Axiomatic Domain Theory*

Marcelo P. Fiore

December 1995

The *denotational semantics* approach to the semantics of programming languages interprets the language constructions by assigning elements of mathematical structures to them. The structures form so-called *categories of domains* and the study of their closure properties is the subject of *domain theory* [Sco70, Sco82, Plo83, GS90, AJ94].

Typically, categories of domains consist of suitably complete partially ordered sets together with continuous maps. But, what is a category of domains? The main aim of *axiomatic domain theory* is to answer this question by axiomatising the structure needed on a mathematical universe so that it can be considered a category of domains. Criteria required from categories of domains can be of the most varied sort. For example, we could ask them to

- have a rich collection of type constructors: sums, products, exponentials, power-domains, dependent types, polymorphic types, etc;
- have fixed-point operators for programs and type constructors;
- have only computable maps [Sco76, Smy77, Mul81, McC84, Ros86, Pho90, Lon95];
- have a Stone dual providing a logic of observable properties [Abr87, Vic89, Zha91].

An additional aim of the axiomatic approach is to relate these *mathematical* criteria with *computational* criteria.

As we indicate below an axiomatic treatment of various of the above aspects is now available but much research remains to be done.

Developments

In the beginning, the axiomatic treatment of domain theory was scattered; it concentrated on fixed-points, mainly of endofunctors but also of endomorphisms.

Concerning fixed-points for endofunctors, already in [Sco72], Scott mentions a suggestion by Lawvere aiming at providing a categorical framework for performing the D_∞ construction. But it was not until [Wan79] that the solution of recursive type equations in categories of domains was first treated *abstractly*, in the sense that no commitment to a particular category of domains was required. Subsequently this approach was developed in [SP82]. The approach was very much appreciated as a *unification* of the techniques for solving recursive type equations in categories of domains, but its *axiomatic* character remained overlooked. For instance, it lead Lehmann and Smyth [LS81]

*Appears in [FJM+96, § Axiomatic Domain Theory].

to outline the first abstract setting for specifying both algebraic (in the ADJ jargon) and recursive types, but these ideas were not pursued further. In [Fre90], aiming at an axiomatic treatment of recursive types, Freyd revisited the previous approaches. And, in [Fre91, Fre92], he proposed a *universal* approach (in the category-theoretic sense) for solving recursive type equations. There he introduced *algebraically compact* categories and established their fundamental property: that bifunctors on them have canonical and minimal fixed-points. This has been a first important step towards an *axiomatic theory of recursive types* (see [Sim92] and [Fio94a, Chapters 6–8]). Other work on algebraic compactness can be found in [Adá93, Bar92].

Concerning fixed-points of endomorphisms, it was noticed by [HP90], after studying the work of [Law64, Law69], that in the presence of cartesian closure they are inconsistent with coproducts (empty or binary). Also algebraic compactness (which yields zero objects) is inconsistent with cartesian closure. This, in principle, precludes a unified treatment of sums, products, exponentials and recursive types via the usual universal properties. However, it was via a direct semantic analysis of non-terminating computations [Plo85] involving *categories of partial maps* [RR88] and, in particular, via the notion of *partial cartesian closure* [LM84] that an appropriate categorical setting emerged.

With this background it was possible, for the first time, to consider categorical models for a rich type theory with recursive types. In [Fio94a, FP94], a notion of categorical model for the metalanguage FPC —a type theory with sums, products, exponentials and recursive types [Plo85, Gun92, Win93]— was defined. Very roughly, categorical models of FPC are algebraically compact partial cartesian closed categories with binary coproducts.

Impact of axiomatic domain theory

In relating operational and denotational semantics. The investigation of the relation between operational and denotational semantics started with a question of Scott in [Sco69]. An answer for PCF (a higher-order functional programming language with fixed-point operators, and base type and arithmetic operations for natural numbers) was given by Plotkin in [Plo77]; there he proved the *soundness* and *adequacy* of the standard semantics of PCF for a call-by-name evaluator. A computational soundness and adequacy result is a correspondence theorem between *operational termination* and *denotational existence*; it generally states that a program terminates according to the operational semantics if and only if its denotational semantics denotes a value. The first axiomatic version of such a result was provided by Berry in [Ber79] (see also [BCL85]) for PCF with respect to a class of models including both the standard one and the *stable* one. In this vein, [Bra94] considered a term language for intuitionistic propositional linear logic extended with fixed-point operators and provided a computational soundness and adequacy result with respect to a categorical semantics.

Concerning languages with recursive types, in [Plo85], FPC was considered as a programming language with a call-by-value operational semantics and a denotational semantics in **pCpo** (the category of cpos and partial continuous functions) and a computational soundness and adequacy result was proved. (Related results can be found in [ML83, MC88].) In [Fio94a], the author gave an axiomatic version of Plotkin's proof for an axiomatisation of absolute non-trivial domain-theoretic models of FPC. From this result it follows that not only the standard model **pCpo** provides a computationally sound and adequate interpretation for FPC but so also do many full subcategories of domains and functor categories over **pCpo**.

In reasoning principles for recursive types. The universal approach taken by the category-theoretic solution of recursive type equations provides, as a by-product, reasoning principles for recursive types. Lehmann and Smyth, and Plotkin [LS81, Plo83], exploited the initiality property of the solution of covariant recursive types (like lists, trees, etc.) to formulate an abstract *induction principle* generalising that of the natural numbers. Motivated by the introduction of algebraic compactness, [Smy92] studied the dual reasoning principle, *coinduction*, in some particular cases. In [Pit94, Pit93], Pitts pursued this line of research further. Drawing upon the initiality/finality universal property given by the compactness axiom, he established a mixed induction/coinduction property of abstract relations on recursive domains in $\mathbf{Cppo}_\perp$ (the category of pointed cpos and strict continuous functions). Abstract category-theoretic accounts of these issues can be found in [Fio93, HJ95].

In type theory. In [CP92], Crole and Pitts introduced a higher-order typed predicate logic for fixed-point computations. This was done by exploiting Moggi's treatment of computations using monads [Mog91], and by introducing the key notion of *fixpoint object*. Fixpoint objects were partly inspired by Martin-Löf's non-standard "iteration type" [ML83], and give a categorical characterisation of general recursion at higer types similar to the characterisation of primitive recursion at higher types in terms of Lawvere's concept of *natural number object* [LS86].

A type-theoretic approach to domain theory is that of [Plo93]. There, rather than considering directly possible categorical structure, the idea is to work within a type theory pursuing the analogies: intuitionistic exponential = function space, and linear exponential = strict function space. More precisely, the basic setting is that of second-order intuitionistic linear type theory enriched with a fixed-point operator for endomorphisms. Then, in the presence of a modified form of Reynold's parametricity the category of linear maps is shown to be algebraically compact with respect to definable endofunctors.

In [Mog95], Moggi describes monadic and incremental approaches to denotational semantics. There, incorporating ideas from axiomatic and synthetic domain theory, typed metalanguages are used to capture relevant structure of semantic categories. Then, by translation into the metalanguage, a variety of programming languages can be interpreted.

In models of domain theory. New non-order-theoretic models of domain theory have been found in connection with research towards establishing a *representation theory* for domains (see '§ Representation theory for domains' below). For illustrative purposes we present one such model. Domains are spaces equipped with a notion of *approximation* (the information order) and a notion of *passage-to-the-limit* (the sup operator). We now consider objects with an *algebraic* (rather than universal) notion of passage-to-the-limit. To this purpose we introduce a notion of *formal-sup operator* due to the author and Gordon Plotkin. Let $\mathcal{D}$ be a cartesian closed category (think of **Poset**, the category of posets and monotone functions), let w be an object in $\mathcal{D}$ (think of ω, the first infinite ordinal), and let $s : w \to w$ be a morphism in $\mathcal{D}$ (think of *succ*, the successor function). A *formal-sup operator* (for diagrams of shape w under the invariance s) on an object D

in $\mathcal{D}$ is given by a map $\bigvee : D^w \to D$ satisfying the following three algebraic laws:

(Constant)
$$\begin{tikzcd} D \arrow[r, "\lambda(\pi_1)"] \arrow[dr, "id_D"'] & D^w \arrow[d, "\bigvee"] \\ & D \end{tikzcd}$$
$\big($i.e. $\bigvee_n\langle x\rangle = x\big)$,

(Diagonal)
$$\begin{tikzcd} D^{w\times w} \cong (D^w)^w \arrow[r, "\bigvee^w"] \arrow[d, "D^{\Delta}"'] & D^w \arrow[d, "\bigvee"] \\ D^w \arrow[r, "\bigvee"'] & D \end{tikzcd}$$
$\big($i.e. $\bigvee_n\langle\bigvee_m\langle x_{m,n}\rangle\rangle = \bigvee_n\langle x_{n,n}\rangle\big)$,

(Shift)
$$\begin{tikzcd} D^w \arrow[r, "D^s"] \arrow[dr, "\bigvee"'] & D^w \arrow[d, "\bigvee"] \\ & D \end{tikzcd}$$
$\big($i.e. $\bigvee_n\langle x_n\rangle = \bigvee_n\langle x_{s(n)}\rangle\big)$.

A *formal cpo* is an object equipped with a formal-sup operator. For formal cpos $(P, \bigvee_P)$ and $(Q, \bigvee_Q)$, a map $f : P \to Q$ in $\mathcal{D}$ is said to be *continuous* if it satisfies the following law:

(Continuity)
$$\begin{tikzcd} P^w \arrow[r, "f^w"] \arrow[d, "\bigvee_P"'] & Q^w \arrow[d, "\bigvee_Q"] \\ P \arrow[r, "f"'] & Q \end{tikzcd}$$
$\big($i.e. $f(\bigvee_P\langle x_n\rangle) = \bigvee_Q\langle f(x_n)\rangle\big)$.

We write $\bigvee_{(w,s)}\mathcal{D}$ for the category of formal cpos and continuous maps. The reader can readily check that our running example $\bigvee_{(\omega,succ)}$**Poset** is **Cpo**, the category of cpos and continous functions. What happens if we replace **Poset** by **Preo** (the category of preorders and monotone functions)? Nothing, $\bigvee_{(\omega,succ)}$**Preo** is again **Cpo**. The surprise comes when we consider the construction for structures with not only a one-dimensional notion of approximation (as the above examples) but with *higher-dimensional* notions of approximation. For instance, consider $\mathbf{Poset}_\wedge$, the category of posets with pullbacks (here the pullback "squares" provide a "two-dimensional" notion of approximation) and stable functions. Then, $\bigvee_{(\omega,succ)}\mathbf{Poset}_\wedge$ cannot be enriched over **Cpo** in a relevant sense [Fio94b] but in it the constructions of domain theory (as, for example, the existence of uniform fixed-point operators and the solution of recursive domain equations) are possible (see '§ Representation theory for domains' below). The above formal-cpo construction has been studied in generality by the author, Plotkin and Power.

Directions

Representation theory for domains. A *representation theorem* is a result that classifies the models of mathematical structures in terms of more concrete models; allowing the study of the general through the study of the particular. (E.g. one such result is Cayley's theorem for groups stating that every group is isomorphic to a subgroup of a group of permutations.) The purpose of setting up a representation theory for domains is to understand the extent to which axiomatisations constrain their models. Work in this direction can be found in [Fio94b]. There the author provided a strong axiomatisation for which enrichment and representation theorems in **Cpo** were proved. Corresponding enrichment and representation theorems for weaker axiomatisations are being explored by the author, Plotkin and Power. This has uncovered a new range of models (among which $\bigvee_{(\omega,succ)}\mathbf{Poset}_\wedge$) based on higher-dimensional *geometric* structures. In fact, these models yield *models of domain theory* in the sense of [Plo95]. According to Plotkin, these consist of a monoidal adjunction between a cartesian closed category and a symmetric monoidal closed one (i.e. a model of intuitionistic linear type theory [BBHdP93]) together with natural axioms for recursion; and, as he shows, they admit the standard techniques for solving recursive domain equations via a version of the limit/colimit coincidence theorem.

Axiomatic and synthetic domain theory. Axiomatic domain theory and synthetic domain theory [Mul81, Ros86, Hyl91, Pho90, Tay91, RS94, Lon95, Ros95] are complementary approaches. On the one hand, synthetic domain theory tries to identify domains, complying with the requirements of axiomatic domain theory, within a universe of sets. On the other hand, as we exemplify below, it is conceivable that one could embed models of axiomatic domain theory in a universe of sets, along the lines prescribed by synthetic domain theory. For example, building upon [Fio94b], for a strong axiomatisation, the author has obtained a representation theorem of the form: every small model has a full and faithful representation in a model of cpos and continuous functions in a presheaf topos; furthermore, the representing model is a full reflective subcategory of the topos. Analogous embeddings for weaker axiomatisations are under investigation by the author, Plotkin and Power.

Type theory. An important direction of research is the formalisation of semantic developments in logical frameworks [dB80, Con86, CH88, HHP92, ACN90] for their subsequent use in machine-assisted reasoning about programs. To this purpose it might be helpful to close the gap between the language of category theory and that of type theory. Recent work in program verification in synthetic domain theory using the LEGO proof checker [Pol95] can be found in [Reu95].

Relating models of FPC. It is an interesting and rather straightforward observation that two interpretations of the simply typed λ-calculus in a cartesian closed category, for which the base types get isomorphic objects, are essentially the same; in that the interpretations of types are canonically natural isomorphic and the interpretations of terms are interdefinable (via the canonical natural isomorphisms). A corresponding result (and generalisations involving interpretations in different models) for FPC would be worth investigating. For instance, this question is relevant to abstract proofs of adequacy (see [Fio94a, Section 10.3]). The main difficulty here is that in the presence of recursive types one has to overcome a kind of *coherence* problem.

Notions of computation. In [Fio94a] an axiomatic setting for partiality was developed. A similar attempt for other notions of computation has not been pursued. A natural step would be to consider non-determinism. One may regard non-determinism as a free construction [HP79] and hence as a notion of computation in the sense of Moggi [Mog91]; alternatively one may develop an abstract setting of observable properties in the vein of [Abr87, Rob87] (see also [Vic89, Joh82]).

Polymorphism and recursion. It would be interesting to develop axiomatic frameworks for polymorphism with recursion (accommodating both domain-theoretic models [CGW89] and per models [AP90, FMRS90]) together with a corresponding representation theory.

Game semantics. The application of *game theory* to semantics has provided new insights and results [AMR94, HO95b, AM95]. In particular, [AMR94] and [HO95a] have constructed *intensionally fully abstract semantic models* of PCF which yield fully abstract models by extensional collapse. As advocated by Abramsky, axiomatic studies of game semantics may lead to abstract full abstraction results (as had happened with adequacy results).

Acknowledgements

I thank P. Cenciarelli and G. Plotkin for their comments.

References

[Abr87] S. Abramsky. *Domain Theory and the Logic of Observable Properties.* PhD thesis, University of London, 1987.

[ACN90] L. Augustsson, Th. Coquand, and B. Nordstrom. A short description of Another Logical Framework. In *Informal Proceedings of the First Workshop on Logical Frameworks, Antibes*, 1990.

[Adá93] J. Adámek. Data types in algebraically ω-complete categories. To appear in Information and Computation, 1993.

[AJ94] S. Abramsky and A. Jung. Domain theory. To appear in the Handbook of Logic in Computer Science, 1994.

[AM95] S. Abramsky and G. McCusker. Games and full abstraction for the lazy λ-calculus. In *10^{th} LICS Conf.*, pages 234–243. IEEE, Computer Society Press, 1995.

[AMR94] S. Abramsky, P. Malacaria, and Jagadeesan R. Full abstraction for PCF (extended abstract). In M. Hagiya and J.C. Mitchell, editors, *Theoretical Aspects of Computer Software*, pages 1–15. Springer-Verlag, 1994. (Full version available from http://theory.doc.ic.ac.uk:80/tfm/papers/MalacariaP/PCFfullabs.dvi.Z).

[AP90] M. Abadi and G.D. Plotkin. A per model of polymorphism and recursive types. In *5^{th} LICS Conf.*, pages 498–507. IEEE, Computer Society Press, 1990.

[Bar92] M. Barr. Algebraically compact functors. *Journal of Pure and Applied Algebra*, 82:211–231, 1992.

[BBHdP93] P.N. Benton, G.M. Bierman, J.M.E. Hyland, and V.C.V. de Paiva. Linear lambda calculus and categorical models revisited. In E. Börger et al., editor, *Selected Papers from Computer Science Logic '92*, volume 702 of *Lecture Notes in Computer Science*. Springer-Verlag, 1993.

[BCL85] G. Berry, P.-L. Curien, and J.-J. Lévy. Full abstraction for sequential languages: the state of the art. In M. Nivat and J.C. Reynolds, editors, *Algebraic methods in semantics*, pages 89–132. Cambridge University Press, 1985.

[Ber79] G. Berry. *Modèles Complètement Adéquats et Stables des Lambda-Calculus Typés*. PhD thesis, L'Université Paris VII, 1979.

[Bra94] T. Braüner. A general adequacy result for a linear functional language. Technical Report RS-94-22, BRICS Research Series, August 1994. (Appears in the proceedings of MFPS'94).

[CGW89] Th. Coquand, C.A. Gunter, and G. Winskel. Domain theoretic models of polymorphism. *Information and Computation*, 81:123–167, 1989.

[CH88] Th. Coquand and G. Huet. The calculus of constructions. *Information and Computation*, 76:95–120, 1988.

[Con86] R.L. Constable et al. *Implementing Mathematics with the NuPrl Proof Development System*. Prentice-Hall, 1986.

[CP92] R.L. Crole and A.M. Pitts. New foundations for fixpoint computations: FIX-hyperdoctrines and the FIX-logic. *Information and Computation*, 98:171–210, 1992.

[CPR91] A. Carboni, M.C. Pedicchio, and G. Rosolini, editors. *Category Theory*, volume 1488 of *Lecture Notes in Mathematics*. Springer-Verlag, 1991.

[dB80] N.G. de Brujin. A survey of the project AUTOMATH. In J.R. Hindley and J.P. Seldin, editors, *To H.B. Curry: Essays in Combinatory Logic, Lambda Calculus and Formalisms*, pages 589–606, 1980.

[Fio93] M.P. Fiore. A coinduction principle for recursive data types based on bisimulation. In 8^{th} *LICS Conf.* IEEE, Computer Society Press, 1993. (Full version to appear in Information and Computation special issue for LICS93).

[Fio94a] M.P. Fiore. *Axiomatic Domain Theory in Categories of Partial Maps*. PhD thesis, University of Edinburgh, 1994. (To be published by Cambridge University Press in the Distinguished Dissertations Series).

[Fio94b] M.P. Fiore. First steps on the representation of domains (extended abstract). Manuscript available from http://www.dcs.ed.ac.uk/home/mf/path.dvi (full version in preparation), December 1994.

[FJM+96] M.P. Fiore, A. Jung, E. Moggi, P. O'Hearn, J. Riecke, G. Rosolini, and I. Stark. Domains and denotational semantics: History, accomplishments and open problems. Technical Report CSR-96-2, The University of Birmingham, School of Computer Science, 1996. (Available from ftp://ftp.cs.bham.ac.uk/pub/tech-reports/1996/CSR-96-02.ps.gz).

[FJP92] M.P. Fourman, P.T. Johnstone, and A.M. Pitts, editors. *Applications of Categories in Computer Science*, volume 177 of *London Mathematical Society Lecture Note Series*. Cambridge University Press, 1992.

[FMRS90] P.J. Freyd, P. Mulry, G. Rosolini, and D.S. Scott. Recursive types reduced to inductive types. In *5^{th} LICS Conf.*, pages 498–507. IEEE, Computer Society Press, 1990.

[FP94] M.P. Fiore and G.D. Plotkin. An axiomatisation of computationally adequate domain-theoretic models of FPC. In *9^{th} LICS Conf.*, pages 92–102. IEEE, Computer Society Press, 1994.

[Fre90] P.J. Freyd. Recursive types reduced to inductive types. In *5^{th} LICS Conf.*, pages 498–507. IEEE, Computer Society Press, 1990.

[Fre91] P.J. Freyd. Algebraically complete categories. In *[CPR91]*, pages 131–156, 1991.

[Fre92] P.J. Freyd. Remarks on algebraically compact categories. In *[FJP92]*, pages 95–106, 1992.

[GS90] C.A. Gunter and D.S. Scott. Semantic domains. *Handbook of Theoretical Computer Science*, pages 633–674, 1990.

[Gun92] C.A. Gunter. *Semantics of Programming Languages: Structures and Techniques*. The MIT Press, 1992.

[HHP92] R. Harper, F. Honsell, and G. Plotkin. A framework for defining logics. *Journal of the ACM*, 40(1):143–184, 1992.

[HJ95] C. Hermida and B. Jacobs. Induction and coinduction via subset types and quotient types. In P. Dybjer and R. Pollack, editors, *Informal proceedings of the Joint CLICS-TYPES Workshop on Categories and Type Theory*, May 1995. (Available as Technical Report 85, Programming Methodology Group, Göteborg University and Chalmers University of Technology).

[HO95a] J.M.E. Hyland and C.-H.L. Ong. On full abstraction for PCF. Manuscript (available from http://theory.doc.ic.ac.uk:80/tfm/papers/OngCLH/pcf.ps.gz), 1995.

[HO95b] J.M.E. Hyland and C.-H.L. Ong. Pi-calculus, dialogue games and PCF. In *7th Annual ACM Conference on Functional Programming Languages and Computer Architecture, La Jolla, California*, 1995.

[HP79] M. Henessy and G. Plotkin. Full abstraction for a simple parallel programming language. *Lecture Notes in Computer Science*, 74, 1979.

[HP90] H. Huwig and A. Poigné. A note on inconsistencies caused by fixpoints in a cartesian closed category. *Theoretical Computer Science*, 73:101–112, 1990.

[Hyl91] J.M.E. Hyland. First steps in synthetic domain theory. In *[CPR91]*, pages 95–104, 1991.

[Joh82] P.T. Johnstone. *Stone Spaces*. Cambridge University Press, 1982.

[Law64] F.W. Lawvere. An elementary theory of the category of sets. *Proc. Nat. Acad. Sci.*, 52, 1964.

[Law69] F.W. Lawvere. Diagonal arguments and cartesian closed categories. In *Category Theory, Homology Theory and their Applications II*, volume 92 of *Lecture Notes in Mathematics*. Springer-Verlag, 1969.

[LM84] G. Longo and E. Moggi. Cartesian closed categories of enumerations for effective type-structures. In G. Kahn, D. MacQueen, and G. Plotkin, editors, *Symposium on Semantics of Data Types*, volume 173 of *Lecture Notes in Computer Science*. Springer-Verlag, 1984.

[Lon95] J. Longley. *Realizability Toposes and Language Semantics*. PhD thesis, University of Edinburgh, 1995.

[LS81] D.J. Lehmann and M.B. Smyth. Algebraic specification of data types: A synthetic approach. *Math. Systems Theory*, 14:97–139, 1981.

[LS86] J. Lambek and P.J. Scott. *Introduction to Higher Order Categorical Logic*. Cambridge University Press, 1986.

[MC88] A.R. Meyer and S.S. Cosmadakis. Semantical paradigms: Notes for an invited lecture. In *3rd LICS Conf.*, pages 236–253. IEEE, Computer Society Press, 1988.

[McC84] D.C. McCarty. *Realizability and Recursive Mathematics*. PhD thesis, University of Oxford, 1984. (Also CMU-CS-84-131, Carnegie-Mellon University).

[ML83] P. Martin-Löf. The domain interpretation of type theory. In Programming Methodology Group, editor, *Workshop on the Semantics of Programming Languages*, pages 21–48, 1983. University of Göteborg and Chalmers University of Technology.

[Mog91] E. Moggi. Notions of computation and monads. *Information and Computation*, 93(1):55–92, 1991.

[Mog95] E. Moggi. Metalanguages and applications. Manuscript (available from http://theory.doc.ic.ac.uk:80/tfm/papers/MoggiE/ML-notes.dvi.gz), 1995.

[Mul81] P. Mulry. Generalized banach-mazur functionals in the topos of recursive sets. *Journal of Pure and Applied Algebra*, 26:71–83, 1981.

[Pho90] W. Phoa. *Domain Theory in Realizability Toposes*. PhD thesis, Trinity College, Cambridge, 1990. (Also CST-82-91, University of Edinburgh).

[Pit93] A.M. Pitts. Relational properties of recursively defined datatypes. In *8th LICS Conf.* IEEE, Computer Society Press, 1993. (Full version to appear in Information and Computation special issue for LICS93).

[Pit94] A.M. Pitts. A co-induction principle for recursively defined domains. *Theoretical Computer Science*, 124:195–219, 1994.

[Plo77] G.D. Plotkin. LCF considered as a programming language. *Theoretical Computer Science*, 5:223–256, 1977.

[Plo83] G.D. Plotkin. Domains. Department of Computer Science, University of Edinburgh, 1983.

[Plo85] G.D. Plotkin. Denotational semantics with partial functions. Lecture at C.S.L.I. Summer School, 1985.

[Plo93] G.D. Plotkin. Type theory and recursion (extended abstract). In *8^{th} LICS Conf.*, page 374. IEEE, Computer Society Press, 1993.

[Plo95] G.D. Plotkin. Algebraic completeness and compactness in an enriched setting. Invited lecture given at ldpl95 (Workshop on LOGIC, DOMAINS, AND PROGRAMMING LANGUAGES), 1995.

[Pol95] R. Pollack. *The Theory of LEGO: A Proof Checker for the Extended Calculus of Constructions.* PhD thesis, University of Edinburgh, 1995.

[Reu95] B. Reus. Program verification in synthetic domain theory. Thesis draft, November 1995.

[Rob87] E. Robinson. Logical aspects of denotational semantics. In *Category Theory and Computer Science*, number 283 in Lecture Notes in Computer Science, pages 238–253. Springer-Verlag, September 1987.

[Ros86] G. Rosolini. *Continuity and Effectiveness in Topoi.* PhD thesis, University of Oxford, 1986.

[Ros95] G. Rosolini. Notes on synthetic domain theory. Manuscript (available from http://theory.doc.ic.ac.uk:80/tfm/papers/RosoliniG/notes_sdt.ps.gz), 1995.

[RR88] E. Robinson and G. Rosolini. Categories of partial maps. *Information and Computation*, 79:95–130, 1988.

[RS94] B. Reus and T. Streicher. Naïve synthetic domain theory. Manuscript, 1994.

[Sco69] D.S. Scott. A type-theoretical alternative to CUCH, ISWIM, OWHY. Manuscript (available as [Sco93]), 1969.

[Sco70] D.S. Scott. Outline of a mathematical theory of computation. Technical Report PRG-2, Oxford Univ. Computing Lab., 1970.

[Sco72] D.S. Scott. Continuous lattices. In F.W. Lawvere, editor, *Toposes, Algebraic Geometry and Logic*, volume 274 of *Lecture Notes in Mathematics*, pages 97–136. Springer-Verlag, 1972.

[Sco76] D.S. Scott. Data types as lattices. *SIAM Journal of Computing*, 5:522–587, 1976.

[Sco82] D.S. Scott. Domains for denotational semantics. In M. Nielsen and E.M. Schimdt, editors, *9^{th} Colloquium on Automata, Languages and Programming*, volume 140 of *Lecture Notes in Computer Science*, pages 577–613. Springer-Verlag, 1982.

[Sco93] D.S. Scott. A type-theoretical alternative to CUCH, ISWIM, OWHY. *Theoretical Computer Science*, 121:411–440, 1993. (Reprint of a manuscript).

[Sim92] A.K. Simpson. Recursive types in Kleisli categories. Manuscript (available from http://www.dcs.ed.ac.uk/home/als/kleisli.dvi.Z), 1992.

[Smy77] M.B. Smyth. Effectively given domains. *Theoretical Computer Science*, 5:257–274, 1977.

[Smy92] M.B. Smyth. I-categories and duality. In *[FJP92]*, pages 270–287, 1992.

[SP82] M.B. Smyth and G.D. Plotkin. The category-theoretic solution of recursive domain equations. *SIAM Journal of Computing*, 11(4):761–783, 1982.

[Tay91] P. Taylor. The fixed point property in synthetic domain theory. In *6^{th} LICS Conf.*, pages 152–160. IEEE, Computer Society Press, 1991.

[Vic89] S. Vickers. *Topology via Logic.* Cambridge University Press, 1989. Volume 5 of Tracts in Theoretical Computer Science.

[Wan79] M. Wand. Fixed point constructions in order-enriched categories. *Theoretical Computer Science*, 8:13–30, 1979.

[Win93] G. Winskel. *The Formal Semantics of Programming Languages: An Introduction.* The MIT Press, 1993.

[Zha91] G.-Q. Zhang. *Logic of Domains.* Birkhäuser, 1991.