Peirce Algebras*

Chris Brink[†] Katarina Britz[†]
Renate A. Schmidt[†§]

[†]Department of Mathematics, University of Cape Town
7700 Rondebosch, South Africa

[§]Max-Planck-Institut für Informatik
66123 Saarbrücken, Germany

Extended Abstract

In its modern form the algebra of relations has been under investigation by mathematicians since Tarski's seminal paper of (1941). The main line of development has been the study of a class of algebras called *relation algebras* (Chin and Tarski 1951, Jónsson 1982), in parallel with developments such as Boolean algebras with operators (Jónsson and Tarski 1951/1952) and cylindric algebras (Henkin, Monk and Tarski 1985). Since the early seventies the algebra of relations has increasingly become of interest to computer scientists. Just as the notion of a partial function provides a natural model for deterministic programs, so the more general notion of a (binary) relation provides a natural model for nondeterministic programs. This idea has been exploited by various authors. For example, it is evident in Floyd-Hoare logic for program verification, it has been extended to specification in Hoare and He, Jifeng (1987), it figures in logics of programs such as dynamic logic (Parikh 1981, Harel 1984), and it was used in the early seventies to model recursive procedures (de Bakker and de Roever 1973, Hitchcock and Park 1972). Recently the algebra of relations has been extensively used in a graph-theoretic approach to programs by Schmidt and Ströhlein (1991). In modal logic, relation algebra features strongly in the Dutch-Hungarian cooperation on van Benthem's (1991) new *arrow logic* (see *Logic at Work, Proceedings of the Applied Logic Conference* (1992)). Venema (1992) is another interdisciplinary study of relation algebra and multi-modal logic. The proof theory of relations is also of interest to computer scientists, and several relational inference systems are available (Wadge 1975, Hennessy 1980, Maddux 1983, Orlowska 1991).

In many applications it has become clear that we need, not just an algebra of relations as distinct from an algebra of sets, but an algebra of relations *interacting* with sets. (For example, if we view a program as effecting a transition on a state space, we may wish to model this by a binary relation acting on a set of states.) Such an algebra was presented in Brink (1981) under the name of *Boolean modules.* A Boolean module is defined (Brink 1988) as a two-sorted algebra $\mathcal{M} = (\mathcal{B}, \mathcal{R}, :)$, where $\mathcal{B}$ is a Boolean algebra, $\mathcal{R}$ is a relation algebra and $:$ is a mapping $\mathcal{R} \times \mathcal{B} \longrightarrow \mathcal{B}$ written $r : a$ such that for any $r, s \in \mathcal{R}$ and $a, b \in \mathcal{B}$:

*The full version of this paper is due to appear in 'Formal Aspects of Computing', vol. 6, Springer-Verlag, London (1994).

M1 $r:(a+b) = r:a + r:b$
M2 $(r+s):a = r:a + s:a$
M3 $r:(s:a) = (r\,;s):a$
M4 $e:a = a$
M5 $0:a = 0$
M6 $r^{\smile}:(r:a)' \leq a'$.

The symbols $+$, $:$, $;$, e, 0, ${}^{\smile}$, $'$ and $\leq$ respectively denote join, Peirce product, relational composition, identity, zero, converse, complementation and the usual partial ordering. Let A be any subset of some non-empty set U and let R, S be any binary relations over U. In the standard models (i.e., in *proper Boolean modules*) the join is set-theoretic union. The *Peirce product* $R:A$ is the set of elements x related by R to some element y in A. The relational composition $R;S$ is the set of pairs (x,y) for which there is a z such that $(x,z) \in R$ and $(z,y) \in S$. The identity is the identity relation over U. The zero is the empty set. The converse of a relation R is the set $R^{\smile}$ of pairs (y,x) for which $(x,y) \in R$. Complementation of sets (respectively relations) is with respect to U (respectively $U \times U$). And, $\leq$ is interpreted as the subset relation.

Though independent of the computer science context, Boolean modules are very similar to *dynamic algebras*, introduced by Kozen (1980) as the algebraic version of dynamic logic. And both of these are quite similar to the *extended relation algebras* introduced by Suppes (1976) in a linguistic context. However, Boolean modules and dynamic algebras both have the drawback of not treating relations (programs) and sets equally: there is a set-forming operator on relations, but no relation-forming operator on sets. Extended relation algebras do not have this drawback, but they do have the drawback of being as yet unformalized as algebras.

We present here a two-sorted algebra, called a *Peirce algebra*, of relations and sets interacting with each other. In a Peirce algebra, sets (or rather, the variables representing sets) can combine with each other as in a Boolean algebra, relations can combine with each other as in a relation algebra, and in addition we have both a set-forming operator on relations and a relation-forming operator on sets. The former is the Peirce product used in Boolean modules; the latter is the operation of *cylindrification*. Peirce algebras thus present a natural next step after Boolean algebras, relation algebras and Boolean modules.

Formally, we define a *Peirce algebra* to be a Boolean module $(\mathcal{B}, \mathcal{R}, :)$ enriched with an operation ^{c} from the underlying Boolean algebra $\mathcal{B}$ to the underlying relation algebra $\mathcal{R}$ such that for every $a \in \mathcal{B}$ and $r \in \mathcal{R}$:

P1 $a^{c}:1 = a$
P2 $(r:1)^{c} = r\,;1$.

In the standard models (i.e., in *proper Peirce algebras*) applying the cylindrification operation to a set A yields the relation A^{c} given by the Cartesian product $A \times U$. An example of a Peirce algebra is any extended relation algebra. Another example is any relation algebra. We show that the underlying Boolean algebra $\mathcal{B}$ of any Peirce algebra can be embedded in its underlying relation algebra $\mathcal{R}$ in two ways: as the Boolean algebra of so-called right ideal elements in $\mathcal{R}$, and as the Boolean algebra of elements below the identity of $\mathcal{R}$. These results reiterate the point made by Maddux (1990) that Peirce algebra is not a mathematical requisite for modelling interactions between relations and sets, in the sense that these can be modelled in relation algebras (as interactions with right ideal elements, for example). However, we argue that Peirce algebra provides a more natural framework for doing so. In a Peirce algebra one can actually manipulate both sets and relations simultaneously. From an applications-oriented point of view this is an advantage, and we present two (sets of) sample applications to substantiate this point.

The first shows how three programming constructs in the calculus of weakest prespecification of Hoare and He, Jifeng (1987) can be modelled naturally in Peirce algebras. This comes about through the isomorphism in any Peirce algebra $(\mathcal{B}, \mathcal{R}, :, ^c)$ between the Boolean algebra $\mathcal{B}$ and the Boolean algebra of right ideal elements of the relation algebra $\mathcal{R}$ and the isomorphism between $\mathcal{B}$ and the Boolean algebra of identity elements in $\mathcal{R}$. First, Hoare and He, Jifeng (1987) use right ideal elements to model conditional statements in logics representing programs as binary relations. Second, subsets of the identity relation are used to model a test operation (Parikh 1981). Third, left ideal elements can be used to model the initialization of abstract data types as defined in Hoare, He, Jifeng and Sanders (1987).

The second application points out that the so-called *terminological logics* arising in knowledge representation based on the system KL-ONE (Woods and Schmolze 1992) have evolved a semantics best described as a calculus of relations interacting with sets. Brink and Schmidt (1992) show that the terminological representation language $\mathcal{ALC}$ of Schmidt-Schauß and Smolka (1991) can be captured in the context of Boolean modules. In this paper we extend this idea and use Peirce algebra to accommodate terminological representation languages even more expressive than $\mathcal{ALC}$.

Terminological representation languages have two syntactic primitives, called *concepts* and *roles*. Concepts are usually interpreted as sets and roles as binary relations. As sets and relations have simple calculi that can be presented, respectively, in the context of Boolean algebra and relation algebra, concepts can be modelled in Boolean algebra and roles in relation algebra. Concepts and roles also interact in certain ways, and these can be modelled as interactions between relations and sets. More specifically, concept-forming operations on roles can be interpreted as variants of Peirce product (with two exceptions), and an algebraic characterization for such interactions are Boolean modules. (The exceptions involve numerical quantification.) Role-forming operators on concepts can be interpreted in terms of cylindrification. A natural algebraic presentation for such interactions is then Peirce algebra. The advantages for doing so are: First, Peirce algebra provides a formal mathematical framework for KL-ONE-based knowledge representation, the development of which has, by and large, been implementation-driven and rather *ad hoc*. Second, Peirce algebra provides a natural (equational) axiomatization for reasoning about information represented in a terminological language. Third, terminological representation can be linked to other areas of application of Peirce algebra. Schmidt (1993), for example, exploits the link between Peirce algebra and extended relation algebra and shows how terminological representation can benefit from Suppes' (1976) linguistic analysis of English language sentences.

References

Brink, C. (1981), Boolean modules, *Journal of Algebra* **71**(2), 291–313.

Brink, C. (1988), On the application of relations, *S. Afr. J. Philos.* **7**(2), 105–112.

Brink, C. and Schmidt, R. A. (1992), Subsumption computed algebraically, *Computers and Mathematics with Applications* **23**(2–9), 329–342.

Chin, L. H. and Tarski, A. (1951), Distributive and modular laws in the arithmetic of relation algebras, *Univ. Calif. Publ. Math.* **1**(9), 341–384.

de Bakker, J. W. and de Roever, W. P. (1973), A calculus for recursive program schemes, *in* M. Nivat (ed.), *Symposium on Automata, Formal Languages and Programming*, North Holland, Amsterdam.

Harel, D. (1984), Dynamic logic, *in* D. Gabbay and F. Guenther (eds), *Handbook of Philosophical Logic*, Vol. II, Reidel Publ. Co., Dordrecht, Holland, pp. 497–604.

Henkin, L., Monk, J. D. and Tarski, A. (1985), *Cylindric Algebras: Part II*, Vol. 115 of *Studies in Logic and the Foundations of Mathematics*, North-Holland, Amsterdam.

Hennessy, M. C. B. (1980), A proof-system for the first-order relational calculus, *Journal of Computer and System Sciences* **20**, 96–110.

Hitchcock, P. and Park, D. (1972), Induction rules and termination proofs, *in* M. Nivat (ed.), *Automata, Languages and Programming*, North-Holland, Amsterdam.

Hoare, C. A. R. and He, Jifeng (1987), The weakest prespecification, *Information Processing Letters* **24**, 127–132.

Hoare, C. A. R., He, Jifeng and Sanders, J. W. (1987), Prespecification in data refinement, *Information Processing Letters* **25**, 71–76.

Jónsson, B. (1982), Varieties of relation algebras, *Algebra Universalis* **15**(3), 273–298.

Jónsson, B. and Tarski, A. (1951/1952), Boolean algebras with operators, Part I/II, *American Journal of Mathematics* **73/74**, 891–939/127–162.

Kozen, D. (1980), A representation theorem for models of $*$-free PDL, *in* J. De Bakker and J. van Leeuwen (eds), *Automata, Languages and Programming*, Vol. 85 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, pp. 351–362.

Logic at Work, Proceedings of the Applied Logic Conference (1992), University of Amsterdam. Preprint. To appear.

Maddux, R. D. (1983), A sequent calculus for relation algebras, *Annals of Pure and Applied Logic* **25**, 73–101.

Maddux, R. D. (1990). Personal communication with C. Brink.

Orlowska, E. (1991), Relational interpretation of modal logic, *in* H. Andréka, J. D. Monk and I. Németi (eds), *Algebraic Logic*, Vol. 54 of *Colloquia Mathematica Societatis János Bolyai*, North-Holland, Amsterdam, pp. 443–471.

Parikh, D. (1981), Propositional dynamic logic of programs: A survey, *in* E. Engeler (ed.), *Logic of Programs*, Vol. 125 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, pp. 102–144.

Schmidt, G. and Ströhlein, T. (1991), *Relations and Graphs*, Springer-Verlag, Berlin.

Schmidt, R. A. (1993), Terminological representation, natural language & relation algebra, *in* H. J. Ohlbach (ed.), *Proceedings of the sixteenth German AI Conference (GWAI-92)*, Vol. 671 of *Lecture Notes in Artificial Intelligence*, Springer-Verlag, Berlin, pp. 357–371.

Schmidt-Schauß, M. and Smolka, G. (1991), Attributive concept description with complements, *Artificial Intelligence* **48**, 1–26.

Suppes, P. (1976), Elimination of quantifiers in the semantics of natural language by use of extended relation algebras, *Rev. Int. de Philosophie* **30**(3–4), 243–259.

Tarski, A. (1941), On the calculus of relations, *Journal of Symbolic Logic* **6**(3), 73–89.

van Benthem, J. (1991), Logic and the flow of information, *Technical Report, ILLC Prepublication Series for Logic, Semantics and Philosophy of Language LP-92-11*, Institute for Logic, Language and Computation, University of Amsterdam. To appear.

Venema, Y. (1992), *Many-Dimensional Modal Logic*, PhD thesis, University of Amsterdam.

Wadge, W. W. (1975), A complete natural deduction system for the relational calculus, *Theory of Computation Report 5*, University of Warwick.

Woods, W. A. and Schmolze, J. G. (1992), The KL-ONE family, *Computers and Mathematics with Applications* **23**(2–5), 133–177.