Notes on the Chu construction and Recursion

Gordon Plotkin

January 9, 2005

1 Introduction

We consider two kinds of recursive equations in categories of the form $\mathbf{Chu}(K, x)$: type equations and value equations. Throughout it is assumed that K is monoidal closed, complete and co-complete.

Type equations have the form

$$A \cong F(A)$$

where F is a given composition of some of a given set of functors over $\mathbf{Chu}(K, x)$ and we seek A, an object in $\mathbf{Chu}(K, x)$. This is of particular interest when we consider a Chu object $A = (X_A, Y_A, m)$ as a kind of event structure (think of X_A as the set of events, and Y_A as the set of configurations).

Value equations have the form

$$a = f \circ a$$

where, for some object x in $\mathbf{Chu}(K, x)$, $f : x \to x$ is given and we seek $a : 1 \to x$. This is of particular interest when we consider a Chu object as a type of computations (think of X_A as the set of elements of the type, and Y_A as the collection of open sets).

We are particularly interested in $\mathbf{Chu}(\mathbf{Set}, 2)$, and so in Sections 2 and 3 we consider type equations in $\mathbf{Chu}(\mathbf{Set}, 2)$ where the allowed functors are covariant. Even so, as we see in Section 2 these equations may not have solutions. In Section 3 we see a case where they do. In Sections 4 and 5 we consider value equations, obtaining negative results in Section 4 for $\mathbf{Chu}(\mathbf{Set}, 2)$ and similar cases, and positive results in Section 5 if instead K already allows value recursion. Finally, in Section 6, we obtain positive results for general type equations, assuming suitable enrichment of K.

2 Covariant Type Equations: A Negative Result

We consider type equations where F is composed from constants, sum, product, tensor and $\oplus$. It turns out that they do not all have solutions because, it seems, of the interaction between tensor and $\oplus$.

First, consider this equation in $\mathbf{Chu}(\mathbf{Set}, 1)$:

$$A \cong ((B \oplus (B \otimes A)) + B) \times B$$

where B is, say, $(2, 3)$—the third component being omitted as it is determined. Then, setting $A = (X, Y)$, we have that

$$X \cong ((2^{(3^X \times Y^2)} \times (2 \times X)^3) + 2) \times 2$$

and

$$Y \cong ((3 \times (3^X \times Y^2)) \times 3) + 3$$

But then neither X nor Y are empty and X has greater or equal cardinality than 2^{3^X}, which is impossible.

Turning to $\mathbf{Chu}(\mathbf{Set}, 2)$, consider the same equation with, now, B being the image of $(2, 3)$ in $\mathbf{Chu}(\mathbf{Set}, 2)$, that is $(2, 3, m)$, where m is constantly 1. Then the equation does not have a solution in $\mathbf{Chu}(\mathbf{Set}, 2)$ either. For if (X, Y, m) is a solution, then $(X, \{y \in Y \mid \forall x \in X.m(x, y) = 1\})$ is a solution to the first equation.

3 Covariant Type Equations: A Positive Result

We consider equations where F is composed from constants, sum, product and tensor. We restrict ourselves to the case where the constants are extensional Chu spaces (the general case is likely also possible, but seems not to be of much interest).

It is easy to see that if A and B are extensional, so are their sum, product and tensor. It is convenient to consider the full subcategory **Sys** of *set systems* which is equivalent to the full subcategory of the extensional spaces. A set system is a space (X, Y, m) where Y is a collection of subsets of X and $m(x, y) = 1$ iff $x \in y$; when writing such spaces we will omit the (determined) third component.

By the above remarks, the constructions sum, product and tensor can be defined also on **Sys**. More technically, it has sums and products, and there is a tensor which is naturally equivalent to the tensor on $\mathbf{Chu}(\mathbf{Set}, 2)$, modulo the evident inclusion functor $I : \mathbf{Sys} \to \mathbf{Chu}(\mathbf{Set}, 2)$.

Here are the definitions:

Finite Products
The terminal object is $(1, 0)$. The product of (X, Y) and (X', Y') is $(X \times X', Z)$ where $Z = \{y \times X' \mid y \in Y\} \cup \{X \times y' \mid y' \in Y'\}$

Finite Sums
The initial object is $(0, \{\emptyset\})$. The sum of (X, Y) and (X', Y') is $(X + X', Z)$ where $Z = \{y + y' \mid y \in Y, y' \in Y'\}$

Tensor Products
The unit is: $(1, \{1\})$. The tensor product of (X, Y) and (X', Y') is $(X \times X', Z)$ where $Z = \{W \subset X \times X' \mid \forall x' \in X'.W^{-1}x' \in Y', \forall x \in X.Wx \in Y\}$

We now proceed by: considering **Sys** as a large cpo; showing all the operators F on **Sys**, built up from constants, and finite sums, products and tensor products, are continuous; and then using the usual least-fixed point apparatus to solve the type equation.

The partial order on **Sys** is defined by:

$$(X, Y) \leq (X', Y') \text{ iff } X \subset X' \text{ and } \forall y' \in Y'.y' \cap X \in Y$$

One easily verifies that this defines a large partial order. The least element is $0_{\mathbf{Sys}}$. If (X_λ, Y_λ) is a directed system, its lub is (X, Y) where X is the union of the X_λ and Y is $\{y \subset X \mid \forall \lambda.y \cap X_\lambda \in Y_\lambda\}$.

The proofs that sum, product and tensor product are continuous monotone operators on **Sys** are omitted (as are all proofs in this note!). Let F be a continuous operator on **Sys**; let F_∞ be the lub of the sequence of its iterates, $F^n(0_{\mathbf{Sys}})$. This is the least fixed-point of F, and so solves the equation $A = F(A)$ (up to equality!).

Let us now consider the categorical aspects of the solution of the equation. First to each inclusion $(X, Y) \leq (X', Y')$ in **Sys** there is an inclusion morphism $\iota : (X, Y) \to (X', Y')$ where ι has first component the inclusion of X in X' (and recall that morphisms with extensional domain are determined by their first component; they are monos iff the first component is 1-1). These inclusion morphisms are preserved by sum, product and tensor product. So in considering continuous operators F on **Sys** let us now assume they are also functors on **Sys** that preserve inclusion morphisms.

Now consider a directed system $A_\lambda = (X_\lambda, Y_\lambda)$. This yields a system Δ of inclusion morphisms. There is a cone $\rho : \Delta \to \bigvee_\lambda A_\lambda$ where the ρ_λ are inclusion morphisms. This cone is universal in **Sys** (and also in $\mathbf{Chu}(\mathbf{Set}, 2)$). Evidently, F preserves such colimiting cones of inclusion morphisms, as it preserves inclusion morphisms and is continuous. It follows by the Basic Lemma in [SP82] that

$$\eta_F : F(F_\infty) \to F_\infty$$

is the initial F-algebra where η_F is the identity.

Next, suppose that F extends to a functor F' on $\mathbf{Chu}(\mathbf{Set}, 2)$, in the sense that is $F' \circ I$ is naturally equivalent to $I \circ F$. Then $\eta_F : F(F_\infty) \to F_\infty$ is also the initial F'-algebra, again by applying the Basic Lemma.

There other operators possible, such as choice:

$$(X, Y) \uplus (X', Y') = (X + X', \{inl(y) \mid y \in Y\} \cup \{inr(y') \mid y' \in Y'\})$$

where $inl : X \to X + X'$ and $inr : X' \to X + X'$ are the usual injection functions. There should be many more possible operators; for example operations on schedules include some kind of semi-colon. It is not clear what the right generalisation of this is. An interesting question: what are the automata-definable operations; do they coincide with the functors over "the system of subsets" part of $\mathbf{Chu}(\mathbf{Set}, 2)$ that are continuous in the above sense? The idea of using a (large) cpo here goes back to Scott's method of solving domain equations using information systems.

4 Recursion at the level of values: Negative Results

In this and the next section, fix L to be a category with a terminal object.

Definition 1 *An object x has the fixed-point property if every $f : x \to x$ has an fixed-point, that is an $a : 1 \to x$ such that $f \circ a = a$. The category L is said to have fixed-points if every object in it has the fixed-point property.*

Proposition 1 *Suppose that L has an initial object that has the fixed-point property. Then the initial and terminal objects are isomorphic. Further, every object has the fixed-point property, in the trivial sense that the fixed-point of a morphism is the unique map from the initial object.*

Thus as the initial and terminal objects in $\mathbf{Chu}(\mathbf{Set}, 2)$ are distinct, $\mathbf{Chu}(\mathbf{Set}, 2)$ does not have the fixed-point property. On the other hand the proposition also shows that there is a general difficulty, as $\mathbf{Chu}(K, x)$ has initial and terminal objects. So, by the proposition, recursion will be generally available in only a trivial sense.

According to the view I put forwards in my LICS paper, one way to go now is to think of the maps here as *linear* and expect fixed-points of the *continuous* ones, which are defined relative to a co-monad, ! (as in linear logic). So suppose that ! is a co-monad over L.

Definition 2 *L has !-fixed-points if every $f : !x \to x$ has a !-fixed-point, that is an $a : !1 \to x$ such that $f \circ a = a$, where composition in the co-Kleisli category is intended.*

In other words, L has !-fixed-points iff the co-Kleisli category has fixed-points.

Proposition 2 *Suppose that L is a monoidal closed category with an initial object. Let $! : L \to L$ be a co-monad, and suppose that L has !-fixed-points. Then if there is a morphism from I to $!1$, the initial and terminal objects are isomorphic.*

So we cannot even make $\mathbf{Chu}(\mathbf{Set}, 2)$ into a category with !-fixed-points in any reasonable way (and we shall argue next why the assumption that there is a morphism from I to $!1$ is reasonable).

5 Recursion at the level of values: Positive Results

We show that we can transfer good recursive structure from K to $\mathbf{Chu}(K, x)$. Let us assume that we have a co-monad ! on K which makes it a model of intuitionistic linear type theory, in the sense of Seely [See89]. (So, in particular, we make the assumption $!1 \cong I$, rather stronger than that considered above). Write $G : \mathbf{Chu}(K,\mathrm{x}) \rightarrow K$ for the evident forgetful functor, and F for its left adjoint.

Proposition 3 $\mathbf{Chu}(K, x)$ *is a model of classical linear logic, in the sense of Seely [See89] with co-monad* $F \circ ! \circ G$. *Further, if* K *has* !*-fixed-points, then* $\mathbf{Chu}(K, x)$ *has* $F \circ ! \circ G$*-fixed-points.*

An example would be to take K to be $\mathbf{CPO}_\perp$ the category of cpos with bottom and strict continuous maps, and ! to be the lifting functor. And there are many more similar examples, e.g. varying the notion of continuity, passing to categories rather than partial orders, or taking certain categories of algebras.

It seems very likely that many more properties transfer, but I do not consider them here, as I am still working on what the properties associated to recursion should be.

6 General Recursive Type Equations

Here we would like to solve type equations in $\mathbf{Chu}(K, x)$ where F is built up out of constants, product, tensor and $(\cdot)^\perp$. This implies that $\mathbf{Chu}(K, x)$ has fixed-points and so a zero object. It is appropriate to use the $\mathbf{CPO}_\perp$-enriched theory of [SP82]. This says that to solve equations of the above form in a category L it suffices that:

- L is $\mathbf{CPO}_\perp$-enriched
- L has ω-limits
- F is built up out of *locally continuous* functors, which may be of mixed variance.

A functor is locally continuous if it preserves lubs of increasing ω-sequences, that is if it is **CPO**-enriched.

So let us assume that K is $\mathbf{CPO}_\perp$-enriched and that products, sum and tensor are locally continuous. Then all the requirements hold:

Proposition 4 *1* $\mathbf{Chu}(K, x)$ *is* $\mathbf{CPO}_\perp$*-enriched*

2 $\mathbf{Chu}(K, x)$ *has* ω*-limits*

3 Product, tensor and $(\cdot)^\perp$ *are locally continuous*

In particular one can take K to be $\mathbf{CPO}_\perp$. A generalisation of this proposition would be a test for a more general theory of enrichment and recursive type equations.

References

[SP82] Gordon Plotkin and Mike Smyth. The Category-Theoretic Solution of Recursive Domain Equations. SIAM Journal on Computing, Vol. 11, No. 4, pp. 761-783, 1982.

[See89] Robert A. G. Seely. Linear logic, ⋆-autonomous categories and co-free algebras. John Gray and André Scedrov, eds., Categories in Computer Science and Logic, (Proc. A.M.S. Summer Research Conference, June 1987), Contemporary Mathematics **92**, (Am. Math. Soc. 1989).