A parametric Type theory

May 6, 2014

Presheaf models of type theory

A group in a presheaf model over a category $\mathcal{C}$ can be seen as a functor from $\mathcal{C}^{opp}$ to the category of groups.

One way to present type theory is to use a generalized algebraic theory with 4 sorts: contexts, types, terms and substitutions. Models of type theory form then a category.

If $\mathcal{C}$ is a category we define then a presheaf model over $\mathcal{C}$ to be a functory from $\mathcal{C}^{opp}$ to the category of models of type theory. This notion can itself be expressed as a generalized algebraic theory, where each of the sorts: contexts, types, terms and substitutions is now indexed by an object of $\mathcal{C}$ and we add constants to express the functoriality laws.

One way to think of this is as a kind of "non-standard" extension of type theory. The main application is that we can now add new constants expressing operations that cannot be expressed in the standard version. The goal of this note is to provide a concrete example of this phenomena, where we can formulate laws expressing an internal version of parametricity.

1 A special case

We consider the category which has for objects finite set of symbols $I, J, \ldots$ not containing 0 and maps $f : I \to J$ are set-theoretic functions $f : I \to J \cup \{0\}$ such that $f(i) = f(j)$ implies $i = j$ whenever $f(i)$ and $f(j)$ are in J. This can also be described as the category of partial bijection. We use the notations $f, g, h, \ldots$ for the maps of this category. The associated presheaf category is equivalent to the category of "nominal restriction sets" studied in section 9.1 of the book [3].

If x is not in I we write $\iota_x : I \to I, x$ the canonical injection. The map $(x0) : I, x \to I$ which sends x to 0 is a retraction of ι_x. It follows that $u \longmapsto u\iota_x$ is injective. Any map f can be written as a composition of "elementary" maps that are $(x0)$, setting x to 0, the maps ι_x, and the maps (xy), renaming x to a fresh variable y.

We are going to describe presheaf models over the opposite of this category.

We believe that this gives a semantics to the parametricity part of the work on "Type Theory in Colors" [2].

$$\frac{\Gamma \vdash_I}{1 : \Gamma \to_I \Gamma} \qquad \frac{\sigma : \Delta \to_I \Gamma \quad \delta : \Theta \to_I \Delta}{\sigma\delta : \Theta \to_I \Gamma}$$

$$\frac{\Gamma \vdash_I A \quad \sigma : \Delta \to_I \Gamma}{\Delta \vdash_I A\sigma} \qquad \frac{\Gamma \vdash_I t : A \quad \sigma : \Delta \to_I \Gamma}{\Delta \vdash_I t\sigma : A\sigma}$$

$$\frac{}{() \vdash_I} \qquad \frac{\Gamma \vdash_I \quad \Gamma \vdash_I A}{\Gamma.A \vdash_I} \qquad \frac{\Gamma \vdash_I A}{\mathsf{p} : \Gamma.A \to_I \Gamma} \qquad \frac{\Gamma \vdash_I A}{\Gamma.A \vdash_I \mathsf{q} : A\mathsf{p}}$$

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

$$\frac{\Gamma.A \vdash_I B}{\Gamma \vdash_I \Pi\ A\ B} \qquad \frac{\Gamma.A \vdash_I B \quad \Gamma.A \vdash_I b : B}{\Gamma \vdash_I \lambda b : \Pi\ A\ B}$$

$$\frac{\Gamma \vdash_I w : \Pi\ A\ B \quad \Gamma \vdash_I u : A}{\Gamma \vdash_I \mathsf{app}(w, u) : B[u]}$$

The equations are

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

$$A1 = A \qquad (A\sigma)\delta = A(\sigma\delta) \qquad u1 = u \qquad (u\sigma)\delta = u(\sigma\delta)$$

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

$$(\Pi\ A\ B)\sigma = \Pi\ (A\sigma)\ (B(\sigma\mathsf{p}, \mathsf{q}))$$

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

We have used the defined operation $[u] = (1, u)$

So far, this is like describing a collection of models, indexed by the objects I. We add the new rules, the *restriction rules*, which connect these models

$$\frac{\Gamma \vdash_I}{\Gamma f \vdash_J} \qquad \frac{\Gamma \vdash_I A}{\Gamma f \vdash_J Af} \qquad \frac{\Gamma \vdash_I a : A}{\Gamma f \vdash_J af : Af} \qquad \frac{\sigma : \Delta \to_I \Gamma}{\sigma f : \Delta f \to_J \Gamma f}$$

for $f : I \to J$. We add also the equations which express that we have a functor from the category of names to the category of models of type theory

$$\Gamma 1 = \Gamma \qquad A1 = A \qquad a1 = a \qquad (\Gamma f)g = \Gamma(fg) \qquad (Af)g = A(fg) \qquad (af)g = a(fg)$$

$$\sigma 1 = \sigma \qquad (\sigma f)g = \sigma(fg) \qquad (\sigma\delta)f = \sigma f(\delta f)$$

$$(A\sigma)f = Af(\sigma f) \qquad (a\sigma)f = af(\sigma f)$$

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

$$(\Pi\ A\ B)f = \Pi\ Af\ Bf$$

$$(\lambda b)f = \lambda(bf) \qquad \mathsf{app}(w, u)f = \mathsf{app}(wf, uf)$$

Theorem 1.1 *The rule*

$$\frac{\Gamma \vdash_I w : \Pi\ A\ B \qquad \Gamma \vdash_I u : A}{\Gamma \vdash_I \mathsf{app}(w, u) : B[u]}$$

is equivalent *(modulo the other rules) to*

$$\frac{\Gamma \vdash_I w : \Pi\ A\ B \qquad \Gamma f \vdash_J u : Af}{\Gamma f \vdash_J \mathsf{app}(wf, u) : Bf[u]} \quad f : I \to J$$

Proof. Indeed, if we assume the first rule and we have $\Gamma \vdash_I w : \Pi\ A\ B$ and $f : I \to J$ then we get $\Gamma \vdash_J wf : (\Pi\ A\ B)f$ by restriction. But we also have $(\Pi\ A\ B)f = \Pi\ Af\ Bf$ and if $\Gamma f \vdash_J u : Af$ then we get $\mathsf{app}(wf, u) : Bf[u]$.

Conversely, if we assume the second rule, in the special case where $I = J$ and $f = 1 : I \to I$ we get the first rule since $A1 = A$ and $w1 = w$. □

If $\Gamma \vdash_I A$ we can think of Γ as a context and A as a type dependent on some quantities represented by the symbols in I, quantities which may get the value 0.

The maps $u \longmapsto u\iota_x$ are injective since $\iota_x(x0) = 1$.

We *identify systematically* Γ with $\Gamma\iota_x$ and A with $A\iota_x$ and a with $a\iota_x$ and σ with $\sigma\iota_x$. In particular, we have $\Gamma \vdash_{I,x} a : A$ if $\Gamma \vdash a : A$.

If for instance $\Gamma \vdash_I A$ and $\Gamma \vdash_{I,x} u : A$ and there exists a such that $\Gamma \vdash_I a : A$ and $u = a\iota_x = a$, we can express this by saying that a is *independent of* the symbol x.

If $\vdash_x A$ we think of A as a *line* starting from the type $\vdash A(x0)$.

2 Transforming a predicate in a line

In order to internalize parametricity, we add the following "non-standard" operations

$$\frac{\Gamma \vdash_I A \qquad \Gamma \vdash_I P : A \to U}{\Gamma \vdash_{I,x} A \times_x P} \qquad \frac{\Gamma \vdash_I a : A \qquad \Gamma \vdash_I p : P\ a}{\Gamma \vdash_{I,x} (a,_x p) : A \times_x P}$$

and

$$\frac{\Gamma \vdash_{I,x} w : A \times_x P}{\Gamma \vdash_I w.x : P[w(x0)]}$$

with the defining equations

$$(A \times_x P)(x0) = A \qquad (a,_x p)(x0) = a \qquad (a,_x p).x = p$$

and

$$(A \times_x P)(xz) = A \times_z P \qquad (a,_x p)(xz) = a,_z p$$

and

$$(w.x)(yz) = (w(yz)).x \qquad (w.x)(yx) = (w(xz)(yx)).z$$

The operation $A \times_x P$ transforms a *predicate* over the type A to a *line* starting from A. This is similar (but much simpler) to the univalence axiom, which transforms an equivalence between two types in a line joining these types.

We notice that the type $A \times_x P$ behaves like a *telescope* (in deBruijn's terminology). If we have $t : A \times_x P \to B \times_x Q$ then $t(x0) : A \to B$ and if $u : A,\ p : P\ u$ then $(t\ (u,_x p)).x : Q\ (t(x0)\ u)$. So a term of type $t : A \times_x P \to B \times_x Q$ defines an element in $(\Sigma f : A \to B)(\Pi u : A)P\ u \to Q\ (f\ u)$.

With these rules, we can internalize parametricity. For instance, using a notation with names for context, we can build a term of type $P\ (f\ A\ a)$ in the context

$$f : (\Pi X : U)X \to X, A : U, P : A \to U, a : A, b : P\ a$$

namely the term $(f\ (A \times_x P)\ (a,_x b)).x$ where x is a fresh symbol. Indeed we have

$$(f\ (A \times_x P)\ (a,_x b))(x0) = f(x0)\ (A \times_x P)(x0)\ (a,_x b)(x0) = f\ A\ a$$

In this way, we have a "non-standard" proof of $P\ (f\ A\ x)$. Notice that there is no "standard" proof of $P\ (f\ A\ x)$ in this context.

Another example is

$$L = (\Pi X : U)X \to (X \times_x A \to X) \to X$$

which is such that

$$L(x0) = N = (\Pi X : U)X \to (X \to X) \to X$$

Notice that the type L is a possible type for lists of elements in A while N is the type of Church numerals. For instance if $a_0\ a_1 : A$ then

$$\lambda X \lambda a \lambda f\ \ f\ (f\ (a,_x a_0),_x a_1)$$

is an element $l : L$ representing the list $[a_0, a_1]$ and such that

$$l(x0) = \lambda X \lambda a \lambda f\ \ f\ (f\ a)$$

3 Semantics

A context $\Gamma \vdash_I$ is interpreted by a family of sets Γf for $f : I \to J$ with restriction maps $\Gamma f \to \Gamma fg,\ \rho \longmapsto \rho g$ satisfying $\rho 1 = \rho$ and $(\rho g)h = \rho(gh)$. A type $\Gamma \vdash_I A$ is interpreted by giving for each $f : I \to J$ and ρ in Γf a set $A(f, \rho)$ with restriction maps $A(f, \rho) \to A(fg, \rho g),\ u \longmapsto ug$ satisfying $u1 = u$ and $(ug)h = u(gh)$. The judgement $\Gamma \vdash_I a : A$ is interpreted by giving a family $a(f, \rho)$ in $A(f, \rho)$ such that

$(a(f,\rho))g = a(fg,\rho g)$. Finally a substitution $\sigma : \Delta \to_I \Gamma$ is interpreted by giving a family of maps $\sigma : \Delta f \to \Gamma f$ such that $(\sigma\rho)g = \sigma(\rho g)$.

In particular here is the interpretation of the rule $\Gamma \vdash_{I,x} A \times_x P$ assuming $\Gamma \vdash_I$, $\Gamma \vdash_I A$, $\Gamma \vdash_I P : A \to U$. Given $f : I, x \to J$ and ρ in the set $\Gamma\iota_x f$, we have to define a set $(A \times_x P)(f,\rho)$. The definition is by case whether or not $f(x) = 0$ or not.

If $f(x) = 0$ then we define $(A \times_x P)(f,\rho) = A(\iota_x f, \rho)$.

If $f(x) = y$ then we define $(A \times_x P)(f,\rho)$ to be the set of pairs (u, v) with u in $A(\iota_x f(y0), \rho(y0))$ and v in $P(\iota_x f(y0), \rho(y0))(u)$. We use that $\rho(y0)$ is in the set $\Gamma\iota_x f(y0)$.

A similar interpretation holds for $\Gamma \vdash_{I,x} (a,_x p) : A \times_x P$. Given $f : I, x \to J$ and ρ in the set $\Gamma\iota_x f$, we have to define $(a,_x p)(f,\rho)$ which should be an element of the set $(A \times_x P)\rho$. The definition is by case whether or not $f(x) = 0$ or not.

If $f(x) = 0$ then we define $(a,_x p)(f,\rho)$ to be $a(\iota_x f, \rho)$.

If $f(x) = y$ then we define $(a,_x p)(f,\rho)$ to be the pair $a(\iota_x f(y0), \rho(y0)), p(\iota_x f(y0), \rho(y0))$.

We justify the rule $\Gamma \vdash_I w.x : P\ (w(x0))$ for $\Gamma \vdash_I$ and $\Gamma\iota_x \vdash_{I,x} w : A \times_x P$. We take $g : I \to J$ and ρ in the set Γg and we have to define $(w.x)(g,\rho)$. We choose y not in J and define $(g, x = y) : I, x \to J, y$. We have $\rho\iota_y$ in the set $\Gamma\iota_x(g, x = y)$ since $\iota_x(g, x = y) = g\iota_y$. The element $w(\iota_x(g, x = y), \rho\iota_y)$ is of the form u, v and we define $(w.x)(g,\rho) = v$.

We justify the rule $\Gamma.A \vdash_I$ if $\Gamma \vdash_I A$. For this we take $f : I \to J$ and we have to define a set $(\Gamma.A)f$. We define this to be the set of pairs ρ, u with ρ in Γf and u in $A\rho$. If $g : J \to K$ we define $(\rho, u)g = \rho g, ug$.

We define next $\Gamma \vdash_I \Pi\ A\ B$ if $\Gamma.A \vdash_I B$. Given $f : I \to J$ and ρ in Γf we define $(\Pi\ A\ B)\rho$ to be the set of families wg with $g : J \to K$ such that

$$wg \in \prod_{u \in A\rho g} B(\rho g, u)$$

and $(wg(u))h = w(gh)(uh)$ if $h : K \to L$.

If $\Gamma.A \vdash_I b : B$ we can then define the interpretation of $\Gamma \vdash_I \lambda b : \Pi\ A\ B$. Given $f : I \to J$ and ρ in Γf and $g : J \to L$ and u in $A\rho g$ we define

$$(\lambda b)(f,\rho)g(u) = b(fg, (\rho g, u))$$

since we have $(\rho g, u)$ in $(\Gamma.A)fg$.

4 The unit interval

We can introduce a non-standard type $\mathbf{I}$ with the rules

$$\frac{\Gamma \vdash_I}{\Gamma \vdash_I \mathbf{I}} \qquad \frac{\Gamma \vdash_I}{\Gamma \vdash_I 0 : \mathbf{I}} \qquad \frac{\Gamma \vdash_I}{\Gamma \vdash_I x : \mathbf{I}}\ x \in I$$

and the equalities

$$\mathbf{I}f = \mathbf{I} \qquad 0f = 0 \qquad x(y0) = x \qquad x(x0) = 0 \qquad x(xy) = y \qquad x(yz) = x$$

5 Nominal presentation

We have defined a *family* of models M_I connected by homomorphisms. We can associate to this *one* model M^*. An object of M^* (which can be a context, or a type, or a term, or a substitution) is a pair (I, u) where I is a finite set of symbols and u an object of M_I. We identify (I, u) and (J, v) if u and v become equal in $M_{I \cup J}$.

Intuitively an object v of M^* depends on finitely many symbols. We can define the independence relation $x\#v$ to mean that $v = (I, u)$ for some I not containing the symbol x. The model M^* has an endomorphism $(x0)$ for each symbol x, and automorphisms (xy) for x and y distinct symbols.

6 Type-checking

We write $A, B, \ldots$ for type values and $u, v, \ldots$ for values and $t, T, \ldots$ for terms. Type-checking is specified by two relations

$$I, \rho, \Gamma \vdash t \Downarrow A \qquad\qquad I, \rho, \Gamma \vdash t \Uparrow$$

For the relation $I, \rho, \Gamma \vdash t \Downarrow A$ we have I, ρ, Γ, t given and A is inferred. For the relation $I, \rho, \Gamma \vdash t \Uparrow$ we check that t has the right given type A.

In these relations Γ gives type values to variables, so Γ is of the form $x_1 : A_1, \ldots, x_n : A_n$ while ρ gives values to variables and is of the form $x_1 = u_1, \ldots, x_n = u_n$. The first argument I is a set of names/symbols $i_1, \ldots, i_m$.

If I is a set of symbols we write $\mathsf{id}_I : I \to I$ the corresponding identity function.

We have an evaluation function $t\ f\ \rho$ which takes one term t which has been type-checked using I, one function $f : I \to J$ and one environment of J-values ρ, and which produces a J-value.

The rules are then

$$\frac{x : A\ in\ \Gamma}{I, \rho, \Gamma \vdash x \Downarrow A}$$

$$\frac{I, \rho, \Gamma \vdash t_0 \Downarrow \Pi\ A\ F \qquad I, \rho, \Gamma \vdash t_1 \Uparrow A}{I, \rho, \Gamma \vdash t_0\ t_1 \Downarrow F\ (t_1\ \mathsf{id}_I\ \rho)}$$

$$\frac{I, \rho, \Gamma \vdash t_0 \Downarrow U.i\ A \qquad I, \rho, \Gamma \vdash t_1 \Uparrow A}{I, \rho, \Gamma \vdash t_0\ t_1 \Downarrow U}$$

$$\frac{I, \rho, \Gamma \vdash t_0 \Downarrow (\Pi\ A\ F).i\ c \qquad I, \rho, \Gamma \vdash t_1 \Uparrow A(i0) \qquad I, \rho, \Gamma \vdash t_2 \Uparrow A.i\ (t_1\ \mathsf{id}_I\ \rho)}{I, \rho, \Gamma \vdash t_0\ t_1\ t_2 \Downarrow F\ (t_1\ \mathsf{id}_I\ \rho,_i t_2\ \mathsf{id}_I\ \rho)}$$

$$\frac{}{I, \rho, \Gamma \vdash U \Downarrow U}$$

$$\frac{I, (\rho, x = X(I)), \Gamma, x : A \vdash t : F\ X(I)}{I, \rho, \Gamma \vdash \lambda x.t \Uparrow \Pi\ A\ F}$$

$$\frac{I, (\rho, x = X(I)), \Gamma, x : A \vdash t : U}{I, \rho, \Gamma \vdash \lambda x.t \Uparrow U.i\ A}$$

$$\frac{I, (\rho, x = X(I), y = Y(I)), \Gamma, x : A(i0), y : A.i\ X(I) \vdash t : (F\ (X(I),_i Y(I)).i\ (c\ X(I))}{I, \rho, \Gamma \vdash \lambda x\ y.t \Uparrow (\Pi\ A\ F).i\ c}$$

$$\frac{(I, i), \rho, \Gamma \vdash t \Downarrow (A,_i P)}{I, \rho, \Gamma \vdash t.i \Downarrow P\ ((t\ \mathsf{id}_{I,i}\ \rho)(i0))}$$

$$\frac{I, \rho, \Gamma \vdash t \Downarrow A \qquad A = B}{I, \rho, \Gamma \vdash t \Uparrow B}$$

$$\frac{I, \rho, \Gamma \vdash T \Uparrow U \qquad I, (\rho, x = X(I)), \Gamma, x : T\ \mathsf{id}_I\ \rho \vdash T' \Uparrow U}{I, \rho, \Gamma \vdash \Pi\ T\ (\lambda x.T') \Uparrow U}$$

$$\frac{I, \rho(i0), \Gamma(i0) \vdash t \Uparrow A(i0) \qquad I, \rho(i0), \Gamma(i0) \vdash p \Uparrow A.i\ (t\ \mathsf{id}_I\ \rho(i0))}{(I, i), \rho, \Gamma \vdash (t,_i p) \Uparrow A}$$

The evaluation is defined by

$$x\ f\ \rho = \rho(x) \qquad (t'\ t)\ f\ \rho = t'\ f\ \rho\ (t\ f\ \rho)$$

$$(t,_i p)\ f\ \rho = t\ (f - i)\ \rho(j0),_j p\ (f - i)\ \rho(j0)$$

if $f(i) = j$ and

$$(t,_i p)\ f\ \rho = t\ (f - i)\ \rho$$

if $f(i) = 0$.

We also have an application

$$(\lambda x.t)\ f\ \rho\ v = t\ f\ (\rho, x = v)$$

A new operation is $A.i$ with

References

[1] M.Bezem, Th. Coquand and S. Huber. A model of type theory in cubical sets. Preprint, 2013.

[2] J.P. Bernardy, G. Moulin. Type Theory in Color, internalizing parametricity and erasure in type theory, ICFP, 2013.

[3] A. M. Pitts. Nominal Sets. Names and Symmetry in Computer Science. Cambridge Tracts in Theoretical Computer Science, 2013.

[4] A. M. Pitts. An Equivalent Presentation of the Bezem-Coquand-Huber Category of Cubical Sets. Manuscript, 17 September 2013.

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

[6] P.L. Curien. Substitutions up to isomorphisms. Fundamenta Informaticae, Volume 19, 1993, p. 51-85.

[7] P. Dybjer. Internal Type Theory. in *Types for Programs and Proofs*, Springer, 1996.