ELSEVIER

Science of Computer Programming 35 (1999) 249–253

Science of Computer Programming

www.elsevier.nl/locate/scico

Book review*

Roberto M. Amadio and Pierre-Louis Curien, *Domains and Lambda-Calculi*, Cambridge Tracts in Theoretical Computer Science 46, Cambridge University Press, Cambridge (UK), 1998, xvi + 484 pages, hardback, ISBN 0 521 62277 8.

It is sometimes necessary, for a piece of mathematics that is steadily progressing and whose applicability is continually extended, to organize its current achievements into a comprehensive report that is useful both to *cognoscenti* as a reference and to those who work in related fields in order to learn its basic techniques and concerns.

This is the task accomplished by the monograph under review with respect to *domain theory*, the study of the mathematical structures needed for the semantics of programming languages in the denotational style of Scott and Strachey. In the 30 years that have passed since its introduction, the research in this area has produced a wealth of results and techniques that relate in complex ways to each other and to advanced parts of set theory and category theory going well beyond the basic combinatorial techniques used in the naïve operational approach to programming. Thus, the monograph is correspondingly complex and requires on the part of the reader, in order to be appreciated, a fluency in the set-theoretic language used in most research papers in theoretical computer science.

We present below an overview of the contents of the monograph, organized around a few themes that correspond, we believe, to the topics of interest to a typical potential reader.

1. Semantics of pure λ-calculus

The minimal structure that allows to define the basic constructions needed for the denotational semantics of programming languages and of λ-calculus, in particular, is the directed complete partial order (dcpo), often assumed to have a least element (cpo). Monotonic functions between dcpo's that preserve the least upper bound of directed subsets are the (Scott-) continuous functions: these are continuous with respect to a T_0 topology (the *Scott topology*) that can be defined on any dcpo. The basic structural properties of these partial orders, and the constructions on dcpo's of interest

* Review copies of books which might be of interest to the readers of *Science of Computer Programming* should be sent to Prof. K. Apt (address: see inside front cover). Proceedings of conferences will not normally be reviewed.

PII: S0167-6423(99)00032-5

for denotational semantics are studied in Chapter 1. These include, as usual: the construction of least fixed points for continuous endofunctions over a dcpo (Kleene fixpoint theorem), used in the definition of a direct and a continuation semantics for two simple imperative languages; functors over the category of dcpo's and continuous functions that describe standard constructions in denotational semantics, like products and function spaces, giving rise to a cartesian closed category; the notion of a *basis* of compact elements for a dcpo, used in defining the *algebraic* dcpo's as those, loosely speaking, that are isomorphic to the ideal completion of their basis.

The basic syntactic theory of the λ-calculus, including the Church–Rosser theorem for β-reduction, standardization, finite developments, syntactic continuity and sequentiality theorems, is presented in Chapter 2. This is a precious little introduction to these fundamental results, containing a succinct account of Lévy's labelled λ-calculus, that is used as a technical tool throughout the chapter and whose recently discovered relations with Girard's Geometry of Interaction are also briefly discussed.

Chapter 3 contains the details of Scott's construction, in a generic category of domains, of the solution to the recursive domain equation $D \cong [D \to D]$, i.e. of a domain isomorphic to the domain of its continuous endofunctions, as a projective limit of a denumerable sequence of domains. The chapter also contains a discussion of the relations between syntactic and semantic properties of λ-terms and, notably, an alternative construction of models of the λ-calculus using filters over type structures (the *filter models*), a logically oriented way of presenting domains that has several further developments elsewhere in the monograph. (For example, in Chapter 8 (Section 8.4) a fully abstract filter model for a λ-calculus with a join operator $\sqcup$ for the parallel evaluation of terms is constructed out of a type structure suited to the call-by-value λ-calculus, following work of Boudol.)

2. Typed λ-calculi

Chapter 4 contains the details of how the simply typed λ-calculus (described in Section 4.1) can be interpreted in arbitrary cartesian closed categories (introduced in Section 4.2). The semantical study of the simply typed λ-calculus is carried out by using logical relations, whose fundamental properties are proved and whose use is illustrated through several important applications, including Friedman's completeness theorem, the result of Jung and Tiuryn on definability and the notion of Sieber-sequentiality.

Many fundamental developments in domain theory have been motivated by the full abstraction problem for PCF, a simply typed λ-calculus with arithmetic primitives and fixed-point operator, which is studied in detail in Chapter 6. This also contains a discussion of the problem of full abstraction, together with a first approach to sequentiality and the description of the fully abstract model for PCF in terms of Böhm trees its relation to game semantics.

Going to higher-order typed λ-calculi, dependent types in the Edinburgh Logical Framework (LF) and second-order types in Girard's system F are studied in Chapter

11 from the syntactical and semantical points of view. This study includes an example of coding of logical systems in LF, the coding of free algebras and iterative function in system F and the strong normalization theorem for the latter system, using the powerful technique of reducibility candidates.

The whole Chapter 15 is devoted to the realizability interpretation of second-order λ-calculus, and constitutes a rather self-contained (and quite useful) introduction to some difficult topics in synthetic domain theory, that would otherwise be difficult to find in the literature. Partial equivalence relations are defined both over partial combinatory algebras (Section 15.1), and also over D_∞ models of the λ-calculus (Section 15.7), providing models for theories of subtyping that play an important role in modelling object-oriented languages (Section 15.8).

3. Domain theory

This is of course a leading theme of the whole monograph. The word 'domain' is often used as a generic name for several species of order-theoretic structures: in the original works of Dana Scott at the end of the 1960s domains were understood as particular complete lattices; later Plotkin suggested to drop the top element of lattices, that can hardly be given a computational meaning, while preserving the completeness property for directed subsets or chains. Algebraic dcpo's with a countable basis and continuous functions form what is often considered in the literature as the category of domains in the strict sense. This category is not cartesian closed, and this is unfortunate as cartesian closure is needed for the interpretation of λ-abstraction and application: Chapter 5 is entirely devoted to the problem of finding the largest cartesian closed subcategory of the category of algebraic dcpo's (with a countable basis) and some of its variants. These lead to the introduction of several other categories of domains, including continuous dcpo's, L-domains and bifinite domains (also known as SFP objects).

A logical view of domains, somewhat related to the type structures used in building the filter models for the λ-calculus mentioned above, is described in Chapter 10 and arises from considering the elements of a domain as programs and the Scott-open subsets as program properties. The techniques of pointless topology allow to establish a duality (Stone duality) between categories of domains and categories of locales, where a locale represents a logical theory (this is Abramsky's program of "domain theory in logical form").

The solution of recursive domain equations is treated in Chapter 7, where the solutions are obtained as limits in O-categories following the standard technique studied originally by Wand, Smyth and Plotkin (Section 7.1), and also as special retracts (called *projections*) of *universal* domains (Section 7.3). The construction of universal domains is studied in an abstract setting following work of Gunter, Jung and especially Droste and Göbel, and exploits the property of amalgamation through a generalization of the Fraissé-Jónsson theorem in model theory in the context of algebroidal categories of

domains (most of them are) that allows to obtain *homogeneous* universal domains, uniquely determined up to isomorphism.

4. Stability, sequentiality and linearity

Stable functions have been introduced by Berry in order to reflect semantically the stability property of the operational behavior of λ-terms (this is a consequence of the theory developed in Chapter 2), and give rise to several cartesian closed categories of domains whose morphisms restrict Scott-continuous functions to match better the operational behavior of programs in sequential languages (notably PCF). The monograph studies several of these in Chapter 12, notably *dI-domains* that can be generated as partial orders of configurations of stable event structures. Here the view of a domain as a collection of partial elements whose order is determined by their information content changes somewhat, and is replaced by the conception of computation as progressing through states, or *configurations*, identified with collections of computational *events*. Accordingly, there are different requirements on the structure of domains: for example, now a compact element is required to dominate only finitely many other compact elements (axiom I).

Sequentiality is a further step in the same direction. Although the notion of sequential first-order function was introduced earlier than stability by Milner and Vuillemin, the first cartesian closed category related to sequentiality was defined later by Berry and Curien in terms of *sequential algorithms* over *concrete data structures*, emphasizing even more than in the stable case the event-like aspects of computational steps. The whole subject of sequentiality is covered in depth in Chapter 14, that also contains a full abstraction result for an extension of PCF with the control operator *catch* inspired by the corresponding operator of the programming language Scheme. The interpretation of algorithms as strategies in games is also described (Section 14.3), providing other pointers to game semantics.

A somewhat minimal framework for the definition of stable functions is that of coherence spaces, introduced by Girard as simplifications of quantitative domains arising in his approach to proof-theoretic ordinals. These domains provide a fine analysis of the logical connectives, interpreting logical formulas as coherence spaces and the connectives as constructors of coherence spaces. In particular it is possible to analyze the intuitionistic formula (or type, or coherence space) $A \to B$ as $!A \multimap B$, where $\multimap$ (*linear implication*) and ! are new connectives. Linear logic studies the resulting logical system, in which the application of structural rules (weakening, contraction) is explicitly controlled and becomes part of the formulae by exploiting the new connectives. The computational relevance of linear logic was suggested by Girard since its invention, but only recently there have been formal developments of connections with concurrency theory, domain theory and the implementation of functional languages. Models of linear logic are presented in Chapter 13, in particular coherence spaces (Section 13.1), and Chu spaces (Section 13.5), that can be interpreted as generalized

event structures or as a generalized calculus of relations, after the work of Pratt. There is also an interesting link with sequentiality and stability through Ehrhard's notion of *hypercoherence* (Section 13.3), and a description of the categories in which linear logic can be modelled (Section 13.2).

5. Non-determinism and process calculi

The denotational semantics of formalisms that arise in the study of concurrency, especially process calculi, is much less developed than their operational semantics. This topic is covered in Chapter 9, through a treatment of powerdomains that are used to give a semantics in the denotational style to Milner's Calculus of Communicating Systems, building on work of Abramsky. In Chapter 16, the notion of function is related to that of process by means of the π-calculus, with a special attention to concurrent functional languages. The treatment here is somewhat more operational in style, and a denotational treatment is still a research problem.

We have found the monograph extremely useful as a reference for classical results and techniques in general domain theory, very balanced in the treatment of mathematical properties of domains and their operational motivations and applications, and always deep in the choice of topics and in unravelling the connections between them. Of course, a book of this kind tends to be encyclopedic, and finding an access path through it to a specific topic of interest is not always easy, until the reader becomes rather well acquainted with its contents. However, it is reasonable to believe that the book can be profitably used also as a textbook for graduate students biased towards theoretical issues, by using shortcuts or making suitable selection in a material that lends itself to the monographic treatment of several issues in semantics.

Felice Cardone,
Dipartimento di Informatica, Sistemistica e Comunicazione
Università di Milano-Bicocca
Via Bicocca degli Arcimboldi, 8, I-20126, Milano, Italy
E-mail address: cardone@disco.unimib.it