LFCS
Laboratory for Foundations of Computer Science
Department of Computer Science - University of Edinburgh

Equational Characterization of Binding (Extended Abstract)

Equational Characterization of Binding
(Extended Abstract)

by

Sun, Yong

LFCS Report Series

ECS-LFCS-89-94
(also published as CSR-312-89)

LFCS
Department of Computer Science
University of Edinburgh
The King's Buildings
Edinburgh EH9 3JZ

September 1989

Equational Characterization of Binding* (Extended Abstract)

Sun, Yong†

September, 1989

Brief Overview

Binding appears in logic, programming and concurrency, e.g, it appears in lambda calculus [Church 41, Barendregt 84] say the lambda abstraction of lambda calculus. However, the binding in lambda calculus is unary. Certainly, we can generalize the idea of the unary binding to an arbitrary finite number of binding. Algebraically, we can consider an extension of the framework of universal algebra[1] *to accommodate the new feature of binding. Therefore, the new extended signature is of second-order instead of first-order. Like the well-known Church-Rosser property of lambda calculus leads to equational presentation of lambda calculus, we seek an equational characterization of binding in the new framework.*

Peter Aczel has given a Church-Rosser theorem for bos[2] *in [Aczel 78] and a definition for binding algebras in [Aczel 80], which is in Birkhoff's approach. Naturally, we would like to characterize binding in this Birkhoff's approach. Unfortunately, the semantics given in this way for binding equations does not work. Therefore, we have to find either a remedy for it or a new semantics model for binding. We will present a solution for each.*

(a) For a remedy, we discover the admissible condition for the Birkhoff's approach to work. This condition is necessary and sufficient. A deduction system of admissible binding equations can be supplied. Some problems are remained open.

(b) For a new semantics model, we will give a new definition for binding algebras. The new binding algebras are intentional, since the previous definition is extensional. A sound and complete equational calculus for intentional binding algebras will be provided. Examples of its application can be given as well.

Due to space limit, this extended abstract mainly serves as an introduction to new concepts and results. Since every new framework has to defend itself in the beginning, this abstract can be regarded to serve this purpose and it defends in an intuitive (or informal) way.

---

*The work presented is the core of the work in [Sun 89?] (in preparation, 1989) and it has been presented in the Jumelage meeting (typed lambda calculus workshop) in Edinburgh, September 1989.

†LFCS, Department of Computer Science, University of Edinburgh, King's Buildings, Mayfield Road, Edinburgh EH9 3JZ, U.K. E-mail : sun%ed.lfcs@nsf-relay.ac.uk

[1] If you prefer categorical terminology, you should note that universal algebra (or free algebra) can be recovered by an adjuction or a monad, see [Maclane 71, Rydeheard 85] for more details.

[2] It stands for *binding operators*.

Contents

1 Introduction to Binding

Semantics is the essence; and languages, expressions and mathematical machinery obtain their remarkable power by abstracting semantics (or meanings) to syntactical level. I substantiate this claim by a few examples, say, the existential quantifier in first-order logic [Dalen 83, Hu 82], the let-expression in ML [Gordon 79], the lambda-abstraction in lambda calculus [Church 41, Barendregt 84], input command in CCS [Milner 80, Milner 89] or CSP [Hoare 78, Zhou 81], ... etc. Formally they can be represented as the existential quantifier : $\exists x.P(x)$, let-expression : *let x be d in e*, λ-expression : $\lambda x.M$, and the input command : $\alpha?x.B(x)$. However, these examples are used here to bring in the idea of binding. Observing carefully, you would discover that there is a general representation of $\exists$, *let*, λ and α-*in*, i.e. binding operators (bo for short).

To put this point of bo more explicitly, I denote a binding as $\langle \_ : \_ \rangle$, where the first _ shows the bound variables in the second _ (term), and alter the previous examples accordingly, i.e. the existential quantifier : $\exists(\langle x : P(x)\rangle)$, let-expression : $let(\langle x : e\rangle, d)$, λ-expression : $\lambda(\langle x : M\rangle)$, the input command : $\alpha\text{-}in(\langle x : B(x)\rangle)$. Nevertheless, $\langle \_ : \_ \rangle$ is a unary binding. It binds the free variable x in terms $P(x)$, e, M and $B(x)$ to make the function terms $\langle x : P(x)\rangle$, $\langle x : e\rangle$, $\langle x : M\rangle$ and $\langle x : B(x)\rangle$ respectively.

It is quite easy to generalize the idea of unary binding $\langle \_ : \_ \rangle$ to arbitrary finite binding $\langle \underbrace{\_, \_, ..., \_}_{n\ times} : \_ \rangle$ $(n \in Nat)$, even to infinite binding, say countably infinite (∞)

$\langle \underbrace{\_, \_, \ldots\ldots}_{\infty\ times} : \_ \rangle$. But I only consider the arbitrary finite binding in this paper[3].

The above observation lead us to consider binding in a broad sense. The wide spread examples above, which run over logic, programming and concurrency, suggest that binding deserves a more general framework for investigation. A framework with great generality in our mind is universal algebras. We can think of extending the framework of universal algebras to include binding. In next section, we proceed this idea and work out what is the signature for binding in such a framework.

2 Signature for Binding

The signature for binding can be thought of as an extension of ordinary signature Σ to include all arbitrary finite binding operators. Therefore, we have to raise the order of the language from first order to second order, i.e. the new extended Σ^{bo} has operations of second order.

We know that second order logic is incomplete [Boolos 80] and that there is a potential inconsistency in equational theories if we combine λ-abstraction with equational theories [Breazu-Tannen 88]. It is very reasonable for us to accommodate the new feature of binding in some restricted way such that the new extended language will still be very rich.

We let the new extended signature Σ^{bo} be indexed by $(S^* \times S)^* \times S^* \times S$ or more meaningful $(S^* \Rightarrow S)^* \times S^* \rightarrow S$, where S is $Sort(\Sigma^{bo})$ and $*$ is the Kleene star operator. Such a Σ^{bo} is powerful enough to include all previous mentioned examples. Let us demonstrate this for the single-sorted case, the existential quantifier : $\exists \in \Sigma^{bo}_{(\bullet \Rightarrow \bullet) \rightarrow \bullet}$, let-expression : $let \in \Sigma^{bo}_{(\bullet \Rightarrow \bullet) \times \bullet \rightarrow \bullet}$, λ-expression : $\lambda \in \Sigma^{bo}_{(\bullet \Rightarrow \bullet) \rightarrow \bullet}$, the input command : $\alpha\text{-}in \in \Sigma^{bo}_{(\bullet \Rightarrow \bullet) \rightarrow \bullet}$. Note that the meaning of $\bullet \Rightarrow \bullet$ is different from $\bullet \rightarrow \bullet$. The former intuitively is a sort of function space and the latter shows an object being mapped from one domain to another under interpretations[4].

Based on the establishment of the signature for binding above, we can turn to the language (or binding terms) for the intended framework.

3 Language of Finitary Binding

For a start, we concentrate on single-sorted algebras of finitary binding operators. There are three reasons for this. (i) Philosophically, we are thinking of some kind of type-free theory; (ii) Notationally, this brings us a lot of simplicity; (iii) The last one is quite realistic, i.e. what kind of relations among sorts (or types) should be represented in the many-sorted signature is a delicated issue. Of course, if we neglect the inter-relationship among sorts, a possible extension to many-sorted cases can be worked out accordingly and is left out.

Since it is single-sorted, the indices of Σ^{bo} can be reduced to $Nat^* \times Nat$, where Nat is the set of natural numbers. Let V be a set of countable infinite (ordinary) variables

[3] Countably infinite binding is actually being considered in cylindric algebras [Henkin 71, Henkin 85]. We will come back to cylindric algebras in Section 7.

[4] The meaning of this statement will make more sense after "interpretations" being defined.

with a (linear, or complete partial) order, x, y, z, ... range over V. FV be a family of FV_m, the set of function variables with arity $m \geq 0$, f, ... range over FV. For every $m \geq 0$, FV_m and V are disjointed with each other.

Definition 1.2.1 (binding terms) : Let T be the set of terms and FT_m be the set of function terms with arity m ($m \geq 0$). They are defined as the least sets such that they are closed under the following :

1. $\frac{x \in V}{x \in T}$;
2. $\frac{f \in FV_{|\vec{t}|};\, t_j \in T}{f(\vec{t}) \in T}$,

   where $\vec{t}(j) = t_j$ ($\vec{t}(j)$ means the jth element of $\vec{t}$);
3. $\frac{t \in T;\, x_j \in V;\, x_i \neq x_j (i \neq j)}{\langle \vec{x} : t \rangle \in FT_{|\vec{x}|}}$;
4. $\frac{\sigma \in \Sigma^{bo}_{<\vec{m}, |\vec{t}|>};\, ft_i \in FT_{m_i} (1 \leq i \leq |\vec{m}|);\, t_j \in T (1 \leq j \leq |\vec{t}|)}{\sigma(\vec{ft}, \vec{t}) \in T}$,

   for $<\vec{m}, |\vec{t}|> \in Nat^* \times Nat$ and where $\vec{ft}(i) = ft_i$.

The above definition deserves some explanation : (a) by conceptual reason, we keep zero binding terms different from ordinary terms; (b) when $|\vec{t}| = 0$, the second condition above becomes $\frac{f \in FV_0}{f() \in T}$; (c) when $|\vec{x}| = 0$, the third condition becomes $\frac{t \in T}{\langle \varepsilon : t \rangle \in FT_0}$, where ε is a special symbol and denotes the empty list; (d) when $|\vec{m}| = 0$, the fourth condition becomes $\frac{\sigma \in \Sigma_{<\varepsilon, |\vec{t}|>};\, t_j \in T}{\sigma(\vec{t}) \in T}$; further if also $|\vec{t}| = 0$, it will become $\frac{\sigma \in \Sigma_{<\varepsilon, 0>}}{\sigma() \in T}$; another case is when $|\vec{t}| = 0$, it becomes $\frac{\sigma \in \Sigma_{<\vec{m}, 0>};\, ft_i \in FT_{m_i} (1 \leq i \leq |\vec{m}|)}{\sigma(\vec{ft}) \in T}$. Also, we obviously have $FV_m \cap FT_m = \emptyset$ ($m \geq 0$) and $V \subseteq T$.

Later, we will use *binding terms* to refer either to terms or to function terms.

4 Equational Examples with BOs

So far, we have set up the syntactical part of the intended framework. With this syntactical set-up, it is possible to treat many theories inside this framework.

For example, the β-conversion of λ-calculus can be expressed as

$$app(\lambda(\langle x : f(x) \rangle), y) \simeq f(y)$$

where $\simeq$ expresses "is equal to". Because of the well-known Church-Rosser property, λ-calculus can be equationally captured. For the functional programming language ML, it has an equivalence relation with λ-calculus, simply

$$app(\lambda(\langle x : f(x) \rangle), y) \simeq let(\langle x : f(x) \rangle, y).$$

This leads to equational presentation of ML inherited from the one for λ-calculus. However, we can certainly develop the equational presentation of ML on its own. For instance, $let(\langle x : h(f(x), z) \rangle, y) \simeq h(let(\langle x : f(x) \rangle, y), z)$, it expresses a property of ML, i.e. if the variable x is globally declared in h but has a local usage in f, then it can be locally declared and used in f. This property can be used to reduce the number of global variables in order to eliminate the potentiality of unnecessary side-effect.

These leads us generally to characterize binding by equations, like the equational characterization of universal algebras. The work in [Aczel 78] gives a generalized Church-Rosser property for the framework of binding and convinces us more on the idea of equational characterization of binding.

Going along this idea, we are further interested in whether there exists an inference rule system for such equational characterization, like Birkhoff's Theorems of equational logic in universal algebras. If such inference rule system does exist and if we are able to capture it precisely, then we will provide the equational characterization of binding once for all, and then it is up to the user to specify which individual system he want to work in, say in λ-calculus, ML, first-order logic or CCS, provided that he can give proper axioms for each.

However, before we can possibly answer those questions above, we have to know what is a *binding algebra*. In another word, *what is semantics for binding?* Without semantics, we can never know whether an equational characterization is sound, and needless to say of its completeness.

5 Semantics for Binding

With regards to semantics (or meanings of binding terms), We motivate the intentional aspect and the extension aspect of the meanings of binding by an example of a polynomial in number theory. A polynomial $x^2 + x + 1$ can be viewed as a "transformation" which relates the object a to the object $a^2 + a + 1$ ($a \in Nat$), and the same polynomial can be viewed as the object which stores the whole "transformation". The latter view is the *intention* of the polynomial and the former is the *extension* of it. So, the first "obvious" attempt to semantics is the extensional one.

5.1 Extensional Binding Algebras (eBAs)

This subsection is to exploit the extensional aspect and to establish the extensional binding algebras. Firstly, we will give a definition for binding algebras (see Definition 5.1.1), which is borrowed from Aczel's Frege Structure [Aczel 80] with slight modification. It follows by the definition of interpretations, see Definition 5.1.2.

Let $\mathcal{F}_m$ be a subset of function space $A^m \to A$ for $m \geq 0$. A family $\mathcal{F} = \, < A', \{\mathcal{F}_m(A) | m \in Nat\} >$ ($A' \subseteq A$[5]) is called *explicitly closed* iff

1. for each $m \geq 0$ and $a \in A'$, there is a unique function $C_{m,a} \in \mathcal{F}_m$ such that $C_{m,a}(\vec{a}) = a$ for $\vec{a} \in A^m$.

2. for each $m > 0$ and $1 \leq i \leq m$, there is a unique function, named $\pi_{m,i} \in \mathcal{F}_m$, such that $\pi_{m,i}(\vec{a}) = a_i$ for all $\vec{a} \in A^m$.

3. for $m > 0$, $k \geq 0$ given $g \in \mathcal{F}_m$ and $g_i \in \mathcal{F}_k$ ($1 \leq i \leq m$), there is a unique function $h \in \mathcal{F}_k$ such that $h(\vec{a}) = g(g_1(\vec{a}), g_2(\vec{a}), ..., g_m(\vec{a}))$ or $h = g \odot < \vec{g} >$; sometimes, it can be further abbreviated as $g(\vec{g})$.

[5]The reason for A' being a subset of A rather than just A is to accommodate binding subalgebras with a same concept of "*explicitly closedness*". However, we will not formally introduce binding subalgebras in this extended abstract. You can look it up in [Sun 89?] if you would like to.

Note that we purposely use $\odot$ as the usual compositional functional, instead of $\circ$ to avoid a potential confusion with another use of $\circ$ in the future.

Intuitively, the explicitly-closedness of a family means that the family is closed under constants, projections, and function compositions. Such a closure is not necessarily preserved by any map over the family. We, therefore, introduce the concept of uniformity over the family.

For $\sigma \in \Sigma_{<\vec{m},m>}$, an interpretation $\mathcal{A}_\sigma : \mathcal{F}_{\vec{m}} \times A^m \to A$ (or $\mathcal{F}_{m_1} \times \mathcal{F}_{m_2} \times ... \times \mathcal{F}_{m_{|\vec{m}|}} \times A^m \to A$) of σ is *uniform* over $\mathcal{F}$ if for any $k \geq 0$, given $g_i \in \mathcal{F}_{k+m_i}$, and $h_j \in \mathcal{F}_k$, there is a unique function $h \in \mathcal{F}_k$ such that for all $a_j \in A$ $h(\vec{a}) = \mathcal{A}_\sigma(\vec{g'}, \vec{b})$, where $b_j = h_j(\vec{a})$ and $g'_i(\vec{a^i}) = g_i(\vec{a}, \vec{a^i})$ for all $\vec{a^i} \in A^{m_i}$.

Now, we are able to define an extensional binding algebra as following.

Definition 5.1.1 (extensional binding algebra — eBA) : An extensional Σ^{bo}-algebra (or extensional binding algebra, eBA for short) $\mathbf{A}$ consists of

- an explicitly closed family $\mathcal{F} =< A', \{\mathcal{F}_m(A) | m \in Nat\} >$, where $A = A'$, and
- for each $\sigma \in \Sigma$, $\mathbf{A}_\sigma$ is uniform over $\mathcal{F}$, sometimes denoted as $\mathcal{A}_\sigma$, $\sigma^{\mathbf{A}}$, or even σ.

Let $\mathbf{A}$ be an eBA. A *valuation* $\vec{\rho}$ of V and FV on $\mathbf{A}$ is a family of maps ρ from V to A and ρ_k from FV_k to $\mathcal{F}_k$ for $k \geq 0$. Later, we will denote a valuation $\vec{\rho}$ as a pair of $< \rho, \varphi >$ where φ is a family of φ_k ($\varphi_k = \rho_k$, $k > 0$), simply denote it as $< \rho, \varphi >$. Sometimes, we call a valuation $\vec{\rho}$ an *environment*.

Let $< \rho, \varphi >$ be a valuation of V and FV on $\mathbf{A}$. Then for any $x \in V$ and any $a \in A$, $\rho[a/x]$ is defined by $\rho[a/x](y) =_{df} \rho(y)$ if $y \neq x$ and $\rho[a/x](y) =_{df} a$ if $y = x$. By this definition, we have that $\rho[a/x][b/x] = \rho[b/x]$ for all $a \in A$ and that $\rho[a/x][b/y] = \rho[b/y][a/x]$ if $x \neq y$, for all $a, b \in A$, where $< \rho, \varphi >$ is a valuation of V and FV on $\mathbf{A}$. Therefore, we can define $\rho[\vec{a}/\vec{x}] =_{df} \rho[a_1/x_1][a_2/x_2]...[a_{|\vec{a}|}/x_{|\vec{x}|}]$ for distincive variables $x_1, x_2, ..., x_{|\vec{x}|}$, where $|\vec{a}| = |\vec{x}|$.

Definition 5.1.2 (interpretations in an eBA) : Let $\mathbf{A}$ be an eBA, and $< \rho, \varphi >$ be a valuation of V and FV on $\mathbf{A}$. An interpretation $\mathcal{A}$ of binding terms over $\mathbf{A}$ is defined inductively as

1. $\mathcal{A}[[x]](\rho, \varphi) =_{df} \rho(x)$ for $x \in V$.
2. $\mathcal{A}[[f(\vec{t})]](\rho, \varphi) =_{df} \varphi(f)(\mathcal{A}[[\vec{t}]](\rho, \varphi))$ for $f \in FV_m$ and $t_j \in T$ $(1 \leq j \leq m)$, where $\mathcal{A}[[\vec{t}]](\rho, \varphi) = \mathcal{A}[[t_1]](\rho, \varphi), \mathcal{A}[[t_2]](\rho, \varphi), ..., \mathcal{A}[[t_{|\vec{t}|}]](\rho, \varphi)$.
3. $\mathcal{A}[[\sigma(\vec{ft}, \vec{t})]](\rho, \varphi) =_{df} \sigma^{\mathbf{A}}(\mathcal{A}[[\vec{ft}]](\rho, \varphi), \mathcal{A}[[\vec{t}]](\rho, \varphi))$ for $\sigma \in \Sigma_{<\vec{m},m>}$, $ft_i \in FT_{m_i}$ $(1 \leq i \leq |\vec{m}|)$ and $t_j \in T$ $(1 \leq j \leq m)$, where $\mathcal{A}[[\vec{ft}]](\rho, \varphi) = \mathcal{A}[[ft_1]](\rho, \varphi), \mathcal{A}[[ft_2]](\rho, \varphi), ..., \mathcal{A}[[ft_{|\vec{ft}|}]](\rho, \varphi)$.
4. $\mathcal{A}[[\ \langle \vec{x} : t \rangle\ ]](\rho, \varphi) =_{df} g$, where $g(\vec{a}) =_{df} \mathcal{A}[[t]](\rho[\vec{a}/\vec{x}], \varphi)$ and $|\vec{a}| = |\vec{x}|$.

We should be aware of that the well-definedness of above definition is not obvious. This is left out and can be looked up in [Sun 89?].

In general, a binding term p "is equal to" another binding term q, written as $p \simeq q$, if and only if (or iff) all possible interpretations of the two terms are the same, i.e. they can not be distinguished from one and another. Formally,

Definition 5.1.3 (Binding Equation and Binding Indistinguishability) : Two binding terms p, q are indistinguishable by an eBA $\mathbf{A}$ iff all possible interpretations of p and q in $\mathbf{A}$ are the same, denoted as $\mathbf{A} \models p \simeq q$. Also, $p \simeq q$ is called a binding equation (or BE).

Our aim is trying to capture the indistinguishability of eBAs syntactically. If possible, we will try to spell out the syntactical calculus (or inference rule system) which totally (or almost totally) capture the indistinguishability

As usual, we can construct the term eBA $\mathbf{T}$ from binding terms. The key point of the construction is to build an explicitly-closed family $\mathcal{F}^{\mathbf{T}}$ from binding terms as the carriers of $\mathbf{T}$ with an uniform interpretations of the operations in Σ^{bo} (i.e. bos), where $\bullet_{[p]}$ denotes the element in $\mathcal{F}^{\mathbf{T}}$ corresponding to the binding term p. A binding homomorphism (or an eBH) from an eBA $\mathbf{A}$ to another eBA $\mathbf{B}$ is a map from $\mathbf{A}$ to $\mathbf{B}$ such that it primarily preserves (a) constants, (b) projections, (c) compositions and (d) interpretations of bos. Then, we would arrive at that $\mathbf{A} \models p \simeq q$ iff $\beta(\bullet_{[p]}) = \beta(\bullet_{[q]})$ for every eBH $\beta : \mathbf{T} \rightarrow \mathbf{A}$. This shows the importance of eBHs, and implicitly demonstrates the significance of their kernels.

However, this traditional Birkhoff's approach does not work as expected. It breaks down on the general commutative property, say the commutative property of an eBH β from an eBA $\mathbf{A}$ to another eBA $\mathbf{B}$, the natural eBH ν_β from $\mathbf{A}$ to quotient eBA $\mathbf{A}/Ker(\beta)$, and the eBH $\hat{\beta}$ from $\mathbf{A}/Ker(\beta)$ to $\mathbf{B}$[6]. That is, there does not exist a commutative diagram below, see figure 5.1.1.

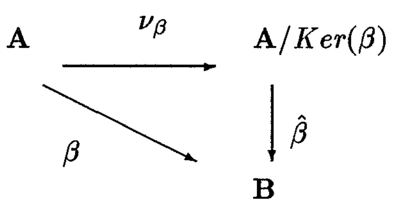

figure 5.1.1

This break-down leads us either to exploit the intentional aspect of "transformation" as explained in the beginning of Section 5 or to find a remedy for the "obvious" extensional attempt to the semantics. Actually, we find out two solutions in [Sun 89?], one for each.

For the extensional one, we discover a necessary and sufficient condition on interpretations, called admissible condition of having the commutative property. The essence of the admissible condition is extensionality, i.e. an eBH $\beta : \mathbf{A} \rightarrow \mathbf{B}$ is *admissible* iff its image of $\mathbf{A}$ is a perfect sub-eBA of $\mathbf{B}$, where an image of an eBH is always a sub-eBA and a perfect sub-eBA is a sub-eBA with extensionality. This admissibility leads us to study admissible BEs, i.e. $p \dot{\simeq} q$'s where the dot on top of $\simeq$ shows the admissibility.

[6] $Ker(\beta)$ denotes the kernel of β and $\mathbf{A}/Ker(\beta)$ is the quotient eBA of $\mathbf{A}$ over $Ker(\beta)$. Their precise definitions can be found in [Sun 89?].

Definition 5.1.4 (Admissible Equation and Admissible Indistinguishability) : Two binding terms p, q are admissibly indistinguishable by an eBA **A** iff all admissible interpretations of p and q are the same, denoted as $\mathbf{A} \models p \dot{\simeq} q$. Also, $p \dot{\simeq} q$ is called an admissible BE.

We succeed in providing a completeness for the calculus $\dot{D}_b$ of admissible BEs, but fail to characterize admissible substitutions syntactically in general. Therefore, more work need to be done in this area. One thing is certain that the admissible completeness is the best result we can have in the traditional Birkhoff's approach.

The intentional (or non-extensional) one is the subject of next subsection.

5.2 Intentional Binding Algebras (iBAs)

The intentional aspect of semantics has been introduced in the beginning of the this section (Section 5), which is essentially like treating all free variables in binding terms as bound. From the extensional binding algebras, we understand that an intentional binding algebra must contain a collection of $\vec{A}$, $pr_{k,i}(1 \leq i \leq k)$, $\circ_{m,k}(m, k \geq 0)$, and $\sigma_k^{\mathbf{A}}(k \geq 0)$ where $\vec{A}$ is a family of $A_k(k \geq 0)$, $pr_{k,i}$ is an element in A_k, $\circ_{m,k}$ is an operation of $A_m \times A_{k^m} \rightarrow A_k$ (or $A_m \times \underbrace{A_k \times A_k \times ... \times A_k}_{m\ times} \rightarrow A_k$), and $\sigma_k^{\mathbf{A}}$ is an operation of $A_{k+\vec{m}} \times A_{k^m} \rightarrow A_k$ (or $A_{k+m_1} \times A_{k+m_2} \times ... \times A_{k+m_{|\vec{m}|}} \times \underbrace{A_k \times A_k \times ... \times A_k}_{m\ times} \rightarrow A_k$). This collection is said to be a *pre-algebra*. Intuitively, it can be thought of that there is a certain carrier $\mathring{A}$ such that (a) A_k is the function space of $\mathring{A}^k \rightarrow \mathring{A}$, (b) $pr_{k,i}$ is a projection map, i.e. $pr_{k,i} : \mathring{A}^k \rightarrow \mathring{A}$, and (c) $\circ_{|\vec{g}|,k}$ is a compositional operation, which will be used in a more readable way $g \circ_{|\vec{g}|,k} <\vec{g}>$ than $\circ_{|\vec{g}|,k}(g, \vec{g})$, and even $g(\vec{g})$ leaving the choice of k to be determined by the context. We also write $\vec{g}(\vec{h})$ for the sequence $g_1(\vec{h}), g_2(\vec{h}), ..., g_{|\vec{g}|}(\vec{h})$, and write $Pr_{i,j}^k$ for $pr_{k,i}, pr_{k,i+1}, ..., pr_{k,j}$ $(1 \leq i \leq j \leq k)$.

Definition 5.2.1 (Intentional Binding Algebra — iBA): **A** is said to be an iBA iff it is a (binding) pre-algebra and it satisfies the following

1. (Associativity)

   Suppose $f \in A_k$, $\vec{g} \in A_{l^k}$ and $\vec{h} \in A_{m^l}$, then $(f(\vec{g}))(\vec{h}) = f(\vec{g}(\vec{h})) : A_m$;

2. (Left Projection)

   Suppose $\vec{f} \in A_{k^l}$, then $pr_{l,i}(\vec{f}) = f_i : A_k$;

3. (Right Projection)

   Suppose $f \in A_k$, then $f(Pr_{1,k}^k) = f$;

4. (Uniformity)

   Suppose $\sigma \in \Sigma_{<\vec{m},m>}$, $\vec{f} \in A_{k+\vec{m}}$, $\vec{g} \in A_{k^m}$ and $\vec{h} \in A_{l^k}$, then

   $(\sigma_k^{\mathbf{A}}(\vec{f}), \vec{g}))(\vec{h}) = \sigma_l^{\mathbf{A}}(\vec{f'}, \vec{g}(\vec{h})) : A_l$,

   where $f_i' = f_i(Pr_{1,m_i}^{m_i+l}, \vec{h}(Pr_{m_i+1,m_i+l}^{m_i+l}))$ for $i = 1, 2, ..., |\vec{m}|$.

The last law of the above is to preserve functional composition of an iBA. So, its presence sounds of a technical reason. However, we will comment on its profound role with binding after defining interpretations.

Unlike the case of the eBAs we can not go straight to interpretations of binding terms. The reason for this comes from how to interpretate function terms. For example, intuitively we know that α-convertible terms in λ-calculus, which are different from each other by their bound variables, have a same denotation. This should be true for binding terms in general. But we deliberatively ignore this kind of technical preparation here, and in case of any doubt, you can check this in [Sun 89?].

Definition 5.2.2 (interpretation in an iBA) : Let $\vec{\varphi}$ be a family of $\varphi_m : FV_m \to A_m$ $(m \geq 0)$. For $t \in T$ and/or given any $m \geq 0$, for $ft \in FT_m$, for any given let $\{\vec{x}\} \subseteq V$[7] and $x_i \neq x_j$ $(i \neq j)$ such that $Free(t) \cap V \subseteq \{\vec{x}\}$ and/or $Free(ft) \cap V \subseteq \{\vec{x}\}$[8], we can define an interpretation $\mathcal{A}_{\vec{x}}$ over the function environments $\vec{\varphi}$ on t and/or ft, $\mathcal{A}_{\vec{x}}[[t]]_\varphi$ and $\mathcal{A}_{\vec{x}}[[t]]_\varphi$, by

1. $\mathcal{A}_{\vec{x}}[[x]]_\varphi =_{df} pr_{|\vec{x}|,i}$, for $x = x_i \in \{\vec{x}\}$.
2. $\mathcal{A}_{\vec{x}}[[f(\vec{t})]]_\varphi =_{df} \varphi_{|\vec{t}|}(f) \circ_{|\vec{t}|,|\vec{x}|} < \mathcal{A}_{\vec{x}}[[\vec{t}]]_\varphi >$,
   where $\mathcal{A}_{\vec{x}}[[\vec{t}]]_\varphi = \mathcal{A}_{\vec{x}}[[t_1]]_\varphi, \mathcal{A}_{\vec{x}}[[t_2]]_\varphi, \ldots, \mathcal{A}_{\vec{x}}[[t_{|\vec{t}|}]]_\varphi$.
3. $\mathcal{A}_{\vec{x}}[[\sigma(\vec{ft}, \vec{t})]]_\varphi =_{df} \sigma_{|\vec{x}|}(\mathcal{A}_{\vec{x}}[[\vec{ft}]]_\varphi, \mathcal{A}_{\vec{x}}[[\vec{t}]]_\varphi)$
4. $\mathcal{A}_{\vec{x}}[[\langle \vec{y} : t \rangle]]_\varphi =_{df} \mathcal{A}_{\vec{x},\vec{z}}[[t\vec{t}[\vec{y} := \vec{z}]\ ]]_\varphi$,
   where z_j is the least $z \in V$ such that $z \notin (Free(t) - \{\vec{y}\}) \cup \{\vec{x}\} \cup \{\vec{z}\lceil_{j-1}\}$.

We should point out that the well-definedness of the above definition is not obvious either, e.g. whether $\mathcal{A}[[t]]_\varphi$ is in $A_{|\vec{x}|}$ and whether $\mathcal{A}[[ft]]_\varphi$ is in $A_{|\vec{x}|+m}$ are not clear at all. However, the well-definedness of interpretation in iBAs is left out and it is examined in [Sun 89?].

The role of binding operators is represented by the index of operations, say the index $|\vec{x}|$ of $\sigma_{|\vec{x}|}$. Such indices are not random, they are semantically inter-related with each other by the laws presented in the definition of iBAs, say the last law in Definition 5.2.1. We should also be aware of that the definition for iBAs is not usual. It is a kind of extension of the usual one. That is, each operation σ has a group of inter-related interpretations $\{\sigma_n^{\mathbf{A}} | n \in Nat\}$ rather than just one interpretation as is the case for the first-order algebras[9].

Analogous to Definition 5.1.3, under the context of iBAs we give a definition for binding equations below. The central idea of it is to bind free variables in arbitrary ways.

Definition 5.2.3 (BEs and Binding Indistinguishability) : Two binding terms p, q are indistinguishable by an iBA $\mathbf{A}$ iff all possible interpretations of p and q are the same, written as $\mathbf{A} \models p \simeq q$, i.e. for all $\{\vec{x}\} \subseteq V$ and $(Free(p) \cup Free(q)) \cap V \subseteq \{\vec{x}\}$, $\mathcal{A}_{\vec{x}}[[p]] = \mathcal{A}_{\vec{x}}[[q]]$.

On the other hand, the intuition of pre-algebras for binding can be exploited because they only involve projections $pr_{k,i}$, compositions $\circ_{l,k}$ and indexed operations σ_k. Further

[7] $\{\vec{x}\}$ means the set having and only having all the elements x_j of $\vec{x}$, i.e. $\{\vec{x}\} = \{y | y = \vec{x}(j)\}$. Hence, $\{\}$ behaves like an forgetful functor from the category of monoids to the category of sets.

[8] $Free$ is the map which provides all free variables and free function variables in a given binding term.

[9] See Chapter 2 in [Sun 89?] and many others for more details about the usual first-order case.

more, the indexing k of operations σ_k gives us a clue to bring down the order of languages, i.e. from the second-order to the first-order. This is formalized as *b-clones* in [Sun 89], which are certain kind of many-sorted algebras. In turn, it leads us to the idea of reducing the equational calculus D_b for binding to the equational calculus $\tilde{D}$ of the usual first-order many-sorted algebras. The reduction is established by discovering two translations between the two calculi, which preserve deductivities from one to the other.

6 Overview of Equational Calculi of eBAs and of iBAs

We commence our research on bos, at equational logics (or calculi) of eBAs and of iBAs Since the usual Birkhoff's approach does not work (although it is remediable), we have to seek an alternative. The one we present is an intentional approach, which reduces the equational calculus of iBAs to the one of many-sorted algebras. The remedy for the Birkhoff's approach is also discovered. Nevertheless, the whole work primarily composes of two parts.

Part 1 is to follow the usual Birkhoff's approach under the binding circumstance and to achieve a similar result as the one in usual (many-sorted) first-order algebras [Sun 89]. Due to the fact that function spaces are carriers in an eBA, we have to define two concepts for binding subalgebras, say *sub-eBAs* and *perfect sub-eBAs*, instead of one as is the case of the usual first-order (many-sorted) algebras. The difference between these two is extensionality. The extensionality also plays a role in the binding congruences (or eBCs) and the kernels of eBHs. Most significantly, it is the essence of the admissibility where the "admissible" concept is very important because the diagram, see figure 5.1.1, commutes iff the eBH β is admissible. Because it is necessary and sufficient, we claim that *Admissible Completeness* for BEs (or Completeness for admissible BEs) of the deduction system, denoted as $\dot{D}_b$, is the best result we can ever have in Birkhoff's approach. Comparing with *logical relations* in literature, the admissible condition is weaker than logical relations in the sense that an eBH must be admissible if it is logical. For a reference to logical relations, see [Plotkin 80, Statman 82].

Nevertheless, about this deduction system $\dot{D}_b$, it is almost the same as D_b except the deduction rule related to substitutions. In this extensional approach (or Birkhoff's one), the substitution rule is only applicable if the substitution map is an *admissible substitution map*[10]. But what is the syntactical characterization of an admissible substitution map is yet to be investigated.

Part 2 is to capture the indistinguishability of iBAs. The central idea is to relate this indistinguishability with the corresponding indistinguishability of the usual first-order many-sorted algebras. The connections between them are established by discovering two translations between binding terms and the terms of b-clones, and by verifying that they preserve the indistinguishabilities and the deductivities between iBAs and b-clones. Hence, the soundness and completeness of the deduction system of iBAs follow accordingly from their counter-parts in the usual first-order many-sorted algebras.

We present the calculus D_b and its completeness as follows. Let Γ_b be a collection of Σ^{bo}-equations i.e. each element in the collection Γ_b is of form $p \simeq q$ such that either

[10] An admissible substitution means that it preserves extensionality if it is considered as a binding homomorphism. However, the exact definition is left out and can be found in [Sun 89?].

$p, q \in T$ or $p, q \in FT_k$ for some $k \geq 0$. We say that a substitution map $\vec{\varrho}$ is a *functional substitution* map iff for all $m \geq 0, \forall f \in FV_m.\vec{\varrho}[f] \cap V = \emptyset \wedge \forall x \in V.\vec{\varrho}(x) \in V$. Then, the equational logic (or calculus) of iBAs is given below.

Theorem 6.1 (Equational Logic D_b of iBAs) : The following calculus D_b of iBAs is sound and complete.

Given any collection Γ_b of binding equations, D_b contains, as usual, (a) the identity rule, (b) the reflectivity rule, (c) the symmetricity rule, (d) the transitivity rule, (e) the axiom-introduction rule, and (f) Modus Ponens. Also it has the extra rules in the following.

1. (α-conversion) $\frac{t \in T; \{\vec{x}\} \subset V; \{\vec{y}\} \subset V}{\vdash \langle \vec{x} : t \rangle \simeq \langle \vec{y} : t\vec{\imath}[\vec{x} := \vec{y}] \rangle}$,
   where $y_j \in V$ and $y_j \notin (Free(t) - \{\vec{x}\}) \cup \{\vec{y} \lceil_{j-1}\}$;

2. (ξ-rule) $\frac{\Gamma_b \vdash t \simeq u}{\Gamma_b \vdash \langle \vec{x} : t \rangle \simeq \langle \vec{x} : u \rangle}$,
   where $t, u \in T$ and $x_j \in V$ $(1 \leq j \leq |\vec{x}|)$ and $x_i \neq x_j$ $(i \neq j)$;

3. (ξ^{-1}-rule) $\frac{\Gamma_b \vdash \langle \vec{y} : t \rangle \simeq \langle \vec{z} : u \rangle}{\Gamma_b \vdash t\vec{\imath}[\vec{y} := \vec{x}] \simeq u\vec{\imath}[\vec{z} := \vec{x}]}$,
   where $\{\vec{x}\} \cap Free(\langle \vec{y} : t \rangle) = \emptyset$ and $\{\vec{x}\} \cap Free(\langle \vec{z} : u \rangle) = \emptyset$;

4. (functional substitution) $\frac{\Gamma_b \vdash p \simeq q}{\Gamma_b \vdash p\vec{\varrho} \simeq q\vec{\varrho}}$,
   where $\vec{\varrho}$ is a functional substitution map;

5. (function composition) $\frac{f \in FV_{|\vec{t}|}; \Gamma_b \vdash t_j \simeq u_j \ (1 \leq j \leq |\vec{t}| = |\vec{u}|)}{\Gamma_b \vdash f(\vec{t}) \simeq f(\vec{u})}$,
   where $t_j, u_j \in T$ $(1 \leq j \leq |\vec{t}|)$;

6. (functional composition) $\frac{\Gamma_b \vdash ft_i \simeq fu_i, t_j \simeq u_j \ (1 \leq i \leq |\vec{m}|, 1 \leq j \leq m)}{\Gamma_b \vdash \sigma(\vec{ft}, \vec{t}) \simeq \sigma(\vec{fu}, \vec{u})}$,
   where $\sigma \in \Sigma_{<\vec{m}, m>}$, $ft_i, fu_i \in FT_{m_i}$ $(1 \leq i \leq |\vec{m}| = |\vec{ft}| = |\vec{fu}|)$ and $t_j, u_j \in T$ $(1 \leq j \leq m)$.

One interesting thing about the completeness of D_b is that ξ^{-1}-rule is not needed when the given Γ_b does not involve function terms. Nevertheless, the two rules of ξ and ξ^{-1} show an important phenomenon in programming, i.e. when we declare a procedure, we break off the binding and write down the body of the procedure; and when we call the procedure, we actually put back the binding and use it as an object.

In contrast to extensional approach (or Birkhoff's approach), although the substitution rule is restricted in intentional approach, i.e. the images of function part of a substitution map (being used in substitution rule) must be closed terms, the restriction is totally captured syntactically.

The applications of the obtained calculi are left out. You can find some of them, say equationalization of first-order logic and equational characterization of finite CCS with data-dependency, in [Sun 89?].

Details of the work in this abstract and others will be collected in my forthcoming thesis [Sun 89?]

7 Review with Related Work

As far as I know, bos have not been well studied in an algebraic form regardless of whether the study is systematical or not. The work relating to binding in literature is mostly restricted to *variable binding term operators* (vbtos). In general, the signature Σ^{vbto} of vbtos is indexed by elements in $\bigcup_{m\in Nat}(S^m \times S)^* \times S$ (or $\bigcup_{m\in Nat}(S^m \Rightarrow S)^* \rightarrow S$) where $S = Sort(\Sigma^{vbto})$. That is, bos are technically more primitive than vbtos and $\Sigma^{vbto} \subsetneq \Sigma^{bo}$ strictly. For example, $\forall, \lambda$ and α-*in* are vbtos and *let* is not a vbto. Therefore, the work in [Abar 86, Hacther 82, Costa 80, Corcoran 72], dealing only with vbtos, is a kind of special cases in our framework.

Closely to binding, there is work on cylindric algebras, monadic algebras and polyadic algebras, see [Halmos 63, Henkin 71, Henkin 85]. this kind of work can simply be put in the spirit of taking algebras out of the first-order logic. so, the essential part is the treatment of the quantifiers. In turn, the feature of the binding being dealt with is more or less the kind of vbto's binding.

Technically, cylindric algebras are a special kind of binding algebras, i.e. infinite binding algebras. To see the relation between cylindric algebras and binding algebras more closely, we observe that all terms appeared in cylindric algebras are closed function terms in binding algebras. they are all bound by $\vec{V}$, i.e. the list of all variables in V queued in a complete order among them. since $|V|$ is assumed to be infinite ω, cylindric algebras are infinite binding algebras, say ω-binding. the dimension of cylindric algebras is corresponding to the cardinal number $|V|$. The kth cylindrifier c_k over a term t, say $c_k(t)$, behaves like an existential quantifier, say $\exists_k^{cyl}$, over the corresponding function term $\langle \vec{V} : t\rangle$ like $\exists_k^{cyl}(\langle \vec{V} : t\rangle)$. The index k of $\exists_k^{cyl}$ has the meaning to bind the kth variable in the order among variables. Its behaviour can also be viewed as a cylinder along the kth axis in the ω-dimension geometry. (k, j)-diagonal element $d_{k,j}$ can be expressed by a binding quasi-dependent equation[11] $\circ^{cyl}(\langle \vec{V} : x_k\rangle, ft) \simeq \circ^{cyl}(\langle \vec{V} : x_j\rangle, ft) \hookrightarrow ft \simeq d_{k,j}$, where $\circ^{cyl}$ is composition functional, $ft = \langle \vec{V} : t\rangle$, $\langle \vec{V} : x_k\rangle$ and $\langle \vec{V} : x_j\rangle$ are kth and jth projection functions respectively.

Monadic Algebras and Polyadic Algebras, see [Halmos 54, Halmos 62, Henkin 85], are alternatives to cylindric algebras. Monadic algebras are special case of Polyadic algebras. Both of them are a special kind of binding algebras as well. To illustrate our point, let us look at monadic algebras first. A monadic algebra is a boolean algebra extended to include one existential quantifier $\exists^{mon}$, where $\exists^{mon}(\langle x : t\rangle)$ can be viewed as short for $\langle x : \exists_1(\langle x : t\rangle)\rangle$ and $\exists_1$ is the usual unary existential quantifier and the index 1 is to emphasize the arity. For polyadic algebras, $\exists_{\vec{x}}$ ($\{\vec{x}\} \subseteq V$) can be thought as $\langle \vec{x} : \exists_{|\vec{x}|}(\langle \vec{x} : t\rangle)\rangle$. where $\exists_{|\vec{x}|}$ is the usual $|\vec{x}|$-ary existential quantifier.

The relationship between cylindric algebras and polyadic algebras can be found in [Henkin 85]. There are other ways of algebraize the first-order logic, say projective algebras (a special case of cylindric algebras) [Bednarek 78, Henkin 85] and relation algebras (another special case of cylindric algebras) [Jonsson 82, Henkin 85]. I have no attempt to claim that the references mentioned are exhausted. The reason of choosing them here is that they are more likely available from a library and the other references

[11] A *"quasi-dependent"* equation $\bullet \hookrightarrow \bullet$ is commonly callled as a *"conditional"* equation or a *"quasi"*-equation. The reason for choosing such a terminology can be found in [Sun 89?]. So, a binding quasi-dependent equation is a quasi-dependent equation involving bos.

can find out through their references.

The significant departing point of our work, especially the work of Chapter 8 in [Sun 89?], is to reduce the first-order logic to algebras, rather than another way around. The final output of our work on the first-order logic is the equationalization of it.

The work in [Barnes 75] and the work in [Reynolds 89] are alternatives to ours in the field of the first-order logic, although the approach of [Barnes 75] is algebraic and the one of [Reynolds 89] is category-oriented, but their attempts ignore the existence of function terms and present them as ordinary terms with certain precautions in handling them. For instance, their treatment of the existential quantifier $\exists$ is $\exists(\langle x : \_\rangle)$, where $\_$ is an arbitrary meta-hole to be filled in, i.e. the variable x in t will be bound if the hole $\_$ is filled with t. Therefore, their approaches are similar to the intentional approach, as you may feel.

— — —

Our framework for binding is very powerful as demonstrate above although the language for bos is of restricted second-order. People might wonder whether it is over powerful since none of mentioned work has treated binding in a so general form. Therefore, it is worthwhile to point out a potential usage of the extra power in the framework presented here.

Since a framework of vbtos (or the like) can not treat first-order objects and ordinary objects at the same time, the vbtos' framework suffers a set-back on dealing with natural languages. That is, it may not be a proper framework for natural languages. As we know, dealing with natural languages is not a easy task. To show the difficulty, we just need to remind you of many semantical paradoxes, say the liar paradox. Comparing with the vbtos' framework, the framework for bos presented here does not have this disadvantage. This aspect of dealing with first-order objects and ordinary objects at the same is being considered as a very crucial point in providing a semantical framework for natural languages, see Barwise argued in his situation theory [Barwise 84] for instance. Therefore, I am quite optimistic in predicting that the binding framework presented here would have a bright future related to natural languages. An easy exercise to start with might be to deal with general quantifiers in natural languages, see [Fenstad 85, Westerstahl 86] for a reference to general quantifiers.

8 Future Development

I have to admit that the work presented here is just an opening for a wide area of research. To justify this, I foresee some of future development briefly.

Equational theories are quite powerful, e.g. λ-calculus and Combinatory Logic [Curry 58] can be presented equationally. O'Donnell and others have shown that many interesting functions can be naturally be described by equations [O'Donnell 77, Johnson 83, O'Donnell 85, Sun 86]. More strikingly, Matijasevic has shown that every recursively enumerable predicate is *diophantine* [Matijasevic 71] (this gave a negative answer to Hilbert's 10th problem, see [Davis 76]), which puts equational theories outstanding.

However, pure equational theories are rather weak since their languages only contain assertions of the form $p \simeq q$. For instance, three well-known natural number systems

(Presburger Arithmetic, Peano Arithmetic, Robison Arithmetic) are not totally presented in equational forms. Naturally, one might wish to unite first-order logic with equational theories. In mechanical theorem proving, such a wish appeared and being taken into action first in [Plotkin 72], as far as I notice. Such a "union" can not be taken for granted without questioning how (first-order) logic can semantically sound to unite with equational theories, since quantifiers, say the existential quantifier, are not first-order objects. It is a quite common practice in introducing higher-order types (or sorts) instead of raising the order of languages (or syntax, signature) to resolve the problem of non-first-order objects. The new framework of binding suggested in this abstract can be regarded to go (or to lead) the other way around. We can apply the work of [Sun 89] to here and get a complete equationalization of first-order logic, see [Sun 89?] for more details. This is quite original. Nevertheless, we can still introduce higher-order types into the binding framework. This subject is remained to be investigated.

Another area might be fruitful is to connect the work here with category theory [Maclane 71, Arbib 75]. This is far from trivial, since the usual category is in nature of first-order, see [Manes 76] for the connection between the usual first-order case and category theory. It is very nice to see how eBAs and iBAs present in categorical framework.

Overall, from a theoretical computer science point of view, we would like to know what is the relationship between transition systems, say operational semantics (see [Plotkin 81] for a reference), and equational systems, say denotational semantics (see [Stoy 78] for a reference). The core of this relationship is the full abstraction, which was investigated first by Plotkin in [Plotkin 77], also see [Milner 77, Stoughton 86, Mulmuley 85]. Nevertheless, all of the mentioned cases are not general enough from the point of view of the framework for bos. Hence, a generalized full abstraction in the framework for bos deserves our attention.

9 Acknowledgement

I would like to thank G. Plotkin who brought binding into my attention and had since encouraged me to work on it. He also provides me many precious insight, guidance and supervision. I owe many thanks to him. Also, I would like to mention that LFCS in University of Edinburgh provides an ideal working environment of doing research, and I get many stimulations from the environment. It is hard for me to single out every benefit obtained and I have to reserve the right of pointing out each of them.

10 References

Abar 86 : C. A. A. P. Abar and M. Yamashita, "Remarks on Variable Binding Term Operators", Polish Acad. Sci. Inst. Philos. Social. Bull. SECT LOGIC, vol.15, No.4, pp 145-151, 1986.

Aczel 78 : P. Aczel, "A General Church-Rosser Theorem", unpublished note, Sept. 1978.

Aczel 80 : P. Aczel, "Frege Structure and Notations of Propositions, Truth and Set", The Kleene Symposium, Studies in Logic Series, 1980.

Arbib 75 : M. A. Arbib and E. G. Manes, "Arrows, Structures and Functors : the categorical imperatives", Academic Press, 1975.

Barendregt 84 : H.P. Barendregt, "The Lambda Calculus, its Syntax and Semantics", revised ed., Studeis in Logic Series, 1984.

Barnes 75 : D. W. Barnes and J. M. Mack, "An Algebraic Introduction to Mathematical Logic", GTM vol.22, Springer-Verlag, 1975.

Barwise 84 : J. Barwise, "The Situation in Logic, I, II, III", CSLI Report No. 2, 21 and 26, CSLI, Stanford University, 1984.

Bednarek 78 : A. Bednarek and S. Ulam, "Projective Algebra and the calculus of relation", J. Symb. Logic, vol.43, 1978.

Church 41 : A. Church, "The Calculi of Lambda-conversion", Annals of Mathematics Studies No.6, 1941.

Boolos 80 : G. S. Boolos and R. C. Jeffrey, "Computability and Logic", Cambridge University Press, 2nd ed. 1980.

Breazu-Tannen 88 : V. Breazu-Tannen, "Combining Algebra and Higher-order Types", Dept. of Computer and Information Science, School of Engineering and Applied Science, University of Pennsylvania, MS-CIS-88-21, LINC LAB 107, March 1988.

Corcoran 72 : J. Corcoran, W. Hatcher and J. Herring, "Variable Binding Term Operators", Zeitschr. f. math. Logik und Grundlagen d. Math., vol. 18, pp 177 - 182, 1972.

Costa 80 : N. C. A. da Costa, "A Model-theoretical Approach to Variable Binding Operators", Mathematical Logic in Latin American, Studies in Logic vol. , North-Holland, 1980.

Curry 58 : H. B. Curry and R. Feys, "Combinatory Logic, vol. I", Studies in Logic vol. , North-Holland 1958.

Dalen 83 : D. van Dalen, "Logic and Structure", 2nd ed., Springer-Verlage, 1983.

Davis 76 : M. Davis, Y. Matijasevic and J. Robinson, "Hilbert's 10th Problem, Diophantine Equations : positive aspects of a Negative Solution", AMS Proc. Symp. in Pure Math. vol.28, 1976.

Fenstad 85 : J. E. Fenstad, P. Halvorsen, T. Langholm and J. von Benthem, "Equations, Schemata and Situations : A Framework for Linguistic Semantics", CSLI, Standford University, CSLI Report No.29, 1985.

Gordon 79 : M. J. C. Gordon, R. Milner and C. Wadsworth, "Edinburgh LCF, a mechanical logic of computation", LNCS vol.78, 1979.

Halmos 54 : P. R. Halmos, "Polyadic Boolean Algebras", Proc. of National Academy of Sciences, vol.40, 1954.

Halmos 62 : P. R. Halmos, "Algebraic Logic", Chelsea, New York, 1962.

Hatcher 82 : W. Hatcher, "The Logical Foundations of Mathematics", 1982.

Henkin 71 : L. Henkin, J. D. Monk and A. Tarski, "Cylindric Algebras, Part I", Studies in Logic series, vol.64, 1971.

Henkin 85 : L. Henkin, J. D. Monk and A. Tarski, "Cylindric Algebras, Part II", Studies in Logic series, vol.115, 1985.

Hoare 78 : C.A.R. Hoare, "Communicating Sequential Processes", CACM vol.28, 1978.

Hu 82 : Hu Shi-hua and Lu Zong-wan, "Introduction to Mathematical Logic", vol.1 and vol.2, series of introductions to modern mathematics, Academic Press, Beijing 1982 (in Chinese).

MacLane 71 : S. MacLane, "Categories for the Working Mathematician", GTM vol.5, 1971.

Manès 76 : E. G. Manes, "Algebraic Theories", GTM vol.26, 1976.

Matijasevic 71 : Y. V. Matijasevic, "Diophantine Representation of Recursively Enumerable Predicates", Proc. of the 2nd Scandinavian Logic Symposium, Studies in Logic vol.63, 1971.

Milner 77 : R. Milner, "Fully Abstract Models of Typed λ-calculus", TCS vol.4, pp 1-22, 1977.

Milner 80 : R. Milner, "A Calculus of Communicating Systems", LNCS vol.92, 1980.

Milner 89 : R. Milner, "Communication and Concurrency", Prentice-Hall, 1989.

Mulmuley 85 : K. Mulmuley, "Fully Abstraction and Semantics Equivalence", Ph. D. Thesis, Computer Science Dept., Carnegie-Mellon Univ., 1985.

O'Donnell 77 : M. J. O'Donnell, "Computing in Systems Described by Equations", LNCS vol. 58, 1977.

O'Donnell 85 : M. J. O'Donnell, "Equational Logic as a Programming Language", The MIT Press, 1985.

Plotkin 72 : G.D. Plotkin, "Building-in Equational Theories", Machine Intelligence No.7, pp 73-90, Edinburgh University Press, 1972.

Plotkin 77 : G. D. Plotkin, "LCF considered as a Programming Language", TCS vol.5, 1977.

Plotkin 80 : G. D. Plotkin, "Lambda Definability in the Full Type Hiearachy", in "To H. B. Curry : Essays on Combinatory Logic, Lambda Calculus and Formalism", Academic Press, 1980.

Plotkin 81 : G. D. Plotkin, "A Structural Approach to Operational Semantics", Computer Science Dept., Aarhus Univ., DAIMI FN-19, Sept. 1981.

Reynolds 89 : J. C. Reynolds, "Variable Binding", chapter 4 in "Semantics as a Design Tool" (advanced topics in theory), January 1989.

Rydeheard 85 : D. E. Rydeheard and R. M. Burstall, "Monads and Theories : a survey for computation", Algebraic Methods in Semantics, ed. M. Nivat and J. C. Reynolds, Cambridge University Press, 1985.

Statman 82 : R. Statman, "Logical Relations and the Typed Lambda Calculus", Information and Control, 1982.

Stoy 77 : J. Stoy, "Denotational Semantics : the Scot-Strachey Approach to Programming Language Theory", The MIT Press, 1977.

Sun 86 : Y. Sun, "Completeness of Kleene Ternary Logic for VLSI at gate-level" unpublished note, 1986.

Sun 89 : Y. Sun, "Equations, Dependent Equations and Quasi-dependent Equations — on Their Unification", to appear in LFCS (green) report, 1989.

Sun 89? : Y. Sun, "Indistinguishabilities of Binding — their calculi and applications", Dept. of Computer Science, University of Edinburgh, Ph.D thesis (in preparation), 1989.

Westerstahl 86 : D. Westerstahl, "Quantifiers in Formal and Natural Languages", CSLI Report No. 55, CSLI, Stanford University, 1986.

Zhou 81 : Chaochen Zhou and C. A. R. Hoare, "Partial Correctness of Communicating Sequential Processes", the 2nd International Conference on Distributed Computing Systems, Paris, April 1981.