Algebraic Meta-Theories and Synthesis of Equational Logics

Research Programme

Marcelo Fiore
University of Cambridge Computer Laboratory

Synopsis

Fiore and Hur [18] recently introduced a novel methodology—henceforth referred to as ***Sol***—for the ***S***ynthesis ***o***f equational and rewriting ***l***ogics from mathematical models.

In [18], ***Sol*** was successfully applied to rationally reconstruct the traditional equational logic for universal algebra of Birkhoff [3] and its multi-sorted version [26], and also to synthesise a new version of the *Nominal Algebra* of Gabbay and Mathijssen [41] and the *Nominal Equational Logic* of Clouston and Pitts [8] for reasoning about languages with name-binding operators.

Based on these case studies and further preliminary investigations, we contend that ***Sol*** can make an impact in the problem of engineering logics for modern computational languages. For example, our proposed research on second-order equational logic will provide foundations for designing a second-order extension of the Maude system [37], a first-order semantic and logical framework used in formal software engineering for specification and programming.

Our research strategy can be visualised as follows:

$$\overset{\text{(I)}}{\rightsquigarrow}\ \begin{array}{c}\textit{Algebraic}\\ \textit{Meta-theory}\end{array} \rightsquigarrow \overset{\text{(II)}}{\boxed{\textbf{\textit{Sol}}}} \rightsquigarrow \begin{array}{c}\textit{Equational}\\ \textit{Logic}\end{array}$$

and consists of two main activities:

- (I) the development of mathematical models of computational languages in the form of algebraic meta-theories, and
- (II) the systematic use of these to synthesise formal deduction systems for equational reasoning according to ***Sol***.

In this context, two further points are worth noticing.

- (*i*) The algebraic meta-theories to be developed, even though ultimately intended to serve as input to ***Sol***, are of interest in their own right and will be thus investigated; *e.g.*, to devise new syntactic structures for formal language specification, to provide new notions of definition by structural recursion, to derive new induction principles, to feedback into programming and meta-programming languages, to induce notions of theory translation, to build algebraic models for higher-order rewriting.
- (*ii*) The equational logics output by ***Sol*** are guaranteed to be sound with respect to a canonical model theory induced by the input algebraic meta-theory. ***Sol*** also provides a framework for analysing completeness, which typically leads to canonical equational logics.

In order to make substantial progress in the area, the proposal targets a host of key features of languages used for formalising, specifying, programming, and reasoning about computation. These features are: *binders*, *metavariables*, *linearity*, *sharing*, *graphical structure*, *type dependency*, *substitution*. Our research programme is planned in a stepwise fashion so that the various feature combinations can be treated modularly. Expected outcomes of our work include algebraic meta-theories for languages with

- (I.1) variable binding and metavariables (with and without linearity constraints);
- (I.2) type dependency (with and without variable binding and metavariables);
- (I.3) sharing (with and without variable binding and metavariables)

together with corresponding

- (II.1) sound and complete logics for equational reasoning, and
- (II.2) syntactic and semantic meta-theories of translations between equational theories.

All in all, the mathematical theory will provide new algebraic foundations for sophisticated computational structures; the equational logics will serve as the basis of computer-assisted systems for formal methods.

1. Background and motivation

We introduce the general scientific background for the proposal and provide specific motivation for our programme. We briefly recall the role and importance of meta-theories in computer science, and argue in favour of founding their development on mathematical models. Readers taking such views for granted can directly move on to Section 2, where the core of the research proposal is presented.

Background. Challenged by both technological and theoretical developments, our view of computation is still evolving. Here “computation” should be understood broadly, to include both classical notions (such as reduction, feasibility, concurrency, communication, interaction, probability, randomisation) and non-classical ones (quantum computation, bio information, *etc.*). The *concrete theories* for specific computational phenomena that are emerging encompass three aspects: (*1*) the study of specification and programming languages for describing computations; (*2*) mathematical structures for modelling computations; and (*3*) logics for reasoning about properties of computations. The interaction between these three strands is often as complex as it is fruitful, with each strand informing and enriching the other two. To make sense of this complexity, and also to compare and/or relate different concrete theories, *meta-theories* have been built. These meta-theories are used for the study, formalisation, specification, prototyping, and testing of concrete theories.

It is this distinction between *concrete theories* and *meta-theories* that plays a prominent role in this proposal, where we focus on the investigation of meta-theories to provide systems that better support the formalisation of concrete theories. The dichotomy between concrete theories and meta-theories is not new. Mathematicians, for instance, are both concerned with the development of the concrete theories of groups, rings, *etc.*, as well as with the universal algebra of these structures. However, the situation in computer science seems to be of a richer nature: The development of concrete theories typically stimulates that of meta-theories, which themselves may turn into concrete theories that feedback into the development. This is best appreciated with an example: The functional core of the programming language ML [45] is a concrete theory of typed functional computation that arose as the meta-language underlying the LCF proof-checking system [28], which in turn arose as a meta-theory for a Logic for Computable Functions [54] based on the mathematical theory of computation provided by Domain Theory [53].

Our research programme is part of this general scientific enterprise. Indeed, we aim to develop algebraic meta-theories for certain ubiquitous computational structures and to synthesise equational logics to reason about them.

Motivation. The meta-theories to be developed aim at syntactic structures for describing languages with the following features: *variable binding*, *meta-variables*, *linearity*, *sharing*, *graphical representation*, *type dependency*, *substitution*. All of these play a central role across areas such as programming-language theory, computer-assisted reasoning, formal software engineering, rewriting theory, type theory, *etc.*; and it is with these applications in mind that our research will be conducted.

Our ambition is both to build and to experiment with a mathematical theory for the design, specification, implementation, and study of formal systems—specifically in the form of equational logics—as needed for reasoning about languages with the aforementioned features. As advocated and pioneered in computer science by Scott [54], Plotkin [49] and others, we argue that such a mathematical theory should account for both syntactic (*i.e.* language theoretic) and semantic (*i.e.* model theoretic) aspects. However, we go a step further and propose an algebraic framework in which both the language syntax and the deduction system are derived. This becomes important as we move from the traditional equational logic of universal algebra [3, 26] to less familiar settings, such as dependent type theory [10, 46, 6, 11] and beyond.

Our approach is novel and contrasts with much work on type theory and logical frameworks, which is mainly developed and validated on a proof-theoretic (*i.e.* syntactic) basis. In this respect, we believe that the lack of a model theory hinders the foundations and applicability of the subject. For instance, syntactic features like those at which we aim here are often not explained but rather encoded, typically by means of analogous features at the meta-level. On the other hand, all of the following can be justified on model-theoretic grounds.

- The implementation of variable binding via de Bruijn indices or levels [20].
- The availability of definitions by structural recursion and the derivability of induction proof principles [13, 48].
- The specification, derivation, and correctness-proof of notions of substitution [20, 15, 17].
- The validity of equational reasoning [18].
- The notion of theory translation [25, 34].
- The combination of equational theories [22, 31, 30].

Within our research programme, such model theories become tools that provide principles for guiding and aiding the design and implementation of meta-theories.

2. Methodology and programme

Philosophy. A distinctive aspect of our approach is the commitment to investigate and develop meta-theories within an *algebraic framework*. For this to be possible, one has to use the mathematical theory of categories [36, 4] to allow for a sufficiently general notion of "algebraic" that still supports equational reasoning in the context of the various language features under consideration. The commitment to the algebraic framework is a direct consequence of adopting the well-established view that the essential syntactic structure of a phrase—its *abstract syntax*—should reflect semantic import, viewed in the light of the following first main thesis for the project:

The mathematical structure of abstract syntax is algebraic.

In fact, we will be adhering to, but significantly extending, the *initial-algebra semantics framework* of the ADJ group [27]. In this framework, (*i*) models for syntax are canonically given as algebras, and (*ii*) syntax is understood abstractly (*viz.* independently of any specific representation) as an initial model equipped with a homomorphic (*i.e.* compositional) interpretation in all models. It follows from (*i*) that the notion of congruence, on which compositional reasoning is founded, is canonically given; and it follows from (*ii*) that syntax supports definitions by *structural recursion*—a generalised form of primitive recursion—with an associated *induction principle* [5, 35]. Thus the algebraic framework accounts for both syntactic and semantic aspects of languages.

With the above set-up in place, a crucial novelty of our research proposal is that of *synthesising equational logics* as prescribed by a mathematical methodology. The second main thesis driving our approach is:

There is a universal abstract deduction system for equational reasoning underlying all concrete equational logics for algebraic structures.

This is analogous to Chomsky's thesis [7] postulating a universal grammar that abstracts all concrete natural-language grammars.

Consequently, our development of equational logics will be profoundly based on model theories. This is in direct line with the development of the equational logic of universal algebra by Birkhoff [3], who aimed at a sound and complete deduction system for reasoning about equality in traditional algebra. Indeed, here we will be pursuing the same kind of programme for the modern algebra needed in current applications to computer science.

2.1. Framework and methodology

Algebraic meta-theories. As we have already mentioned, we rely on the mathematical theory of categories to provide a notion of algebraic structure that is general enough to encompass the sophisticated language features that we are interested in. We briefly recall how categorical models are used to achieve this and mention some recent models by the authors and collaborators that play a central role in the consideration of one of these features: variable binding.

The main conceptual step is to regard a category as a mathematical universe of discourse within which algebraic structure is considered. For the usual category of sets and functions, this leads to the traditional notion of algebraic structure of universal algebra. However, by suitably varying the universe of discourse, broader notions of algebraic structure may be obtained. The reason is that more sophisticated universes of discourse allow for the consideration of algebras A with operations $A^a \to A$ of arity a for suitable objects a other than natural numbers. For instance, in the model $\boldsymbol{\mathcal{F}}$ of Fiore, Plotkin and Turi [20] there is an object of variables V for which operations $A^{\mathrm{V}} \to A$ of arity V provide interpretations of variable-binding operators modulo α-equivalence. Two related universes of discourse are the model of nominal sets $\boldsymbol{\mathcal{N}}$ of Gabbay and Pitts [42] that also accommodates variable-binding arities, but by means of supporting an intrinsic notion of freshness, and the model $\boldsymbol{\mathcal{R}}$ of Fiore and Staton [21, Definition 2.4] which somehow lays in between the models $\boldsymbol{\mathcal{F}}$ and $\boldsymbol{\mathcal{N}}$.

It is by fine-tuning universes of discourse that progress will be made to accommodate further syntactic features.

Synthesis of equational logics. Our methodology ***Sol*** for the synthesis of equational logics from algebraic meta-theories such as the above consists of four phases.

[***Sol*1**] *Select a category $\mathscr{S}$ as universe of discourse and consider within it a syntactic notion of signature such that every signature Σ gives rise to a monad $\mathcal{S}$ on $\mathscr{S}$.*

The category $\mathscr{S}$ provides the ambient mathematical universe for the model theory; the monad $\mathcal{S}$ embodies algebraic structure (with notions of variable and substitution).

As mentioned above, the universe of discourse should be carefully chosen to consist of mathematical objects with enough internal structure to allow for the algebraic realisation of the syntactic constructs that are being modelled.

[***Sol*2**] *Select a class of coarity-arity pairs (c, a) of objects in the universe of discourse and give a description of the Kleisli maps $c \to \mathcal{S}(a)$ as syntactic terms.*

In this context, Kleisli maps are regarded as generalised terms. The need for terms with both arities and coarities is well-established in categorical algebra, but somehow new in applications to computer science. In the setting of the algebraic models for variable binding mentioned above, the role of coarities and arities respectively corresponds to that of variables and metavariables [18, 17].

A syntactic notion of equational presentation—as a set of pairs of syntactic terms—is thus obtained. For these, ***Sol*** provides a canonical algebraic model theory for the validity of equational assertions. Models of equational presentations are Eilenberg-Moore algebras satisfying the equations.

From the model theory, an equational logic for reasoning about the equality of Kleisli maps has been extracted [18, Section 3]. This logic is sound by construction.

[***Sol*3**] *Synthesise a deduction system for equational reasoning on syntactic terms with rules arising as syntactic counterparts of the rules for the logic of Kleisli maps.*

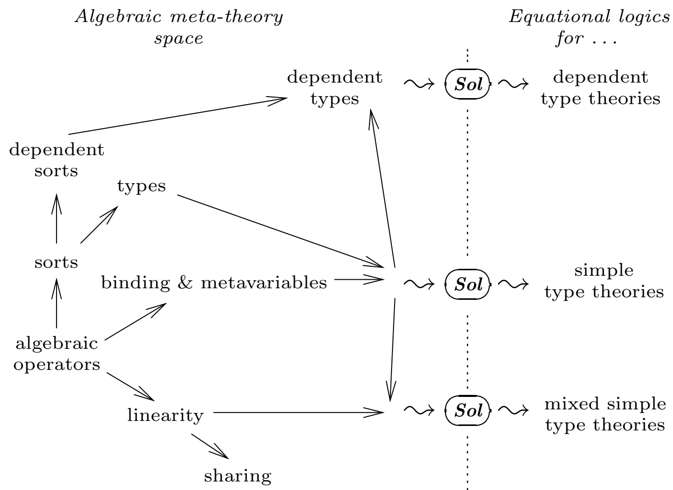
Again by construction, the synthesised equational logic will be guaranteed to be sound. ***Sol*** also provides a framework for completeness based on an inductive construction of free algebras [19].

[***Sol*4**] *Analyse the construction of free algebras so as either to establish the completeness of the synthesised equational logic, or to get insight into how to extend it to make it complete.*

The resulting equational logic is thus synthesised from an algebraic meta-theory by means of first principles.

2.2. Research programme

Our proposed investigations in the context of the above philosophy, framework and methodology are put forward. As a prologue, the diagram below gives a screenshot of the research to be described. The left hand side of this diagram, presents the space of algebraic meta-theories to be explored; the right hand side, singles out some possibilities for equational logics resulting from the application of ***Sol***.



Overall, thus, we propose a programme to develop a coherent mathematical theory for prominent and ubiquitous features of modern computational languages and thereby to synthesise canonical equational logics for reasoning about them.

The research to be undertaken on the various regions of the algebraic meta-theory space is now expounded upon.

1. Variable binding and metavariables

Second-order syntax. By *second-order* syntactic structures we understand languages with *variables*, *variable binding*, and *metavariables*; the latter being essentially contexts (or holes) with term parameters. The terminology *second-order* stems from the fact that in the context of languages with higher-order constructs (like *e.g.* CRS [33] and HOAS [47]) metavariables can be encoded by second-order variables. However, this is conceptually and technically unsatisfactory. Conceptually, because any understanding of the mathematical structure of metavariables is postponed to that of the syntactic higher-order mechanisms of the meta-language; technically, because one has to work with syntax up to the β-equivalence of the meta-language. The model-theoretic approach of Fiore [17] directly addresses the conceptual problem and, in doing so, solves the technical issue. Furthermore, it opens up new possibilities for research.

The initial task of our work here will be to provide a full account of the extended abstract [17, Part I], expanding it from the mono-sorted setting to the multi-sorted one. Building on this, we are to pursue the following lines of investigation.

[***1.a***] The second-order syntactic structures arising from the mathematical model are more general than those that have been considered in applications in at least two respects: they allow operators that are both *equivariant* (as in [8]) and *parameterised* (as in [50]). Suitable concrete syntax needs to be developed for these; accompanied with case studies validating its applicability, specially in the context of the development of *second-order equational logic* (see below).

[***1.b***] The aforementioned work is carried over in the model $\boldsymbol{\mathcal{F}}$ for variable binding. The question arises as to whether such a development can also be carried through in the model $\boldsymbol{\mathcal{R}}$—recall §*Algebraic meta-theories* in Section 2.1. If so, we would then pursue the use of *nominal techniques* for second-order syntax. This may lead to a novel syntactic theory.

[***1.c***] Our adopted algebraic view lends itself to the consideration of structural recursion and associated induction principles. Thus, we would aim for principles of the same expressive power as the *α-structural recursion/induction* of Pitts [48]. Indeed, if the above task is successful we would expect these principles to be subsumed by this work.

Second-order equational logic. The application of [***Sol*2**] to the above algebraic second-order syntax will provide *second-order equational presentations.* For these, the application of [***Sol*3**] will give a *second-order equational logic*; whose completeness will be analysed by the methods of [***Sol*4**]. The outcome of this research is summarised in the following item.

[***1.d***] A logical framework for specifying and reasoning about presentations of simple type theories, that is sound and complete for a canonical algebraic model theory.

2. Type dependency

The problem of providing algebraic models for abstract syntax (with or without variable binding) in the presence of type dependency has been open for around ten years. Recent progress on the subject for first-order languages with dependent sorts by Fiore [17] strongly suggests the possibility of setting up a complete mathematical theory. Our research proposal in this direction is to reconsider this work and proceed to incorporate (term and type) variable-binding, and capture-avoiding and metavariable substitution. Specifically, we will proceed as follows.

First-order dependent syntax. Our initial task concerning the abstract syntax of first-order languages with dependent sorts is to provide a full account of the extended abstract [17, Part II], which mainly addresses *simple sort dependency*, and proceed to extend this work along the following lines so as to reach at a complete theory.

[***2.a***] Mathematical structures for modelling the general case of sort dependency need to be investigated. Two lines of investigation are to be pursued: (*1*) *sketches* as graphical representations of dependently-sorted signatures, with sorts together with their dependencies and operators [17, Part II]; and (*2*) Cartmell's categories with attributes (see *e.g.* [29]). The former model seems to accommodate the notion of *simultaneous* substitution; the latter one that of *single-variable* substitution.

[***2.b***] A general theory of free constructions for the algebraic structures arising from first-order dependently-sorted languages will be investigated and developed. This will serve as the foundations for initial-algebra semantics and structural recursion/induction in this context.

Second-order dependent syntax. The main challenge to be addressed in the development of second-order dependently-typed abstract syntax is summarised in the following research task.

[***2.c***] Unify the algebraic meta-theories of second-order abstract syntax [***1.a***] and of first-order dependently-sorted abstract syntax [***2.a***], and extend them to incorporate type variable-binding.

Equational logics. The objective above is to build the necessary algebraic meta-theories for the application of [***Sol*1**] to dependently-typed languages. We will then proceed as follows.

[*2.d*] The consideration of [*2.a*] within [***Sol*2**–***Sol*4**] will lead to studies within the realm of *(first-order) dependently-sorted algebraic theories*.

Our research will analyse the two views of dependently-sorted languages provided by Cartmell's *generalised algebraic theories* [6] and by Freyd's *essentially algebraic theories* [23]. Relationships to *cartesian logic* [43, 24] may arise and, if so, will be pursued.

[*2.e*] The application of [***Sol*2**–***Sol*4**] to [*2.c*] will lead to *Algebraic Type Theory*: a body of work providing algebraic models and equational logics for dependent type theories.

3. Graphical structure

The considerations on syntax in [*1*–*2*] can be classified as concerning *cartesian syntax*. Cartesian syntax is roughly characterised by the following two aspects: (*i*) the *wiring* of variables in terms allows for the operations of weakening, contraction, and permutation; and (*ii*) the *graph* structure of terms is given as an ordered tree.

These two combinations of wiring and graph structure for syntax are only one of the possibilities of interest. Along the wiring axis, for instance, the notion of *linearity* (of relevance in logic [2, 1], rewriting theory, and formal languages) corresponds to wiring structure that only supports the operation of permutation. Along the graphical-structure axis, examples of graph structure more general than that of trees arise as *term graphs* and *bigraphs* (*e.g.* in the contexts of rewriting theory [57, 52] and of concurrency theory [9, 44]).

Our approach to the investigation of algebraic meta-theories for graphical syntax will consider the wiring and graph structures in this order. The main methodological reason for this is that the two axes do not seem to be orthogonal to each other, but rather the latter requires aspects of the former.

Wiring structure. We will firstly restrict attention to the investigation of wiring structure in the context of tree structure, fully developing a purely *linear setting* from which we can then proceed to develop a *mixed setting*.

[*3.a*] The first step will be to extend the theory of (first-order) *linear* abstract syntax with variable binding of Tanaka [55] to provide an algebraic meta-theory for second-order linear abstract syntax along the lines of [17, Part I] and [*1.a*].

[*3.b*] A general algebraic meta-theory for second-order *mixed models* (linear, affine [32], relevant [32], cartesian) will be subsequently developed. In this respect, the categorical notion of PRO (which stands for PROduct category [36]) seems to provide the right mathematical concept for modelling the various combinations of wiring structure from which mixed models can be built [15, 16]. A central problem to be addressed here is the development of a general theory of substitution; stepping stones for which are [15, 16] and the work of Power and Tanaka [56].

These developments will build the necessary algebraic meta-theories for the application of [***Sol*1**] to mixed models and pave the way for the following.

[*3.c*] The application of [***Sol*2**–***Sol*4**] to [*3.b*] will lead to *Mixed Simple Type Theory*: a body of work providing algebraic models and equational logics for simple type theories with mixed wiring structure (as *e.g.* Barber and Plotkin's Dual Intuitionistic Linear Logic [1]).

Graph structure. Our study of more general graph structures will start with the development of an algebraic meta-theory for *term graphs* [57, 52]. Our main aim is:

[*3.d*] to give a characterisation of the abstract syntax of term graphs with variable binding as initial algebras, thereby providing algebraic models together with an initial-algebra semantics; and

[*3.e*] to investigate structural recursion and associated induction principles.

This development is ambitious and, if successful, will also make an impact on programming techniques for the algorithmic manipulation of data structures such as directed acyclic graphs.

Logical systems for graphical structure should arise from the application of ***Sol*** to graphical syntax. We propose the following first step from which to start exploring the topic.

[*3.f*] Apply [***Sol*2**–***Sol*4**] to [*3.d*] to extract a rewriting logic for term graphs, that is sound and complete for the canonical model theory.

Restricting attention to first-order term graphs, investigate the relationship between the obtained deductive system and the deductive system for term graphs of Corradini, Gadducci, Kahl, and König [9].

4. Theories and translations

Having developed algebraic meta-theories for general classes of equational logics based on presentations, it is natural to investigate two further related topics: (*i*) *theories* [34], as invariant (*i.e.* presentation independent) versions of equational presentations; and (*ii*) *translations*, as means of relating presentations or theories.

The various notions of equational theory and of translations between them are to arise in complete accordance with the canonical model theory, which will thus guarantee their correctness.

[*4.a*] Notions of theory for the equational logics of [*1*–*3*] are to be investigated. Ideally, a classification will arise and the development of a general mathematical theory for them all could be attempted.

[*4.b*] A framework for translations between equational presentations and/or theories is to be developed.

With a mathematical theory of translations in place, general criteria for achieving conservative-extension results (as *e.g.* in [14, Section 3]) will be sought.

These developments are important in the ever more pressing problem of organising and relating theories of computation.

3. Related work

Particularly relevant to our project is the work of Plotkin [51], who advocated an *algebraic framework* extending the equational logic of universal algebra in the two orthogonal dimensions provided by the addition of variable binding and of type dependency. Our proposed work in this specific context gives a conceptual framework and mathematical methodology for the realisation of this programme by synthesis from algebraic meta-theories.

We will also contribute to the research programme on *internal type theory* of Dybjer [11], which aims at the formalisation of (the meta-theory of) type theory. In this context, the relationship between the type-theoretic approach to induction-recursion [12, 13] and the algebraic approach to structural recursion is a main problem to be investigated.

Our approach shares the basic foundations of the *enriched algebraic theories* of Power *et al.* [40, 38, 39, 30]. However, there are substantial differences: whilst we focus on equational presentations and logics, Power concentrates on theories and on operations for combining them. Extending the common ground of both approaches to a unified theory is a substantial open problem.

References

[1] A. Barber. *Linear Type Theories, Semantics and Action Calculi.* PhD thesis, LFCS, The University of Edinburgh, 1997.

[2] N. Benton, G. Bierman, V. de Paiva, and M. Hyland. Linear λ-calculus and categorical models revisited. In *CSL'92*, LNCS, 1993.

[3] G. Birkhoff. On the structure of abstract algebras. *Proc. of the Cambridge Philosophical Society*, 31:433–454, 1935.

[4] F. Borceux. *Handbook of Categorical Algebra I.* Cambridge University Press, 1994.

[5] R. Burstall. Proving properties of programs by structural induction. *The Computer Journal*, 12(1):41–48, 1969.

[6] J. Cartmell. Generalised algebraic theories and contextual categories. *Annals of Pure and Applied Logic*, 32:209–243, 1986.

[7] N. Chomsky. *Knowledge of Language: Its Nature, Origin, and Use.* Praeger, 1986.

[8] R. Clouston and A. Pitts. Nominal equational logic. *ENTCS*, 172:223–257, 2007.

[9] A. Corradini, F. Gadducci, W. Kahl, and B. König. Inequational deduction as term graph rewriting. *ENTCS*, 72(1):31–44, 2007.

[10] N. de Bruijn. AUTOMATH, a language for mathematics. Technical Report 68-WSK-65, Eindhoven University of Technology, 1968.

[11] P. Dybjer. Internal type theory. In *TYPES'95*, pages 120–134. Springer-Verlag, 1996.

[12] P. Dybjer. A general formulation of simultaneous inductive-recursive definitions in type theory. *JSL*, 65(2):525–549, 2000.

[13] P. Dybjer and A. Setzer. Induction-recursion and initial algebras. *Annals of Pure and Applied Logic*, 124:1–47, 2003.

[14] M. Fiore, , R. Di Cosmo, and V. Balat. Remarks on isomorphisms in typed lambda calculi with empty and sum types. In *LICS'02*, 2002.

[15] M. Fiore. On the structure of substitution. Invited address for MFPS XXII, 2006.

[16] M. Fiore. Towards a mathematical theory of substitution. Invited talk for CT 2007, 2007.

[17] M. Fiore. Second-order and dependently-sorted abstract syntax. In *LICS'08*, 2008.

[18] M. Fiore and C.-K. Hur. Term equational systems and logics. In *Proc. MFPS XXIV*, ENTCS, pages 171–192, 2008.

[19] M. Fiore and C.-K. Hur. On the construction of free algebras for equational systems. In TCS special issue for ICALP'07, 2009.

[20] M. Fiore, G. Plotkin, and D. Turi. Abstract syntax and variable binding. In *LICS'99*, pages 193–202, 1999.

[21] M. Fiore and S. Staton. A congruence rule format for name-passing process calculi from mathematical structural operational semantics. In *LICS'06*, pages 49–58, 2006.

[22] P. Freyd. Algebra-valued functors in general and tensor products in particular. *Colloq. Math. Wroclaw*, 14:89–106, 1966.

[23] P. Freyd. Aspects of topoi. *Bull. Austral. Math. Soc.*, 7, 1972.

[24] P. Freyd. Cartesian logic. *TCS*, 278(1–2):3–21, 2002.

[25] T. Fujiwara. On mappings between algebraic systems. *Osaka Math. J.*, 11:153–172, 1959.

[26] J. Goguen and J. Meseguer. Completeness of many-sorted equational logic. *Houston Journal of Mathematics*, 11:307–334, 1985.

[27] J. Goguen, J. Thatcher, and E. Wagner. An initial algebra approach to the specification, correctness and implementation of abstract data types. In *Current Trends in Programming Methodology.* 1978.

[28] M. Gordon, R. Milner, and C. Wadsworth. Edinburgh LCF: A mechanised logic of computation. LNCS 78, 1979.

[29] M. Hofmann. Syntax and semantics of dependent types. In *Semantics and Logics of Computation*, pages 79–130. CUP, 1997.

[30] M. Hyland and A.J. Power. Discrete Lawvere theories and computational effects. *TCS*, 366:144–162, 2006.

[31] M. Hyland, G. Plotkin, and A.J. Power. Combining computational effects: Commutativity and sum. In *ICTCS*, 2002.

[32] B. Jacobs. Semantics of weakening and contraction. *APAL*, 69(1):73–106, 1994.

[33] J. Klop. *Combinatory Reduction Systems.* PhD thesis, Mathematical Centre Tracts 127, CWI, Amsterdam, 1980.

[34] F. Lawvere. Functorial semantics of algebraic theories. Republished in: Reprints in TAC, No. 5, pp. 1–121, 2004.

[35] D. Lehmann and M. Smyth. Algebraic specification of data types: A synthetic approach. *Math. Systems Theory*, 14:97–139, 1981.

[36] S. MacLane. Categorical algebra. *Bull. Amer. Math. Soc.*, 71:40–106, 1965.

[37] The Maude System. `<http://maude.cs.uiuc.edu/>`.

[38] A.J. Power. Enriched Lawvere theories. *TAC*, 6:83–93, 1999.

[39] A.J. Power. Discrete Lawvere theories. In *CALCO'05*, LNCS 3629, pages 348–363, 2000.

[40] G.M. Kelly and A.J. Power. Adjunctions whose counits are coequalizers, and presentations of finitary enriched monads. *JPAA*, 1993.

[41] M.J. Gabbay and A. Mathijssen. A formal calculus for informal equality with binding. In *WoLLIC'07*, 2007.

[42] M.J. Gabbay and A. Pitts. A new approach to abstract syntax with variable binding. *Formal Aspects of Computing*, 13:341–363, 2002.

[43] C. McLarty. Left exact logic. *JPAA*, 41:63–66, 1986.

[44] R. Milner. *The space and motion of communicating agents*. CUP, 2009. (To appear).

[45] R. Milner, M. Tofte, R. Harper, and D. MacQueen. *The Definiton of Standard ML (Revised)*. The MIT Press, 1997.

[46] B. Nordström, K. Petersson, and J. Smith. *Programming in Martin-Löf's Type Theory: An introduction.* OUP, 1990.

[47] F. Pfenning and C. Elliott. Higher-order abstract syntax. In *Proc. of the ACM SIGPLAN PLDI'88*, 1988.

[48] A. M. Pitts. Alpha-structural recursion and induction. *Journal of the ACM*, 53:459–506, 2006.

[49] G. Plotkin. A metalanguage for predomains. In *Workshop on the Semantics of Programming Languages*, 1985.

[50] G. Plotkin. Some varieties of equational logic (extended abstract). In *Algebra, Meaning and Computation*, LCNS 4060, 2006.

[51] G. Plotkin. An algebraic framework for logics and type theories. *Colloquium in Honor of Gérard Huet*, 2007.

[52] D. Plump. Term graph rewriting. In *Handbook of graph grammars and computing by graph transformation*, pages 3–61. 1999.

[53] D. Scott. Outline of a mathematical theory of computation. O.U.C.L. Programming Research Group, 1970.

[54] D. Scott. A type-theoretical alternative to ISWIM, CUCH, OWHY. Typescript 1969.

[55] M. Tanaka. Abstract syntax and variable binding for linear binders. In *MFPS XXV*, LNCS 1893, pages 670–679, 2000.

[56] M. Tanaka and A.J. Power. A unified category-theoretic formulation of typed binding signatures. In *MERLIN'05*, pages 13–24, 2005.

[57] Terese. *Term Rewriting Systems*, chapter Term graph rewriting by E. Barendsen, pages 712–743. CUP, 2003.