Weak Type Theory

1 Definitional equality

The first published version of type theory [3] contains a version which does not allow the ξ-rule. The syntax is non standard and is described by P. Aczel as "unusual, complicated syntax of defined combinators to avoid contracting a redex inside an abstraction that binds a variable in the redex". The goal of this note is to provide an alternative presentation.

2 Model of type theory

A model is given by a collection of *contexts*. If Γ, Δ are context we have a collection $\Delta \to \Gamma$ of *substitutions* from Δ to Γ. This should form a category: we have a substitution $1 : \Gamma \to \Gamma$ and a composition operator $\sigma\delta : \Theta \to \Gamma$ if $\delta : \Theta \to \Delta$ and $\sigma : \Delta \to \Gamma$. Furthermore we should have $\sigma 1 = 1\sigma = \sigma$ and $(\theta\sigma)\delta = \theta(\sigma\delta)$. If Γ is a context we have a collection of *types over* Γ. We write $\Gamma \vdash A$ to express that A is a type over Γ. If $\Gamma \vdash A$ and $\sigma : \Delta \to \Gamma$ we should have $\Delta \vdash A\sigma$. Furthermore $A1 = A$ and $(A\sigma)\delta = A(\sigma\delta)$. If $\Gamma \vdash A$ we have also a colection of *elements of type* A. We write $\Gamma \vdash a : A$ to express that a is an element of type A. If $\Gamma \vdash a : A$ and $\sigma : \Delta \to \Gamma$ we should have $\Delta \vdash a\sigma : A\sigma$. Furthermore $a1 = a$ and $(a\sigma)\delta = a(\sigma\delta)$.

We have a *context extension operation*: if $\Gamma \vdash A$ then we have a new context $\Gamma.A$. Furthermore there is a projection $\mathsf{p} \in \Gamma.A \to \Gamma$ and a special element $\Gamma.A \vdash \mathsf{q} : A\mathsf{p}$. If $\sigma : \Delta \to \Gamma$ and $\Gamma \vdash A$ and $\Delta \vdash a : A\sigma$ we have an extension operation $(\sigma, a) : \Delta \to \Gamma.A$. We should have $\mathsf{p}(\sigma, a) = \sigma$ and $\mathsf{q}(\sigma, a) = a$ and $(\sigma, a)\delta = (\sigma\delta, a\delta)$ and $(\mathsf{p}, \mathsf{q}) = 1$.

If $\Gamma \vdash a : A$ we write $[a] = (1, a) : \Gamma \to \Gamma.A$. Thus if $\Gamma.A \vdash B$ and $\Gamma \vdash a : A$ we have $\Gamma \vdash B[a]$. If furtermore $\Gamma.A \vdash b : B$ we have $\Gamma \vdash b[a] : B[a]$. Models are usually presented by giving a class of special maps (fibrations), in our case they are the maps $\mathsf{p} : \Gamma.A \to \Gamma$, and the elements are the sections of these fibrations, in our case the maps $[a] : \Gamma \to \Gamma.A$ determined by an element $\Gamma \vdash a : A$.

We suppose furthermore one operation $\Pi\ A\ B$ such that $\Gamma \vdash \Pi\ A\ B$ if $\Gamma \vdash A$ and $\Gamma.A \vdash B$. We should have $(\Pi\ A\ B)\sigma = \Pi\ (A\sigma)\ (B\sigma^+)$ where $\sigma^+ = (\sigma\mathsf{p}, \mathsf{q})$. We have an abstraction operation λb such that $\Gamma \vdash \lambda b : \Pi\ A\ B$ if $\Gamma.A \vdash b : B$. We have an application operation such that $\Gamma \vdash \mathsf{app}(c, a) : B[a]$ if $\Gamma \vdash a : A$ and $\Gamma \vdash c : \Pi\ A\ B$. These operations should satisfy the equations

$$\mathsf{app}(\lambda b, a) = b[a], \qquad c = \lambda(\mathsf{app}\ c^+), \qquad (\lambda b)\sigma = \lambda(b\sigma^+), \qquad \mathsf{app}(c, a)\sigma = \mathsf{app}(c\sigma, a\sigma)$$

where we write $c^+ = (c\mathsf{p}, \mathsf{q})$ and $\sigma^+ = (\sigma\mathsf{p}, \mathsf{q})$.

To define a model of type theory with one universe, we assume that we have a special type $\Gamma \vdash U$ such that $U\sigma = U$ and $\Gamma \vdash A$ whenever $\Gamma \vdash A : U$. Furthermore we assume that $\Gamma \vdash \Pi\ A\ B : U$ whenever $\Gamma \vdash A : U$ and $\Gamma.A \vdash B : U$.

All equations we have been using can be grouped together in the equations of *C-monoid* [2]. There are the following equations of a monoid with a special constants $\mathsf{p}, \mathsf{q}, \mathsf{app}$ and operations (x, y) and λx

$$(xy)z = x(yz) \qquad x1 = 1x = x$$

$$\mathsf{p}(x, y) = x \qquad \mathsf{q}(x, y) = y \qquad (x, y)z = (xz, yz) \qquad 1 = (\mathsf{p}, \mathsf{q})$$

$$\mathsf{app}(\lambda x, y) = x[y] \qquad (\lambda x)y = \lambda(xy^+) \qquad 1 = \lambda\ \mathsf{app}$$

where we define $[y] = (1, y)$ and $x^+ = (x\mathsf{p}, \mathsf{q})$. We have $x^+(y, z) = (xy, z)$ and $x^+y^+ = (xy)^+$ and $x^+[y] = (x, y)$.

We can add also descibe a model of type theory with *dependent sums*. We should have $\Gamma \vdash \Sigma\ A\ B$ if $\Gamma \vdash A$ and $\Gamma.A \vdash B$. If $\sigma : \Delta \to \Gamma$ we should have $(\Sigma\ A\ B)\sigma = \Sigma\ (A\sigma)\ (B\sigma^+)$. If $\Gamma \vdash a : A$ and

$\Gamma \vdash b : B[a]$ we should have $\Gamma \vdash (a,b) : \Sigma\ A\ B$. We require the equation $(a,b)\sigma = a\sigma, b\sigma$. We ask also for two operations $\Gamma \vdash \mathsf{p}c : A$ and $\Gamma \vdash \mathsf{q}c : B[\mathsf{p}c]$ if $\Gamma \vdash c : \Sigma\ A\ B$ and the equations $\mathsf{p}(a,b) = a$ and $\mathsf{q}(a,b) = b$.

3 Model for weak conversion

When implementing λ-calculus, one does not usually reduce under an abstraction and it is natural to consider a version of type theory which follows this restriction. The first published version of MLTT [3] had actually this restriction. The conversion rules are

$$(xy)z = x(yz) \qquad x1 = 1x = x$$

$$\mathsf{p}(x,y) = x \qquad \mathsf{q}(x,y) = y \qquad (x,y)z = (xz, yz)$$

$$\mathsf{app}((\lambda x)\sigma, y) = x(\sigma, y)$$

For the typing rules, we remove the conversion rule $(\Pi\ A\ B)\sigma = \Pi\ (A\sigma)\ (B\sigma^+)$ and have instead the following rules

$$\frac{\Gamma \vdash A \quad \Gamma.A \vdash B \qquad \sigma : \Delta \to \Gamma \qquad \Delta \vdash w : (\Pi\ A\ B)\sigma \qquad \Delta \vdash u : A\sigma}{\Delta \vdash \mathsf{app}(w,u) : B(\sigma,u)}$$

and the conversion rule is

$$\frac{\Gamma \vdash A \quad \Gamma.A \vdash B \qquad \sigma : \Delta \to \Gamma \qquad \Gamma.A \vdash b : B \quad \Delta \vdash u : A\sigma}{\Delta \vdash \mathsf{app}((\lambda b)\sigma, u) = b(\sigma,u) : B(\sigma,u)}$$

We call WMLTT this version of Type Theory and the rules are presented in Figure 2.1.

4 Computation rule

There is a natural computation system associated to this version of type theory.

$$\sigma 1 \to \sigma \qquad 1\sigma \to \sigma \qquad (\sigma\delta)\nu \to \sigma(\delta\nu)$$

$$(\sigma,u)\delta \to (\sigma\delta, u\delta) \qquad \mathsf{p}(\sigma,u) \to \sigma \qquad \mathsf{q}(\sigma,u) \to u$$

$$\mathsf{app}(w,u)\delta \to \mathsf{app}(w\delta, u\delta) \qquad \mathsf{app}((\lambda b)\sigma, u) \to b(\sigma,u)$$

References

[1] J. Cartmell. Generalised algebraic theories and contextual categories. Ann. Pure Appl. Logic 32 (1986), no. 3, 209–243.

[2] J. Lambek and P.J. Scott. *Introduction to higher order categorical logic.* Cambridge studies in advanced mathematics 7, 1986.

[3] P. Martin-Löf. An intuitionistic theory of types: predicative part. Logic Colloquium, 1973.

$$\frac{\Gamma \vdash}{1 : \Gamma \to \Gamma} \qquad \frac{\sigma : \Delta \to \Gamma \quad \delta : \Theta \to \Delta}{\sigma\delta : \Theta \to \Gamma}$$

$$\frac{\Gamma \vdash A \quad \sigma : \Delta \to \Gamma}{\Delta \vdash A\sigma} \qquad \frac{\Gamma \vdash t : A \quad \sigma : \Delta \to \Gamma}{\Delta \vdash t\sigma : A\sigma}$$

$$\frac{}{\vdash} \qquad \frac{\Gamma \vdash \quad \Gamma \vdash A}{\Gamma.A \vdash} \qquad \frac{\Gamma \vdash A}{\mathsf{p} : \Gamma.A \to \Gamma} \qquad \frac{\Gamma \vdash A}{\Gamma.A \vdash \mathsf{q} : A\mathsf{p}}$$

$$\frac{\sigma : \Delta \to \Gamma \quad \Gamma \vdash A \quad \Delta \vdash u : A\sigma}{(\sigma, u) : \Delta \to \Gamma.A}$$

$$\frac{\Gamma.A \vdash B}{\Gamma \vdash \Pi\ A\ B} \qquad \frac{\Delta.A\sigma \vdash b : B(\sigma\mathsf{p}, \mathsf{q})}{\Delta \vdash \lambda b : (\Pi\ A\ B)\sigma}$$

$$\frac{\Gamma.A \vdash B}{\Gamma \vdash \Sigma\ A\ B} \qquad \frac{\Gamma.A \vdash B \quad \sigma : \Delta \to \Gamma \quad \Delta \vdash u : A\sigma \quad \Delta \vdash v : B(\sigma, u)}{\Delta \vdash (u, v) : (\Sigma\ A\ B)\sigma}$$

$$\frac{\sigma : \Delta \to \Gamma \quad \Delta \vdash w : (\Pi\ A\ B)\sigma \quad \Delta \vdash u : A\sigma}{\Delta \vdash \mathsf{app}(w, u) : B(\sigma, u)}$$

$$\frac{\Delta \vdash w : (\Sigma\ A\ B)\sigma}{\Delta \vdash \mathsf{p}w : A\sigma} \qquad \frac{\Delta \vdash w : (\Sigma\ A\ B)\sigma}{\Delta \vdash \mathsf{q}w : B(\sigma, \mathsf{p}w)}$$

$$\sigma 1 = \sigma \qquad 1\sigma = \sigma \qquad (\sigma\delta)\nu = \sigma(\delta\nu)$$

$$(\sigma, u)\delta = (\sigma\delta, u\delta) \qquad \mathsf{p}(\sigma, u) = \sigma \qquad \mathsf{q}(\sigma, u) = u$$

$$\mathsf{app}(w, u)\delta = \mathsf{app}(w\delta, u\delta) \qquad \mathsf{app}((\lambda b)\sigma, u) = b(\sigma, u)$$

Figure 1: Rules of WMLTT