 = [total_{M_1,M_2};x]([ev;z]([j_{M_3,M'_1}],o'_1,T),o_2) \succ$$

$$[ev;z]([j_{max(M_3,M_2),max(M_1,M_2)}],[total_{M_3,M_2};x](o'_1,o_2),\mathcal{U}_{max(M_1,M_2)}) = E'_2$$

iii. jMM-reduction at $[j_{M_0,M'_1}]$. Then $M_0 \equiv_{\mathcal{A}} M'_1$ and

$$E_1 = [ev;z]([j_{max(M_0,M_2),max(M_1,M_2)}],[total_{M_0,M_2};x](o_1,o_2),\mathcal{U}_{max(M_1,M_2)}) \succ$$

$$[ev;z]([\lambda;x](\mathcal{U}_{max(M_0,M_2)},[x]),[total_{M_0,M_2};x](o_1,o_2),\mathcal{U}_{max(M_1,M_2)}) \succ$$

$$[total_{M_0,M_2};x](o_1,o_2) = E'_1$$

$$E_2 = [total_{M_1,M_2};x]([ev;x]([\lambda;x](\mathcal{U}_{M_0},[x]),o_1,T),o_2) \succ$$

$$[total_{M_1,M_2};x](o_1,o_2) = E'_2$$

(b) totaljb-reduction at the root. $E = [total_{M_1,M_3};x](o_1,[ev;z]([j_{M_2,M'_3}],o_2,T))$ such that $M'_3 \equiv_{\mathcal{A}} M_3$ and

$$E_1 = [ev;z]([j_{max(M_1,M_2),max(M_1,M_3)}],[total_{M_1,M_2};x](o_1,o_2),\mathcal{U}_{max(M_1,M_3)}),$$

Exposed nodes where second reduction may occur are $[total_{M_1,M_3};x]$, $[ev;z]$ and $[j_{M_2,M'_3}]$ with the following actual cases:

i. totalja-reduction at $[total_{M_1,M_3};x]$. Then $o_1 = [ev;z']([j_{M_0,M'_1}],o'_1,T')$ where $M'_1 \equiv_{\mathcal{A}} M_1$ and

$$E_1 = [ev;z]([j_{max(M_1,M_2),max(M_1,M_3)}],$$

$$[total_{M_1,M_2};x]([ev;z']([j_{M_0,M'_1}],o'_1,T'),o_2),\mathcal{U}_{max(M_1,M_3)}) \succ$$

$$[ev;z]([j_{max(M_1,M_2),max(M_1,M_3)}],$$

$$[ev;z']([j_{max(M_0,M_2),max(M_1,M_2)}],[total_{M_0,M_2};x](o'_1,o_2),\mathcal{U}_{max(M_0,M_2)}),$$

$$\mathcal{U}_{max(M_1,M_3)}) \succ$$

$$[ev;z]([j_{max(M_0,M_2),max(M_1,M_3)}],[total_{M_0,M_2};x](o'_1,o_2),\mathcal{U}_{max(M_1,M_3)}) = E'_1$$

$$E_2 = [ev;z']([j_{max(M_0,M_3),max(M_1,M_3)}],$$

$$[total_{M_0,M_3};x](o'_1,[ev;z]([j_{M_2,M'_3}],o_2,T'),\mathcal{U}_{max(M_1,M_3)}) \succ$$

$$[ev;z']([j_{max(M_0,M_3),max(M_1,M_3)}],$$

$$[ev;z]([j_{max(M_0,M_2),max(M_0,M_3)}],[total_{M_0,M_2};x](o'_1,o_2),\mathcal{U}_{max(M_0,M_3)}),$$

$$\mathcal{U}_{max(M_1,M_3)}) \succ$$

$$[ev;z']([j_{max(M_0,M_2),max(M_1,M_3)}],[total_{M_0,M_2};x](o'_1,o_2),\mathcal{U}_{max(M_1,M_3)}) = E'_2$$

ii. jj-reduction at $[ev;z]$. Then $o_2 = [ev;z']([j_{M_0,M_2'}], o_2', T')$ where $M_2' \equiv_{\mathcal{A}} M_2$ and

$$E_1 = [ev;z]([j_{max(M_1,M_2),max(M_1,M_3)}],$$
$$[total_{M_1,M_2};x](o_1,[ev;z']([j_{M_0,M_2'}],o_2',T')),\mathcal{U}_{max(M_1,M_3)}) \succ$$
$$[ev;z]([j_{max(M_1,M_2),max(M_1,M_3)}],$$
$$[ev;z']([j_{max(M_1,M_0),max(M_1,M_2)}],[total_{M_1,M_0};x](o_1,o_2'),\mathcal{U}_{max(M_1,M_2)}),$$
$$\mathcal{U}_{max(M_1,M_3)}) \succ$$
$$[ev;z]([j_{max(M_1,M_0),max(M_1,M_3)}],[total_{M_1,M_0};x](o_1,o_2'),\mathcal{U}_{max(M_1,M_3)}) = E_1'$$
$$E_2 = [total_{M_1,M_3};x](o_1,[ev;z]([j_{M_0,M_3'}],o_1',T')) \succ$$
$$[ev;z]([j_{max(M_1,M_0),max(M_1,M_3)}],[total_{M_1,M_0};x](o_1,o_2'),\mathcal{U}_{max(M_1,M_3)}) = E_2'$$

iii. jMM-reduction at $[j_{M_2,M_3'}]$. Then $M_2 \equiv_{\mathcal{A}} M_3'$ and

$$E_1 = [ev;z]([j_{max(M_1,M_2),max(M_1,M_3)}],[total_{M_1,M_2};x](o_1,o_2),\mathcal{U}_{max(M_1,M_3)}) \succ$$
$$[ev;z]([\lambda;x](\mathcal{U}_{max(M_1,M_2)},[x]),[total_{M_1,M_2};x](o_1,o_2),\mathcal{U}_{max(M_1,M_3)}) \succ$$
$$[total_{M_1,M_2};x](o_1,o_2) = E_1'$$
$$E_2 = [total_{M_1,M_3};x](o_1,[ev;x]([\lambda;x](\mathcal{U}_{M_2},[x]),o_2,T')) \succ$$
$$[total_{M_1,M_3};x](o_1,o_2) = E_2'$$

7 Adding pairwise disjoint unions - system TS3

7.1 TS3 terms and typing function

Definition 7.1.1 *[**d31**] The following labels are permitted in the expressions of TS3 - the labels permitted in TS2,* $\amalg$*, coprod,* $ii1$*,* $ii2$ *and* $[sum;x]$*,* $[\amalg;x_1,x_2]$*.*

The notions of u-level expressions, T- and o- terms in TS3 are defined as follows:

Definition 7.1.2 *[**d32**]*

1. *expressions with the root node caring a TS2-label is an u-level expression, o-expression or a T-expression according to the rules of Definition 5.1.2,*
2. *expressions with the root of the form* $\amalg$ *and* $[\amalg;x_1,x_2]$ *are T-expressions,*
3. *expressions with the root of the form coprod,* $ii1$*,* $ii2$ *and sum. are o-expressions.*

Definition 7.1.3 *[**d33**] A TS3-term is a TS3-expression such that:*

1. *any node caring one of the TS2-labels satisfies the conditions of Definition 5.1.3,*

2. *any node of the form* $[\amalg]$ *has valency* 2 *and its two branches are T-expressions,*

3. *any node of the form* $[\amalg; x_1, x_2]$ *has valency* 5*, its first four branches are T-expressions and the fifth branch is an o-expression, its first, second and fifths branches do not depend on* x_1, x_2*, its third branch does not depend on* x_2 *and its fourth branch does not depend on* x_1*,*

4. *any node of the form* $[coprod]$ *has valency* 4*, its first two branches are u-level expressions and the last two branches are o-expressions,*

5. *any node of the form* $[ii1]$ *has valency 3 and its first two branches are T-expressions and the third one is an o-expression,*

6. *any node of the form* $[ii2]$ *has valency 3 and its first two branches are T-expressions and the third one is an o-expression.*

7. *any node of the form* $[sum; x]$ *has valency 6, its first two branches are T-expressions next three are o-expressions all of which do not contain* x *and the last branch is a T-expression.*

Definition 7.1.4 *[***d34***] A node in TS3 term is called non-essential if it satisfies the conditions of Definition 6.1.4. The same applies to the definition of essential nodes, essential and non-essential subexpressions and of* $Ess(E)$ *is extended to TS3-terms in the obvious way.*

We let $TS3$ denote the set of TS3-terms and $TT3$ and $oT3$ denote the subsets of T-terms and o-terms. The obvious analog of Lemma 3.1.7 holds for TS3-expressions. We extend to TS3 the abbreviations introduced for TS2-expressions, abbreviate $[coprod](M_1, M_2, o_1, o_2)$ as $[coprod_{M_1,M_2}](o_1, o_2)$ and write $T_1 \amalg T_2$ for $[\amalg](T_1, T_2)$.

Definition 7.1.5 *[***dtau3***] Under the assumptions of Definition 3.1.8 we extend the typing function* τ_Γ *to o-terms of TS3 as follows:*

1. *the value of* τ_Γ *on an o-term whose root node carries a TS2-label is computed according to the rules of Definition 5.1.5.*

2. $\tau_\Gamma([coprod_{M_1,M_2}](o_1, o_2)) = \mathcal{U}_{max(M_1,M_2)}$*,*

3. $\tau_\Gamma([ii1](T_1, T_2, o)) = [\amalg](T_1, T_2)$*,*

4. $\tau_\Gamma([ii2](T_1, T_2, o)) = [\amalg](T_1, T_2)$*,*

5. $\tau_\Gamma([sum; x](T_1, T_2, s_1, s_2, o, S)) = S[o/x]$*.*

We have the obvious analog of Proposition 3.1.9 for TS3.

7.2 Equivalence relations $\equiv_{\mathcal{A}}$ and $\sim_{\mathcal{A}}$ on TS3-terms

The relations $\equiv_{\mathcal{A}}$ and $\sim_{\mathcal{A}}$ are extended to TS1-terms in the obvious way. We also have obvious analogs of all the statements of Section 3.2.

7.3 Reducibility relation on TS3 terms

Definition 7.3.1 *[**drd3**] Let $UC = (Fu, \mathcal{A})$ be universe context and FV a sequence of T-variables. Define a relation $\succ_{\mathcal{A}}$ on $TS3(Fu, FV, Fv)$ for any Fv as the transitive closure of the union of the following reduction relations:*

1. *Reductions of Definition 5.5.1.*
2. *Elcoprod-reduction. An essential subterm of the form $[El][coprod_{M_1,M_2}](o_1, o_2)$ reduces to $[\amalg]([El](o_1), [El](o_2))$.*
3. *amalg-reductions.*
   (a) *amalg1-reduction. An essential sub term of the form*
   $$[\amalg; x_1, x_2](T_1, T_2, S_1, S_2, [ii1](T_1', T_2', o))$$
   such that $T_1 \sim_{\mathcal{A}} T_1'$ and $T_2 \sim_{\mathcal{A}} T_2'$ reduces to $S_1[o/x_1]$.
   (b) *amalg1-reduction. An essential sub term of the form*
   $$[\amalg; x_1, x_2](T_1, T_2, S_1, S_2, [ii2](T_1', T_2', o))$$
   such that $T_1 \sim_{\mathcal{A}} T_1'$ and $T_2 \sim_{\mathcal{A}} T_2'$ reduces to $S_2[o/x_2]$.
4. *sum-reductions.*
   (a) *sum1-reduction. An essential subterm of the form*
   $$[sum; x](T_1, T_2, s_1, s_2, [ii1](T_1', T_2', o))$$
   such that $T_1 \sim_{\mathcal{A}} T_1'$ and $T_2 \sim_{\mathcal{A}} T_2'$ reduces to $[ev; x](s_1, o, [\amalg](T_1, T_2))$.
   (b) *sum2-reduction. An essential subterm of the form*
   $$[sum; x](T_1, T_2, s_1, s_2, [ii2](T_1', T_2', o))$$
   such that $T_1 \sim_{\mathcal{A}} T_1'$ and $T_2 \sim_{\mathcal{A}} T_2'$ reduces to $[ev; x](s_2, o, [\amalg](T_1, T_2))$.
5. *coprodj-reductions:*
   (a) *coprodj1-reduction. An essential subterm of the form*
   $$[coprod_{M_1,M_2}]([ev; z]([j_{M_0,M_1'}], o_1, T), o_2)$$
   where $M_1 \equiv_{\mathcal{A}} M_1'$ reduces to
   $$[ev; z]([j_{max(M_0,M_2),max(M_1,M_2)}], [coprod_{M_0,M_2}](o_1, o_2), \mathcal{U}_{max(M_1,M_2)})$$

(b) coprodj2-reduction. An essential subterm of the form

$$[coprod_{M_1,M_3}](o_1, [ev; z]([j_{M_2,M'_3}], o_2, T))$$

where $M_3 \equiv_{\mathcal{A}} M'_3$ *reduces to*

$$[ev; z]([j_{max(M_1,M_2),max(M_1,M_3)}], [coprod_{M_1,M_2}](o_1, o_2), \mathcal{U}_{max(M_1,M_3)})$$

The obvious analogs of Lemmas 4.1.3-4.1.6 hold for TS3.

7.4 Local confluence for general terms of TS3

There are eight new confluence situations associated with disjoint union. They are:

1. Elcoprod-reduction at the root with:
   (a) coprodj1-reduction at $[coprod]$.
   (b) coprod2-reduction at $[coprod]$.
2. coprodj1-reduction at the root with:
   (a) coprodj2-reduction at $[coprod]$.
   (b) jj-reduction at $[ev; z]$.
   (c) jMM-reduction at $[j]$.
3. coprodj2-reduction at the root with:
   (a) coprodj1-reduction at $[coprod]$.
   (b) jj-reduction at $[ev; z]$.
   (c) jMM-reduction at $[j]$.

which are handled in exactly the same way as similar confluence situations for $[forall]$ and $[total]$.

7.5 Derivation rules of TS3

The derivation rules for contexts and judgements of TS3 are the derivation rules for TS2 together with the following additional ones:

$$\frac{\Gamma \vdash o_1 : \mathcal{U}_{M_1} \quad \Gamma \vdash o_1 : \mathcal{U}_{M_2}}{\Gamma \vdash [coprod_{M_1,M_2}] : \mathcal{U}_{max(M_1,M_2)}}$$

$$\frac{\Gamma, x_1 : T_1 \rhd \quad \Gamma, x_2 : T_2 \rhd}{\Gamma, x : [\amalg](T_1, T_2) \rhd}$$

$$\frac{\Gamma \vdash a : [Ш](T_1, T_2) \quad \begin{array}{c}\Gamma, x_1 : T_1, s_1 : S_1 \rhd \\ \Gamma, x_2 : T_2, s_2 : S_2 \rhd\end{array}}{\Gamma, s : [Ш; x_1, x_2](T_1, T_2, S_1, S_2, a) \rhd}$$

$$\frac{\Gamma \vdash o_1 : T_1 \quad \Gamma, x_2 : T_2 \rhd}{\Gamma \vdash [ii1](T_1, T_2, o_1) : [Ш](T_1, T_2)}$$

$$\frac{\Gamma, x_1 : T_1 \rhd \quad \Gamma \vdash o_2 : T_2}{\Gamma \vdash [ii2](T_1, T_2, o_2) : [Ш](T_1, T_2)}$$

$$\frac{\Gamma, x : [Ш](T_1, T_2), y : S \rhd \quad \begin{array}{c}\Gamma \vdash s_1 : [\prod; x_1](T_1, S[[ii1](T_1, T_2, x_1)/o]) \\ \Gamma \vdash s_2 : [\prod; x_2](T_2, S[[ii2](T_1, T_2, x_2)/o])\end{array} \quad \Gamma \vdash o : [Ш](T_1, T_2)}{\Gamma \vdash [sum; x](T_1, T_2, s_1, s_2, o, S) : S[o/x]}$$

8 Adding the empty type - system TS4

8.1 TS4 terms and typing function

Definition 8.1.1 *[***d41***] The following labels are permitted in the expressions of TS4 - the labels permitted in TS3, empty, $empty_r$.*

The notions of u-level expressions, T- and o- terms in TS4 are defined as follows:

Definition 8.1.2 *[***d42***]*

1. *expressions with the root node caring a TS3-label is an u-level expression, o-expression or a T-expression according to the rules of Definition 7.1.2,*
2. *expressions with the root of the form* [empty] *and* $[empty_r]$ *are o-expressions,*

Definition 8.1.3 *[***d43***] A TS4-term is a TS4-expression such that:*

1. *any node caring one of the TS3-labels satisfies the conditions of Definition 7.1.3,*
2. *any node of the form* [empty] *has valency* 0*,*
3. *any node of the form* $[empty_r]$ *has valency* 2*, its first branch is a T-expression and its second branch an o-expression.*

Definition 8.1.4 *[***d44***] A node in TS4 term is called non-essential if it satisfies the conditions of Definition 6.1.4. The same applies to the definition of essential nodes, essential and non-essential subexpressions and of $Ess(E)$ is extended to TS4-terms in the obvious way.*

We let $TS4$ denote the set of TS4-terms and $TT4$ and $oT4$ denote the subsets of T-terms and o-terms. The obvious analog of Lemma 3.1.7 holds for TS4-expressions. We extend to TS4 the abbreviations introduced for TS3-expressions and also abbreviate $[El][empty]$ as $\emptyset$.

Definition 8.1.5 *[**dtau4**] Under the assumptions of Definition 3.1.8 we extend the typing function τ_Γ to o-terms of TS4 as follows:*

1. *the value of τ_Γ on an o-term whose root node carries a TS3-label is computed according to the rules of Definition 7.1.5.*
2. $\tau_\Gamma([empty]) = \mathcal{U}_0$,
3. $\tau_\Gamma([empty_r](T, o) = T$

We have the obvious analog of Proposition 3.1.9 for TS3.

8.2 Equivalence relations $\equiv_{\mathcal{A}}$ and $\sim_{\mathcal{A}}$ on TS4-terms

The relations $\equiv_{\mathcal{A}}$ and $\sim_{\mathcal{A}}$ are extended to TS1-terms in the obvious way. We also have obvious analogs of all the statements of Section 3.2.

8.3 Reducibility relation on TS4 terms

There are no new reductions for TS4 compared to TS3. The obvious analogs of Lemmas 4.1.3-4.1.6 hold for TS4.

8.4 Local confluence for general terms of TS4

There are no new confluence situations for TS4 compared with TS3.

8.5 Derivation rules of TS4

The derivation rules for contexts and judgements of TS4 are the derivation rules for TS3 together with the following additional ones:

$$\frac{\Gamma \rhd}{\Gamma \vdash empty : \mathcal{U}_0}$$

$$\frac{\Gamma, x : T \rhd \qquad \Gamma \vdash o : \emptyset}{\Gamma \vdash empty_r(T, o) : T}$$

This derivations rules show in particular correctness of the following definition

$$Def. \quad fromempty(T : Type) := \lambda\, x : \emptyset, [empty_r](T, x) : \prod x : \emptyset, T$$

9 Adding generalized W-types - system TS5

9.1 TS5 terms and typing function

Definition 9.1.1 *[**d51**] The following labels are permitted in the expressions of TS5 - the labels permitted in TS4,* $[IC; x, y, z]$*,* $[c; x, y, z]$*,* $[IC_r; x, y, z, x', v]$*,* $[ic; x, y, z]$*.*

The notions of u-level expressions, T- and o- terms in TS5 are defined as follows:

Definition 9.1.2 *[**d52**]*

1. *expressions with the root node caring a TS4-label is an u-level expression, o-expression or a T-expression according to the rules of Definition 8.1.2,*

2. *expressions with the root of the form* $[IC; x, y, z]$ *are T-expressions,*

3. *expressions with the root of the form* $[c; x, y, z]$*,* $[IC_r; x, y, z, x', v]$ *and* $[ic; x, y, z]$ *are o-expressions.*

Definition 9.1.3 *[**d53**] A TS5-term is a TS5-expression such that:*

1. *any node caring one of the TS4-labels satisfies the conditions of Definition 7.1.3,*

2. *any node of the form* $[IC; x, y, z]$ *has valency* 5*, its first branch, third and fourth branches are T-expressions and second and fifths branches are o-expressions, the first and second branch do not depend on* x, y, z*, the third branch does not depend on* y, z *and the fourth branch does not depend on* z*,*

3. *any node of the form* $[c; x, y, z]$ *has valency* 7 *with the same conditions on the first five branches as for* $[IC; x, y, z]$*, the sixth and seventh branches are o-expressions which do not depend on* x, y *and* z*,*

4. *any node of the form* $[IC_r; x, y, z, x', v]$ *has valency* 8 *with the same conditions on the first five branches as for* $[IC; x, y, z]$ *and additional condition that they do not depend on* x', v*, its sixth and eighths branches are o-expressions, its seventh branch is a T-expression, the sixth and eighths branches do not depend on* x, y, z, x', v*, the seventh branch does not depend on* x, y, z*,*

5. *any node of the form* $[ic; x, y, z]$ *has valency* 8*, its first three branches are u-level expressions and the last five branches are o-expressions, the fourth and fifth branches do not depend on* x, y, z*, the sixth branch does not depend on* y, z *and the seventh branch does not depend on* z*.*

Definition 9.1.4 *[**d54**] A node in TS5 term is called non-essential if it satisfies the conditions of Definition 6.1.4. The same applies to the definition of essential nodes, essential and non-essential subexpressions and of* $Ess(E)$ *is extended to TS5-terms in the obvious way.*

We let $TS5$ denote the set of TS5-terms and $TT5$ and $oT5$ denote the subsets of T-terms and o-terms. The obvious analog of Lemma 3.1.7 holds for TS5-expressions. We extend to TS5 the abbreviations introduced for TS4-expressions.

Definition 9.1.5 *[**dtau5**] Under the assumptions of Definition 3.1.8 we extend the typing function τ_Γ to o-terms of TS5 as follows:*

1. *the value of τ_Γ on an o-term whose root node carries a TS4-label is computed according to the rules of Definition 8.1.5.*
2. $\tau_\Gamma([c; x, y, z](A, a, B, D, q, b, f)) = [IC; x, y, z](A, a, B, D, q)$,
3. $\tau_\Gamma([IC_r; x, y, z, x', v](A, a, B, D, q, i, S, t)) = S[a/x', i/v]$,
4. $\tau_\Gamma([ic; x, y, z](M_1, M_2, M_3, o_A, a, o_B, o_D, q)) = \mathcal{U}_{max(M_1,M_2,M_3)}$.

We have the obvious analog of Proposition 3.1.9 for TS5.

9.2 Equivalence relations $\equiv_{\mathcal{A}}$ and $\sim_{\mathcal{A}}$ on TS5-terms

The relations $\equiv_{\mathcal{A}}$ and $\sim_{\mathcal{A}}$ are extended to TS5-terms in the obvious way. We also have obvious analogs of all the statements of Section 3.2.

9.3 Reducibility relation on TS5 terms

Definition 9.3.1 *[**drd5**] Let $UC = (Fu, \mathcal{A})$ be universe context and FV a sequence of T-variables. Define a relation $\succ_{\mathcal{A}}$ on $TS5(Fu, FV, Fv)$ for all Fv as the transitive closure of the union of the following reduction relations:*

1. *Reductions of Definition 7.3.1.*
2. *Elic-reduction. An essential sub term of the form $[El][ic; x, y, z](M_1, M_2, M_3, o_A, a, o_B, o_D, q)$ reduces to $[IC; x, y, z]([El](o_A), a, [El](o_B), [El](o_D), q)$.*
3. *ICiota-reduction. An essential subterm of the form*

$$[IC_r; x, y, z, x', v](A, a, B, D, [c; x, y, z](A', a', B', D', q, b, f), S, t)$$

where $A \sim_{\mathcal{A}} A'$, $a \sim_{\mathcal{A}} a'$, $B \sim_{\mathcal{A}} B'$ and $D \sim_{\mathcal{A}} D'$ reduces to

$$t\ a\ b\ f\ [\lambda; d](D[a/x, b/y], [IC_r; x, y, z, x', v](A, q[a/x, b/y, d/z], B, D, q, (f\ d)))$$

The obvious analogs of Lemmas 4.1.3-4.1.6 hold for TS5.

9.4 Recursive equality in TS5

In the following two rules we use the notation $qd = q[a/x, b/y, z/d]$.

$$\frac{\begin{array}{l}\Gamma, x : A, y : B, z : D \vdash q : A \\ \Gamma, a : A, i : [IC; x, y, z](A, a, B, D, q), t_1 : T_1 \rhd \\ \Gamma, a : A, i : [IC; x, y, z](A, a, B, D, q), t_2 : T_2 \rhd \\ (\Gamma, a : A, b : B[a/x], d : D[a/x, b/y], i : [IC; x, y, z](A, qd, B, D, q) \vdash T_1[qd/a] =_{\mathcal{A}} T_2[qd/a]) \Rightarrow \\ \qquad (\Gamma, a : A, i : [IC; x, y, z](A, a, B, D, q) \vdash T_1 =_{\mathcal{A}} T_2)\end{array}}{\Gamma, a : A, i : [IC; x, y, z](A, a, B, D, q) \vdash T_1 =_{\mathcal{A}} T_2}$$

$$\frac{\begin{array}{l}\Gamma, x : A, y : B, z : D \vdash q : A \\ \Gamma, a : A, i : [IC; x, y, z](A, a, B, D, q) \vdash f : T \\ \Gamma, a : A, i : [IC; x, y, z](A, a, B, D, q) \vdash g : T \\ (\Gamma, a : A, b : B[a/x], d : D[a/x, b/y], i : [IC; x, y, z](A, qd, B, D, q) \vdash f[qd/a] =_{\mathcal{A}} g[qd/a]) \Rightarrow \\ \qquad (\Gamma, a : A, i : [IC; x, y, z](A, a, B, D, q) \vdash f =_{\mathcal{A}} g)\end{array}}{\Gamma, a : A, i : [IC; x, y, z](A, a, B, D, q) \vdash f =_{\mathcal{A}} g}$$

9.5 Local confluence for general terms of TS5

There are no new confluence situations for TS5 compared with TS5.

9.6 Derivation rules of TS5

The derivation rules for contexts and judgements of TS5 are the derivation rules for TS4 together with the following additional ones:

$$\frac{\Gamma \vdash a : A \quad \Gamma, x : A, y : B, z : D \vdash q : A}{\Gamma, w : [IC; x, y, z](A, a, B, D, q) \rhd}$$

$$\frac{\Gamma \vdash o_A : \mathcal{U}_{M_1} \quad \Gamma \vdash a : [El](o_A) \quad \begin{array}{l}\Gamma, x : [El](o_A) \vdash o_B : \mathcal{U}_{M_2} \\ \Gamma, x : [El](o_A), y : [El](o_B) \vdash o_D : \mathcal{U}_{M_3} \\ \Gamma, x : [El](o_A), y : [El](o_B), z : [El](o_D) \vdash q : [El](o_A)\end{array}}{\Gamma \vdash [ic; x, y, z](o_A, a, o_B, o_D, q) : \mathcal{U}_{max(M_1, M_2, M_3)}}$$

$$\frac{\Gamma \vdash a : A \quad \Gamma, x : A, y : B, z : D \vdash q : A \quad \Gamma \vdash b : B[a/x] \quad \Gamma \vdash f : [\prod; z'](D[a/x, b/y], ICqa)}{\Gamma \vdash [c; x, y, z](A, a, B, D, q, b, f) : [IC; x, y, z](A, a, B, D, q)}$$

where

$$ICqa := [IC; x, y, z](A, q[a/x, b/y, z'/z], B, D, q)$$

and

$$\frac{\begin{array}{l}\Gamma \vdash a : A \\ \Gamma, x : A, y : B, z : D \vdash q : A \\ \Gamma \vdash i : [IC; x, y, z](A, a, B, D, q) \\ \Gamma, x' : A, v : ICx', s : S \rhd \\ \Gamma \vdash t : \prod(x' : A)(y' : B[x'/x])(w : [\prod; z'](D[x'/x, y'/y], ICq')), \\ [\prod; d](D[x'/x, y'/y], S[q[x'/x, y'/y, d/z]/x', (w\, d)/v]) \to S[(cx'y'w)/v]\end{array}}{\Gamma \vdash [IC_r; x, y, z, x', v](A, a, B, D, q, i, S, t) : S[a/x', i/v]}$$

where

$$ICx' := [IC; x, y, z](A, x', B, D, q)$$

$$ICq' := [IC; x, y, z](A, q[x'/x, y'/y, z'/z], B, D, q)$$

$$cx'y'w := [c; x, y, z](A, x', B, D, q, y', w)$$

Remark 9.6.1 Our inductive type $[IC; x, y, z](A, a, B, D, q)$ corresponds to the following inductive definition in Coq:

Inductive IC(A:Type)(a:A)(B:A->Type)(D:forall x:A, (B x -> Type))(q:forall x:A, forall y:B x, forall z: D x y, A):= c: forall b:B a, forall f : (forall d: D a b, IC A (q a b d) B D q), IC A a B D q .

with $[IC_r; \ldots]$ being a direct analog of the eliminator IC_r and the reduction rule the direct analog of the iota-reduction for this inductive type.

10 Datatypes

The basic constructions of TS5 allow one to define various "datatypes" i.e. types with associated eliminators and computation rules which in Coq are introduced by strictly positive inductive definitions without pseudo-parameters. In this section we consider only natural numbers and binary trees but it should be possible to write down a simple algorithm which would translate a general strictly positive inductive definition without pseudo-parameters into the language of TS5.

10.1 Natural numbers

The object of type $\mathcal{U}_0$ corresponding to natural numbers is defined as follows:

$$Def. \quad nat := [ic; x, y, z](o_A, tt, o_B, o_D, tt) : \mathcal{U}_0$$

where

$$o_A := pt \quad o_B := [coprod_{0,0}](pt, pt)$$

$$o_D := [sum; y'](Pt, Pt, (\lambda\, z' : Pt, empty), (\lambda\, z' : Pt, pt), y, \mathcal{U}_0)$$

We will write $\mathbf{N}$ for $[El](nat)$. The two standard constructors for $\mathbf{N}$ are defined as:

$$Def. \quad O := [c; x, y, z](A, tt, B, D, tt, [ii1](Pt, Pt, tt), (fromempty\, \mathbf{N})) : \mathbf{N}$$

$$Def. \quad S := \lambda\, n : \mathbf{N}, [c; x, y, z](A, tt, B, D, tt, [ii2](Pt, Pt, tt), (\lambda\, t : Pt, n)) : \mathbf{N} \to \mathbf{N}$$

where

$$A := [El](o_A) = Pt \quad B := [El](o_B) = Pt \amalg Pt \quad D = [El](o_D)$$

The eliminator associated with natural numbers is defined as:

$$[nat_r; v](v : \mathbf{N}, v' : T \triangleright)(o_O : T[O/v])(o_S : \prod(v : \mathbf{N})(y : T), T[(S\, v)/v])(n : \mathbf{N}) :=$$

$$= [IC_r; x, y, z, x', v](A, tt, B, D, tt, n, T, t) : T[n/v]$$

where

$$t : \prod(x' : Pt)(y' : Pt \amalg Pt)(w : D[y'/y] \to \mathbf{N})(s : \prod d : D[y'/y], T[(w\, d)/v]),$$

$$T([c; x, y, z](A, x', B, D, tt, y', w)/v)$$

using eliminators for Pt and $Pt \amalg Pt$ we see that it is sufficient to describe $t\, tt\, [ii1](tt)$ and $t\, tt\, [ii2](tt)$ whose types are

$$t\, tt\, [ii1](tt) : \prod(w : \emptyset \to \mathbf{N})(s : \prod d : \emptyset, T[(w\, d)/v]), T([c; x, y, z](A, tt, B, D, tt, [ii1](tt), w)/v)$$

and

$$t\, tt\, [ii2](tt) : \prod(w : Pt \to \mathbf{N})(s : \prod d : Pt, T[(w\, d)/v]), T([c; x, y, z](A, tt, B, D, tt, [ii2](tt), w)/v)$$

$$= \lambda(x : Pt)(y : Pt \amalg Pt)(w : \prod z' : D, \mathbf{N})(s : \prod d : D, T[w\, d/v]), [sum; y'](Pt, Pt, f_1, f_2, y, (\prod(d : D[y'/y])$$

11 Adding the identity types - system TS6

11.1 TS6 terms and typing function

Definition 11.1.1 *[**d61**] The following labels are permitted in the expressions of TS6 - the labels permitted in TS5, Id, $paths$, $refl$, $[J; x, e]$.*

The notions of u-level expressions, T- and o- terms in TS6 are defined as follows:

Definition 11.1.2 *[**d62**]*

1. *expressions with the root node caring a TS5-label is an u-level expression, o-expression or a T-expression according to the rules of Definition 15.1.2,*

2. *expressions with the root of the form* $[Id]$ *are T-expressions,*

3. *expressions with the root of the form* $[paths]$*,* $[relf]$ *and* $[J;x,e]$ *are o-expressions,*

Definition 11.1.3 *[**d63**] A TS6-term is a TS6-expression such that:*

1. *any node caring one of the TS5-labels satisfies the conditions of Definition 15.1.3,*

2. *any node of the form* $[Id]$ *has valency* 3*, its first branch is a T-expression and the last two branches are o-expressions,*

3. *any node of the form* $[paths]$ *has valency* 4*, its first branch is a u-level expression and the following three branches are o-expressions,*

4. *any node of the form* $[refl]$ *has valency* 2*, its first branch is a T-expression and its second branch is an o-expression,*

5. *any node of the form* $[J;x,e]$ *has valency* 6*, its first branch is a T-expression, the following 4 branches are o-expressions all of them do not contain* x, e *and the last branch is a T-expression.*

Definition 11.1.4 *[**d64**] A node in TS6 term is called non-essential if it satisfies the conditions of Definition 6.1.4. The same applies to the definition of essential nodes, essential and non-essential subexpressions and of* $Ess(E)$ *is extended to TS6-terms in the obvious way.*

We let $TS6$ denote the set of TS6-terms and $TT6$ and $oT6$ denote the subsets of T-terms and o-terms. The obvious analog of Lemma 3.1.7 holds for TS6-expressions. We extend to TS6 the abbreviations introduced for TS5-expressions, abbreviate $[paths](M, t, o_1, o_2)$ as $[paths_M](t, o_1, o_2)$, $[Id](T, o_1, o_2)$ as $Id\,T\,o_1\,o_2$ and write $Id\,x\,y$ for $Id\,\tau_\Gamma(x)\,x\,y$ and $refl\,o$ for $[refl](\tau_\Gamma(o), o)$.

Definition 11.1.5 *[**dtau6**] Under the assumptions of Definition 3.1.8 we extend the typing function* τ_Γ *to o-terms of TS6 as follows:*

1. *the value of* τ_Γ *on an o-term whose root node carries a TS5-label is computed according to the rules of Definition 15.1.5.*

2. $\tau_\Gamma([paths_M](t, o_1, o_2)) = \mathcal{U}_M$*,*

3. $\tau_\Gamma([refl](T, o)) = [Id](T, o, o)$*,*

4. $\tau_\Gamma([J;x,e](T, a, b, q, i, S)) = S[o_1/x, o_2/y, o_3/e]$

We have the obvious analog of Proposition 3.1.9 for TS6.

11.2 Equivalence relations $\equiv_{\mathcal{A}}$ and $\sim_{\mathcal{A}}$ on TS6-terms

The relations $\equiv_{\mathcal{A}}$ and $\sim_{\mathcal{A}}$ are extended to TS1-terms in the obvious way. We also have obvious analogs of all the statements of Section 3.2.

11.3 Reducibility relation on TS6 terms

Definition 11.3.1 *[***drd6***] Let $UC = (Fu, \mathcal{A})$ be universe context and FV a sequence of T-variables. Define a relation $\succ_{\mathcal{A}}$ on $TS6(Fu, FV, Fv)$ for all Fv as the transitive closure of the union of the following reduction relations:*

1. *Reductions of Definition 7.3.1.*
2. *Elpaths-reduction An essential subterm of the form $[El][paths_M](t, o_1, o_2)$ reduces to $[Id]([El](t), o_1, o_2)$,*
3. *pathsj-reduction An essential subterm of the form $[paths_{M_2}]([ev; x]([j_{M_1,M_2'}], t,), o_1, o_2)$ such that $M_2 \equiv_{\mathcal{A}} M_2'$ reduces to $[ev; x]([j_{M_1,M_2'}], [paths_{M_1}](t, o_1, o_2), M_2')$,*
4. *Jiota-reduction. An essential subterm of the form $[J; x, e](T, a, a', q, [refl](T', a''), S)$ such that $a \sim_{\mathcal{A}} a' \sim_{\mathcal{A}} a''$ and $T \sim_{\mathcal{A}} T'$ reduces to q,*

The obvious analogs of Lemmas 4.1.3-4.1.6 hold for TS6.

11.4 Local confluence for general terms of TS6

There are the following new confluence situations in TS6 compared with TS5:

1. Elpaths-reduction at the root and pathsj-reduction at $[paths]$,
2. pathsj-reduction at the root and
   (a) jj-reduction at $[ev; x]$,
   (b) jMM-reduction at [j].

These cases are dealt with in the same way as similar cases for $[forall]$, $[total]$ etc.

11.5 Derivation rules of TS6

The derivation rules for contexts and judgements of TS6 are the derivation rules for TS5 together with the following additional ones:

$$\frac{\Gamma \vdash t : \mathcal{U}_M \quad \Gamma \vdash o_1 : [El][t] \quad \Gamma \vdash o_2 : [El][t]}{\Gamma \vdash [paths_M](t, o_1, o_2) : \mathcal{U}_M}$$

$$\frac{\Gamma \vdash o_1 : T \quad \Gamma \vdash o_2 : T}{\Gamma, y : [Id](T, o_1, o_2) \rhd}$$

$$\frac{\Gamma \vdash o : T}{\Gamma \vdash [refl](T, o) : [Id](T, o, o)}$$

$$\frac{\Gamma \vdash a : T \qquad \begin{array}{c} \Gamma, x : T, e : [Id](T, a, x), z : S \rhd \\ \Gamma \vdash q : S[a/x, [refl](T, a)/e] \end{array} \qquad \begin{array}{c} \Gamma \vdash b : T \\ \Gamma \vdash i : [Id](T, a, b) \end{array}}{\Gamma \vdash [J; x, e](T, a, b, q, i, S) : S[b/x, i/e]}$$

12 Adding resizing rules RR0 and RR1 - system TS7

In order to introduce the resizing rules we will need the folioing definitions:

1. $Def.\quad Iscontr(X : Type) := \sum x : X, \prod y : X, Id\, y\, x$
2. $Def.\quad Hfiber\{X, Y : Type\}(f : X \to Y)(y : Y) := \sum x : X, Id\,(f\,x)\,y$
3. $Def.\quad Isweq\{X\,Y\,Type\}(f : X \to Y) := \prod y : Y, Iscontr(Hfiber\,f\,y)$
4. $Def.\quad Weq(X\,Y\ : Type) := \sum f : X \to Y, Isweq\,f$
5. $Def.\quad Isaprop(X : Type) := \prod x : X, \prod x' : X, Iscontr(Id\,x\,x')$
6. $Def.\quad Isaset(X : Type) := \prod x : X, \prod x' : X, Isaprop(Id\,x\,x')$
7. $Def.\quad idfun(X : Type) := \lambda x : X, x$
8. $Th.\quad idisweq(X : Type) : Isweq\,X$
9. $Def.\quad idweq(X : Type) := [pair; f](Isweq\,f, idfun\,X, idisweq\,X) : Weq\,X\,X$

12.1 TS7 terms and typing function

Definition 12.1.1 *[***d71***] The following labels are permitted in the expressions of TS7 - the labels permitted in TS6,* $rr0$*,* $rr1$*.*

The notions of u-level expressions, T- and o- terms in TS7 are defined as follows:

Definition 12.1.2 *[***d72***]*

1. *expressions with the root node caring a TS5-label is an u-level expression, o-expression or a T-expression according to the rules of Definition 11.1.2,*
2. *expressions with the root of the form* $rr0$ *and* $rr1$ *are o-expressions,*

Definition 12.1.3 *[***d73***] A TS7-term is a TS7-expression such that:*

1. *any node caring one of the TS6-labels satisfies the conditions of Definition 11.1.3,*
2. *any node of the form* $[rr0]$ *has valency* 5*, its first two branches are u-level expressions and the last three are o-expressions,*
3. *any node of the form* $[rr1]$ *has valency* 3*, its first branch is an u-level expression and the last two branches are o-expressions,*

Definition 12.1.4 *[d74] A node in TS7 term is called non-essential if it satisfies the conditions of Definition 6.1.4 or if it belongs to the subexpression t or e of a subexpression of the form $[rr0](M_1, M_2, s, t, e)$ or if it belongs to subexpression p of a subexpression of the form $[rr1](M, a, p)$.*

The definition of essential nodes, essential and non-essential subexpressions and of $Ess(E)$ is extended to TS6-terms in the obvious way.

We let $TS7$ denote the set of TS7-terms and $TT7$ and $oT7$ denote the subsets of T-terms and o-terms. The obvious analog of Lemma 3.1.7 holds for TS7-expressions. We extend to TS7 the abbreviations introduced for TS6-expressions and also abbreviate $[rr0](M_2, M_1, s, t, e)$ as $[rr0_{M_2,M_1}](s, t, e)$ and $[rr1](M, a, p)$ as $[rr1_M](a, p)$.

Definition 12.1.5 *[dtau7] Under the assumptions of Definition 3.1.8 we extend the typing function τ_Γ to o-terms of TS7 as follows:*

1. *the value of τ_Γ on an o-term whose root node carries a TS6-label is computed according to the rules of Definition 11.1.5.*
2. $\tau_\Gamma([rr0_{M_2,M_1}](s, t, e)) = \mathcal{U}_{M_1}$,
3. $\tau_\Gamma([rr1_M](a, p)) = \mathcal{U}_0$,

We have the obvious analog of Proposition 3.1.9 for TS7.

12.2 Equivalence relations $\equiv_\mathcal{A}$ and $\sim_\mathcal{A}$ on TS7-terms

The relations $\equiv_\mathcal{A}$ and $\sim_\mathcal{A}$ are extended to TS7-terms in the obvious way. We also have obvious analogs of all the statements of Section 3.2.

12.3 Reducibility relation on TS7 terms

Definition 12.3.1 *[drd7] Let $UC = (Fu, \mathcal{A})$ be universe context and FV a sequence of T-variables. Define a relation $\succ_\mathcal{A}$ on $TS7(Fu, FV, Fv)$ for all Fv as the transitive closure of the union of the following reduction relations:*

1. *Reductions of Definition 11.3.1.*
2. *Elrr0-reduction. An essential subterm of the form $[El][rr0_{M_2,M_1}](s, t, e)$ reduces to $[El](s)$.*
3. *forallrr0a-reduction. An essential subterm $S = [forall_{M_1,M_2}; x]([rr0_{M_3,M_1'}](s, t, e), o_2)$ such that $M_1 \equiv_\mathcal{A} M_1'$ and $M_1' \leq_\mathcal{A} M_3$ reduces to*

$$[rr0_{max(M_3,M_2),max(M_1,M_2)}]([forall_{M_3,M_2}; x](s, o_2), S, idweq\, S)$$

4. *forallrr0b-reduction. An essential subterm* $S = [forall_{M_1,M_2}; x](o_1, [rr0_{M_3,M_2'}](s,t,e))$ *such that* $M_2 \equiv_{\mathcal{A}} M_2'$ *and* $M_2' \leq_{\mathcal{A}} M_3$ *reduces to*

$$[rr0_{max(M_1,M_3),max(M_1,M_2)}]([forall_{M_1,M_3}; x](o_1, s), S, idweq\, S)$$

5. *totalrr0a-reduction. An essential subterm* $S = [total_{M_1,M_2}; x]([rr0_{M_3,M_1'}](s,t,e), o_2)$ *such that* $M_1 \equiv_{\mathcal{A}} M_1'$ *and* $M_1' \leq_{\mathcal{A}} M_3$ *reduces to*

$$[rr0_{max(M_3,M_2),max(M_1,M_2)}]([total_{M_3,M_2}; x](s, o_2), S, idweq\, S)$$

6. *totalrr0b-reduction. An essential subterm* $S = [total_{M_1,M_2}; x](o_1, [rr0_{M_3,M_2'}](s,t,e))$ *such that* $M_2 \equiv_{\mathcal{A}} M_2'$ *and* $M_2' \leq_{\mathcal{A}} M_3$ *reduces to*

$$[rr0_{max(M_1,M_3),max(M_1,M_2)}]([total_{M_1,M_3}; x](o_1, s), S, idweq\, S)$$

7. *coprodrr0a-reduction. An essential subterm* $S = [coprod_{M_1,M_2}]([rr0_{M_3,M_1'}](s,t,e), o_2)$ *such that* $M_1 \equiv_{\mathcal{A}} M_1'$ *and* $M_1' \leq_{\mathcal{A}} M_3$ *reduces to*

$$[rr0_{max(M_3,M_2),max(M_1,M_2)}]([coprod_{M_3,M_2}](s, o_2), S, idweq\, X)$$

8. *coprodrr0b-reduction. An essential subterm* $S = [coprod_{M_1,M_2}](o_1, [rr0_{M_3,M_2'}](s,t,e))$ *such that* $M_2 \equiv_{\mathcal{A}} M_2'$ *and* $M_2' \leq_{\mathcal{A}} M_3$ *reduces to*

$$[rr0_{max(M_1,M_3),max(M_1,M_2)}]([coprod_{M_1,M_3}](o_1, s), S, idweq\, S)$$

9. *pathsrr0-reduction. An essential subterm* $S = [paths_{M_1}]([rr0_{M_2,M_1'}](s,t,e), o_1, o_2)$ *such that* $M_1 \equiv_{\mathcal{A}} M_1'$ *and* $M_1' \leq_{\mathcal{A}} M_2$ *reduces to*

$$[rr0_{M_2,M_1}]([paths_{M_2}](s, o_1, o_2), S, idweq\, S)$$

10. *rr1rr0-reduction. An essential subterm of the form* $[rr1_M](a,p)$ *reduces to*

$$[rr0_{M,0}](a, [rr1_M](a,p), idweq\, [rr1_M](a,p)).$$

11. *jrr0j-reduction. An essential subterm of the form* $j_{M_1',M_3}\, [rr0_{M_2,M_1}](s,t,e)$ *such that* $M_1' \equiv_{\mathcal{A}} M_1$ *and* $M_1 \leq_{\mathcal{A}} M_2 \leq_{\mathcal{A}} M_3$ *reduces to* $j_{M_2,M_3}\, s$.

12. *jrr0rr0-reduction. An essential subterm of the form* $j_{M_1',M_2}\, [rr0_{M_3,M_1}](s,t,e)$ *such that* $M_1' \equiv_{\mathcal{A}} M_1$ *and* $M_1 \leq_{\mathcal{A}} M_2 \leq_{\mathcal{A}} M_3$ *reduces to* $[rr0_{M_3,M_2}](s, j_{M_1,M_2}\, t, e)$.

13. *rr0jj-reduction. An essential subterm of the form* $[rr0_{M_2,M_1}]((j_{M_0,M_2'}\, s), t, e)$ *such that* $M_2 \equiv_{\mathcal{A}} M_2'$ *and* $M_0 \leq_{\mathcal{A}} M_1 \leq_{\mathcal{A}} M_2$ *reduces to* $j_{M_0,M_1}\, s$.

14. *rr0jrr0-reduction. An essential subterm of the form* $[rr0_{M_3,M_1}]((j_{M_2,M_3'}\, s), t, e)$ *such that* $M_3 \equiv_{\mathcal{A}} M_3'$ *and* $M_0 \leq_{\mathcal{A}} M_1 \leq_{\mathcal{A}} M_2$ *reduces to* $[rr0_{M_2,M_1}](s,t,e)$.

15. *rr0rr0-reduction. An essential subterm of the form* $[rr0_{M_2,M_1}]([rr0_{M_3,M_2'}](s, t_2, e_2), t_1, e_1)$ *such that* $M_2 \equiv_{\mathcal{A}} M_2'$ *and* $M_1 \leq_{\mathcal{A}} M_2 \leq_{\mathcal{A}} M_3$ *reduces to* $[rr0_{M_3,M_1}](s, t_1, e_1)$.

16. *rr0MM-reduction. An essential subterm of the form* $[rr0_{M,M'}](s,t,e)$ *such that* $M \equiv_{\mathcal{A}} M'$ *reduces to* s.

The obvious analogs of Lemmas 4.1.3-4.1.6 hold for TS7.

12.4 Local confluence for general terms of TS7

There are, if I have not missed anything, 91 new confluence cases related to the addition of resizing rules RR0 and RR1. We consider them below by naming the corresponding root reduction and the reduction at an exposed essential sub-term. The proofs in all cases seem to be straightforward.

1. Elj-reduction at the root.
   (a) jrr0j.
   (b) jrr0rr0.
2. Elforall-reduction at the root.
   (a) forallrr0a.
   (b) forallrr0b.
3. jj-reduction at the root.
   (a) jrr0j.
   (b) jrr0rr0.
4. forallja-reduction at the root.
   (a) jrr0j.
   (b) jrr0rr0.
5. foralljb-reduction at the root.
   (a) jrr0j.
   (b) jrr0rr0.
6. Eltotal-reduction at the root.
   (a) totalrr0a.
   (b) totalrr0b.
7. totalja-reduction at the root.
   (a) jrr0j.
   (b) jrr0rr0.
8. totaljb-reduction at the root.
   (a) jrr0j.
   (b) jrr0rr0.

9. Elcoprod-reduction at the root.
    (a) coprodrr0a.
    (b) coprodrr0b.
10. coprodja-reduction at the root.
    (a) jrr0j.
    (b) jrr0rr0.
11. coprodjb-reduction at the root.
    (a) jrr0j.
    (b) jrr0rr0.
12. Elpaths-reduction at the root.
    (a) pathsrr0.
13. pathsj-reduction at the root.
    (a) jrr0j.
    (b) jrr0rr0.
14. Elrr0-reduction at the root.
    (a) rr0jj.
    (b) rr0jrr0.
    (c) rr0rr0.
    (d) rr0MM.
15. forallrr0a/forallrr0b-reductions at the root.
16. forallrr0a-reduction at the root.
    (a) rr0jj.
    (b) rr0jrr0.
    (c) rr0rr0.
    (d) rr0MM.
17. forallrr0b-reduction at the root.
    (a) rr0jj.
    (b) rr0jrr0.
    (c) rr0rr0.
    (d) rr0MM.

18. totalrr0a/totalrr0b-reductions at the root.
19. totalrr0a-reduction at the root.
    (a) rr0jj.
    (b) rr0jrr0.
    (c) rr0rr0.
    (d) rr0MM.
20. totalrr0b-reduction at the root.
    (a) rr0jj.
    (b) rr0jrr0.
    (c) rr0rr0.
    (d) rr0MM.
21. coprodrr0a/coprodrr0b-reductions at the root.
22. coprodrr0a-reduction at the root.
    (a) rr0jj.
    (b) rr0jrr0.
    (c) rr0rr0.
    (d) rr0MM.
23. coprodrr0b-reduction at the root.
    (a) rr0jj.
    (b) rr0jrr0.
    (c) rr0rr0.
    (d) rr0MM.
24. pathsrr0-reduction at the root.
    (a) rr0jj.
    (b) rr0jrr0.
    (c) rr0rr0.
    (d) rr0MM.
25. jrr0j-reduction at the root.
    (a) rr0jj.
    (b) rr0jrr0.

(c) rr0rr0.

(d) rr0MM.

26. jrr0rr0-reduction at the root.

    (a) jrr0j.
    (b) rr0jj.
    (c) rr0jrr0.
    (d) rr0rr0.
    (e) rr0MM.

27. rr0jj-reduction at the root.

    (a) jrr0j.
    (b) jrr0rr0.
    (c) rr0jrr0.
    (d) rr0MM.

28. rr0jrr0-reduction at the root.

    (a) jrr0j.
    (b) jrr0rr0.
    (c) rr0jj.
    (d) rr0MM.

29. rr0rr0-reduction at the root.

    (a) rr0jj.
    (b) rr0jrr0.
    (c) rr0rr0.
    (d) rr0MM(a).
    (e) rr0MM(b).

30. rr0MM-reduction at the root.

    (a) rr0jj.
    (b) rr0jrr0.
    (c) rr0rr0.

12.5 Derivation rules of TS7

1.
$$\frac{\Gamma \vdash s : \mathcal{U}_{M_2} \quad \Gamma \vdash t : \mathcal{U}_{M_1} \quad M_1 \leq_{\mathcal{A}} M_2 \quad \Gamma \vdash e : Weq * s * t}{\Gamma \vdash [rr0_{M_2,M_1}](s,t,e) : \mathcal{U}_{M_1}}$$

2.
$$\frac{\Gamma \vdash a : \mathcal{U}_M \quad \Gamma \vdash p : Isaprop * a}{\Gamma \vdash [rr1](a,p) : \mathcal{U}_0}$$

12.6 Main meta-theorems and conjectural meta-theorems for TS6

In what follows we fix a universe context UC and a sequence of T-variables FV and consider all the notions relative to this context and this sequence.

Lemma 12.6.1 *[***lm000***] In TS6 one has:*

1. *Any derivation tree for a context of the form $\Gamma, \Gamma' \rhd$ has a rooted sub-tree whose leaves are contexts of the form $\Gamma \rhd$.*
2. *Any derivation tree for a judgement of the form $\Gamma, \Gamma' \vdash o : T$ has a rooted sub-tree whose leaves are contexts of the form Γ.*

Proof: Induction on the length of the derivation tree. In the first case if Γ' is empty then there is nothing to prove. Looking at the derivation rules we see that the premises for any derivation rule for a context of the form $\Gamma, \Gamma' \rhd$ where Γ' is non-empty has either the same form or of the form $\Gamma, \Gamma' \vdash o : T$ or equals $\Gamma \rhd$. The premises for any derivation rule for a judgement of the form $\Gamma, \Gamma' \vdash o : T$ is either of the same form or of the form $\Gamma, \Gamma' \rhd$ where Γ' is non-empty or equals $\Gamma \rhd$.

Definition 12.6.2 *[***d000***] For a context of the form $\Gamma, \Gamma' \rhd$ or a judgement of the form $\Gamma, \Gamma' \vdash o : T$ the derivation depth relative to Γ is the minimum over all derivation trees of the depth of the smallest sub-tree with all leaves equal $\Gamma \rhd$.*

Note that the derivation depth relative to Γ is 0 only for $\Gamma \rhd$. Note also that for any context or judgement of the form considered in Definition 12.6.2 there exists a derivation rule which produce this context or judgement such that all the premises are again of the same form and their derivation depth relative to Γ is strictly less than for the original context or judgement.

Lemma 12.6.3 *[***lm00***] One has:*

1. *Let $x_1 : T_1, \ldots, x_n : T_n \rhd$ be a derivable context in TS6. Then for any $i \leq n$ the context $x_1 : T_1, \ldots, x_i : T_i \rhd$ is derivable.*

2. *Let $x_1 : T_1, \ldots, x_n : T_n \vdash o : T$ be a derivable judgement in TS6. Then for any $i \leq n$ the context $x_1 : T_1, \ldots, x_i : T_i \rhd$ is derivable.*

Proof: The statements:

1. Let $x_1 : T_1, \ldots, x_n : T_n \rhd$ be a derivable context in TS6 of derivation depth $\leq N$. Then for any $i \leq n$ the context $x_1 : T_1, \ldots, x_i : T_i \rhd$ is derivable.
2. Let $x_1 : T_1, \ldots, x_n : T_n \vdash o : T$ be a derivable judgement in TS6 of derivation depth $\leq N$. Then for any $i \leq n$ the context $x_1 : T_1, \ldots, x_i : T_i \rhd$ is derivable.

are immediately provable by induction on N from the form of the derivation rules.

12.7 Main conjectural meta-theorems for TS0

Let us fix a universe context $UC = (Fu, \mathcal{A})$ and a sequence of T-variables FV.

Definition 12.7.1 *[**bnd**] A term E is said to be bounded relative to $\mathcal{A}$ if there exists $d \in \mathbf{N}$ such that for any sequence of the form $E \succ_{\mathcal{A}} E_1 \succ_{\mathcal{A}} E_2 \succ_{\mathcal{A}} \ldots \succ_{\mathcal{A}} E_n$ one has $n \leq d$. The smallest d satisfying this property is called the reduction depth of E.*

Definition 12.7.2 *[**nf**] A term E' is said to be a normal form of a term E relative to $\mathcal{A}$ if there is a sequence of the form $E \succ_{\mathcal{A}} E_1 \succ_{\mathcal{A}} \ldots \succ_{\mathcal{A}} E_d = E'$ and the reduction depth of E' is 0. A term is said to be in the normal form if it is a normal form of itself.*

In our system there is no chance for the actual uniqueness of the normal form. Instead the normal form of derivable terms (see below) is expected to be unique up to $\equiv_{\mathcal{A}}$.

Definition 12.7.3 *[**derivable**] A T-term T is called derivable (relative to UC) if there is a derivable context of the form $\Gamma, x : T \rhd$. An o-term o is called derivable if there is a derivable judgement of the form $\Gamma \vdash o : T$.*

Conjecture 1 *[**c1**] If E is a derivable term then any subterm of E is derivable.*

Conjecture 2 *[**c5**] If $\Gamma \vdash o : T$ is derivable then $\Gamma \vdash o : \tau_{\Gamma}(o)$ is derivable.*

Conjecture 3 *[**c2**] If E is a derivable term and $E \succ_{\mathcal{A}} E'$ then E' is derivable.*

Conjecture 4 *[**c3**] If E is a derivable term, $E \succ_{\mathcal{A}} E_1$, $E \succ_{\mathcal{A}} E_2$ then there exist E'_1, E'_2 such that $E'_1 \equiv_{\mathcal{A}} E'_2$ and $E_1 \succeq_{\mathcal{A}} E'_1$, $E_2 \succeq_{\mathcal{A}} E'_2$.*

Conjecture 5 *[**c4**] If $\mathcal{A} \neq \emptyset$ and E is a derivable term then E is bounded.*

The following pre-theorems are theorems modulo the conjectures stated above.

Pretheorem 12.7.4 *[***pt1***] Let $\mathcal{A} \neq \emptyset$, E be a derivable term and $E \succeq_{\mathcal{A}} E_1$, $E \succ_{\mathcal{A}} E_2$ then there exist E_1', E_2' such that $E_1' \equiv_{\mathcal{A}} E_2'$ and $E_1 \succeq_{\mathcal{A}} E_1'$, $E_2 \succeq_{\mathcal{A}} E_2'$.*

Proof: ???

Pretheorem 12.7.5 *[***pt2***] Let $\mathcal{A} \neq \emptyset$ and E be a derivable term. Then E has a normal form $N(E)$ and for any two normal fors $N_1(E)$, $N_2(E)$ of E one has $N_1(E) \equiv_{\mathcal{A}} N_2(E)$.*

Proof: ???

Definition 12.7.6 *[***deq***] Let $\mathcal{A} \neq \emptyset$ and E, E' be derivable terms. We say that $E =_{\mathcal{A}} E'$ if for normal forms $N(E)$, $N(E')$ of E and E' respectively one has $N(E) \equiv_{\mathcal{A}} N(E')$.*

Note that our pre-theorems imply that if $\mathcal{A} \neq \emptyset$ then $=_{\mathcal{A}}$ is a decidable equivalence relation on derivable terms.

13 Appendix A. Complete list of derivation rules

1.
$$\frac{}{\rhd}$$

2.
$$\frac{\Gamma}{\Gamma, x : X \rhd \quad for \quad X \in FV}$$

3.
$$\frac{\Gamma, x : T, \Gamma' \rhd}{\Gamma, x : T, \Gamma' \vdash x : T}$$

4.
$$\frac{\Gamma \vdash o : T \quad \Gamma, x : T' \rhd \quad T \succ_{\mathcal{A}} T'}{\Gamma \vdash o : T'}$$

5.
$$\frac{\Gamma \vdash o : T \quad \Gamma, x : T' \rhd \quad T' \succ_{\mathcal{A}} T}{\Gamma \vdash o : T'}$$

6.
$$\frac{\Gamma \vdash o : T \quad T \sim_{\mathcal{A}} T'}{\Gamma \vdash o : T'}$$

7.
$$\frac{\Gamma, x : T, x' : T' \rhd}{\Gamma, x'' : [\prod; x](T, T') \rhd}$$

8.
$$\frac{\Gamma \vdash o : \mathcal{U}_M}{\Gamma, x : [El](o) \rhd}$$

9.
$$\frac{\Gamma \rhd}{\Gamma \vdash u_M : \mathcal{U}_{M+1}}$$

10.
$$\frac{\Gamma, x : T_1 \vdash o : T_2}{\Gamma \vdash [\lambda; x](T_1, o) : [\prod; x](T_1, T_2)}$$

11.
$$\frac{\Gamma \vdash o_1 : [\prod; x](T_1, T_2) \quad \Gamma \vdash o_2 : T_1}{\Gamma \vdash [ev; x](o_1, o_2, T_2) : T_2[o_2/x]}$$

12.
$$\frac{\Gamma \rhd \quad M_1 \leq_{\mathcal{A}} M_2}{\Gamma \vdash [j_{M_1, M_2}] : [\prod; x](\mathcal{U}_{M_1}, \mathcal{U}_{M_2})}$$

13.
$$\frac{\Gamma \vdash o_1 : \mathcal{U}_{M_1} \quad \Gamma, x : [El](o_1) \vdash o_2 : \mathcal{U}_{M_2}}{\Gamma \vdash [forall_{M_1, M_2}; x](o_1, o_2) : \mathcal{U}_{max(M_1, M_2)}}$$

14.

$$\frac{\Gamma, x : T_1, y : T_2 \rhd}{\Gamma, y : [\sum; x](T_1, T_2) \rhd}$$

15.

$$\frac{\Gamma, x : T_1, y : T_2 \rhd \quad \Gamma \vdash o_1 : T_1 \quad \Gamma \vdash o_2 : T_2[o_1/x]}{\Gamma \vdash [pair; x](o_1, o_2, T_2) : [\sum; x](T_1, T_2)}$$

16.

$$\frac{\Gamma \vdash a : [\sum; x](T_1, T_2)}{\Gamma \vdash [pr1; x](T_1, T_2, a) : T_1}$$

17.

$$\frac{\Gamma \vdash a : [\sum; x](T_1, T_2)}{\Gamma \vdash [pr2; x](T_1, T_2, a) : T_2[[pr1; x](T_1, T_2, a)/x]}$$

18.

$$\frac{\Gamma \vdash o_1 : \mathcal{U}_{M_1} \quad \Gamma, x : [El](o_1) \vdash o_2 : \mathcal{U}_{M_2}}{\Gamma \vdash [total_{M_1.M_2}; x](o_1, o_2) : \mathcal{U}_{max(M_1,M_2)}}$$

19.

$$\frac{\Gamma \rhd}{\Gamma \vdash pt : \mathcal{U}_0}$$

20.

$$\frac{\Gamma \rhd}{\Gamma \vdash tt : Pt}$$

21.

$$\frac{\Gamma, x : Pt, y : T \rhd \quad \Gamma \vdash o : T[tt/x]}{\Gamma \vdash [pt_r; x](o, T) : [\prod; x](Pt, T)}$$

22.

$$\frac{\Gamma \vdash o_1 : \mathcal{U}_{M_1} \quad \Gamma \vdash o_1 : \mathcal{U}_{M_2}}{\Gamma \vdash [coprod_{M_1,M_2}](o_1, o_2) : \mathcal{U}_{max(M_1,M_2)}}$$

23.

$$\frac{\Gamma, x_1 : T_1 \rhd \quad \Gamma, x_2 : T_2 \rhd}{\Gamma, x : [\amalg](T_1, T_2) \rhd}$$

24.

$$\frac{\Gamma \vdash o_1 : T_1 \quad \Gamma, x_2 : T_2 \rhd}{\Gamma \vdash [ii1](T_1, T_2, o_1) : [\amalg](T_1, T_2)}$$

25.

$$\frac{\Gamma, x_1 : T_1 \rhd \quad \Gamma \vdash o_2 : T_2}{\Gamma \vdash [ii2](T_1, T_2, o_2) : [\amalg](T_1, T_2)}$$

26.

$$\frac{\Gamma, x : [\amalg](T_1, T_2), y : S \rhd \quad \begin{array}{c} \Gamma \vdash s_1 : [\prod; x_1](T_1, S[[ii1](T_1, T_2, x_1)/o]) \\ \Gamma \vdash s_2 : [\prod; x_2](T_2, S[[ii2](T_1, T_2, x_2)/o]) \end{array} \quad \Gamma \vdash o : [\amalg](T_1, T_2)}{\Gamma \vdash [sum; x](T_1, T_2, s_1, s_2, o, S) : S[o/x]}$$

27.

$$\frac{\Gamma \rhd}{\Gamma \vdash empty : \mathcal{U}_0}$$

28.

$$\frac{\Gamma, x : T \rhd \qquad \Gamma \vdash o : \emptyset}{\Gamma \vdash empty_r(T, o) : T}$$

29.

$$\frac{\Gamma \rhd}{\Gamma \vdash nat : \mathcal{U}_0}$$

30.

$$\frac{\Gamma \rhd}{\Gamma \vdash O : \mathbf{N}}$$

31.

$$\frac{\Gamma \rhd}{\Gamma \vdash S : [\prod; x](\mathbf{N}, \mathbf{N})}$$

32.

$$\frac{\Gamma, x : \mathbf{N}, y : T \rhd \qquad \begin{array}{c} \Gamma \vdash o_1 : T[O/x] \\ \Gamma \vdash o_2 : [\prod; x](\mathbf{N}, [\prod; y](T, T[[ev; z](S, x, \mathbf{N})/x])) \end{array} \qquad \Gamma \vdash n : \mathbf{N}}{\Gamma \vdash [nat_r; x](o_1, o_2, n, T) : T[n/x]}$$

33.

$$\frac{\Gamma \vdash t : \mathcal{U}_M \quad \Gamma \vdash o_1 : [El][t] \quad \Gamma \vdash o_2 : [El][t]}{\Gamma \vdash [paths_M](t, o_1, o_2) : \mathcal{U}_M}$$

34.

$$\frac{\Gamma \vdash o_1 : T \quad \Gamma \vdash o_2 : T}{\Gamma, y : [Id](T, o_1, o_2) \rhd}$$

35.

$$\frac{\Gamma \vdash o : T}{\Gamma \vdash [refl](T, o) : [Id](T, o, o)}$$

36.

$$\frac{\Gamma \vdash a : T \qquad \begin{array}{cc} \Gamma, x : T, e : [Id](T, a, x), z : S \rhd & \Gamma \vdash b : T \\ \Gamma \vdash q : S[a/x, [refl](T, a)/e] & \Gamma \vdash i : [Id](T, a, b) \end{array}}{\Gamma \vdash [J; x, e](T, a, b, q, i, S) : S[b/x, i/e]}$$

37.

$$\frac{\Gamma \vdash s : \mathcal{U}_{M_2} \quad \Gamma \vdash t : \mathcal{U}_{M_1} \quad M_1 \leq_{\mathcal{A}} M_2 \quad \Gamma \vdash e : Weq * s * t}{\Gamma \vdash [rr0_{M_2, M_1}](s, t, e) : \mathcal{U}_{M_1}}$$

38.

$$\frac{\Gamma \vdash a : \mathcal{U}_M \quad \Gamma \vdash p : Isaprop * a}{\Gamma \vdash [rr1_M](a, p) : \mathcal{U}_0}$$

14 Appendix B. Complete list of reductions

1. Elforall-reduction.

$$[El][forall_{M_1,M_2};x](o_1,o_2) \succ_{\mathcal{A}} [\prod;x]([El](o_1),[El](o_2))$$

2. Eltotal-reduction.

$$[El][total_{M_1,M_2};x](o_1,o_2) \succ_{\mathcal{A}} [\sum;x]([El](o_1),[El](o_2))$$

3. Elcoprod-reduction.

$$[El][coprod_{M_1,M_2}](T_1,T_2) \succ_{\mathcal{A}} [\amalg]([El](o_1),[El](o_2))$$

4. Elpaths-reduction.

$$[El][paths_M](t,o_1,o_2) \succ_{\mathcal{A}} [Id]([El](t),o_1,o_2)$$

5. Elj-reduction

$$[El][ev;y]([j_{M_1,M_2}],o,T) \succ_{\mathcal{A}} [El](o)$$

6. jMM-reduction.

$$[j_{M_1,M_2}] \succ_{\mathcal{A}} [\lambda;x](\mathcal{U}_{M_1},[x]) \quad iff \quad M_1 \equiv_{\mathcal{A}} M_2$$

7. beta-reduction.

$$[ev;z]([\lambda;x](T_1,o_1),o_2,T_2) \succ_{\mathcal{A}} o_1[o_2/x]$$

8. jj-reduction.

$$[ev;z]([j_{M_2',M_3}],[ev;y]([j_{M_1,M_2}],o_1,T),T') \succ_{\mathcal{A}} [ev;z]([j_{M_1,M_3}],o_1,T')$$

$$iff \quad M_2' \equiv_{\mathcal{A}} M_2$$

9. eta-reduction.

$$[\lambda;x](T_1,[ev;y](o,[x],T_2)) \succ_{\mathcal{A}} o \quad iff\, f\, and\, T\, do\, not\, depend\, on\, x$$

10. foralllja-reduction.

$$[forall_{M_1,M_2};x]([ev;z]([j_{M_0,M_1'}],o_1,T),o_2) \succ_{\mathcal{A}}$$

$$[ev;z]([j_{max(M_0,M_2),max(M_1,M_2)}],[forall_{M_0,M_2};x](o_1,o_2),\mathcal{U}_{max(M_1,M_2)})$$

$$iff \quad M_1' \equiv_{\mathcal{A}} M_1$$

11. foralljb-reduction.

$$[forall_{M_1,M_3};x](o_1,[ev;z]([j_{M_2,M_3'}],o_2,T)) \succ_{\mathcal{A}}$$
$$[ev;z]([j_{max(M_1,M_2),max(M_1,M_3)}],[forall_{M_1,M_2};x](o_1,o_2),\mathcal{U}_{max(M_1,M_3)})$$
$$iff \quad M_3' \equiv_{\mathcal{A}} M_3$$

12. iotatotal1-reduction.

$$[pr1;y](T_1,T_2,[pair;x](o_1,o_2,T_2')) \succ_{\mathcal{A}} o_1 \quad iff \quad T_2[y/x] \sim_{\mathcal{A}} T_2'$$

13. iotatotal2-reduction.

$$[pr2;x](T_1,T_2',[pair;y](o_1,o_2,T_2')) \succ_{\mathcal{A}} o_2 \quad iff \quad T_2[y/x] \sim_{\mathcal{A}} T_2'$$

14. etatotal-reduction.

$$[pair;x]([pr1;y](T_1',T_2',o'),[pr2;y](T_1'',T_2'',o''),T_2) \succ_{\mathcal{A}} o'$$
$$iff \quad o \equiv_{\mathcal{A}} o' \quad and \quad T_2'[x/y] \sim_{\mathcal{A}} T_2 \sim_{\mathcal{A}} T_2''[x/y]$$

15. totalja-reduction.

$$[total_{M_1,M_2}]([ev;z]([j_{M_0,M_1'}],o_1,T),o_2) \succ_{\mathcal{A}}$$
$$[ev;z]([j_{max(M_0,M_2),max(M_1,M_2)}],[total_{M_0,M_2};x](o_1,o_2),\mathcal{U}_{max(M_1,M_2)})$$
$$iff \quad M_1' \equiv_{\mathcal{A}} M_1$$

16. totaljb-reduction.

$$[total_{M_1,M_3}](o_1,[ev;z]([j_{M_2,M_3'}],o_2,T)) \succ_{\mathcal{A}}$$
$$[ev;z]([j_{max(M_1,M_2),max(M_1,M_3)}],[total_{M_1,M_2};x](o_1,o_2),\mathcal{U}_{max(M_1,M_3)})$$
$$iff \quad M_3' \equiv_{\mathcal{A}} M_3$$

17. iotapt-reduction.

$$[ev;x]([pt_r;y](o;T),tt,T') \succ_{\mathcal{A}} o \quad iff \quad T' \sim_{\mathcal{A}} T[x/y]$$

18. sum1-reduction.

$$[sum;x](T_1,T_2,s_1,s_2,[ii1](T_1',T_2',o)) \succ_{\mathcal{A}} [ev;x](s_1,o,[\text{II}](T_1,T_2))$$

19. sum2-reduction.

$$[sum;x](T_1,T_2,s_1,s_2,[ii2](T_1',T_2',o)) \succ_{\mathcal{A}} [ev;x](s_2,o,[\text{II}](T_1,T_2))$$

20. coprodja-reduction.

$$[coprod_{M_1,M_2}]([ev;z]([j_{M_0,M_1}],o_1,T),o_2) \succ_{\mathcal{A}}$$

$$[ev;z]([j_{max(M_0,M_2),max(M_1,M_2)}],[coprod_{M_0,M_2}](o_1,o_2),\mathcal{U}_{max(M_1,M_2)})$$

$$iff \quad M_1 \equiv_{\mathcal{A}} M_1'$$

21. coprodjb-reduction.

$$[coprod_{M_1,M_3}](o_1,[ev;z]([j_{M_2,M_3'}],o_2,T)) \succ_{\mathcal{A}}$$

$$[ev;z]([j_{max(M_1,M_2),max(M_1,M_3)}],[coprod_{M_1,M_2}](o_1,o_2),\mathcal{U}_{max(M_1,M_3)})$$

$$iff \quad M_3 \equiv_{\mathcal{A}} M_3'$$

22. natO-reduction.

$$[nat_r;x](o_1,o_2,O,T) \succ_{\mathcal{A}} o_1$$

23. natS-reduction.

$$[nat_r;x](o_1,o_2,[ev;y](S,n,\mathbf{N}),T) \succ_{\mathcal{A}} [ev;x]([ev;y](o_2,n,\mathbf{N}),[nat_r;x](o_1,o_2,n,T),T)$$

24. pathsj-reduction.

$$[paths_{M_2}]([ev;x]([j_{M_1,M_2'}],t,T),o_1,o_2) \succ_{\mathcal{A}} [ev;x]([j_{M_1,M_2'}],[paths_{M_1}](t,o_1,o_2),T)$$

$$iff \quad M_2 \equiv_{\mathcal{A}} M_2'$$

25. Jiota-reduction.

$$[J;x,e](T,a,a',q,[refl](T',a''),S) \succ_{\mathcal{A}} q \quad iff \quad a \equiv_{\mathcal{A}} a' \equiv_{\mathcal{A}} a'' \quad and \quad T \sim_{\mathcal{A}} T'$$

26. Elrr0-reduction.

$$[El][rr0_{M_2,M_1}](s,t,e) \succ_{\mathcal{A}} [El](s)$$

27. forallrr0a-reduction.

$$S = [forall_{M_1,M_2};x]([rr0_{M_3,M_1'}](s,t,e),o_2)$$

such that $M_1 \equiv_{\mathcal{A}} M_1'$ and $M_1' \leq_{\mathcal{A}} M_3$ reduces to

$$[rr0_{max(M_3,M_2),max(M_1,M_2)}]([forall_{M_3,M_2};x](s,o_2),S,idweq\,S)$$

28. forallrr0b-reduction.

$$S = [forall_{M_1,M_2};x](o_1,[rr0_{M_3,M_2'}](s,t,e))$$

such that $M_2 \equiv_{\mathcal{A}} M_2'$ and $M_2' \leq_{\mathcal{A}} M_3$ reduces to

$$[rr0_{max(M_1,M_3),max(M_1,M_2)}]([forall_{M_1,M_3};x](o_1,s),S,idweq\,S)$$

29. totalrr0a-reduction.

$$S = [total_{M_1,M_2};x]([rr0_{M_3,M_1'}](s,t,e),o_2)$$

such that $M_1 \equiv_{\mathcal{A}} M_1'$ and $M_1' \leq_{\mathcal{A}} M_3$ reduces to

$$[rr0_{max(M_3,M_2),max(M_1,M_2)}]([total_{M_3,M_2};x](s,o_2),S,idweq\,S)$$

30. totalrr0b-reduction.

$$S = [total_{M_1,M_2};x](o_1,[rr0_{M_3,M_2'}](s,t,e))$$

such that $M_2 \equiv_{\mathcal{A}} M_2'$ and $M_2' \leq_{\mathcal{A}} M_3$ reduces to

$$[rr0_{max(M_1,M_3),max(M_1,M_2)}]([total_{M_1,M_3};x](o_1,s),S,idweq\,S)$$

31. coprodrr0a-reduction.

$$S = [coprod_{M_1,M_2}]([rr0_{M_3,M_1'}](s,t,e),o_2)$$

such that $M_1 \equiv_{\mathcal{A}} M_1'$ and $M_1' \leq_{\mathcal{A}} M_3$ reduces to

$$[rr0_{max(M_3,M_2),max(M_1,M_2)}]([coprod_{M_3,M_2}](s,o_2),S,idweq\,X)$$

32. coprodrr0b-reduction.

$$S = [coprod_{M_1,M_2}](o_1,[rr0_{M_3,M_2'}](s,t,e))$$

such that $M_2 \equiv_{\mathcal{A}} M_2'$ and $M_2' \leq_{\mathcal{A}} M_3$ reduces to

$$[rr0_{max(M_1,M_3),max(M_1,M_2)}]([coprod_{M_1,M_3}](o_1,s),S,idweq\,S)$$

33. pathsrr0-reduction.

$$S = [paths_{M_1}]([rr0_{M_2,M_1'}](s,t,e),o_1,o_2)$$

such that $M_1 \equiv_{\mathcal{A}} M_1'$ and $M_1' \leq_{\mathcal{A}} M_2$ reduces to

$$[rr0_{M_2,M_1}]([paths_{M_2}](s,o_1,o_2),S,idweq\,S)$$

34. rr1rr0-reduction.

$$[rr1_M](a,p) \succ_{\mathcal{A}} [rr0_{M,0}](a,[rr1_M](a,p),idweq\,[rr1_M](a,p))$$

35. jrr0j-reduction.

$$j_{M_1',M_3}\,[rr0_{M_2,M_1}](s,t,e) \succ_{\mathcal{A}} j_{M_2,M_3}\,s$$
$$iff \quad M_1' \equiv_{\mathcal{A}} M_1 \quad and \quad M_1 \leq_{\mathcal{A}} M_2 \leq_{\mathcal{A}} M_3$$

36. jrr0rr0-reduction.

$$j_{M_1',M_2}[rr0_{M_3,M_1}](s,t,e) \succ_{\mathcal{A}} [rr0_{M_3,M_2}](s, j_{M_1,M_2}\, t, e)$$

$$iff \quad M_1' \equiv_{\mathcal{A}} M_1 \quad and \quad M_1 \leq_{\mathcal{A}} M_2 \leq_{\mathcal{A}} M_3$$

37. rr0jj-reduction.

$$[rr0_{M_2,M_1}]((j_{M_0,M_2'}\, s), t, e) \succ_{\mathcal{A}} j_{M_0,M_1}\, s$$

$$iff \quad M_2 \equiv_{\mathcal{A}} M_2' \quad and \quad M_0 \leq_{\mathcal{A}} M_1 \leq_{\mathcal{A}} M_2$$

.

38. rr0jrr0-reduction.

$$[rr0_{M_3,M_1}]((j_{M_2,M_3'}\, s), t, e) \succ_{\mathcal{A}} [rr0_{M_2,M_1}](s,t,e)$$

$$iff \quad M_3 \equiv_{\mathcal{A}} M_3' \quad and \quad M_0 \leq_{\mathcal{A}} M_1 \leq_{\mathcal{A}} M_2$$

39. rr0rr0-reduction.

$$[rr0_{M_2,M_1}]([rr0_{M_3,M_2'}](s,t_2,e_2),t_1,e_1) \succ_{\mathcal{A}} [rr0_{M_3,M_1}](s,t_1,e_1)$$

$$iff \quad M_2 \equiv_{\mathcal{A}} M_2' \quad and \quad M_1 \leq_{\mathcal{A}} M_2 \leq_{\mathcal{A}} M_3$$

40. rr0MM-reduction.

$$[rr0_{M,M'}](s,t,e) \succ_{\mathcal{A}} s \quad iff \quad M \equiv_{\mathcal{A}} M'$$

15 Adding the natural numbers - system TS5

15.1 TS5 terms and typing function

Definition 15.1.1 *[**d51**] The following labels are permitted in the expressions of TS5 - the labels permitted in TS4, nat, O, S, nat_r.*

The notions of u-level expressions, T- and o- terms in TS5 are defined as follows:

Definition 15.1.2 *[**d52**]*

1. *expressions with the root node caring a TS4-label is an u-level expression, o-expression or a T-expression according to the rules of Definition 8.1.2,*

2. *expressions with the root of the form $[nat]$, O, S and $[nat_r; x]$ are o-expressions,*

Definition 15.1.3 *[**d53**] A TS5-term is a TS5-expression such that:*

1. *any node caring one of the TS4-labels satisfies the conditions of Definition 7.1.3,*

2. *any node of the form* $[nat]$ *has valency* 0*,*

3. *any node of the form* $[O]$ *has valency* 0*,*

4. *any node of the form* $[S]$ *has valency* 0*,*

5. *any node of the form* $[nat_r]$ *has valency* 4*, its first three branches are o-expressions which do not contain* x *and the fourth branch a T-expression.*

Definition 15.1.4 *[***d54***] A node in TS5 term is called non-essential if it satisfies the conditions of Definition 6.1.4. The same applies to the definition of essential nodes, essential and non-essential subexpressions and of* $Ess(E)$ *is extended to TS5-terms in the obvious way.*

We let $TS5$ denote the set of TS5-terms and $TT5$ and $oT5$ denote the subsets of T-terms and o-terms. The obvious analog of Lemma 3.1.7 holds for TS5-expressions. We extend to TS5 the abbreviations introduced for TS4-expressions and also abbreviate $[El][nat]$ as $\mathbf{N}$.

Definition 15.1.5 *[***dtau5***] Under the assumptions of Definition 3.1.8 we extend the typing function* τ_Γ *to o-terms of TS5 as follows:*

1. *the value of* τ_Γ *on an o-term whose root node carries a TS4-label is computed according to the rules of Definition 8.1.5.*

2. $\tau_\Gamma([nat]) = \mathcal{U}_0$,

3. $\tau_\Gamma([O]) = \mathbf{N}$,

4. $\tau_\Gamma([S]) = [\prod; x](\mathbf{N}, \mathbf{N})$,

5. $\tau_\Gamma([nat_r; x](o_1, o_2, o_3, T)) = T$.

We have the obvious analog of Proposition 3.1.9 for TS5.

15.2 Equivalence relations $\equiv_{\mathcal{A}}$ and $\sim_{\mathcal{A}}$ on TS5-terms

The relations $\equiv_{\mathcal{A}}$ and $\sim_{\mathcal{A}}$ are extended to TS1-terms in the obvious way. We also have obvious analogs of all the statements of Section 3.2.

15.3 Reducibility relation on TS5 terms

Definition 15.3.1 *[***drd5***] Let* $UC = (Fu, \mathcal{A})$ *be universe context and* FV *a sequence of T-variables. Define a relation* $\succ_{\mathcal{A}}$ *on* $TS5(Fu, FV, Fv)$ *for all* Fv *as the transitive closure of the union of the following reduction relations:*

1. *Reductions of Definition 7.3.1.*

2. *natO-reduction. An essential subterm of the form* $[nat_r; x](o_1, o_2, O, T)$ *reduces to* o_1,

3. *natS-reduction. An essential subterm of the form* $[nat_r; x](o_1, o_2, [ev; y](S, n, \mathbf{N}), T)$ *reduces to* $[ev; x]([ev; y](o_2, n, \mathbf{N}), [nat_r; x](o_1, o_2, n, T), T)$.

The obvious analogs of Lemmas 4.1.3-4.1.6 hold for TS5.

15.4 Local confluence for general terms of TS5

There are no new confluence situations for TS5 compared with TS5.

15.5 Derivation rules of TS5

The derivation rules for contexts and judgements of TS5 are the derivation rules for TS4 together with the following additional ones:

$$\frac{\Gamma \rhd}{\Gamma \vdash nat : \mathcal{U}_0}$$

$$\frac{\Gamma \rhd}{\Gamma \vdash O : \mathbf{N}}$$

$$\frac{\Gamma \rhd}{\Gamma \vdash S : [\prod; x](\mathbf{N}, \mathbf{N})}$$

$$\frac{\Gamma, x : \mathbf{N}, y : T \rhd \qquad \begin{array}{c} \Gamma \vdash o_1 : T[O/x] \\ \Gamma \vdash o_2 : [\prod; x](\mathbf{N}, [\prod; y](T, T[[ev; z](S, x, \mathbf{N})/x])) \end{array} \qquad \Gamma \vdash n : \mathbf{N}}{\Gamma \vdash [nat_r; x](o_1, o_2, n, T) : T[n/x]}$$