Clones and Genoids in Lambda Calculus and First Order Logic

Zhaohua Luo

Abstract

A *genoid* is a category of two objects such that one is the product of itself with the other. A genoid may be viewed as an abstract substitution algebra. It is a remarkable fact that such a simple concept can be applied to present a unified algebraic approach to lambda calculus and first order logic.

Content :

Introduction

A *genoid* (A, G) consists of a monoid G with an element $+$, a right act A of G with an element x, such that for any $a \in A$ and $u \in G$ there is a unique element $[a, u] \in G$ such that $x[a, u] = a$ and $+[a, u] = u$. A genoid represents a category with two objects such that one is the dense product of itself with the other.

Email address: zluo@azd.com.
URL: http://www.algebraic.net/cag.

Denote by Act_G the category of right acts of G. The infinite sequence of finite powers of A in Act_G determines a Lawvere theory $Th(A, G)$:

$$A^0, A,\ A^2,\ A^3,\ ...$$

A genoid may be viewed as a Lawvere theory with extra capacity provided by G.

For any right act P of G, we define a new right act $P^A = (P, \circ)$, which has the same universe as P, but the action for any $u \in G$ is defined by $a \circ u = a[x, u+]$. Let $ev : P^A \times A \to P$ be the map defined by $ev(p, a) = p[a, e]$ for any $p \in P$ and $a \in A$. Then $\Lambda : hom(T \times A, P) \to hom(T, P^A)$ defined by $(\Lambda f)t = f(t+, x)$ for any $t \in T$ is bijective, with the inverse $\Lambda' : hom(T, P^A) \to hom(T \times A, P)$ defined by $(\Lambda' g)(t, a) = ev(g(t), a)$. Thus (P^A, ev) is the exponent in the cartesian closed category Act_G. In particular if $T = P = A$ we obtain a canonical bijection

$$\Lambda : hom(A \times A, A) \to hom(A, A^A).$$

This is the starting point of lambda calculus.

We define an *extensive lambda genoid* to be a genoid (A, G) together with two homomorphisms $\lambda : A^A \to A$ and $\bullet : A \times A \to A$ such that $(\Lambda \bullet)\lambda = id_A$ (β-conversion) and $\lambda(\Lambda \bullet) = id_A$ (η-conversion). This means that A and A^A are isomorphic as right acts of G. Conversely, any genoid (A, G) such that A and A^A are isomorphic determines an extensive lambda genoid.

A *quantifier algebra of a genoid* (A, G) is a Boolean algebra P which is also a right act of G with Boolean algebra endomorphisms as actions, together with a homomorphism $\exists : P^A \to P$ such that $\exists(p \vee q) = (\exists p) \vee (\exists q)$ and $p < (\exists p)+$ for any $p, q \in P$. The study of a first order theory can also be reduced to the study of a quantifier algebra for a genoid (A, G).

We say a genoid (A, G) is a *clone* if G is the countable power A^ω of A. Algebraically the class of clones forms a (non-finitary) variety. A general theory of *clones over any full subcategory of a category* is presented at the end of this paper.

The theory of clones considered in this paper originated from the theory of monads. Two equivalent definitions of monads, namely *monads in clone form* and *monads in extension form* given by E. Mane [7], can be interpreted as only defined over a given subcategory of a category. These are *clones in algebraic form* and *clones in extension form* over a subcategory respectively. It turns out that these two forms of clones are no longer equivalent unless the subcategory is dense. But morphisms of clones, algebras of clones, and morphisms of algebras can all be defined for these two types of clones. Since many familiar algebraic structures, such as monoids, unitary Menger algebras, Lawvere theories, countable Lawvere theories, classical and abstract clones are all special cases of clones over various dense subcategories of **Set**, the syntax and semantics of these algebraic structures can be developed in a unified way, so that it is much easier to extend these results to many-sorted sets.

1 Genoids

A *genoid theory* is a category $(\mathbf{A}, \mathbf{G})$ of two objects together with two morphisms $x : \mathbf{G} \rightarrow \mathbf{A}$ and $+ : \mathbf{G} \rightarrow \mathbf{A}$ such that $(\mathbf{G}, x, +)$ is the product of $\mathbf{A}$ and $\mathbf{G}$, i.e. $\mathbf{G} = \mathbf{A} \times \mathbf{G}$. We also assume that $\mathbf{G}$ is a dense object, although this is not essential. A *left algebra* of $(\mathbf{A}, \mathbf{G})$ is a functor from $(\mathbf{A}, \mathbf{G})$ to the category **Set** of sets preserving the product. A *morphism of left algebras* is a natural transformation.

Recursively we have

$$\mathbf{G} = \mathbf{A}^n \times \mathbf{G} = \mathbf{A} \times ... \times \mathbf{A} \times \mathbf{G}$$

for any positive integer n. If $\mathbf{G}$ is the countable power of $\mathbf{A}$, i.e.

$$\mathbf{G} = \mathbf{A}^\omega = \mathbf{A} \times \mathbf{A} \times ...$$

then we say that $(\mathbf{A}, \mathbf{G})$ is a *clone theory.*

Let $A = hom(\mathbf{G}, \mathbf{A})$ and $G = hom(\mathbf{G}, \mathbf{G})$. Then G is a monoid and A is a right act of G. For any pair $(a, u) \in A \times G$ let $[a, u] : \mathbf{G} \rightarrow \mathbf{G}$ be the unique morphism such that $x[a, u] = a$ and $+[a, u] = u$. Then $u = [xu, +u]$ for any $u \in G$.

Definition 1 *A genoid $(A, G, x, +, [\ ])$ consisting of a monoid (G, e), a right act A of G, $x \in A$, $+ \in G$, and a map $[\ ] : A \times G \rightarrow G$ such that for any $a \in A$ and $u \in G$ we have*

(G1) $x[a, u] = a$.

(G2) $+[a, u] = u$.

(G3) $u = [xu, +u]$.

A genoid is simply denoted by (A, G). Clearly any genoid theory $(\mathbf{A}, \mathbf{G})$ determines a genoid (A, G). Since we assume $\mathbf{G}$ is a dense object, $(\mathbf{A}, \mathbf{G})$ is uniquely determines by (A, G). Conversely if (A, G) is a genoid then the subcategory of right acts of G generated by A and G is a genoid theory. Hence the notions of genoid theory and genoid are equivalent.

Remark 2 *Genoids form a variety of 2-sorted heterogeneous finitary algebras with universes A and G. A genoid (A, G) is called standard if it is generated by A as a 2-sorted algebra.*

Suppose (A, G) is a genoid. We have $e = [x, +]$ and $[a, u]v = [au, vu]$ for any $a \in A$ and $u \in G$. We shall write $[a_1, a_2, ..a_n, u]$ for $[a_1, [a_2, [...[a_n, u]...]]]$.

Let $x_1 = x$, $x_{i+1} = x_i+$ for any $i > 0$. Then axiom (G3) extends to

$$u = [x_1u, x_2u, ..., x_nu, +^nu]$$

for any $n > 0$.

It is easy to define a *many-sorted genoid* for a nonempty set S of sorts:

Definition 3 *An S-genoid theory is a category $(\{\mathbf{A}^s\}_{s\in S}, \mathbf{G})$ of objects $\{\mathbf{A}^s\}_{s\in S}$ and $\mathbf{G}$, together with morphisms $\{x^s : \mathbf{G} \to \mathbf{A}^s\}_{s\in S}$ and $\{+^s : \mathbf{G} \to \mathbf{G}\}_{s\in S}$ such that $(\mathbf{G}, x^s, +^s)$ is the product of $\mathbf{A^s}$ and $\mathbf{G}$ for all $s \in S$, and $x^s+^t = x^s$, $+^s+^t = +^t+^s$ for any distinct $s, t \in S$. We also assume that $\mathbf{G}$ is a dense object. A left algebra of $(\{\mathbf{A}^s\}_{s\in S}, \mathbf{G})$ is a functor from this category to* **Set** *preserving the products.*

Definition 4 *An S-genoid is a pair (A, G) consisting of a monoid G and a set $A = \{(A^s, G, x^s, +^s, [\ ]^s)\}_{s\in S}$ of genoids such that $x^s+^t = x^s$ and $+^s+^t = +^t+^s$ for any two distinct elements $s, t \in S$.*

Suppose (A, G) is an S-genoid. For any $s \in S$ let $\kappa^s_n : G \to (A^s)^n$ be the map sending each $u \in G$ to $[x^s_1u, x^s_2u, ..., x_nu]^s \in (A^s)^n$. Let (T, n) be a pair consisting of a finite subset T of S and an integer $n > 0$. We say an element p of P *has a finite support* (T, n) *(or p has a finite rank n)* if $pu = pv$ for any $u, v \in G$ with $\kappa^s_n(u) = \kappa^s_n(v)$ for any $s \in T$. We say p *has a finite rank* 0 (or p *is closed*) if $pu = pv$ for any $u, v \in G$. An element of P is called *finitary* if it has a finite support. We say P is *locally finitary* if any of its element is finitary. We say (A, G) is a *locally finitary genoid* if A is locally finitary as a right act of G.

Example 1.1 *An algebraic genoid is a monoid G together with two elements $x, + \in G$ such that $xx = +x = x$, and $(xG, G, x, +)$ is a genoid. Algebraic genoids form a finitary variety.*

Example 1.2 *An algebraic S-genoid with a zero element 0 is a monoid G with a zero element 0 together with a set $\{x^s, +^s\}_{s\in S}$ of pairs of elements of G such that*

1. $x^sx^t = 0$ *for any distinct* $s, t \in S$.

2. $+^sx^t = x^tx^t = x^t$ *for any* $s, t \in S$.

3. (A, G) is an S-genoid with $A = \{(x^s G, G, x^s, +^s)\}_{s \in S}$.

Algebraic S-genoids form a finitary variety.

2 Clones

Let $\mathbb{N}$ be the set of positive integers.

Definition 5 *A clone theory over $\mathbb{N}$ is a category $(\mathbf{A}, \mathbf{G})$ of two objects together with an infinite sequence of morphisms $x_1, x_2, ..$ from $\mathbf{G}$ to $\mathbf{A}$ such that $(\mathbf{G}, \{x_1, x_2, ..\})$ is a countable power of $\mathbf{A}$. A left algebra of $(\mathbf{A}, \mathbf{G})$ is a functor from $(\mathbf{A}, \mathbf{G})$ to **Set** preserving the countable power.*

Let $A = hom(\mathbf{G}, \mathbf{A})$ and $G = hom(\mathbf{G}, \mathbf{G})$. Then G is a monoid and A is a right act of G. For any infinite sequence $a_1, a_2, ...$ of elements of A let $[a_1, a_2...] \in G$ be the unique morphism such that $x_i[a_1, a_2...] = a_i$ for any integer $i > 0$. Then $u = [x_1 u, x_2 u, ...]$ for any $u \in G$. Any clone theory determines a genoid theory with $x = x_1$, and $+ = [x_2, x_3, ...]$.

Definition 6 *A clone in extension form over $\mathbb{N}$ is a nonempty set A such that*

(i) The set A^* of all the infinite sequences $[a_1, a_2, ...]$ of elements of A is a monoid with a unit $[x_1, x_2, ...]$.

(ii) A is a right act of A^*.

(iii) $x_i[a_1, a_2, ...] = a_i$ for any $[a_1, a_2, ...]$ and $i > 0$.

Any clone A in extension form determines a genoid $(A, A^*, x_1, +, [\ ])$ with $+ = [x_2, x_3, ...]$ and $[a_1, [b_1, b_2, ..]] = [a_1, b_1, b_2, ..]$. Thus we may speak of locally finitary clones. Conversely, assume (A, G) is a any genoid. Denote by $\mathcal{F}(A)$ the set of finitary elements of A. For any $a \in \mathcal{F}(A)$ with a finite rank $n > 0$ and $[a_1, a_2, ...] \in \mathcal{F}(A)^*$ define

$$a[a_1, a_2, ...] = a[a_1, a_2, ..., a_n, e],$$

which is independent of the choice of n. Let

$$[a_1, a_2, ...][b_1, b_2, ...] = [a_1[b_1, b_2, ...], a_2[b_1, b_2, ...], ...].$$

Then $\mathcal{F}(A)^*$ is a monoid with the unit $[x_1, x_2, ...]$, $\mathcal{F}(A)$ is a right act of A^*, and $x_i[a_1, a_2, ...] = a_i$. Thus $\mathcal{F}(A)$ is a locally finitary clone. If $A = \mathcal{F}(A)$ is locally finitary then we have a canonical homomorphism of genoids $(A, G) \to (A, A^*)$ sending each $u \in G$ to $[x_1 u, x_2 u, ...x_n u, ...]$.

Remark 7 *Clones form a variety of infinitary algebras with universe A.*

Remark 8 *The notion of a locally finitary clone is equivalent to that of a Lawvere theory (without the 0-ary object).*

Definition 9 *Let A be a clone. A left algebra of A (or a left A-algebra) is a set D together with a multiplication $A \times D^{\mathbb{N}} \to D$ such that for any $a \in A$, $[a_1, a_2, ...] \in A^{\mathbb{N}}$ and $[d_1, d_2, ...] \in D^{\mathbb{N}}$*
1. $(a[a_1, a_2, ...])[d_1, d_2, ...] = a([a_1[d_1, d_2, ...], a_2[d_1, d_2, ...], ...]$.
2. $x_i[d_1, d_2, ...] = d_i$.

Remark 10 *Left algebras of clones are main objects of study in universal algebra (cf. [8]).*

Definition 11 *A clone in algebraic form over $\mathbb{N}$ is a nonempty set A such that the set $A^{\mathbb{N}}$ of maps from $\mathbb{N}$ to A is a monoid and $(ru)v = r(uv)$ for any maps $r : N \to N$ and $u, v : N \to A$.*

Remark 12 *Since $\mathbb{N}$ is dense in* **Set***, one can show easily that the two forms of clones over $\mathbb{N}$ are equivalent. Therefore in the following we shall not distinguish these two types of clones.*

3 Binding Algebras

Let (A, G) be a genoid. The map $\delta : G \to G$ sending u to $[x, u+]$ is an endomorphism of monoid G. Let $- = [x, e]$. One can show that $(\delta, +, -)$ is a monad on the one-object category determined by the monoid G, as we have $+- = (\delta+)- = e$ and $-- = (\delta-)-$. The Kleisli category of this monad is the monoid $(G, *)$ with $u * v = u[x_1, v]$.

If P is any right act of G denote by P^A the new right act $(P, \circ)$ of G defined by $p \circ u = p(\delta u) = p[x, u+]$ for any $p \in P$ and $u \in G$. Let $ev : P^A \times A \to P$ be the map defined by $ev(p, a) = p[a, e]$. Define $\Lambda : hom(T \times A, P) \to hom(T, P^A)$ by $(\Lambda f)t = f(t+, x)$ for any $t \in T$, and $\Lambda' : hom(T, P^A) \to hom(T \times A, P)$ by $(\Lambda' g)(t, a) = ev(g(t), a)$. Then both $\Lambda'\Lambda$ and $\Lambda\Lambda'$ are identities (which implies that Λ is bijective). Thus (P^A, ev) is the exponent in the the cartesian closed category Act_G of right acts of G.

Let $\Delta : Act_G \to Act_G$ be the functor sending each act P to P^A, and each morphism $f : P \to Q$ to $f : P^A \to Q^A$. The actions of $+$ and $-$ induces two natural transformations $+ : Id \to \Delta$ and $- : \Delta^2 \to \Delta$. It is easy to see that $(\Delta, +, -)$ is a monad on Act_G.

Definition 13 *A binding operation is a homomorphism $P^A \to P$. A cobind-*

ing operation is a homomorphism $P \to P^A$.

Remark 14 *We assume $y, z, w, ..., y_1, y_2, ..., z_1, z_2, ... \in \{x_1, x_2, x_3...\}$, which are called syntactical variables. Suppose σ is a binding operation. The traditional operation $\sigma x_i : P \to P$ (for each variable x_i) is defined as the derived operation:*

$$\sigma x_i.p = \sigma(p[x_2, x_3, ..., x_i, x_1, +^{i+1}]).$$

If $y = x_i$ then $\sigma y.p$ means $\sigma x_i.p$.

Example 3.1 *For any $p \in P$ we have*

1. $\sigma x_1.p = \sigma(p[x_1, ++]) = (\sigma p)+$.

2. $\sigma p = (\sigma x_1.p)-$.

3. *If p has a finite rank $n > 0$ then σp has a finite rank $n - 1$. Thus $\sigma^n p$ is closed.*

4. *If p is closed then σp is closed.*

Definition 15 *Let S be any set of sorts. Let k be a non-negative integer. An S-arity of rank k is a finite sequence $\alpha =< (s_1, n_1), ..., (s_k, n_k), (s_{k+1}, n_{k+1}) >$ with $s_i \in S$ and $n_i \geq 0$. An α-binding operation on a right act P of an S-genoid (A, G) is a homomorphism of right acts*

$$(\Delta^{s_1})^{n_1} P \times ... \times (\Delta^{s_k})^{n_k} P \to (\Delta^{s_{k+1}})^{n_{k+1}} P$$

(assume $(\Delta^{s_i})^0 P = P$), where Δ^{s_i} is the functor sending each right act Q of G to $Q^{A^{s_i}}$.

Definition 16 *An S-signature is a set Σ of operation symbols such that for each symbol $f \in \Sigma$ an S-arity $ar(f)$ is attached. A Σ-binding algebra for an S-genoid (G, A) is a right act P of G such that for each symbol $f \in \Sigma$ an $ar(f)$-binding operation f^P on P is assigned.*

Remark 17 *We shall drop all the references to the elements of S if S is a singleton. Thus an arity of rank k is simply a finite sequence $\alpha =< n_1, ..., n_k, n_{k+1} >$ of non-negative integers.*

Example 3.2 1. *A binding operation is a $< 1, 0 >$-operation.*

2. *A cobinding is a $< 0, 1 >$-operation.*

3. *A homomorphism $P^2 \to P$ is a $< 0, 0, 0 >$-operation.*

4. *A homomorphism $P^0 \to P$ is a $< 0 >$-operation, which reduces to a closed element of P.*

Lambda genoids and predicate algebras defined below are examples of binding algebras.

4 Lambda Calculus

A genoid (A, G) is *reflexive* if A^A is a retract of A (as right acts of G). It is *extensive* if A^A is isomorphic to A.

A *lambda genoid* is a genoid (A, G) together with two homomorphisms $\lambda : A^A \to A$ and $A^2 \to A$ of right acts of G. If $((\lambda a)+)x = a$ for any $a \in A$ we say A is a *reflexive lambda genoid* (or a λ_β-genoid). If furthermore $\lambda((a+)x) = a$ for any $a \in A$ then we say A is an *extensive lambda genoid* (or a $\lambda_{\beta\eta}$-genoid). Thus a genoid is reflexive (resp. extensive) iff it is the underlying genoid of a reflexive (resp. extensive) lambda genoid.

Remark 18 *Lambda clones (resp. reflective lambda clones, resp. extensive lambda clones) form a variety of (infintary) algebras. The initial lambda clone is precisely the clone determined by terms in $\lambda\sigma$-calculus (cf [2]).*

Remark 19 *Lambda algebraic genoids (resp. reflective lambda algebraic genoids, resp. extensive lambda algebraic genoids) form a variety of finitary algebras.*

Remark 20 *The classical operation $\lambda x_i : A \to A$ (for each variable x_i) is defined as the derived operation:*

$$\lambda x_i.a = \lambda(a[x_2, x_3, ..., x_i, x_1, +^{i+1}]).$$

If $y = x_i$ then $\lambda y.a$ means $\lambda x_i.a$.

Suppose (A, G) is an extensive lambda genoid. Assume $a, b, c \in A$ and $u \in G$. Here are some useful formulas:

(1) $(\lambda a)b = a[b, e]$.

(2) $((\lambda a)u)b = a[b, u]$.

(3) $(\lambda a+)b = a$.

(4) $(\lambda^n a) +^n x_n...x_1 = a$ for any integer $n > 0$.

(5) If a has a finite rank $n > 0$ then $\lambda^n a$ is closed and $(\lambda^n a)x_n...x_1 = a$. Thus

$$(\lambda^n a)a_n...a_1 = (\lambda^n a)x_n...x_1[a_1, ..., a_n, e] = a[a_1, ..., a_n]$$

(6) An element a has a finite rank $n > 0$ if and only if there is a closed element

c such that

$$a = cx_n...x_1.$$

The following closed terms play important roles in lambda calculus (notation: $\lambda y_1...y_n.a = \lambda y_1.(\lambda y_2.(..(\lambda y_n.a)...)).$)

$\mathbf{I} = \lambda y.y = \lambda x_1.$

$\mathbf{K} = \lambda yz.y = \lambda\lambda x_2.$

$\mathbf{S} = \lambda yzw.yw(zw) = \lambda\lambda\lambda x_3x_1(x_2x_1).$

It follows from (5) we have

$\mathbf{I}a = x_1[a, e] = a.$

$\mathbf{K}ab = x_2[b, a, e] = a.$

$\mathbf{S}abc = (x_3x_1(x_2x_1))[c, b, a, e] = ac(bc).$

Definition 21 *Let S be a nonempty set carrying a binary operation $\rightarrow$. An S-simply typed lambda genoid is an S-genoid (A, G) together with homomorphisms $\{\lambda^s : (A^t)^{A^s} \rightarrow A^{s\rightarrow t}\}$ and $\{A^{s\rightarrow t} \times A^s \rightarrow A^t\}$ such that for any $a \in A^t$ and $c \in A^{s\rightarrow t}$ we have $(\lambda^s a) +^s x^s = a$ and $\lambda^s(c +^s x^s) = c$.*

5 First Order Logic

A *predicate algebra of an S-genoid* (A, G) is a right act P of G together with homomorphisms of right acts $\{\exists^s : P^{A^s} \rightarrow P\}_{s\in S}$, $\mathsf{F} : P^0 \rightarrow P$, and $\Rightarrow : P \times P \rightarrow P$.

Define the following derived operations on P:

$\neg p = (p \Rightarrow \mathsf{F}),$

$\mathsf{T} = \neg\mathsf{F},$

$p \vee q = (\neg p) \Rightarrow q,$

$p \wedge q = \neg(p \Rightarrow \neg q).$

We say P is a *reduced predicate algebra* if for any $p, q, \in P$ and $s \in S$

(i) $(\vee, \wedge, \neg, \mathsf{F}, \mathbf{T})$ defines a Boolean algebra P.

(ii) $\exists^s(p \vee q) = (\exists^s p) \vee (\exists^s q).$

(iii) $p < (\exists^s p)+^s$

A reduced predicate algebra is also called a *quantifier algebra*.

Remark 22 *The class of predicate algebras (resp. quantifier algebras) of a genoid is a variety of finitary algebras.*

An *interpretation of a predicate algebra* P is a pair (Q, μ) consisting of a reduced predicate algebra Q and a homomorphism $\mu : P \to Q$ of predicate algebras. We say $p \in P$ is *logical valid* (written $\models p$) if for any interpretation (Q, μ) we have $\mu(p) = \mathsf{T}$. If $p, q \in P$ then we say that p and q are *logically equivalent* (written $p \equiv q$) if $(p \Rightarrow q) \wedge (q \Rightarrow p)$ is logically valid. Then $\equiv$ is a congruence on P. The set of congruence classes of P with respect to the congruence $\equiv$ is a reduced predicate algebra called the *Lindenbaum-Tarski algebra of* P. (see [8] for a further development of the theory of predicate algebras).

Theorem 23 *Suppose A is a locally finitary clone. Any left algebra D of A determines a locally finitary reduced predicate algebra $P(D^{\mathbb{N}})$, where $P(D^{\mathbb{N}})$ is the power set of $D^{\mathbb{N}}$. A locally finitary predicate algebra of A is reduced iff it belongs to the variety generated by predicate algebras $P(D^{\mathbb{N}})$ for all left algebras D of A.*

6 Clones Over A Subcategory

Definition 24 *Let $\mathbf{N}$ be a full subcategory of a category $\mathbf{X}$. A clone (in extension form, or Kleisli triple) over $\mathbf{N}$ is a system $T = (T, \eta, ^* -)$ consisting of functions*

(a) $T : Ob\mathbf{N} \to Ob\mathbf{X}$.

(b) η assigns to each object $A \in \mathbf{N}$ a morphism $\eta_A : A \to TA$.

(c) $*-$ maps each morphism $f : B \to TC$ with $B, C \in \mathbf{N}$ to a morphism $*f : TB \to TC$, such that for any $g : C \to TD$ with $D \in \mathbf{N}$

(i) $*f * g = *(f * g)$.

(ii) $\eta_B * f = f$.

(iii) $*\eta_C = id_{TC}$.

Remark 25 *If $\mathbf{N} = \mathbf{X}$ we obtain the original definition for a Kleisli triple over a category, which is an alternative description of a monad.*

Definition 26 *Let* $\mathbf{N}$ *be a full subcategory of a category* $\mathbf{X}$*. A clone theory in extension form (resp. in algebraic form) over* $\mathbf{N}$ *is a pair* $(\mathbf{K}, T)$ *where* $\mathbf{K}$ *is a category and* T *is a functor* $T : \mathbf{K} \to \mathbf{X}$ *(resp.* T *is a function* $T : Ob\mathbf{K} \to Ob\mathbf{X}$*) such that for any* $A, B, C, D \in \mathbf{N}$

(i) $Ob\mathbf{N} = Ob\mathbf{K}$*.*

(ii) $\mathbf{K}(A, B) = \mathbf{X}(A, TB)$*.*

(iii) $f(Tg) = fg$ *(resp.* $r(fg) = (rf)g$*) for any* $f \in \mathbf{K}(A, B)$*,* $g \in \mathbf{K}(B, C)$ *and* $r \in \mathbf{N}(D, A)$*.*

Remark 27 *If* $\mathbf{N}$ *is dense then these two forms of clone theory are equivalent. In particular, a clone theory over* $\mathbf{N} = \mathbf{X}$ *in both forms corresponds to a monad on* $\mathbf{X}$*.*

Remark 28 *Any clone over* $\mathbf{N}$ *determines a clone theory in extension form over* $\mathbf{N}$*, called its Kleisli category. Conversely any clone theory in extension form over* $\mathbf{N}$ *induces a clone over* $\mathbf{N}$ *(see [8] for details).*

Example 6.1 *Let* $\mathbf{X} = \mathbf{Set}$ *be the category of sets.*

1. A clone over a singleton is equivalent to a monoid.

2. A clone over a finite set is equivalent to a unitary Menger algebra.

3. A clone over a countable set is equivalent to a clone over $\mathbb{N}$ *defined above.*

4. A clone over the subcategory $\{0, 1, 2, ...\}$ *of finite sets is equivalent to a clone in the classical sense (i.e. a Lawvere theory).*

Remark 29 *1. A clone theory over* $\mathbf{N} = \mathbf{X}$ *is equivalent to a monad on* $\mathbf{X}$*.*

2. A clone (or a monad) over a one-object category is called a Kleisli algebra.

3. Any genoid (A, G) *determines a Kleisli algebra since* $(\delta, +, -)$ *is a monad on the one-object category* G*.*

7 Relate Work

In classical universal algebra one studies left algebras of a clone over $\mathbb{N}$. Such a clone can be defined in many different ways (see [3][9][10][15]). Our approach to binding algebras and untyped lambda calculus was greatly inspired by [1]. For other algebraic approaches to untyped lambda calculus see [2][5][6][12][14][16].

Our definition of a quantifier algebra of a genoid was based on Pinter [13] (see also [4][11]).

References

[1] M. Fiore, G. Plotkin, D. Turi, *Abstract syntax and variable binding* Proceedings of the 14th Annual IEEE Symposium on Logic in Computer Science, LICS '99, (1999) 193–202

[2] C. Hankin, *An Introduction to Lambda Calculus for computer Scientists* College Publications (February 2, 2004)

[3] P. M. Cohn, *Universal Algebra*, Kluwer Academic Publishers (1981)

[4] J. Cirulis, *An algebraization of first order logic with terms* Colloq. Math. Soc. J. Bolyai 54, Algebraic logic, 1991, 125146

[5] B. Jacobs, *Simply typed and untyped lambda calculus revisited* In: M.P. Fourman, P.T. Johnstone and A.M. Pitts (eds.) Applications of Category Theory in Computer Science (LMS 177, Camb. Univ. Press, 1992), 119 - 142.

[6] M. Maggesi, A. Hirschowitz, *The algebraicity of the lambda-calculus* preprint 2007 arXiv:0704.2900.

[7] E. Manes, *Algebraic Theories*, Springer-Verlag, 1976.

[8] Z. Luo, *Clone theory, its syntax and semantics, and applications to lambda calculus and algebraic logic* (2007).

[9] W. D. Neumann, *Representing varieties of algebras by algebras*, J. Austral. Math. Soc. **11** (1970), 1–8.

[10] B. Pareigis and H. Rohrl, *Left linear theories – A generalization of module theory*, Applied Categorical Structures. Vol. **11**, No. 2, (1994), 145-171.

[11] B. Plotkin, *Algebraic logic, varieties of algebras and algebraic varieties* arXiv:math/0312420

[12] A. Obtulowicz, A. Wiweger, *Categorical, functorial, and algebraic aspects of the. type-free lambda calculus* Banach Center Publications 9, Warsaw (1982), 399-422

[13] C, C. Pinter, *A simple algebra of first order logic*, Notre Dame Journal of Formal Logic, Vol. **XIV**, No. 3, (1973), 361-366.

[14] A. Salibra, R. Goldblatt *A finite equational axiomatization of the functional algebras for the lambda calculus* Information and Computation, vol. 148, n.1, pp.71-130 (1999).

[15] B. M. Schein, V. S. Trohimenko, *Algebras of multiplace functions*, Semi-group Forum **17**, (1979), 1-64

[16] P. Selinger, *The lambda calculus is algebraic* Journal of Functional Programming (2002), 12: 549-566