Available online at www.sciencedirect.com
ScienceDirect
Electronic Notes in Theoretical Computer Science
ELSEVIER
Electronic Notes in Theoretical Computer Science 162 (2006) 227–231
www.elsevier.com/locate/entcs

# What is algebraic in process theory?

Bas Luttik[1]

*Department of Mathematics and Computer Science*
*Technische Universiteit Eindhoven, The Netherlands*
*CWI, Amsterdam, The Netherlands*

**Abstract**

Process theory started in the 1970's with an emphasis on giving an algebraic treatment of its fundamental concepts. In the 1990's, with the rapid introduction of advanced features (data, time, mobility, probability, stochastics), the algebraic line was largely abandoned. I believe that a thorough abstract algebraic treatment adds a degree of mathematical maturity and elegance to the theory. In this note I discuss what is algebraic in process theory, and what is not (yet).

*Keywords:* Process theory, process algebra, abstract algebra.

## 1 Prologue

In mathematics, sometimes two kinds of *algebra* are distinguished: elementary and abstract. *Elementary algebra* records the properties of the real number system, mostly in the form of equations using symbols to denote constants (particular real numbers) and variables (ranging over all real numbers). Elementary algebra is concrete in the sense that it is about one particular kind of object: the real number. *Abstract algebra* (also known as *modern algebra*) is concerned with the study of (the properties of) the fundamental operations of arithmetic in more generality, e.g., no longer talking about addition of real numbers only, but talking about addition of anything that might be worth adding. The generality is usually achieved by defining the fundamental operations axiomatically.

For an example of an axiomatic definition, consider a theory of two binary operations defined by postulating that the first operation is commutative, associative and idempotent, and that the second operation distributes from the right over the first and is also associative. A ring theorist may tell you that this comes close to a definition of the theory of idempotent semirings, except that a few axioms are

[1] Email: s.p.luttik@tue.nl

1571-0661/$ – see front matter 
doi:10.1016/j.entcs.2005.12.086

surely missing. Most notably, the ring theorist remarks, an axiom expressing that the second operation also distributes from the left over the first ought to be included. A process theorist, however, will recognize that this axiom has been left out on purpose, for what we have here is a definition of the theory of alternative and sequential composition of processes. This particular version of the theory was proposed by Bergstra and Klop in 1982 (see [4,6]); they presented it as a set of formal equations.

## 2 Algebraic process theory

Algebraic process theory started in the 1970's, with the introduction of CSP by Hoare [7,11,12] and of CCS by Milner [15,16]. What is algebraic about CSP and CCS? It is the fact that the emphasis is on studying the properties of a collection of fundamental operations on processes. CSP and CCS for the bigger part agree on what are those fundamental operations, both including sequencing, nondeterministic choice, and parallel composition. Moreover, these constructs turned out to satisfy very similar properties. The main difference between CSP and CCS lies in to what is viewed as the appropriate mathematical representation of the notion of process: in CSP a process is mathematically represented as a set of failures [2], whereas in CCS it is an element of the set of labelled transition systems modulo observation equivalence.

Defining a language of first-order operations on a domain of processes and proving properties of these operations is algebra in the elementary sense of the word. With their seminal paper [10], Hennessy and Milner made an important step in the direction of a more abstract approach, providing a ground-complete [3] equational axiomatisation of observation equivalence in the context of recursion-free CCS. Their axioms could in principle be taken as an abstract algebraic definition. A genuine abstract algebraic approach was first explicitly proposed by Bergstra and Klop [4,6]. One of their methodological concerns when introducing ACP was to present "first a system of axioms for communicating processes [...] and next study its models" (see [6, p. 112]).

Let me try to avoid a misunderstanding here as to why Bergstra and Klop's theory is algebraic in the sense of abstract algebra. It is *not* (or at least not merely) the fact that it uses equational axioms to define the operations. The equations are just a means to realise the real desideratum of abstract algebra, which is to abstract from the nature of the objects under consideration. In the same way as the mathematical theory of rings is about arithmetic without relying on a mathematical definition of *number*, Bergstra and Klop's proposal deals with process theory without relying on a mathematical definition of *process*.

[2] A *failure* is a sequence of events in which a process may engage together with a set of events that it subsequently refuses to engage in.

[3] We call an axiomatisation *ground-complete* if any two behaviourally equivalent *closed* process expressions are provably equal.

**Algebraic achievements**

In the second half of the 1980's the algebraic approach in process theory received quite some attention. We briefly mention three categories of algebraic results (see Aceto's paper [1] for a more elaborate overview with the appropriate references):

**Expressiveness:** Several results were obtained showing that certain combinations of fundamental process theoretic operations are more expressive than others. For instance, it was shown that the behaviour of a *stack* can be specified by means of a finite recursive specification using alternative composition and sequential composition, while this is not possible if sequential composition is replaced by prefix multiplication [5].

**Axiomatisability:** A lot of effort was put into providing satisfactory equational axiom systems for certain combinations of process theoretic operations, and showing that satisfactory equational axiom systems do not exist for other combinations. Here *satisfactory* usually meant *finite* and *ground-complete* with respect to some notion of behavioural congruence.

**Unique decomposition:** For several versions of parallel composition it has been established that every process can be uniquely expressed as a parallel composition of parallel prime processes up to a certain behavioural equivalence. The first such result was obtained by Milner and Moller in [17].

Most of the abovementioned results are algebraic in the same way as elementary algebra is algebraic: they record properties of a collection of operations defined on a predetermined mathematical model of processes (usually, labelled transition systems modulo a behavioural equivalence). The $\omega$-completeness results presented by Moller [18] in his excellent PhD thesis can be considered an exception; they are more abstract algebraic since they are about the quality of the axiom systems themselves and do not rely on a particular predetermined mathematical model of processes.

In a recent paper [14], the author together with Vincent van Oostrom showed that the story of unique decomposition results can be retold in the abstract algebraic setting of commutative monoids. The predominant technique, discovered by Milner, to prove unique decomposition results in process theory was generalised along abstract algebraic lines to the abstract algebraic setting of commutative monoids, yielding a complete axiomatisation of the class of commutative monoids with unique decomposition. The great advantage is that to prove unique decomposition with respect to *some* version of parallel composition up to *some* behavioural equivalence, it now suffices to establish that the induced monoid satisfies the axioms that make the general proof go through.

**Not yet algebraic**

In the 1990's, attention shifted towards the introduction in process theory of sophisticated features such as data, time, mobility, probability and stochastics [2], and less effort was put into providing an algebraic treatment. (A notable exception is the work on recursive operations, see the recent survey [3]). Most of the process

theoretic treatments of these features involved the use of variable binding operations. For instance, the formal process specification language $\mu$CRL [8], which combines process theoretic operations from ACP with abstract data types, involves *choice quantifiers* $\sum_d$. The intuition is that if $p(d)$ is a formal $\mu$CRL expression with a free variable $d$ ranging over the values of some datatype, then $\sum_d p(d)$ denotes an alternative composition with a summand $p(v)$ for every value $v$ of the datatype. The construction can be used to express value-passing.

The choice quantifiers of $\mu$CRL, and binding operations in general, are not algebraic. The reason is that they rely for their definition on the syntactic nature of the objects on which they act, for they are supposed to bind a variable in the objects. Recall the desideratum of abstract algebra: the intrinsic nature of the objects should not matter. Thus, saying that $\mu$CRL is algebraic amounts to saying that a process is an expression, which of course it isn't. In [13] it is shown that it is possible to provide an abstract algebraic treatment of choice quantification much in the same way as existential quantification is treated in algebraic logic [9].

## 3 In conclusion

I believe that a thorough abstract algebraic treatment will add a degree of mathematical maturity and elegance to the field of process theory. Therefore I think that we should further develop the abstract algebraic side of process theory, by giving abstract algebraic treatments of advanced process theoretic concepts (e.g., mobility, time, stochastics) and by considering fundamental process theoretic results and constructions from an algebraic perspective.

## References

[1] Aceto, L., *Some of my favourite results in classic process algebra*, BRICS Report NS-03-2, BRICS, Department of Computer Science, Aalborg University (2003).

[2] Baeten, J. C. M., *A brief history of process algebra*, Theoret. Comput. Sci. **335** (2005), pp. 131–146.

[3] Bergstra, J. A., W. J. Fokkink and A. Ponse, *Process algebra with recursive operations*, in: J. A. Bergstra, A. Ponse and S. A. Smolka, editors, *Handbook of Process Algebra*, Elsevier Science Inc., 2001 pp. 333–389.

[4] Bergstra, J. A. and J. W. Klop, *Fixed point semantics in process algebra*, Technical report IW 280, Mathematical Centre (1982).

[5] Bergstra, J. A. and J. W. Klop, *The algebra of recursively defined processes and the algebra of regular processes*, in: J. Paredaens, editor, *Proceedings of ICALP'84*, LNCS 172, 1984, pp. 82–95.

[6] Bergstra, J. A. and J. W. Klop, *Process algebra for synchronous communication*, Information and Control **60** (1984), pp. 109–137.

[7] Brookes, S. D., C. A. R. Hoare and A. W. Roscoe, *A theory of communicating sequential processes*, J. ACM **31** (1984), pp. 560–599.

[8] Groote, J. F. and A. Ponse, *The syntax and semantics of $\mu$CRL*, in: A. Ponse, C. Verhoef and S. F. M. van Vlijmen, editors, *Algebra of Communicating Processes*, Workshops in Computing (1994), pp. 26–62.

[9] Halmos, P. R., *The basic concepts of algebraic logic*, American Mathematical Monthly **63** (1956), pp. 363–387.

[10] Hennessy, M. and R. Milner, *Algebraic laws for nondeterminism and concurrency*, J. ACM **32** (1985), pp. 137–161.

[11] Hoare, C. A. R., *Communicating sequential processes*, Commun. ACM **21** (1978), pp. 666–677.

[12] Hoare, C. A. R., "Communicating Sequential Processes," Series in Computer Science, Prentice-Hall International, London, 1985.

[13] Luttik, B., "Choice Quantification in Process Algebra," Ph.D. thesis, University of Amsterdam (2002), available from `http://www.win.tue.nl/~luttik`.

[14] Luttik, B. and V. van Oostrom, *Decomposition orders—another generalisation of the fundamental theorem of arithmetic*, Theoret. Comput. Sci. **335** (2005), pp. 147–186.

[15] Milner, R., "A Calculus of Communicating Systems," LNCS 92, Springer-Verlag, Berlin, 1980.

[16] Milner, R., "Communication and Concurrency," Prentice-Hall International, Englewood Cliffs, 1989.

[17] Milner, R. and F. Moller, *Unique decomposition of processes*, Theoret. Comput. Sci. **107** (1993), pp. 357–363.

[18] Moller, F., "Axioms for Concurrency," Ph.D. thesis, University of Edinburgh (1989).