Some Varieties of Equational Logic (Extended Abstract)

Some Varieties of Equational Logic (Extended Abstract)

Gordon Plotkin[1,⋆]

LFCS, School of Informatics, University of Edinburgh, UK.

The application of ideas from universal algebra to computer science has long been a major theme of Joseph Goguen's research, perhaps even *the* major theme. One strand of this work concerns algebraic datatypes. Recently there has been some interest in what one may call algebraic *computation* types. As we will show, these are also given by equational theories, if one only understands the notion of equational logic in somewhat broader senses than usual.

One moral of our work is that, suitably considered, equational logic is not tied to the usual first-order syntax of terms and equations. Standard equational logic has proved a useful tool in several branches of computer science, see, for example, the RTA conference series [9] and textbooks, such as [1]. Perhaps the possibilities for richer varieties of equational logic discussed here will lead to further applications.

We begin with an explanation of computation types. Starting around 1989, Eugenio Moggi introduced the idea of monadic notions of computation [11, 12] with the idea that, for appropriately chosen monads T on, e.g., **Set**, the category of sets, one thinks of $T(X)$ as the type of computations of an element of X. For example, for side-effects one takes the monad $T_S(X) =_{\mathrm{def}} (S \times X)^S$ where S is the set of states. Below, we take $S =_{\mathrm{def}} V^{\mathrm{Loc}}$ where V is a countably infinite set of values such as the natural numbers, and Loc is a finite set of locations. See [2] for a recent exposition of Moggi's ideas, particularly emphasising the connections with functional programming, where the monadic approach has been very influential.

As is well known, equational theories give rise to free algebra monads. For example the free semilattice monad arises from the theory of a binary operation $\cup$ subject to the axioms of associativity, commutativity and idempotence, where the last is the equation $x \cup x = x$. The induced monad $T_N(X)$ is the collection of all non-empty finite subsets of X. In general, the equational theories with operations of finite arity induce exactly those monads which have finite rank, see, e.g., [19].

In denotational semantics one typically employs a category of ordered structures, such as ω-**Cpo**, the category of ω-cpos, which are partial orders with lubs of increasing ω-chains, and with morphisms those monotonic functions preserving the ω-lubs. An ω-**Cpo**-semilattice is a semilattice in ω-**Cpo**, that is an ω-cpo together with a continuous binary function satisfying the semilattice axioms; the free ω-**Cpo**-semilattice monad is (a generalisation of) the convex powerdo-

⋆ This work has been done with the support of EPSRC grant GR/S86372/01 and a Royal Society-Wolfson research merit award.

main monad, originally defined only on a subcategory [5]. There are also lower, or Hoare, and upper, or Smyth, powerdomain monads; these are obtained by adding an additional axiom, viz:

$$x \leq x \cup y$$

for the lower powerdomain, and:

$$x \geq x \cup y$$

for the upper one. Note that these are inequations rather than equations.

This idea was carried further in [15] where similar characterisations were noted for other important monads arising in Moggi's approach, such as those for exceptions, state, input/output, probabilistic nondeterminism and nontermination. One of the main contributions there was an axiomatisation of the state monad employing families of operations of finite or countably infinite arity, as follows. For each location l one assumes given an operation symbol:

$$\mathrm{lookup}_l$$

of arity the countably infinite set V (it is convenient to allow any set to be an arity, not just a cardinal) and for each each location l and value v one assumes given a unary operation symbol:

$$\mathrm{update}_{l,v}$$

The idea is that a term of the form $\mathrm{lookup}_l(\ldots t_v \ldots)$ denotes the computation which looks up the contents of l in the current state and, if this is v, then proceeds according to the computation denoted by the v-th argument, t_v. Similarly a term of the form $\mathrm{update}_{l,v}(t)$ denotes the computation which first updates the contents of the location l to v and then proceeds according to the computation denoted by t.

These ideas have been elaborated into what may be termed the algebraic theory of notions of computation, where the operations and equations are primary and determine the monads. The computational importance of the operations is that it is they that give rise to the effects at hand [16]. Applications include the operational semantics of effects [14], their modular combination [7] and, prospectively, a general logic of effects [17]; see [18] for a survey.

The examples demonstrate that the algebraic theory of computation would benefit from a wider means of expression than is provided by standard equational theories: one also needs to consider parameterization, operations of countable, i.e., denumerable, arity and inequations. As we will see, a unifying rôle is played by Lawvere theories: each such kind of 'equational' theory corresponds to a kind of Lawvere theory, possibly enriched or countable rather than finitary, as standard.

Parameterization This occurs naturally in mathematics, for example in the notion of a vector space over a given field $\mathbb{F}$. There one has the axiom:

$$\lambda(x + y) = \lambda x + \lambda y$$

which involves both field elements and vectors. To treat the notion as an equational theory in the standard sense, one would introduce a unary operation of 'multiplication by λ' for each field element λ and the axiom would be rendered as a family of equations, with one for each field element. We will instead treat the axiom as a single *parametric* equation, with λ a variable ranging over the field and with multiplication by a field element treated as a parametric unary operation on the vector space.

One can go further and allow 'side-conditions,' involving only the parameter variables. For example, in the case of state, treating update as a unary operation parametric over locations and values, one has the following parametric equation:

$$\mathrm{update}_{l,v}(\mathrm{update}_{l',v'}(x)) = \mathrm{update}_{l',v'}(\mathrm{update}_{l,v}(x)) \text{ (if } l \neq l')$$

which has the side condition that $l \neq l'$; the equation states that the order in which one updates distinct locations does not matter.

Such parametric equational theories abbreviate ordinary equational theories, but, by allowing a schema to be replaced by a parametric equation with side conditions, may enable finitary axiomatisation and consequent direct computer implementation. Formally one assumes given an interpretation $\mathfrak{A}$ of a many-sorted first-order signature, the *parameter* signature; for the equational part one further assumes given a *parametric* signature where the operation symbols are assigned a given list of sorts from the parameter signature as well as the usual natural number. There is then a natural notion of parametric term where the parameters are given by standard first-order terms over the parameter signature and so of parametric equation:

$$t = u \ (\varphi)$$

with side condition φ written in first-order logic with equality over the parameter signature. A collection of such equations abbreviate, as indicated above, a standard equational theory over a derived signature.

There is a natural system for deriving these parametric equations from a given collection Th of first-order formulas with equality over the parameter signature, together with another given collection Eqn of parametric equations; the system includes first-order logic with equality for the parameter spaces and equational logic for the parametric equations. One can define whether a parametric equation is a semantic consequence of Th and Eqn relative to the fixed interpretation $\mathfrak{A}$, but, unfortunately, taking Th to be the theory of $\mathfrak{A}$, completeness need not hold. It may, however, hold in particular cases: one such is that of vector spaces mentioned above taking the standard 'ring signature' for the many-sorted first-order signature. On the other hand, fixing Th and Eqn, one can show completeness, if by validity one means with respect to *all* models of Th.

Infinitary operations One can treat operations of countable arity using the evident natural notions of countable equational theory and countable Lawvere theory; the induced monads are those of countable rank. Here is an example of a schema of infinitary equations involving the operation of looking up the contents of a location:

$$\mathrm{lookup}_l(\ldots \mathrm{update}_{l,v}(x)\ldots) = x$$

The equation states that if a location is looked up and then updated with the value found, then that is equivalent to doing nothing.

However it would again be preferable to have a finitary syntax, now for operations of countably infinite arity. To that end, we employ binding on variables of the arity sort, here val (standing for V); the term-forming construction for lookup is then:

$$\mathrm{lookup}_a(v\!:\!\mathrm{val}.t)$$

where a is a parameter term of sort loc (standing for Loc) and t is a parametric term given the environment $v : \mathrm{val}$. With this, the above infinitary schema can be written as the following finitary 'equation':

$$\mathrm{lookup}_l(v\!:\!\mathrm{val}.\mathrm{update}_{l,v}(x)) = x$$

We consider next the following infinitary equation scheme:

$$\mathrm{lookup}_l(\ldots \mathrm{update}_{l',v'}(x_v)\ldots) = \mathrm{update}_{l',v'}(\mathrm{lookup}_l(\ldots x_v \ldots)) \text{ (if } l \neq l')$$

which states that the operations of looking up one location and updating another commute. Notice that it employs a family x_v of variables. If we introduce the notion of a parametric variable (ranging over a suitable collection of functions) this infinitary equation scheme can also be rendered in a finitary fashion:

$$\mathrm{lookup}_l(v\!:\!\mathrm{val}.\mathrm{update}_{l',v'}(x_v)) = \mathrm{update}_{l',v'}(\mathrm{lookup}_l(v\!:\!\mathrm{val}.x_v)) \text{ (if } l \neq l')$$

These two ideas of binding and parametric variables suffice to write down all the parameterized, possibly infinitary, equation schemes for global state given in [15] finitarily.

In the general formalism, we again begin with an interpretation $\mathfrak{A}$ of a parameter signature, as above, except that we assume also given a subcollection of the sorts, called the *arity* sorts. In the parametric signature an operation symbol has m parameter arguments of given parameter sorts, and n argument positions, with the ith being abstracted on k_i arity sorts. A collection of parametric equations abbreviates a countable equational theory, provided that the arity sorts are interpreted by countable sets.

One can then give a logic following the previous lines. An immediate question is whether the logic is complete for global state, where for the many-sorted first-order signature one would take the two sorts, loc and val, and constants for all the elements of Loc, with the evident interpretation using V and Loc. We would also like to know whether we have completeness relative to all interpretations of a given theory, as we do in the simpler case, considered above, of finitary

operations. Positive answers to such questions would demonstrate that valid uniform infinitary equations have uniform proofs.

Inequations These are a natural generalisation of equations and there is an evident notion of inequational, or ordered, equational logic over operations of finite arity, which has a straightforward completeness theorem using posets rather than sets [3]. The resulting ordered equational theories correspond to ordered Lawvere theories, in the sense of [23, 3]. These are not the same as the **Pos**-enriched Lawvere theories of [19], as the latter allow all finite posets as arities of operations, not just the discrete ones. However they are the same as the **Pos**-enriched Lawvere theories of [10], equivalently the **Pos**-enriched discrete Lawvere theories of [20]. There is a natural generalisation to countable inequational logic, and the inequational theories of this logic correspond to the discrete countable **Pos**-theories (the countable case is the main one considered in [20]). In general discrete $\mathbf{V}$-theories of a given rank freely induce $\mathbf{V}$-theories of that rank, in the sense of [19], and the latter induce the $\mathbf{V}$-monads of the same rank; not all such monads arise from discrete theories.

Parameterization, now over given posets, is again an expressive convenience, and there are inequational versions of the two equational deductive systems considered above: one for parametric inequations and the other with finitary syntax for infinitary operations. For the parameter interpretation $\mathfrak{A}$ it is natural to work with enriched first-order structures, which we take to mean here that sorts are interpreted by posets, operations by monotonic functions and relations by subsets; one then naturally works with first-order logic with inequations $a \leq b$, rather than equations, to express parameter conditions. One evidently requires arity sorts to be interpreted by countable discrete partial orders to obtain discrete countable **Pos**-theories from a collection of parametric inequations.

Turning to ω-**Cpo**-enrichment, one can consider discrete finitary or countable ω-**Cpo**-theories. Here parameterization is more than an expressive convenience: it enables one to implicitly write down equations involving sups of increasing chains. One can still work with simple inequations, but rather than finitary or countably infinitary operation symbols, one takes families of such, parameterized over a collection of parameter ω-cpos. They are to be interpreted by functions which are continuous over the parameter ω-cpos as well as the algebra ω-cpo. A natural example is provided by d-cones, which arise when considering powerdomains for mixed ordinary and probabilistic nondeterminism [22]. These are the ω-**Cpo**-semimodules over the semiring $\overline{\mathbb{R}}_+$, which latter is the ω-cpo of the non-negative reals extended with a point at infinity, and endowed with the natural semiring structure [13].

Collections of such inequations induce the discrete finitary or countable ω-**Cpo**-theories, according to the arities of the operation symbols allowed. However there is a question as to what is the appropriate inequational logic. It may be best to introduce an explicit infinitary syntax for sups of increasing ω-sequences, but then sup-terms would only be well-formed if one could prove the sequence was increasing, and that would mean a mutual recursion between the definitions of proofs and well-formed terms. It remains to investigate such a system.

The next question is to what extent one can achieve a useful finitary system. One can clearly investigate analogues of the methods used above to handle parameterization and operations of countably infinite arity. But it is far from clear what to do about the sup-terms. Perhaps one can restrict to considering only least fixed-points and work with a combination of the above ideas and the μ-calculus, for which, and associated logical and categorical results, see [4, 8, 21].

Whatever the difficulties are with finding the right logic, it is at least the case that the combination of parameterization, binding constructions and inequations, interpreted over ω-**Cpo**, is enough to express all the theories of computation types so far considered over that category. We should admit, however, that this is not quite enough to account for all the computation types so far considered. One difficult case is that of the continuations monad. However one can argue that there the types should not be treated as algebraic since the natural operations are not even of the right type to be algebraic operations, and, further, the monad does not have a rank [6].

A more interesting case is that of *local* state, as opposed to the above *global* state, where one can declare new locations. This was treated using a monad over a presheaf category in [15]. The monad was specified by equations, but they involved a mixture of linear and ordinary operations, with the linear structure coming from the Day tensor on the presheaf category. This example feels as if it should be treatable within an algebraic framework, but we do not see the proper notions of Lawvere theory or equational theory. Finally there is also the possibility of employing other semantic categories in place of ω-**Cpo** for the algebraic computational types; we content ourselves here with the remark that for reasonable such categories, one would expect the relevant free algebras still to exist.

References

1. F. Baader & T. Nipkow, *Term Rewriting and All That*, Cambridge University Press, 1998.
2. N. Benton, J. Hughes & E. Moggi, Monads and Effects, *Proc. APPSEM 2000*, LNCS, **2395**, 42–122, Springer-Verlag, 2002.
3. S. L. Bloom, Varieties of Ordered Algebras, *J. Comput. Syst. Sci.*, **13**(2), 200–212, 1976.
4. S. L. Bloom & Z. Ésik, *Iteration Theories: The Equational Logic of Iterative Processes*, EATCS Monographs on Theoretical Computer Science, Springer-Verlag, 1993.
5. M. Hennessy & G. D. Plotkin, Full Abstraction for a Simple Parallel Programming Language, *Proc. 8th. MFCS* (ed. J. Becvár), LNCS, **74**, 108–120, Springer-Verlag, 1979.
6. J. M. E. Hyland, P. B. Levy, G. D. Plotkin & A. J. Power, Combining Continuations with Other Effects, *Proc. 4th. ACM SIGPLAN Continuations Workshop*, 45–47, University of Birmingham Report CSR-04-1, 2004.
7. J. M. E. Hyland, G. D. Plotkin & A. J. Power, Combining Effects: Sum and Tensor, *Theoretical Computer Science*, to appear, 2006.

8. A. J. C. Hurkens, M. McArthur, Y. N. Moschovakis, L. S. Moss & G. T. Whitney, The Logic Of Recursive Equations, *JSL*, **63**(2), 451–478, 1998.
9. J-P. Jouannaud, Twenty Years Later, *Proc. 16th. RTA* (ed. J. Giesl), LNCS, **3467**, 368–375, Springer-Verlag, 2005.
10. J. Meseguer, Varieties of Chain-Complete Algebras, *JPAA*, **19**, 347–383, 1980.
11. E. Moggi, Computational Lambda-Calculus and Monads, *Proc. 4th. LICS*, 14–23, IEEE Press, 1989.
12. E. Moggi, Notions of Computation and Monads, *Inf. & Comp.*, **93**(1), 55–92, 1991.
13. G. D. Plotkin, A Domain-Theoretic Banach-Alaoglu Theorem, Festschrift for Klaus Keimel, *MSCS*, **16**, 1–13, 2006.
14. G. D. Plotkin & A. J. Power, Adequacy for Algebraic Effects, *Proc. 4th. FOSSACS* (eds. F. Honsell & M. Miculan), LNCS, **2030**, 1–24, Springer-Verlag, 2001.
15. G. D. Plotkin & A. J. Power, Notions of Computation Determine Monads, *Proc. 5th. FOSSACS*, LNCS, **2303**, 342–356, Springer-Verlag, 2002.
16. G. D. Plotkin & A. J. Power, Algebraic Operations and Generic Effects, *Applied Categorical Structures*, **11**(1), 69–94, 2003.
17. G. D. Plotkin & A. J. Power, Logic for Computational Effects: Work in Progress, *Proc. 6th. IWFM* (eds. J. M. Morris, B. Aziz & F. Oehl), Electronic workshops in computing, BCS, 2003.
18. G. D. Plotkin & A. J. Power, Computational Effects and Operations: an Overview, *Proc. Domains VI*, ENTCS, **73**, 149–163, Elsevier, 2004.
19. A. J. Power, Enriched Lawvere Theories, *Theory and Applications of Categories*, **6**, 83–93, 2000.
20. A. J. Power, Discrete Lawvere Theories, *Proc. 1st. CALCO* (eds. J. L. Fiadeiro, N. Harman, M. Roggenbach & J. J. M. M. Rutten), LNCS, **3629**, 348–363, Springer-Verlag, 2005.
21. A. Simpson & G. Plotkin, Complete Axioms for Categorical Fixed-point Operators, *Proc. 15th. LICS*, 30–41, IEEE Press, 2000.
22. R. Tix, K. Keimel and G. Plotkin, *Semantic Domains for Combining Probability and Non-Determinism*, ENTCS, Vol. 129, pp. 1–104, Amsterdam: Elsevier, 2005.
23. E. G. Wagner, J. B. Wright, J. A. Goguen & J. W. Thatcher, Some Fundamentals of Order-Algebraic Semantics, *Proc. 5th. MFCS* (ed. A. W. Mazurkiewicz), LNCS, **45**, 153–168, Springer-Verlag, 1976.