A test type system

Vladimir Voevodsky

Started January 25, 2013

Contents

1 Rules . . . 1
2 Derivation trees . . . 5
3 Main structural properties . . . 5
4 TTS as a generalized algebraic theory . . . 10

This note gives some ideas about the test type system TTS with secondary witnessed which Dan Grayson and I have been working on implementing. While TTS by itself has (most likely) decidable definitional equality and typing making secondary witnesses to be formally speaking unnecessary, they become essential for the implementation of more complex systems with undecidable typing such as HTS.

1 Rules

Sequences of expression of the form

$$x_1 : T_1, \ldots, x_n : T_n \rhd$$

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

$$x_1 : T_1, \ldots, x_n : T_n \vdash T \stackrel{d}{=} T'$$

$$x_1 : T_1, \ldots, x_n : T_n \vdash o \stackrel{d}{=} o' : T$$

where $x_1, \ldots, x_n$ are names of variables, T_i is an expression with free variables from $\{x_1, \ldots, x_{i-1}\}$ and o, o', T, T' are expressions with free variables from the set $\{x_1, \ldots, x_n\}$ are called *sentences* of the type system.

Sequences of expressions of the form

$$x_1 : T_1, \ldots, x_n : T_n \rhd$$

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

$$x_1 : T_1, \ldots, x_n : T_n \vdash p : T :\stackrel{d}{=} T'$$

$$x_1 : T_1, \ldots, x_n : T_n \vdash p : o \stackrel{d}{=} o' : T$$

satisfying the same properties as above and such that p is an expression with free variables from $\{x_1, \ldots, x_n\}$ are called extended sentences.

We are aiming at a type system where every derivable extended sentence can be obtained by a unique inference rules such that one gets a bijection between inference trees and α-equivalence classes of derivable extended sentences.

General inference rules

1.

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

2. for each $X \in FV$ and x not in $v(\Gamma)$

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

Note: the condition that x is not one of the variables declared in Γ is essential since otherwise it is possible that a sentence of the form $\Gamma, x : T, \Gamma' \vdash x : T$ can be obtained by two different inference rules of the family of rules given in the next item. A possible alternative(?) is to include in the next item the condition that x is not in $v(\Gamma')$.

3. for each $i \in \mathbf{N}$

$$\frac{\Gamma, x : T, \Gamma' \rhd \qquad T \sim T'}{\Gamma, x : T, \Gamma' \vdash [wd](x) : x : T'}$$

Note: The occurrence of x in $[wd](x)$ is called a special occurrence. When we write $E[o/x]$ this refers to the expression obtained from E by replacing x with o in all non-special occurrences. In all cases when we do that we also replace $[wd](x)$ by some expression other than o.

4.

$$\frac{\Gamma, x : T \rhd \qquad \Gamma, x : T' \rhd \qquad T \sim T'}{\Gamma \vdash [Wrefl] : T \stackrel{d}{=} T'}$$

5.

$$\frac{\Gamma \vdash p : T_1 \stackrel{d}{=} T_2}{\Gamma \vdash [Wsymm](p) : T_2 \stackrel{d}{=} T_1}$$

6.

$$\frac{\Gamma \vdash p12 : T_1 \stackrel{d}{=} T_2 \qquad \Gamma \vdash p23 : T_2 \stackrel{d}{=} T_3}{\Gamma \vdash [Wtrans](p12, p23, T_2) : T_1 \stackrel{d}{=} T_3}$$

7.

$$\frac{\Gamma \vdash p : o : T \qquad \Gamma \vdash p' : o' : T \qquad o \sim o'}{\Gamma \vdash [wrefl](p, p') : o \stackrel{d}{=} o' : T}$$

8.

$$\frac{\Gamma \vdash p : o_1 \stackrel{d}{=} o_2 : T}{\Gamma \vdash [wsymm](p) : o_2 \stackrel{d}{=} o_1 : T}$$

9.

$$\frac{\Gamma \vdash p12 : o_1 \stackrel{d}{=} o_2 : T \qquad \Gamma \vdash p23 : o_2 \stackrel{d}{=} o_3 : T}{\Gamma \vdash [wtrans](p12, p23, o_2) : o_1 \stackrel{d}{=} o_3 : T}$$

10.

$$\frac{\Gamma \vdash p : o : T \qquad \Gamma \vdash p' : T \stackrel{d}{=} T'}{\Gamma \vdash [wconv](p, p', T) : o : T'}$$

11. (*) for each $n \in \mathbf{N}$ and $X \in FV$

$$\frac{\Gamma, x_1 : T_1, \ldots, x_n : T_n \rhd \quad \Gamma \vdash q : S = S'}{\Gamma, x_1 : T_1[S/X], \ldots, x_{n-1} : T_{n-1}[S/X] \vdash q_n : T_n[S/X] = T_n[S'/X][p_1/wd(x_1), \ldots, p_{n-1}/wd(x_{n-1})]}$$

where

$$q_i = Ws(q, S, S', i, X, T_1, \ldots, T_i)$$

$$p_i = wconv(wd(x_i), q_i, T_i[S/X])$$

and Γ does not contain X.

12. (*) for each $n \in \mathbf{N}$ and $X \in FV$

$$\frac{\Gamma, x_1 : T_1, \ldots, x_n : T_n \vdash p : o : T \quad \Gamma \vdash q : S = S'}{\Gamma, x_1 : T_1[S/X], \ldots, x_n : T_n[S/X] \vdash r : o[S/X] = o[S'/X][p_1/wd(x_1), \ldots, p_n/wd(x_n)] : T[S/X]}$$

where

$$q_i = Ws(q, S, S', i, T_1, \ldots, T_i)$$

$$p_i = wconv(wd(x_i), q_i, T_i[S/X])$$

$$r = ws(q, S, S', i, T_1, \ldots, T_n, p, o, T)$$

and Γ does not contain X.

13. (*) for each $n \in \mathbf{N}$

$$\frac{\Gamma, x : S, x_1 : T_1, \ldots, x_n : T_n \rhd \quad \Gamma \vdash p : o : S \quad \Gamma \vdash p' : o' : S \quad \Gamma \vdash r : o = o' : S}{\Gamma, x_1 : To_1, \ldots, x_{n-1} : To_{n-1} \vdash q_n : To_n = To'_n[p_1/wd(x_1), \ldots, p_{n-1}/wd(x_{n-1})]}$$

where

$$To_i = T_i[o/x, p/wd(x)]$$

$$To'_n = T_n[o'/x, p'/wd(x)]$$

$$q_i = ws(r, o, o', S', i, T_1, \ldots, T_i)$$

$$p_i = wconv(wd(x_i), q_i, To_i)$$

14.

$$\frac{\Gamma \vdash p : o \overset{d}{=} o' : T \quad \Gamma \vdash p' : T \overset{d}{=} T'}{\Gamma \vdash [wconveq](p, p', T) : o \overset{d}{=} o' : T'}$$

Universe

1. for x not in $v(\Gamma)$

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

2. for x not in $v(\Gamma)$

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

3.

$$\frac{\Gamma \vdash peq : o = o' : \mathcal{U} \quad \Gamma \vdash p : o : \mathcal{U} \quad \Gamma \vdash p' : o' : \mathcal{U}}{\Gamma \vdash [weleq](peq, p, p') : [El](o, p) = [El](o', p')}$$

Dependent products We let $[wch](x,p)$ denote $[[wconv]([wd](x),[wsymm](p))/[wd](x)]$.

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

2. $$\frac{\Gamma, x : T_1, y : T_2 \rhd \quad \Gamma \vdash p : T_1 \stackrel{d}{=} T_1'}{\Gamma \vdash [wpi1](p) : [\prod; x](T_1, T_2) \stackrel{d}{=} [\prod; x](T_1', T_2[wch](x,p))}$$

3. $$\frac{\Gamma, x : T_1, y : T_2 \rhd \quad \Gamma, x : T_1 \vdash p : T_2 \stackrel{d}{=} T_2'}{\Gamma \vdash [wpi2; x](p) : [\prod; x](T_1, T_2) \stackrel{d}{=} [\prod; x](T_1, T_2')}$$

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

5. $$\frac{\Gamma \vdash p_1 : T_1 \stackrel{d}{=} T_1' \quad \Gamma, x : T_1 \vdash p_2 : o : T_2}{\Gamma \vdash [wl1](p_1, p_2) : [\lambda; x](T_1, o) \stackrel{d}{=} [\lambda; x](T_1', o[wch](x,p)) : [\prod; x](T_1', T_2[wch](x,p))}$$

6. $$\frac{\Gamma, x : T_1 \vdash p : o \stackrel{d}{=} o' : T_2}{\Gamma \vdash [wl2](p) : [\lambda; x](T_1, o) \stackrel{d}{=} [\lambda; x](T_1, o') : [\prod; x](T_1, T_2)}$$

7. $$\frac{\Gamma \vdash p_f : f : [\prod; x](T_1, T_2) \quad \Gamma \vdash p_o : o : T_1}{\Gamma \vdash [wev](p_f, p_o) : [ev; x](f, o, T_1, T_2) : T_2[o/x, p_o/[wd](x)]}$$

8. $$\frac{\Gamma \vdash p_{t1} : T_1 \stackrel{d}{=} T_1' \quad \Gamma \vdash p_f : f : [\prod; x](T_1, T_2) \quad \Gamma \vdash p_o : o : T_1}{\Gamma \vdash [wevt1](p_{t1}, p_f, p_o) : [ev; x](f, o, T_1, T_2) = [ev; x](f, o, T_1', T_2[wch](x, p_{t1})) : T_2[o/x, p_o/[wd](x)]}$$

9. $$\frac{\Gamma, x : T_1 \vdash p_{t2} : T_2 \stackrel{d}{=} T_2' \quad \Gamma \vdash p_f : f : [\prod; x](T_1, T_2) \quad \Gamma \vdash p_o : o : T_1}{\Gamma \vdash [wevt2](p_{t2}, p_f, p_o) : [ev; x](f, o, T_1, T_2) = [ev; x](f, o, T_1, T_2') : T_2[o/x, p_o/[wd](x)]]}$$

10. $$\frac{\Gamma \vdash p_{feq} : f = f' : [\prod; x](T_1, T_2) \quad \Gamma \vdash p_o : o : T_1}{\Gamma \vdash [wevf](p_{feq}, p_o) : [ev; x](f, o, T_1, T_2) = [ev; x](f', o, T_1, T_2) : T_2[o/x, p_o/[wd](x)]]}$$

11. $$\frac{\Gamma \vdash p_f : f : [\prod; x](T_1, T_2) \quad \Gamma \vdash p_{oeq} : o = o' : T_1 \quad \Gamma \vdash p_o : o : T_1}{\Gamma \vdash [wevo](p_f, p_{oeq}, p_o) : [ev; x](f, o, T_1, T_2) = [ev; x](f, o', T_1, T_2) : T_2[o/x, p_o/[wd](x)]]}$$

12. $$\frac{\Gamma \vdash p_1 : o_1 : T_1 \quad \Gamma, x : T_1 \vdash p_2 : o_2 : T_2}{\Gamma \vdash [wbeta](p_1, p_2) : [ev; y]([\lambda; x](T_1, o_2), o_1, T_1, T_2) \stackrel{d}{=} o_2[o_1/x, p_1/[wd](x)] : T_2[o_1/x, p_1/[wd](x)]}$$

13.

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

2 Derivation trees

Definition 2.1 *[**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 2.2 *[**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 2.3 *[**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 2.3 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 four kinds of derivation trees - the ones which define four kinds of sentences.
3. Each vertex of a derivation tree is labelled by the (number or name of) the corresponding inference rule. The kind of the branch corresponding to a given vertex is completely determined by the label of the vertex.

3 Main structural properties

Lemma 3.1 *[**untree**] Let S be a derivable extended sentence of TTS. Then it is obtainable by an exactly one inference rule.*

Proof: Straightforward.

Lemma 3.2 *[**dertree**] The derivation tree for a sentence of one of the following forms:*

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

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

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

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

has (a unique) smallest cutting surface whose elements represent sentences of the form $\Gamma \rhd$*.*

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

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

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

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

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

Remark 3.3 Note that the assertion of Lemma 3.2 is not tautological and really depends on the form of the inference 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 inference rules then Lemma 3.2 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 $x : X \rhd$.

As an immediate corollary of Lemma 3.2 we get the following result.

Lemma 3.4 *[**dertree0**] For any derivable sentence of one of the following forms*

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

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

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

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

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

Lemma 3.5 *[**dertree1**] One has the following properties of derivable sentences:*

1. $$\frac{\Gamma, x_1 : T_1 \rhd \quad \Gamma, \Gamma' \rhd}{\Gamma, x_1 : T_1, \Gamma' \rhd}$$

2. $$\frac{\Gamma, x_1 : T_1 \rhd \quad \Gamma, \Gamma' \vdash p : o : T}{\Gamma, x_1 : T_1, \Gamma' \vdash p : o : T}$$

3. $$\frac{\Gamma, x_1 : T_1 \rhd \quad \Gamma, \Gamma' \vdash p : T = T'}{\Gamma, x_1 : T_1, \Gamma' \vdash p : T = T'}$$

4. $$\frac{\Gamma, x_1 : T_1 \rhd \quad \Gamma, \Gamma' \vdash p : o = o' : T}{\Gamma, x_1 : T_1, \Gamma' \vdash p : o = o' : T}$$

where our notation means that if the sentences above the line are derivable then the sentences below the line are.

Proof: Consider the derivation tree of the right hand side sentence above the line relative to Γ. Replace each Γ with $\Gamma, x_1 : T_1 \vdash$.

Remark 3.6 The key to the validity of the proof of Lemma 3.5 is that for any of the inference rules one of the following possibilities holds:

1. the product sentence of the inference 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 inference 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 inference rule.

These conditions would not hold if we had a generating inference 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

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

or if we had a generating inference 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

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

Lemma 3.7 *[**dertree2**] One has the following properties of derivable sentences:*

1. $$\frac{\Gamma, \Gamma'' \rhd \quad \Gamma, \Gamma' \rhd}{\Gamma, \Gamma', \Gamma'' \rhd}$$

2. $$\frac{\Gamma, \Gamma'' \rhd \quad \Gamma, \Gamma' \vdash p : a : T}{\Gamma, \Gamma', \Gamma'' \vdash p : a : T}$$

3. $$\frac{\Gamma, \Gamma'' \rhd \quad \Gamma, \Gamma' \vdash p : T = T'}{\Gamma, \Gamma', \Gamma'' \vdash p : T = T'}$$

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

Proof: By induction on the length of Γ'' using Lemma 3.5.

Lemma 3.8 *[**dertree3**] One has the following properties of derivable sentences:*

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

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

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

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

Proof: By induction on the depth of the derivation tree of the right hand side sentence above the line relative to $\Gamma, x : S\rhd$. If the depth is zero the statement follows from Lemma 3.4. Further we need to consider each of the inference rules assuming that the context Γ is of the form $\Gamma, x : S, \Gamma'$ and verify that after replacing the context by $\Gamma, \Gamma'[a/x, p/[wd](x)]$, all the components of all the judgements J by $J[a/x, p/[wd](x)]$ and assuming that the sentences above the line and $\Gamma \vdash p : a : S$ are derivable we can show that the sentence below the line is derivable.

For example in the $[wpi1]$ rule we get above the line:

$$\Gamma, \Gamma'[a/x, p/[wd](x)], x' : T_1[a/x, p/[wd](x)], y : T_2[a/x, p/[wd](x)]\rhd$$

and

$$\Gamma, \Gamma'[a/x, p/[wd](x)] \vdash (p' : T_1 = T_1')[a/x, p/[wd](x)]$$

and below the line

$$\Gamma, \Gamma'[a/x, p/[wd](x)] \vdash$$

$$([wpi1](p') : [\prod; x'](T_1, T_2) = [\prod; x](T_1', T_2[[wch](x', p')/[wd](x')]))[a/x, p/[wd](x)]$$

and our claim follows from the fact that since $x \neq x'$ we have

$(E[[wch](x',p')/[wd](x')])[a/x,p/[wd](x)] = (E[a/x,p/[wd](x)])[[wch](x',p'[a/x,p/[wd](x)])/[wd](x')]$

Another example is $[wd](x)$ rule. Then above the line we get

$$\Gamma, \Gamma'[a/x, p/[wd](x)] \rhd$$

and below the line

$$\Gamma, \Gamma'[a/x, p/[wd](x)] \vdash ([wd](x) : x : S)[a/x, p/[wd](x)]$$

which equals

$$\Gamma, \Gamma'[a/x, p/[wd](x)] \vdash p : a : S$$

which is derivable by the inductive assumption and Lemma 3.7.

Lemma 3.9 *[***dertree3.1***] One has the following properties of derivable sentences:*

$$\frac{\Gamma \vdash p : T_1 = T_1' \quad \Gamma, x_1 : T_1, \Gamma' \rhd}{\Gamma, x_1' : T_1', \Gamma'[x_1'/x_1, [wch](x',p)/[wd](x)] \rhd}$$

$$\frac{\Gamma \vdash p : T_1 = T_1' \quad \Gamma, x_1 : T_1, \Gamma' \vdash p' : T_2 = T_2'}{\Gamma, x_1' : T_1', \Gamma'[x_1'/x_1, [wch](x',p)/[wd](x)] \vdash (p' : T_2 = T_2')[x_1'/x_1, [wch](x',p)/[wd](x)]}$$

$$\frac{\Gamma \vdash p : T_1 = T_1' \quad \Gamma, x_1 : T_1, \Gamma' \vdash p' : o : T_2}{\Gamma, x_1' : T_1', \Gamma'[x_1'/x_1, [wch](x',p)/[wd](x)] \vdash (p' : o : T_2)[x_1'/x_1, [wch](x',p)/[wd](x)]}$$

$$\frac{\Gamma \vdash p : T_1 = T_1' \quad \Gamma, x_1 : T_1, \Gamma' \vdash p' : o = o' : T_2}{\Gamma, x_1' : T_1', \Gamma'[x_1'/x_1, [wch](x',p)/[wd](x)] \vdash (p' : o = o' : T_2)[x_1'/x_1, [wch](x',p)/[wd](x)]}$$

Proof: By induction on the depth by the second sentence above the line relative to $\Gamma, x_1 : T_1 \rhd$. If the depth is 0 the assertion is obvious. The only non-trivial inference rule is $[wd](x)$ which is easily checked.

Lemma 3.10 *[***dertree4***] One has the following properties of derivable sentences:*

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

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

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

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

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

where the question mark ? means that there exists an expression which makes the corresponding (extended) sentence derivable.

Proof: Let us add the tautological property $\frac{\Gamma \rhd}{\Gamma \rhd}$ for sentences of the first kind and proceed by induction on the derivation depth of the sentence above the line relative to Γ.

The first non-trivial rule to check is the *wpi*1 rule. The inductive step in this case follows from Lemma 3.9. The next one is *wev* which follows from Lemma 3.8.

General non-essential sub-expressions

1. The sub-expressions with root labels wd, $Wrefl$, $Wsymm$, $Wtrans$, $wrefl$, $wsymm$, $wtrans$, $wconv$ and $wconveq$ are non-essential.

4 TTS as a generalized algebraic theory

This section is written to understand a new connection in type theory which I am just beginning to understand. It is based on the ideas of [?]. Cartmell's paper is about "generalized algebraic theories". At some point he makes an assertion:

"The essentially algebraic theories of Freyd [5] can be seen to have the same descriptive power as generalised algebraic theories, at least as far as the usual set valued models are concerned."

which made me think for a long time that "generalized algebraic" and "essentially algebraic" are interchangeable. In fact it is not so. What seems to be true (I am still working on it) is that any generalized algebraic theory defines an essentially algebraic one but many different generalized algebraic theories may generate the same essentially algebraic one.

Here is an example. Consider an essentially algebraic (in fact algebraic) theory of the form $(S_1, S_2, f : S_1 \to S_2)$ whose models are pairs of sets together with a function between them. There are (al least) two *different* generalized algebraic theories which correspond to it. In the approach of describing generalized algebraic theories suggested by Cartmell these two are given by:

Theory 1

$$\vdash A \quad type$$

$$x : A \vdash B(x) \quad type$$

Theory 2

$$\vdash A_1 \quad type$$

$$\vdash A_2 \quad type$$

$$x : A_1 \vdash f(x) : A_2$$

As far as I understand to choose a generalized algebraic theory corresponding to a given essentially algebraic one has to specify, roughly speaking, which of the operations of the later will be translated as type dependencies (display maps) and which as usual functions.

There is always a choice when all operations are translated as functions. Consider for example an essentially algebraic theory of the form $(S_1, S_2, S_3, f_1, f_2 : S_1 \to S_2, g : \{x | f_1 x = f_2 x\} \to S_3)$. This can be translated as:

Theory 1

$$\vdash A_1 \quad type$$

$$\vdash A_2 \quad type$$

$$\vdash A_3 \quad type$$

$$x : A_1 \vdash f_1(x) : A_2$$

$$x : A_1 \vdash f_2(x) : A_2$$

$$x_1 : A_2, x_2 : A_2 \vdash Eq_{A_2}(x_1, x_2) \quad type$$

$$x_1 : A_2, x_2 : A_2, y_1 : Eq_{A_2}(x_1, x_2), y_2 : Eq_{A_2}(x_1, x2) \vdash y_1 = y_2$$

$$x_1 : A_2, x_2 : A_2, y : Eq_{A_2}(x_1, x_2), \vdash x_1 = x_2$$

$$x : A_1, y : Eq_{A_2}(f_1(x), f_2(x)) \vdash g(x, y) : A_3$$

or as:

Theory 2

$$\vdash A_1 \quad type$$

$$x_1 : A_1, x_2 : A_1 \vdash A_2(x_1, x_2) \quad type$$

$$x : A_1, x' : A_2(x, x) \vdash g(x, x') : A_3$$

Notes 1.

$$f : T_1' \to \mathcal{U}, x_1 : T_1', g : El(f\ x_1, p_1') \to \mathcal{U}, x_2 : El(f\ x_1, p_1'), x_3 : El(g\ x_2, p_2')\triangleright$$

$$p_1' = wev(wd(f), wd(x_1), T_1', \mathcal{U})$$

$$p_2' = wev(wd(g), wd(x_2), El(f\ x_1, p_1'), \mathcal{U})$$

2.

$$f : T_1' \to \mathcal{U} \vdash q_1 : T_1' \overset{d}{=} T_1''$$

$$f : T_1' \to \mathcal{U}, x_1 : T_1'', g : El(f\ x_1, p_1'') \to \mathcal{U}, x_2 : El(f\ x_1, p_1''), x_3 : El(g\ x_2, p_2'')\triangleright$$

where

$$p_1'' = wev(wd(f), wconv(wd(x_1), wsymm(q_1), T_1''), T_1', \mathcal{U})$$

$$p_2'' = wev(wd(g), wd(x_2), El(f\ x_1, p_1''), \mathcal{U})$$

3.

$$f : T_1' \to \mathcal{U}, x_1 : T_1'' \vdash q_2 : (El(f\ x_1, p_1'') \to \mathcal{U}) \to T_2''$$

$$f : T_1' \to \mathcal{U}, x_1 : T_1'', g : T_2'', x_2 : El(f\ x_1, p_1''), x_3 : El(g\ x_2, p_2''')\triangleright$$

$$p_2''' = wev(wconv(wd(g), wsymm(q_2), T_2''), wd(x_2), El(f\ x_1, p_1''), \mathcal{U})$$

4.

$\Gamma = (x_1 : T_1, x_2 : T_2(x_1, wd(x_1)), x_3 : T_3(x_1, wd(x_1), x_2, wd(x_2)), x_4 : T_4(x_1, wd(x_1), x_2, wd(x_2), x_3, wd(x_3)))$

$\Gamma' = (x_1 : T_1', x_2 : T_2'(x_1, wd(x_1)), x_3 : T_3'(x_1, wd(x_1), x_2, wd(x_2)), x_4 : T_4'(x_1, wd(x_1), x_2, wd(x_2), x_3, wd(x_3)))$

How to express $q : \Gamma = \Gamma'$?

$$\vdash q_1 : T_1 = T_1'$$

$$x_1 : T_1 \vdash q_2 : T_2(x_1, wd(x_1)) = T_2'(x_1, wconv(wd(x_1), q_1, T_1))$$

$$x_1 : T_1, x_2 : T_2(x_1, wd(x_1)) \vdash q_3 : T_3(x_1, wd(x_1), x_2, wd(x_2)) = \\ T_3'(x_1, wconv(wd(x_1), q_1, T_1), x_2, wconv(wd(x_2), q_2, T_2))$$