Notes on the Abadi-Plotkin logic for parmetricity

Dan Frumin

Last updated: December 13, 2019

Abadi-Plotkin logic (APL) is a second-order multi-sorted logic where one is allowed to quantify over terms, predicates (types), and relations. The logic is presented in the paper "A Logic for Parametric Polymorphism" by Gordon Plotkin and Martín Abadi [1]. The aim of this document is to fill out some of the details and proof that have been omitted in the paper, as well as give more examples.

1 Types, terms, and relations

Types and terms are those of System F:

$$\text{Types } A ::= X \mid A \to B \mid \forall X.A$$

$$\text{Terms } t ::= x \mid \lambda x : A.t \mid t\ t \mid \Lambda X.t \mid t_A$$

where X and x range over type and term variables, respectively.

Substitution of relations for variables in types is defined recursively. If $A[X]$ is a type with a free variable X, then the substitution $A[R]$ for a relation $R \subseteq B \times C$ is a relation $A[R] \subseteq A[B] \times A[C]$ defined as

- $X[R] = R$ and $Y[R] = Y$ if $Y \neq X$
- $(A \to A')[R] = A[R] \to A'[R]$
- $(\forall Y.A[X, Y])[R] = \forall Y, Z, F \subseteq Y \times Z.A[R, F]$

We write $u[A[R]]t$ for the proposition $(A[R])(u, t)$.

Definition 1. *Given a function $f : A \to B$ we define the* graph *of f as the relation $\langle f \rangle$ s.t. $\langle f \rangle(x, y) \iff f\ x = y$.*

Definition 2. *We denote the* identity relation $\langle \mathrm{id}_X \rangle$ *on a type X as I_X.*

1.1 Positivity and negativity

(.. or covariance and contravariance)

For the next section we will need to distinguish between covariant (positive) occurrences of free variables and contravariant (negative). Essentially, if $A[X]$ is covariant in X and $f : C \to D$ is a function/term, then we have a substitution $A[f] : A[C] \to A[D]$. If A is contravariant in X, then this substitution yields a term $A[f] : A[D] \to A[C]$.

$$\text{pos\_var}\frac{A = X}{A[X]\ \text{pos}}$$

$$\text{neg\_pos\_var\_not}\frac{A = Y \neq X}{A[X]\ \text{pos}\ , A[X]\ \text{neg}}$$

$$\text{pos\_arr}\frac{\begin{array}{c}A[X]\ \text{neg}\\ B[X]\ \text{pos}\end{array}}{(A[X] \to B[X])\ \text{pos}} \quad \text{neg\_arr}\frac{\begin{array}{c}A[X]\ \text{pos}\\ B[X]\ \text{neg}\end{array}}{(A[X] \to B[X])\ \text{neg}}$$

$$\text{pos\_forall}\frac{A[X, Y]\ \text{pos in}\ X}{(\forall Y.A[X, Y])\ \text{pos}} \quad \text{neg\_forall}\frac{A[X, Y]\ \text{neg in}\ X}{(\forall Y.A[X, Y])\ \text{neg}}$$

2 Dinaturality

Let $F[Y, X]$ be covariant in X and contravariant in Y. In other words, if $f : X \to X'$ and $g : Y' \to Y$, $F[g, f] : F[Y, X] \to F[Y', X']$. Particularly,

$$F[\mathrm{id}_X, f] : F[X, X] \to F[X, Y]$$
$$F[f, \mathrm{id}_Y] : F[Y, Y] \to F[X, Y]$$

for $f : X \to Y$.

Dinaturality (for F) states that

$$\forall XY \forall f : X \to Y.F[\mathrm{id}_X, f] \circ (-)_X = F[f, \mathrm{id}_Y] \circ (-)_Y$$

where $(-)_X$ is $\lambda u.u_X$ for $u : \forall X.F[X, X]$. By using dinaturality we can prove properties like canonicity of certain encodings.

Example 3. *The* unit type *can be encoded in System F as* $\mathbf{1} = \forall X.X \to X$*, with an element* $* := \Lambda X.\lambda x : X.x$*.*

The unit typed is obtained from a bifunctor $A[Y, X] = Y \to X$; hence $\mathbf{1} = \forall X.A[X, X]$. By calculation, we have $A[\mathrm{id}_X, f] : (X \to X) \to X \to Y = f \circ -$ and $A[f, \mathrm{id}_Y] : (Y \to Y) \to X \to Y = - \circ f$. Hence, dinaturality for A states that for any $u : \mathbf{1}$

$$f \circ u_X = u_Y \circ f$$

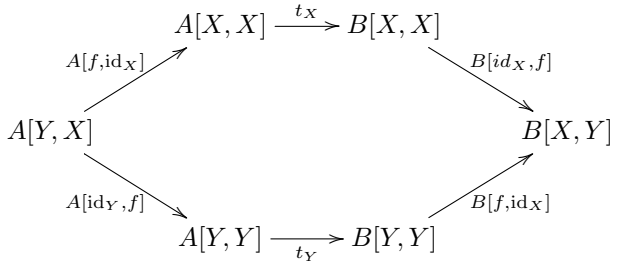
Let X be an arbitrary type and $x_0 : X$ an arbitrary term of that type. The consider dinaturality for $f : X \to X := \lambda x.x_0$.

$$\forall x.f(u_X(x)) = u_X(f(x))$$

in other words, $x_0 = u_X(x_0)$. Because X and x_0 were arbitrary, we can conclude (using the η-rules and congruence rules) that any $u : \mathbf{1}$ "behaves like" the identity $*$. Formally, we can prove $u = *$ in APL.

2.1 Dinaturality categorically

More generally, let $A, B : \mathcal{C}^{op} \times \mathcal{C} \to \mathcal{C}$ be bi(endo)functors. A natural transformation $t : A \Rightarrow B$ is *dinatural* if for any $f : X \to Y$ the following diagram commutes.



By picking A to be a terminal bifunctor we can recover the previously mentioned formula for dinaturality. In this setting, terms $t : \forall X.F[X, X]$ are interpreted as dinatural transformations $t_X : 1 \to F[X, X]$.

3 Parametricity schema

Parametricity states that

$$\forall Y_1, \ldots, Y_n \forall (u : \forall X.A[X, \bar{Y}]).u[\forall X.A[X, I_{Y_1}, \ldots, I_{Y_n}]]u$$

By unfolding the definition of $[\forall X. \ldots]$ and removing the parameters we get a simplified version

$$\forall (u : \forall X.A[X]).\forall Y, Z, R \subseteq Y \times Z.u_Y[A[R]]u_Z$$

Lemma 4 (Identity extension lemma)**.** *For any $A[X]$ it is provable in APL that*

$$\forall X \forall (u, v : A[X]).u[A[I_X]]v \iff u = v$$

Proof. By induction on A, extending the statement to multiple parameters. □

The following lemma is dubbed "logical relations lemma" because it (roughly) states that plugging in related values in a term result in related expressions.

Lemma 5 (Logical relations lemma)**.** *For any term* $x_1 : A_1[X], \ldots, x_n : A_n[X] \vdash t[x_1, \ldots, x_n] : B$ *we have*

$$\forall X, Y \forall R \subset X \times Y \forall x_1 : A_1[X], \ldots, x_n : A_n[X] \forall y_1 : A_1[Y], \ldots, y_n : A_n[Y]$$
$$(\bigwedge_i A[R](x_i, y_i)) \implies B[R](t[x_1, \ldots, x_n], t[y_1, \ldots, y_n])$$

Proof. By induction on the derivation $x_1 : A_1[X], \ldots, x_n : A_n[X] \vdash t[x_1, \ldots, x_n] : B$. □

Lemma 6. *Dinaturality is a consequence of parametricity.*

Proof. Let F be a bifunctor, we are to show

$$\forall XY \forall f : X \to Y. F[\mathrm{id}_X, f] \circ (-)_X = F[f, \mathrm{id}_Y] \circ (-)_Y$$

So let X, Y be types, $f : X \to Y$ be a term, and let $u : \forall X. F[X, X]$. By the η-rule it suffices to show:

$$F[\mathrm{id}_x, f](u_X) = F[f, \mathrm{id}_Y](u_Y)$$

By parametricity we have

$$u_X [F[\langle f \rangle, \langle f \rangle]] u_Y$$

We are going to show $(F[\mathrm{id}_X, f], F[f, \mathrm{id}_Y]) \in [F[\langle f \rangle, \langle f \rangle] \to F[I_X, I_Y]]$; then the statement will follow from the identity extension lemma.

Note that $F[\langle f \rangle, \langle f \rangle] \to F[I_X, I_Y] = F[I_X \to \langle f \rangle, \langle f \rangle \to I_Y]$. By lemma 5 it then suffices to check that $(\mathrm{id}_X, f) \in I_X \langle f \rangle$ and $(f, \mathrm{id}_Y) \in \langle f \rangle \to I_Y$. Both propositions holds by computation. □

4 Functorial matters

Every type $A[X]$ with X occurring positively in A can be seen as a functor. Specifically there is a map $A[-] : \forall XY.(X \to Y) \to A[X] \to A[Y]$.

Lemma 7. *For any type* A *we have* $A[\mathrm{id}_X] = \mathrm{id}_{A[X]}$

Proof. By induction on the structure on A, generalizing X to a list of free variables $\vec{X}$. □

We need to show a more general statement.

Lemma 8 (Graph lemma)**.** *For any functor* $A[X]$*, with* X *occurring positively we have the following statement:*

$$\forall XX' \forall (f : X \to X') \forall (w : A[X])(w' : A[X']).$$
$$A[\langle f \rangle](w, w') \iff \langle A[f] \rangle (w, w')$$

Proof. By parametricity of $A[-]$ we have for any types X, X', Y, Y' and relations $R \subset X \times X', Q \subset Y \times Y'$:

$$A[-][(R \to Q) \to A[R] \to A[Q]]A[-]$$

For the direction $\langle A[f]\rangle \Rightarrow A[\langle f\rangle]$ take $R = I_X, Q = \langle f\rangle$. Since $(\mathrm{id}_X, f) \in (I_X \to \langle f\rangle)$ we have

$$A[\mathrm{id}_A][I_{A[X]} \to A[\langle f\rangle]]A[f]$$

where $I_{A[X]} = A[I_X]$ by lemma 4. Let $(w, w') \in \langle A[f]\rangle$, i.e. $w' = A[f](w)$. Then, $A[\mathrm{id}_A](w) = \mathrm{id}_{A[X]}(w) = w[A[\langle f\rangle]]A[f](w) = w'$.

For the other direction take $R = \langle f\rangle, Q = I_Y$. Because $(f, \mathrm{id}_B) \in (\langle f\rangle \to I_B)$ we have

$$A[f][A[\langle f\rangle] \to I_{A[X]}]\,\mathrm{id}_{A[X]}$$

once again by lemmas 4 and 7. If $(w, w') \in A[\langle f\rangle]$, then $A[f](w) = w'$, i.e. $(w, w') \in \langle A[f]\rangle$. □

Using the graph lemma we can obtain:

Lemma 9. *For any covariant $A[X]$*

$$\forall (f : X \to Y)(g : Y \to Z).A[g \circ f] = A[g] \circ A[f]$$

Proof. We employ the parametricity of $A[-]$:

$$A[-]_{X,Z}[(\langle f\rangle \to I_Z) \to A[\langle f\rangle] \to A[I_Z]]A[-]_{Y,Z}$$

One can verify that $(g \circ f, g) \in (\langle f\rangle \to I_Z)$, and $(u, A[f](u)) \in A[\langle f\rangle]$ for any $u : A[X]$, using the graph lemma. Hence,

$$A[g \circ f](u)[A[I_Z]]A[g](A[f](u))$$

for any $u : A[X]$. We obtain the required result using the identity extension lemma. □

Thus any type $A[X]$ with X occurring only positively is functorial.

5 Encodings of datatypes

We have already seen the unit type encoding $\mathbf{1} = \forall X.X \to X$. We can also show that every inhabitant of $\mathbf{1}$ "behaves like" $*$ using parametricity alone.

Lemma 10. $\forall u : \mathbf{1}.(u = *)$ *is true in APL*

Proof. Let $u : \mathbf{1}$. By the η-rule,

$$u = * \iff \Lambda Z.\lambda x : Z.u_Z(x) = \Lambda Z.\lambda x : Z.x$$

By congruence rules (for the right to left direction) and by the β-rule (for the left to right direction), this is equivalent to

$$\forall Z, a : Z.u_Z(a) = a$$

Thus let Z be a type and $a : Z$. Pick a relation $R = (x : Z, y : Z).y = a$. Then by parametricity we have

$$u_Z[R \to R]u_Z \iff \forall (x, y) \in R.u_Z(x)[R]u_Z(y)$$

Clearly, $(a, a) \in R$. Hence $(u_Z(a), u_Z(a)) \in R$, in other words, $u_Z(a) = a$. □

Example 11. *The empty type* $\mathbf{0}$ *is encoded by* $\mathbf{0} = \forall X.X$.

Using parametricity we can show that $\mathbf{0}$ is uninhabited.

Lemma 12. $\forall (u : \mathbf{0}).\bot$ *is derivable in APL*

Proof. Let $u : \mathbf{0}$. Let X be an arbitrary type. Take $R = (x : X, y : X).\bot$. Then, by parametriciry, $u_X[R]u_X$, which is a contradiction. □

5.0.1 Products

Example 13. *The products are given by* $A \times B = \forall X.((A \to B \to X) \to X)$.

The pairing function $\mathsf{pair}_{A,B} = \lambda a\ b.\Lambda X.\lambda f.f\ a\ b$ is abbreviated as $\langle a, b\rangle = \mathsf{pair}_{A,B}\ a\ b$. The projections are given by $\mathsf{fst}_{A,B} = \lambda p.p_A(K_{A,B})$ and $\mathsf{snd}_{A,B} = \lambda p.p_B(K'_{A,B})$.

From the computational rules one can verify that $\forall x : A \forall y : B.\mathsf{fst}\langle x, y\rangle \wedge \mathsf{snd}\langle x, y\rangle = y$. Using parametricity/dinaturality we can prove that the encoding of the products is categorical. For this we will use the canonicity of the encoding:

Lemma 14. $\forall u : A \times B.\langle \mathsf{fst}\ u, \mathsf{snd}\ u\rangle = u$ *is true in APL. This proposition is also called* surjective pairing *in* λ*-calculus literature.*

The categoricity of the products means that

$$\forall f : X \to A \forall g : X \to B.\exists !(h : X \to A \times B).\mathsf{fst} \circ h = f \wedge \mathsf{snd} \circ h = g$$

Given f, g we provide $h = \langle f, g\rangle := \lambda x.\langle f\ x, g\ x\rangle$. By the computational rules (and η-rules) we have $\mathsf{fst} \circ h = f$ and $\mathsf{snd} \circ h = g$. Suppose that there is another h' with this property. Then for all $x : X$ we have $h'\ x = \langle \mathsf{fst}(h'\ x), \mathsf{snd}(h'\ x)\rangle$ using lemma 14 and consequently $h'\ x = \langle f\ x, g\ x\rangle = h\ x$; hence $h' = h$.

Proof (of lemma 14) using parametricity. By η and computational rules it suffices to show that

$$\forall X \forall e : A \to B \to X.\langle \mathsf{fst}(u), \mathsf{snd}(u)\rangle_X(e) = u_X(e)$$

where the left hand side computes to $e(\mathsf{fst}(u))(\mathsf{snd}(u))$.

Define terms $\bar{e}, \tilde{e} : A \times B \to X$ as

$$\bar{e} := \lambda w.w_X(e)$$

$$\tilde{e} := \lambda w.e(\mathsf{fst}(w))(\mathsf{snd}(w))$$

Thus our goal reduces to showing $\tilde{e}(u) = \bar{e}(u)$. By parametricity we have

$$\forall XY \forall R \subset X \times Y. u_X[(I_A \to I_B \to R) \to R]u_Y$$

Consider $R = \langle \bar{e} \rangle$ the graph of $\bar{e}$; that is, $R(x, y) := \bar{e}(x) = y \equiv x_X(e) = y$. We claim that $(\mathsf{pair}_{A,B}, e) \in [I_A \to I_B \to \langle \bar{e} \rangle]$: for let $a : A, b : B$ be arbitrary, then $(\mathsf{pair}\ a\ b, e\ a\ b) \in \langle \bar{e} \rangle$ iff $\bar{e}\langle a, b\rangle = \langle a, b\rangle_A(e) = e\ a\ b$, where the last equality is purely computational. It follows from parametricity that $u_{A\times B}(\mathsf{pair}_{A,B})[\langle \bar{e} \rangle]u_X(e)$, i.e. $\bar{e}(u_{A\times B}(\mathsf{pair}_{A,B})) = (u_{A\times B}(\mathsf{pair}_{A,B}))_X(e) = u_X(e)$. Since we have showed this equality for an arbitrary e, we can conclude that $u_{A\times B}(\mathsf{pair}_{A,B}) = u$.

On the other hand, we can show that $(\mathsf{pair}_{A,B}, e) \in [I_A \to I_B \to \langle \tilde{e} \rangle]$: as $\tilde{e}\langle a, b\rangle = e\ a\ b$ by computation. Hence, $u_{A\times B}(\mathsf{pair}_{A,B})[\langle \tilde{e} \rangle]u_X(e)$, i.e. $\tilde{e}(u_{A\times B}(\mathsf{pair})) = u_X(e) = \bar{e}(u)$. But from the previous paragraph we know that $u_{A\times B}(\mathsf{pair}) = u$; hence we have $\tilde{e}(u) = \bar{e}(u)$. □

5.0.2 Coproducts

Example 15. *The sums are given by* $A + B = \forall X.((A \to X) \to (B \to X) \to X)$.

With the injections given by $\mathsf{inl} = \lambda x.\Lambda Z.\lambda f\ g.fx$ and $\mathsf{inr} = \lambda y.\Lambda Z.\lambda f\ g.gy$ and pattern matching given by $\mathsf{case}_{A,B} = \Lambda X.\lambda f : A \to X \lambda g : B \to X \lambda u.u_X\ f\ g$ and we write $[f, g]_X(u)$ for $\mathsf{case}_{A,B,X}\ f\ g\ u$.

From the computational rules alone we get $[f, g]_X(\mathsf{inl}(x)) = f\ x$ and $[f, g]_X(\mathsf{inr}(x)) = g\ x$.

Lemma 16. $\forall X \forall h : A + B \to X.h = [h \circ \mathsf{inl}, h \circ inr]_X$.

Proof. First of all we can show, by parametricity, that $\forall u : A+B.u = u_{A+B}\ \mathsf{inl}\ \mathsf{inr}$.

Let X be a type, $e : A \to X$, $e' : B \to X$ be terms. Then $[e, e']_X : A + B \to X$. From the computational rule we obtain $(\mathsf{inl}, e) \in [I_A \to \langle [e, e']_X \rangle]$: as $[e, e']_X(\mathsf{inl}(a)) = e\ a$. Similarly we can show that $(\mathsf{inr}, e') \in [I_B \to \langle [e, e']_X \rangle]$. It then follows by parametricity that for any u

$$u_{A+B}\ \mathsf{inl}\ \mathsf{inr}[\langle [e, e']_X \rangle]u_X\ e\ e'$$

In other words, $[e, e']_X(u_{A+B}\ \mathsf{inl}\ \mathsf{inr}) = (u_{A+B}\ \mathsf{inl}\ \mathsf{inr})_X\ e\ e' = u_X\ e\ e'$. As e, e' and X were arbitrary, we obtain $u_{A+B}\ \mathsf{inl}\ \mathsf{inr} = u$ for any u.

Note that this implies $\mathrm{id}_{A+B} = [\mathsf{inl}, \mathsf{inr}]_{A+B}$.

Then one can either use the dinaturality condition, which says that you can move a function in and out of the destructors:

$$\forall u : A+B \forall XY \forall f : X \to Y \forall e : A \to X \forall e' : B \to X.\ u_Y\ (f \circ e)\ (f \circ e') = f(u_X\ e\ e')$$

or one can use another instance of the the parametricy axiom on case:

$$\mathsf{case}_{A+B}[(I_A \to \langle h \rangle) \to (I_B \to \langle h \rangle) \to I_{A+B} \to \langle h \rangle]\mathsf{case}_X$$

which gives us $\mathsf{case}_{A+B}\ \mathsf{inl}\ \mathsf{inr}\ u[\langle h\rangle]\mathsf{case}_X\ (h{\circ}\mathsf{inl})\ (h{\circ}\mathsf{inr})\ u$, i.e. $h(\mathsf{case}_{A+B}\ \mathsf{inl}\ \mathsf{inr}\ u) = \mathsf{case}_X\ (h \circ \mathsf{inl})\ (h \circ \mathsf{inr})\ u$; and the use η. □

Exercise: derive the categorical property of the sums.

5.0.3 Natural numbers

Example 17. *The encoding of the Church numerals is given by* $\mathbf{N} = \forall X.(X \to X) \to X \to X$

For each natural number $n \in \mathbb{N}$ there is a corresponding numeral $\underline{n} = \Lambda X.\lambda f\ x.f^n(x)$. The successor is given by $S = \lambda n.\lambda X.\lambda f\ x.f(n_X f x)$. We have a recursion operator $\mathsf{rec} = \Lambda X.\lambda f\ z.\lambda n.n_X\ f\ z$. One can derive the standard equalities for the recursor using the computational equalities.

Just like for other datatypes, we can prove the η-rule for Church numbers:

Lemma 18. $\forall n : \mathbf{N}.n_{\mathbf{N}}\ S\ \underline{0} = n$.

Proof. By η for functions it suffices to show $(n_{\mathbf{N}}\ S\ \underline{0})_A\ f\ z = n_A\ f\ z$ for an any type A and any terms $f : A \to A$, $z : A$. So, let A, f, and z be arbitrary. We apply the parametricity principle to n. Pick the relation $R \subseteq \mathbf{N} \times A$ to be $R(x, y) \triangleq x_A\ f\ z = y$. One can verify that $(\underline{0}, z) \in R$ and $(S, f) \in [R \to R]$. Hence, $(n_{\mathbf{N}}\ S\ \underline{0}, n_A\ f\ z) \in R$. In other words, $(n_{\mathbf{N}}\ S\ \underline{0})_A\ f\ z = n_A\ f\ z$. □

Lemma 19 (Induction scheme)**.** *Let* $\Phi(x)$ *be a formula of APL with a free* $x : \mathbf{N}$*. Suppose that* $\Phi(\underline{0})$ *holds, and* $\forall n : \mathbf{N}.\Phi(n) \implies \Phi(S\ n)$ *holds. Then* $\forall n : \mathbf{N}.\Phi(n)$.

Proof. By the η rule it suffices to prove $\Phi(n_{\mathbf{N}}\ S\ \underline{0})$. The result follows by (unary) parametricity. □

If we want to show that all elements of $\mathbf{N}$ are numerals, we need a sort of (actual) natural numbers in our logic. If such a sort exists, then we can actually write down a function $\tau : n \mapsto \underline{n}$.

Lemma 20. *Suppose we have a sort* $\mathbb{N}$ *of natural numbers with the usual arithmetic operations in the underlying logic. Then we can show, inside the logic, that every term of the type* $\mathbf{N}$ *is a numeral:* $\forall \phi : \mathbf{N}.\exists n : \mathbb{N}.\phi = \underline{n}$.

Proof. Let $\phi : \mathbf{N}$. Let X be a type and let $f : X \to X$ and $z : X$. Take $k = \phi_{\mathbb{N}}\ (+1)\ 0$ and a relation $R = \{(n, y) \in \mathbb{N} \times X \mid f^n(z) = y\}$. Then:

1. $(0, z) \in R$ by definition $(f^0 = \mathrm{id}_X)$
2. $((+1), f) \in R$: let $(n, y) \in R$. Then $f^{n+1}(x) = f(f^n(x)) = f(y)$ and hence $(n + 1, f\ y) \in R$

It the follows by parametricity that

$$k[R]\phi_X\ f\ x$$

i.e. $f^k(x) = \phi_X\ f\ x$. □

Exericse: what if we define the successor funciton in another way? What is the dinaturality principle for natural numbers?

5.1 Initial algebras

Definition 21. *Given a covariant functor $A[X]$, an* algebra *for A (also called an A-algebra) is an object X and a map $t : A[X] \to X$. A morphism of algebras $\alpha : (X, t) \to (Y, f)$ is a morphism $\alpha : X \to Y$ such that $\alpha \circ t = f \circ A[\alpha]$:*

$$\begin{array}{ccc} A[X] & \xrightarrow{A[\alpha]} & A[Y] \\ {\scriptstyle t}\downarrow & & \downarrow{\scriptstyle f} \\ X & \xrightarrow[\alpha]{} & Y \end{array}$$

An A-algebra algebra (X, t) is called initial *if for any other algebra (Y, f) there is a unique morphism $\alpha : (X, t) \to (Y, f)$. Such an algebra is called* weakly initial *if the uniqueness condition on α is dropped.*

System F allows for encoding of initial algebras for datatypes with positive holes. For $A[X]$ pos, the initial A-algebra is denoted by $\mu X.A[X]$ (or sometimes μA)

$$\mu X.A[X] := \forall Z.((A[Z] \to Z) \to Z)$$

with the combinators

$$\begin{aligned} &\mathsf{fold} : \forall Z.((A[Z] \to Z) \to \mu X.A[X] \to Z) \\ &\mathsf{fold} = \Lambda Z.\lambda(t : A[Z] \to Z).\lambda z.z_Z(t) \end{aligned}$$

$$\begin{aligned} &\mathsf{in} : A[\mu X.A[X]] \to \mu X.A[X] \\ &\mathsf{in} = \lambda x.\Lambda Z.\lambda f.f(A[\mathsf{fold}_Z(f)](x)) \end{aligned}$$

Note that in the lecture notes [2], Amal Ahmed considers System F_μ with recursive types built-in. The combinator in actually corresponds to the term fold in her lecture notes. In theorem 24 we will see that $\mathsf{fold}_{A[\mu A]}(A[\mathsf{in}]) : \mu X.A[X] \to A[\mu X.A[X]]$ corresponds to the term unfold in her lecture notes.

Under those definitions, $(\mu X.A[X], \mathsf{in})$ is an initial A-algebra. The weak initiality is witnessed by the fold combinator:

Lemma 22. *If $t : A[Z] \to Z$ is an A-algebra, then $\mathsf{fold}_Z(t) : \mu A \to Z$ is a morphism from in to t.*

Proof. By computational equalities one can establish that $\mathsf{fold}_Z(t)(\mathsf{in}(x)) = t(A[\mathsf{fold}_Z(t)](x))$. □

Lemma 23. *$(\mu X.A[X], \mathsf{in})$ is an initial algebra.*

Proof. Suppose $f : A[Z] \to Z$ is an algebra and $h : \mu X.A[X] \to Z$ is a morphism $h : (\mu X.A[X], \mathsf{in}) \to (Z, f)$. We will show that $h = \mathsf{fold}_Z(f)$.

Fist of all, we will show that $\forall x : \mu A.x_{\mu X.A[X]}(\mathsf{in}) = x$. By extensionality it suffices to show $(x_{\mu X.A[X]}(\mathsf{in}))_Z(f) = x_Z(f)$ for any $Z, f : A[Z] \to Z$. In other words it suffices to show

$$(x_{\mu X.A[X]}(\mathsf{in}))[\langle \mathsf{fold}_Z(f)\rangle]x_Z(f)$$

By parametricity it suffices to prove that $\mathsf{in}[A[\langle \mathsf{fold}_Z(f)\rangle] \to \langle \mathsf{fold}_Z(f)\rangle]f$. So let $m[A[\langle \mathsf{fold}_Z(f)\rangle]]n \iff m[\langle A[\mathsf{fold}_Z(f)]\rangle]n$, i.e. $A[\mathsf{fold}_Z(f)](m) = n$. Then $\mathsf{fold}_Z(f)(\mathsf{in}(m)) = f(A[\mathsf{fold}_Z(f)](m)) = f(n)$.

Using this equation we reason, $h(x) = h(x_{\mu A}(\mathsf{in}))$. So it suffices to show that $h(x_{\mu A}(\mathsf{in})) = x_Z(f) = \mathsf{fold}_Z(f)(x)$. For that apply parametricity for x with the relation $\langle h\rangle$:

$$x_{\mu A}[[\langle A[h]\rangle \to \langle h\rangle] \to \langle h\rangle]x_Z \implies$$
$$(\mathsf{in}[\langle A[h]\rangle \to \langle h\rangle]f \implies h(x_{\mu A})(\mathsf{in}) = x_Z(f))$$

□

Theorem 24 (Lambek's theorem)**.** *An initial A-algebra is actually an* isomorphism*. It follows that $\mu X.A[X]$ the smallest fixed point of A.*

Proof. Since $(\mu A, \mathsf{in})$ is an A-algebra, so is $(A[\mu A], A[\mathsf{in}])$. So $\mathsf{fold}_{A[\mu A]}(A[\mathsf{in}])$ is a morphism of algebras. It basically shows us that every element of μA is in the image of in; specifically: $x = \mathsf{in}(\mathsf{fold}_{A[\mu A]}(A[\mathsf{in}])(x)) = \mathsf{in}(x_{A[\mu A]}(A[\mathsf{in}]))$.

Once again by extensionality it suffices to show that $x_Z(f) = \mathsf{in}(x_{A[\mu A]}(A[\mathsf{in}]))_Z(f) = \mathsf{fold}_Z(f)(\mathsf{in}(x_{A[\mu A]}(A[\mathsf{in}])))$. This can be shown by using parametricity with the relation the relation $\langle \mathsf{fold}_Z(f) \circ \mathsf{in}\rangle$ and equations for fold.

To show that $\mathsf{fold}_{A[\mu A]}(A[\mathsf{in}])(\mathsf{in}(y)) = y$ we note that $\mathsf{fold}_{A[\mu A]}(A[\mathsf{in}]) \circ \mathsf{in} = A[\mathsf{in}] \circ A[\mathsf{fold}_{A[\mu A]}(A[\mathsf{in}])] = A[\mathsf{in} \circ \mathsf{fold}_{A[\mu A]}(A[\mathsf{in}])] = A[\mathrm{id}_{\mu A}] = \mathrm{id}_{A[\mu A]}$. □

5.2 Existential types

The encoding of the existential types is defined as follows

$$\exists X.A[X] := \forall Y.(\forall X.A[X] \to Y) \to Y$$

with the combinators

$$\mathsf{pack} : \forall X.(A[X] \to \exists Z.A[Z])$$
$$\mathsf{pack} = \Lambda X.\lambda(x : A[X]).\lambda Y.\lambda(f : \forall Z.A[Z] \to Y).f_X(x)$$

$$\mathsf{unpack} : \exists X.A[X] \to (\forall Y.(\forall X.A[X] \to Y) \to Y)$$
$$\mathsf{unpack} = \lambda u.\Lambda Y.\lambda f.u_Y(f)$$

We also define a version of unpack with the unversal quantifier at the top level:

$$\begin{aligned}\overline{\mathsf{unpack}} &: \forall Y.(\forall X.A[X] \to Y) \to \exists X.A[X] \to Y\\ \overline{\mathsf{unpack}} &= \Lambda Y.\lambda f.\lambda u.\mathsf{unpack}(u)_Y(f)\\ &= \Lambda Y.\lambda f.\lambda u.u_Y(f)\end{aligned}$$

Noe that by computational rules (and without η) it is provable that $\mathsf{unpack}(\mathsf{pack}_X(x))_Y(f) = f_X(x)$ (with all free variables universally quantified).

The logical relation principle for existential types that is presented in Ahmed's notes [2] is more appealing then the parametricity principle for the encoding of existentials in System F: to show that two elements $u, w : \exists X.A[X]$ are in the same relational interpretation of the type $\exists X.A[X]$ it suffices to show that there *exists* a relation $S \subset X \times Y$ relating the *implementations* of u and w. Here we will prove this principle.

For a given type $A[X]$ define $R \subset \exists X.A[X] \times \exists X.A[X]$ to be

$$\begin{aligned}R := (u, w).\exists X, Y \exists (x : A[X]) \exists (y : A[Y]) \exists S \subset X \times Y.\\ u = \mathsf{pack}_X(x) \wedge w = \mathsf{pack}_Y(y) \wedge x[A[S]]y\end{aligned}$$

Lemma 25. $\forall (u, w : \exists X.A[X]).u[R]w \Rightarrow u[\exists X.A[X]]w$

Proof. Suppose that $u[R]w$, i.e. $u = \mathsf{pack}_X(x)$, $w = \mathsf{pack}_Y(y)$ and $x[A[S]]y$ for some X, Y, x, y, S. Let $Q \subset C \times D$. We are to show that $(\mathsf{pack}_X(x))_C[(\forall X.A[X] \to Q) \to Q](\mathsf{pack}_Y(y))_D$. So let $f : \forall X.A[X] \to C$, $g : \forall X.A[X] \to D$ and $f[(\forall X.A[X] \to Q)]g$. Then $(\mathsf{pack}_X(x))_C(f) = f_X(x)$ and $(\mathsf{pack}_Y(y))_D(g) = g_Y(y)$. The result follows from the relatedness of f and g. □

The implication in the other direction (lemma 27) implies a certain canonicity result: every element of $\exists X.A[X]$ is in the image of pack. We split this result into two lemmas.

Lemma 26. $\forall u : \exists X.A[X].u = u_{\exists X.A[X]}(\mathsf{pack})$

Proof. It suffices to show that for all Y and for all $f : \forall X.A[X] \to Y$, $u_Y(f) = (u_{\exists X.A[X]}(\mathsf{pack}))_Y(f)$. Note that the right hand side of the equation is just $\overline{\mathsf{unpack}}_Y(f)(u_{\exists X.A[X]}(\mathsf{pack}))$; the equation is then equivalent to the proposition

$$u_{\exists X.A[X]}(\mathsf{pack})[\langle \overline{\mathsf{unpack}}_Y(f) \rangle]u_Y(f)$$

As usual, by parametricity it suffices to show that $\mathsf{pack}[\forall R.A[R] \to \langle \overline{\mathsf{unpack}}_Y(f) \rangle]f$.

So, suppose K, L are types, $R \subset K \times L$, and $k : K, l : L$, and $k[A[R]]l$. We are to show that $\mathsf{pack}_K(k)[\langle \overline{\mathsf{unpack}}_Y(f) \rangle]f_L(l)$, i.e. $\overline{\mathsf{unpack}}_Y(f)(\mathsf{pack}_K(k)) = f_L$. But by computation, $\overline{\mathsf{unpack}}_Y(f)(\mathsf{pack}_K(k)) = f_K(k)$. By the parametricity of f, it is the case that $f_K[A[R] \to I_Y]f_L$. The result follows immediately. □

Lemma 27. $\forall(u, w : \exists X.A[X]).u[\exists X.A[X]]w \Rightarrow u[R]w$

Proof. Let $u, w : \exists X.A[X]$ such that $u[\exists X.A[X]]w$, i.e. $u[\forall Y.(\forall X.A[X] \to Y) \to Y]w$. Hence, in particular, $u_{\exists X.A[X]}[(\forall X.A[X] \to R) \to R]w_{\exists X.A[X]}$. Our claim is that $\mathsf{pack}[\forall X.A[X] \to R]\mathsf{pack}$.

So let $Q \subset C \times D$ and let $m : A[C], n : A[D]$ such that $m[A[Q]]n$. We are to show $\mathsf{pack}_C(m)[R]\mathsf{pack}_D(n)$. In other words,

$$\exists XY\exists(x : A[X])\exists(y : A[Y])\exists S.\mathsf{pack}_C(m) = \mathsf{pack}_X(x) \wedge \mathsf{pack}_D(n) = \mathsf{pack}_Y(y) \wedge x[A[S]]y$$

Clearly we should take $X = C, Y = D, x = m, y = n, S = Q$ and we are done.

Hence, $u_{\exists X.A[X]}(\mathsf{pack})[R]w_{\exists X.A[X]}(\mathsf{pack})$. Finally, lemma 26, $u_{\exists X.A[X]}(\mathsf{pack}) = u$ and $w_{\exists X.A[X]}(\mathsf{pack}) = w$. □

Exercise: derive the canonicity for the existential types (from parametricity), and prove the categorical characterisation:

$$\forall Y\forall(f : \forall X.A[X] \to Y)\exists!(g : (\exists X.A[X]) \to Y)\forall X.(f_X = g \circ \mathsf{pack}_X)$$

References

[1] Gordon Plotkin, Martín Abadi. *A Logic for Parametric Polymorphism.* Typed Lambda Calculi and Applications. TLCA 1993.

[2] Amal Ahmed. *An Introduction to Logical Relations.* `https://www.cs.uoregon.edu/research/summerschool/summer16/notes/AhmedLR.pdf`, 2015. Communicated by Lau Skorstengaard.