The Concurrency Column

by

Luca Aceto

BRICS, Department of Computer Science
Aalborg University
9220 Aalborg Ø, Denmark
Email: `luca@cs.auc.dk`
`http://www.cs.auc.dk/~luca/BEATCS/`

As frequent readers of this part of the Bulletin may have already noticed, from this volume I have taken over the job of editing the concurrency column. On behalf of the whole concurrency community, I take this opportunity for thanking my predecessor, Mogens Nielsen, for his superb editorial work over the years. In order to follow on Mogens' footsteps I need the assistance of the concurrency community, and I encourage anybody interested in contributing an article to the concurrency column to contact me at the above email address.

This column will be published soon after CONCUR 2003, the 14th International Conference on Concurrency Theory, that was held in Marseille in the period September 3–5, 2003. This was the best attended CONCUR conference to date, and its lively scientific programme witnessed the vitality of our research field. While waiting for a conference report to appear in a future volume of the Bulletin, I encourage those of you who could not travel to Marseille to check the programme of the main conference and its satellite workshops at the URL http://concur03.univ-mrs.fr/.

To inaugurate my stint as editor of the concurrency column, I present this month a personal note devoted to some of my favourite results in classic process algebra and to some open problems in the field. I hope that this piece will contribute to the healthy development of this area of research.

Some of My Favourite Results in Classic Process Algebra

Luca Aceto
BRICS, Department of Computer Science
Aalborg University
9220 Aalborg Ø, Denmark
Email: luca@cs.auc.dk

This note is dedicated to Anna and Róbert.

> "What I had done so far amounted to a mere fraction of nothing at all. It was so much dust, and the slightest wind would blow it away." [13, Page 207]

1 Introduction and Disclaimers

I have been asked a few times what my favourite results in process algebra are, and often thought that it might be a useful exercise to reflect a little on that question, and pen down my thoughts. The spur for attempting this somewhat foolish enterprise at this point in time comes from the workshop on "Process Algebra: Open Problems and Future Directions" that I co-organize with Zoltán Ésik, Wan Fokkink and Anna Ingólfsdóttir. My co-organizers and I hope that, apart from being a celebration of over twenty years of research in process algebra, that workshop will bear witness to the continuing vitality of this field of investigation, and I believe that one of the ways to contribute to the solution of new problems and the development of new avenues for research is to mull over the results that have been achieved so far and the open problems they raise.

Whether the writing of this note means that I am "past the best before date" as an active researcher in the process algebra community I must leave to others to judge—I certainly am if Godfrey Harold Hardy was to be trusted when he wrote

> "It is a melancholy experience for a professional mathematician to find himself writing about mathematics. [...] I write about mathematics because [...] I have no longer the freshness of mind, the energy, or the patience to carry on effectively with my proper job. [43, pp. 61 and 63]"

However, I hope that Gian-Carlo Rota's statement from [71] that

> "Not only is it good for you to write an expository paper once in a while, but such writing is essential for the survival of mathematics."

will prove to be true for me in this case.

It goes without saying that the list of results in process algebra that I will present in what follows is very partial, is based entirely on my personal views of the moment and my tastes, and is limited by my (lack of) knowledge, energy and time. (Indeed, I can guarantee that if I were to draw a similar list in a few months' time, then that list would most likely be different.) I apologize to all the colleagues of mine whose excellent results are not mentioned here, and whose work I know less than I should. There are just so many beautiful results in our field of interest, and the variety is so large, that there is no hope to do justice to even a tiny fragment of them in a piece like this one—the interested reader is invited to browse through [21] to obtain a bird's eye view of research in process algebra. Indeed, if ever there is a message to be gleaned from this note, then it is that the process algebra research community should be proud of its non-trivial achievements.

As the readers of this note will notice, the results mentioned below are not just confined to the algebraic aspects of process theory, and reflect my very generous, and somewhat arbitrary, view of what process algebra is. I believe that algebraic ideas underlie many facets of process theory in that algebra provides *structure and reasoning techniques* that turn out to guide our thoughts even when we work with plain automata based formalisms, logics, testing approaches and so on. Moreover I see the results in process algebra proper, and process theory in general, as belonging to the same body of work regardless of whether algebraic techniques play an explicit role in them or not. That is why this survey mentions results which appear to be purely about logic and/or automata. I hope that this position won't upset any sensibilities amongst my colleagues—indeed, it is meant to bring a sense of unity to the work of what I see as a community of kindred spirits with different tastes and interests in research.

In keeping with the foolish nature of this whole enterprise, I have decided to restate some of the items that populate my "to do" list as problems in this note. Some of them are very specific, some others are very vague and most likely are just wishful thinking. I hope to be able to settle at least one of them in the rest of my career.

The presentation is based on my notes for an imaginary talk on my favourite results in classic process algebra, and is aimed at readers who are familiar with concurrency theory, and process algebra in particular. I have made no real attempt to turn it into a polished piece of scientific writing that is accessible to a wide readership. I hope that the bibliographic references will help the patient uninitiated reader find much better and thorough reviews of the results mentioned in this note.

The structure of the note is as follows. Section 2 presents results and problems related to behavioural equivalences and their relationships with modal logic. Section 3 discusses theorems on the (non-)existence of finite equational axiomatizations for behavioural equivalences over fragments of process algebras. Two results from the meta-theory of process algebras based on structural operational semantics are reviewed in Section 4. Section 5 is devoted to a taste of expressiveness results in classic process algebra. The note concludes with the mention of two theorems highlighting the connections, and the differences, between the theory of process algebra and the classic theory of automata and formal languages (Section 6).

2 Behavioural Equivalences and Logic

Let me start with a classic example of an early result proven by Hennessy and Milner in [45] that has had a monumental impact on many aspects of research in process theory, is based on a cute idea, but does not have a mind-bogglingly complex proof—once one has seen it, of course! It is the kind of result that we regularly teach our students in Aalborg in an introductory concurrency theory course, and whose proof has reached what Paul Erdős might have called the "Book Proof" stage.

Most researchers in process theory will agree that one of the fundamental notions of behavioural equivalence over labelled transition systems is (strong) bisimulation equivalence [60, 66]. This notion of equivalence has many alternative characterizations. Amongst these, I want to mention here the characterization of bisimulation equivalence in terms of a modal logic that is often referred to as *Hennessy-Milner Logic*.

Result 1 (Hennessy and Milner [45]) *Two image finite processes are bisimulation equivalent if, and only if, they afford the same properties that can be expressed in Hennessy-Milner Logic.*

Variations on this result abound in the literature (see, e.g., [2, 11, 33, 39, 53, 58, 78]). Indeed one of the tests for the reasonableness of a behavioural equivalence is that it has a pleasing, natural modal characterization. Modal characterizations of behavioural equivalences and preorders have important uses in, e.g.,

- automatic verification of reactive systems (for instance, in providing a convenient formalism for expressing *distinguishing formulae*—see, e.g., [52, 55]),
- characterizing equivalence classes of (finite) processes with respect to some notion of behavioral equivalence by means of the so-called *characteristic formulae* [49, 77],
- showing finitariness and algebraicity of behavioural preorders [11],
- the characterization of the largest congruences included in completed trace equivalence that are induced by SOS formats—see, e.g., [41, 42], and
- non-finite axiomatizability proofs [9].

The interpretation of Hennessy-Milner Logic that matches notions of bisimulation preorder, as developed in [58, 78], has pleasing connections with intuitionistic logics.

The second result I want to mention here also makes use of Result 1 in the generation of tests from formulae in Hennessy-Milner logic to characterize bisimulation equivalence as a testing equivalence. It is one of the main results in a line of research that was quite popular in the late 1980's and early 1990's, when an effort was being made to justify, or at least clarify, the observational nature of bisimulation equivalence.

Result 2 (Abramsky [1]) *Abramsky's characterization of bisimulation equivalence as a testing equivalence.*

A notable, albeit maybe not so well known, result buried in Abramsky's paper is a theorem to the effect that adding any collection of monotonic tests to the language for tests used in the proof of Result 2 does not increase the distinguishing power of the test language. To my mind, this is possibly the closest we have come to giving a precise solution to the following

Problem 1 *Can one prove in a formal sense that bisimulation equivalence is the finest "reasonable" behavioural equivalence?*

3 Axiomatizations of Behavioural Equivalences

One of the natural outcomes of the algebraic structure of process description languages is that we can formulate general (in)equivalences between process terms that we expect to hold with respect to the chosen notion of behavioral semantics in the linear time-branching time spectrum [39] in terms of (in)equations. Several natural questions immediately arise pertaining to the (non-)existence of (finite) axiomatizations for behavioural equivalences over fragments of process description languages. I will now mention some of my favourite results in this line of research. It will come as no surprise to the readers of this note who know about my work that I am very partial to this type of contribution, and this is reflected by the role they play in this presentation.

3.1 Positive Results

The axiomatization of bisimulation equivalence over the recursion-free fragment of CCS offered in [45] employs an axiom schema, the seminal and well known *expansion theorem*. This begged the question of whether one could replace this axiom schema with a finite set of equations. Bergstra and Klop showed that this can be done by extending the language with two auxiliary operators, the *left merge* and the *communication merge*, that can be used to finitely axiomatize the parallel composition operator in bisimulation semantics.

Result 3 (Bergstra and Klop [19]) *Bergstra and Klop's axiomatization of the merge operator in terms of the auxiliary left merge and communication merge operators.*

Again, this is an example of a seminal result which is based on an ingenious idea, but whose proof is not technically very complicated once the appropriate machinery is in place. An analysis of the reasons why operators like the left merge and the communication merge are equationally well behaved in bisimulation semantics has led to general algorithms for the generation of (finite) equational axiomatizations for behavioural equivalences from their operational semantics—see, e.g., [4] and the references in [10].

Problem 2 *Can one give a finite axiomatization for the weak behavioural preorders studied in [86] over the recursion-free fragment of CCS enriched with the constant* Ω *(to stand for a divergent process), the left merge and the communication merge?*

To my mind, two of the most satisfying results in the theory of processes are:

Result 4 (Milner [59, 61]) *Milner's axiomatizations of bisimilarity and observation congruence over the regular fragment of CCS.*

Even though one can trace the general proof strategy employed in the proofs of those results to Salomaa's axiomatization of the theory of regular expressions [72], the axioms given by Milner are so elegant, and the proofs are so crisp, that those results set high standards for the whole community. Other axiomatization results for regular processes use similar proof strategies—see [12, 37, 70] to list but a few. Let me remark in passing that the axiomatization first presented by Bergstra and Klop in [20] played a seminal role in the discovery of the laws for observational congruence given by Milner in [61]. Indeed, Milner himself states in *op. cit.* that his axioms were inspired by a reading of [20].

Problem 3 *Settle Milner's conjecture in [59] regarding the axiomatization of bisimulation equivalence over the language of regular expressions.*

Problem 4 *Characterize those recursive equations over the regular fragment of CCS that yield a process that can be represented by a regular expression modulo strong bisimulation equivalence. (This open question was originally raised by Milner in [59].) Some work on this problem has been done by Bosscher in [25, Chapter 3] and by De Nicola and Labella in [32].*

Problem 5 *Axiomatize all the equivalences in the linear time-branching time spectrum over regular processes. Obtain axiomatizations for these equivalences that are relative to those for iteration algebras given by Bloom and Ésik—see, e.g., [24].*

Axiomatization results for behavioural equivalences based on interleaving over fragments of process algebras that cannot describe infinite behaviours usually consist of a collection of laws that allow one to reduce terms to finite synchronization trees, and laws that axiomatize the equivalence under consideration over finite synchronization trees. This strategy is not directly applicable when the equivalence under consideration treats parallelism as a primitive notion. To my mind, the most satisfying approach in this setting is to isolate a collection of axioms that explicitly characterize the relevant properties of the parallel composition operator. (See, e.g., [26, 31] for the application of an alternative, very ingenious approach to the problem.) One of the earliest examples of such a set of laws is given by

Result 5 (Hennessy [44]) *Hennessy's axiomatization of the relation of timed congruence (also known as split-2 congruence) over a recursion-free process language.*

This result was published in 1988 by the SIAM Journal on Computing, but was actually obtained in 1981–1982. In particular, I am quite partial to the decomposition result that underlies the completeness proof. This is an example of a result that very few researchers cared about when it was obtained (see the publication delay!), but that became fashionable and relevant at the heyday of the study of non-interleaving equivalences for process description languages. This is also an example of a result whose "proof" employs techniques that are, I believe, seminal, but whose statement turned out to be incorrect—more precisely, one of the proposed axioms is not sound (see, e.g. [3] for a general discussion of this problem, and a correct proof of Hennessy's result). To my mind, this does not diminish the importance of the contribution to the research area, and I still recall the astonishment—maybe common amongst PhD. students—I had when I read that paper the first time at the mere thought that somebody could have proven such a result.

Result 6 (Fokkink [34]) *The completeness proof for bisimulation equivalence over BPA* published by Fokkink in op. cit.*

This completeness theorem was first proven by Fokkink and Zantema in [35], but the proof presented in [34] is probably the "Book Proof" for that result. Its main interest lies in the wealth of proof techniques that it introduces, which have turned out to be useful in establishing related results—see, e.g., [7, 29, 30].

Problem 6 *Is bisimulation equivalence finitely based over the language obtained by adding the empty process to BPA*?*

Problem 7 *Give an equational axiomatization of bisimulation equivalence over the language BPA$^*_\delta$ obtained by adding the constant δ to BPA*?*

Conway showed in [28] that any equational axiomatization for the language of regular expressions must contain an infinite number of equations in two or more variables. Does the same hold for equational axiomatizations of bisimulation equivalence over the language BPA$^*_\delta$?

3.2 Negative Results

Christos Papadimitriou once wrote in [65, Page 2] that "negative results are *the only possible* self-contained theoretical results", and I often like to cite him to "justify" my partiality towards negative results pertaining to the non-existence of finite equational axiomatizations for behavioural equivalences over process description languages. Amongst the extant such results, I would like to mention here

Result 7 (Moller [62, 63]) *Moller's proofs of non-finite axiomatizability for bisimulation equivalence over CCS and over PA without the left-merge.*

To the best of my knowledge, these were the first such results in process theory, and offered a powerful demonstration of the usefulness of proof theoretic techniques in proofs of non-finite axiomatizability for behavioural equivalences over process algebras. Indeed, most of the proofs of such negative results in process algebra I know of have used proof theoretic techniques (two exceptions are the menagerie of non-finite axiomatizability results over BPA^* for all of the behavioural congruences in between ready simulation and completed traces offered in [8]—that are based on Conway's results for regular expressions [28]—, and the toy result for trace and simulation equivalence over BPA with a singleton action set [5, 6]—that, despite its toy nature, has a rather non-trivial proof, if I may say so). Moller's proofs use a very neat unique decomposition result for processes. It is quite remarkable, albeit maybe not wholly unexpected, how useful decomposition results turn out to be in very different settings!

The interested reader will find further examples of non-finite axiomatizability results in, e.g., [9, 22, 74]. The last of these references is notable in that it deals with a language with finite-state recursive definitions.

Problem 8 *Are the left merge and communication merge operators necessary to obtain a finite axiomatization of strong bisimulation equivalence? Can one obtain a finite axiomatization by adding only* one *binary operator to the signature of CCS? In particular, does bisimulation equivalence admit a finite equational axiomatization over the language obtained by adding Hennessy's auxiliary operator from [44] to CCS?*[1]

Problem 9 *Give a model theoretic proof of Moller's theorem.*

Problem 10 *Can one prove that observation congruence is also not finitely based over CCS?*

Problem 11 *Can one come up with sufficient conditions of a reasonably general nature (be they syntactic or semantic) that ensure that bisimulation equivalence is finitely based?*

[1] As was recently pointed out to me by Jos Baeten and Rob van Glabbeek, it is certainly possible to obtain a finite axiomatization of (strong/weak/branching) bisimulation congruence by adding one ternary operator to the signature of CCS.

Problem 12 *Is it decidable whether a "finite process language" is finitely based modulo bisimulation equivalence? (This requires some sort of characterization of a process language.) See McKenzie's solution to Tarski's celebrated finite basis problem in [57].*

4 SOS Theory

Structural operational semantics [68] has played a key role in the development of the theory of process algebras. Moreover, it has been a key tool in the development of the meta-theory of process description languages by providing the technical framework for the generalization of results proven for many different specific process algebras to families of such languages—see [10] for a survey of these results.

Amongst the wealth of theorems that have been developed in this field of research, let me mention a result that was ahead of its time, and offered the blueprint for a series of developments that became very popular from the late 1980's onwards.

Result 8 (de Simone [75]) *Expressive completeness modulo FH-equivalence of Meije-SCCS with respect to the collection of operators that can be described using rules in de Simone's format.*

This result offers possibly the main example of a process algebra, viz. Meije [14], whose design was explicitly guided by a deep analysis of expressiveness considerations. Variants on de Simone's original result are reported in, e.g., [38, 50, 67, 83]. The use of FH-bisimulation—essentially a notion of bisimulation that can be defined directly over open terms, using derived SOS rules for contexts as *formal hypotheses*—in the aforementioned result of de Simone's also offered a sound, albeit incomplete, technique for establishing the validity of equations modulo bisimulation equivalence over de Simone languages. This work was the precursor of that on contexts as action transducers developed in, e.g., [54, 84]. I have the feeling that there is still life in the search for bisimulation-like proof techniques that apply to more generous formats of SOS transition rules for establishing validity of equations using "operational rules as transitions", and this leads me to formulate the following

Problem 13 *Devise variations on FH-bisimulation that apply to formats of operational rules like the tyft/tyxt format [42], and formats allowing for the use of predicates [85].*

In order to support compositional reasoning techniques, and indeed to be considered as a reasonable notion of equality, behavioural relations over process algebras ought to be preserved by the operators in the language under consideration. Establishing such congruence results for different languages and behavioural relations has been a major enterprise in process algebra research. The commonalities amongst many congruence results for bisimulation equivalence was highlighted by the following key result

Result 9 (Groote and Vaandrager [42]) *Bisimulation equivalence is preserved by operators whose operational semantics is given by rules in the tyft/tyxt format. Moreover, the largest congruence contained in completed trace equivalence with respect to that format is* 2*-nested simulation equivalence.*

Together with [23], this paper has generated a veritable industry of results on the meta-theory of SOS and process algebras. (See [10] for a mention of some of these achievements and pointers to the original literature.) The proof techniques used in these results were extremely ingenious, and have paved the way to many similar developments. Again, the role played by the modal characterizations of behavioural equivalences in the proof of the characterizations of the largest congruences is remarkable.

5 Expressiveness Results

One of the classic topics in the theory of computation that is taught in most Computer Science curricula is the Chomsky hierarchy of languages, and the connection between classes of languages and the simplest machines that recognize them. These results are prime examples of *expressiveness results*, and have their counterparts in process algebra. Indeed, the study of the expressiveness of process algebras has a rather long history, and its continuing role in the development of this field of research is witnessed, for instance, by a workshop that is entirely devoted to expressiveness in concurrency, and whose tenth edition took place in September 2003.

Naturally enough, all full blown process algebras are Turing complete, and results to this effect have been amongst the earliest expressiveness theorems in the field. However, as argued eloquently in, e.g., [83], there are other important, and perhaps more interesting, measures of the expressiveness of fragments of process algebras. One of them we have already met in Result 8, viz. the characterization of the collection of operators that can be denoted in a given language modulo bisimulation equivalence. Another measure of the expressiveness of a language, and the one I would like to focus on here, is the study of the collection of processes that can be described using it. Early results in this area that still stand out for the very ingenious proofs needed to establish them are:

Result 10 (Bergstra and Klop [18]) *Bergstra and Klop's theorems to the effect that, modulo strong bisimulation equivalence, there is*

1. *a process, viz. a stack or a counter over a finite data type, that is recursively definable over BPA but not over the regular fragment of CCS,*
2. *a process, viz. a bag over a finite data type with cardinality at least two, that is finitely definable over PA but not over BPA, and*
3. *a process over a binary action alphabet that is finitely definable over ACP but not over PA.*

Further undefinability results of this type may be found, for example, in [15, 16, 83]. A general Chomsky-like hierarchy of process languages has been first proposed by Moller in [64], and further refined by Mayr in [56].

Problem 14 *Establish many more expressiveness results that offer insight on the power of different process description languages, and of the features they are based on.*

6 Connections with Automata and Formal Language Theory

The behaviour of processes can be described by labelled transition systems, which, in some form or the other, are also one of the objects of study in the classic theory of automata and formal languages. Moreover, the usefulness of algebraic techniques has long been recognized in automata theory, a field that arguably uses much more sophisticated notions and results from algebra than our own. (See, e.g., the monograph [81] for a survey of topics in algebraic automata theory, and a taste of the algebraic results used in this line of research.) It is therefore not overly surprising that there are connections between these fields of theoretical computer science. To highlight the connections, and the differences, between automata theory and process algebra, I would like to close this note by mentioning two results pertaining to decidability of behavioural equivalences over classes of infinite state processes and to axiomatic issues related to the theory of languages.

A classic result in automata theory is the fact that language equivalence for context-free languages is undecidable (see, e.g., [48] for a textbook presentation). That bisimulation equivalence for context-free processes enjoys different decidability properties was first highlighted by

Result 11 (Baeten, Bergstra and Klop [17]) *Baeten, Bergstra and Klop's theorem to the effect that strong bisimulation equivalence is decidable for context-free processes without redundant nonterminals.*

This surprising result spurred a flurry of research activity studying the decidability properties of behavioural equivalences in the linear time-branching time spectrum over classes of infinite state processes, and mapping the territory between decidable and undecidable equivalence problems. (See the references [27, 76] for excellent overviews of the wealth of results achieved in this line of study.)

Techniques from the field of process algebra have also been used with remarkable effect to improve upon, or simplify, extant results in automata and formal language theory. I limit myself to mentioning two examples of these applications here.

Hirshfeld, Jerrum and Moller have shown in [47] that strong bisimulation equivalence is decidable in polynomial time over normed BPA. Since strong bisimulation equivalence coincides with language equivalence over deterministic processes, this result dramatically improves upon the doubly exponential upper bound for the equivalence problem for deterministic context-free grammars established by Korenjak and Hopcroft in [51].

A mixture of techniques developed in concurrency and language theory have been employed by Stirling in [79], where he offered a simplification of Sénizergues' remarkable proof from [73] of the decidability of the equivalence problem for deterministic pushdown automata. (Sénizergues was awarded the 2002 Gödel prize for this achievement.) In subsequent work presented in [80], Stirling established a primitive recursive upper bound for this problem.

Problem 15 *Is weak bisimulation equivalence decidable over BPA and BPP?*

Problem 16 *Is strong bisimulation equivalence decidable over the language PA? Hirshfeld and Jerrum have shown in [46] that the answer is positive for normed PA.*

Given the role that axiomatic results have played in this note, it is perhaps fitting for me to bring it to a close by mentioning

Result 12 (Tschantz [82]) *Tschantz's axiomatization of the theory of languages over concatenation and shuffle.*

Where is the theory of processes there, you may ask? Well, to prove this result Steven Tschantz (a pure mathematician at Vanderbilt University) essentially rediscovered the concept of pomset [69]—a model of concurrency based on partial orders whose algebraic aspects have been investigated by Gischer in [36]—, and his proof can be phrased in terms of what some of us would call ST-trace equivalence [40].

Techniques from classic formal language theory have been used in process theory—see, e.g., the Conway type arguments in [8, 36] to name but two examples. I expect that more interaction between the two fields would lead to a useful synergy.

Problem 17 *Can process algebraic techniques be brought to bear on scheduling problems, and cost optimality problems? These are currently tackled purely using automata, and that's great, but in formal language theory regular expressions with multiplicities over semirings play a role, they allow one to define measures of non-determinism in automata etc. Can we contribute?*

The mention of this last result brings me back to the first research meeting I ever attended, viz. the workshop on Concurrency and Compositionality organized in 1988 in Königswinter by van Glabbeek, Goltz and Olderog. It is there that I heard Pratt ask the question that led to Result 12, and met, for the first time, many of the players that have shaped the field of process algebra. I hope that, by writing this note, I have not done a disservice to their work and to the field.

Acknowledgments. I thank all of my co-authors and colleagues who have taught me the little I know about this fascinating field of research. Jos Baeten, Wan Fokkink and Jiří Srba offered incisive comments on previous drafts of this paper. *The opinions expressed in this note, and any infelicity herein, are solely my responsibility.*

References

[1] S. Abramsky, *Observation equivalence as a testing equivalence*, Theoretical Comput. Sci., 53 (1987), pp. 225–241.

[2] ——, *A domain equation for bisimulation*, Information and Computation, 92 (1991), pp. 161–218.

[3] L. Aceto, *On "Axiomatising Finite Concurrent Processes"*, SIAM J. Comput., 23 (1994), pp. 852–863.

[4] L. Aceto, B. Bloom, and F. Vaandrager, *Turning SOS rules into equations*, Information and Computation, 111 (1994), pp. 1–52.

[5] L. Aceto, Z. Ésik, and A. Ingólfsdóttir, *On the two-variable fragment of the equational theory of the max-sum algebra of the natural numbers*, in Proceedings of the 17th International Symposium on Theoretical Aspects of Computer Science, STACS 2000 (Lille), H. Reichel and S. Tison, eds., vol. 1770 of Lecture Notes in Computer Science, Springer-Verlag, Feb. 2000, pp. 267–278.

[6] ——, *The max-plus algebra of the natural numbers has no finite equational basis*, Theoretical Comput. Sci., 293 (2003), pp. 169–188.

[7] L. Aceto and W. Fokkink, *An equational axiomatization for multi-exit iteration*, Information and Computation, 137 (1997), pp. 121–158.

[8] L. Aceto, W. Fokkink, and A. Ingólfsdóttir, *A menagerie of non-finitely based process semantics over BPA*—from ready simulation to completed traces*, Mathematical Structures in Computer Science, 8 (1998), pp. 193–230.

[9] L. Aceto, W. Fokkink, R. van Glabbeek, and A. Ingólfsdóttir, *Nested semantics over finite trees are equationally hard*, Technical Report RS–03–27, BRICS, Department of Computer Science, Aalborg University, August 2003.

[10] L. Aceto, W. Fokkink, and C. Verhoef, *Structural operational semantics*, in Handbook of Process Algebra, North-Holland, 2001, pp. 197–292.

[11] L. Aceto and M. Hennessy, *Termination, deadlock and divergence*, J. Assoc. Comput. Mach., 39 (1992), pp. 147–187.

[12] L. Aceto and A. Jeffrey, *A complete axiomatization of timed bisimulation for a class of timed regular behaviours*, Theoretical Comput. Sci., 152 (1995), pp. 251–268.

[13] P. Auster, *The New York Trilogy—The Locked Room*, Faber and Faber, 1988.

[14] D. Austry and G. Boudol, *Algèbre de processus et synchronisations*, Theoretical Comput. Sci., 30 (1984), pp. 91–131.

[15] J. Baeten and J. Bergstra, *Global renaming operators in concrete process algebra*, Information and Computation, 78 (1988), pp. 205–245.

[16] J. Baeten, J. Bergstra, and J. W. Klop, *On the consistency of Koomen's fair abstraction rule*, Theoretical Comput. Sci., 51 (1987), pp. 129–176.

[17] J. C. Baeten, J. Bergstra, and J. W. Klop, *Decidability of bisimulation equivalence for processes generating context-free languages*, J. Assoc. Comput. Mach., 40 (1993), pp. 653–682.

[18] J. Bergstra and J. W. Klop, *The algebra of recursively defined processes and the algebra of regular processes*, in Proceedings 11^{th} ICALP, *Antwerpen*, J. Paredaens, ed., vol. 172 of Lecture Notes in Computer Science, Springer-Verlag, 1984, pp. 82–95.

[19] ——, *Process algebra for synchronous communication*, Information and Computation, 60 (1984), pp. 109–137.

[20] ——, *A complete inference system for regular processes with silent moves*, in Proceedings Logic Colloquium 1986, F. Drake and J. Truss, eds., Hull, 1988, North-Holland, pp. 21–81.

[21] J. Bergstra, A. Ponse, and S. Smolka, eds., *Handbook of Process Algebra*, North-Holland, 2001.

[22] S. Blom, W. Fokkink, and S. Nain, *On the axiomatizability of ready traces, ready simulation and failure traces*, in Proceedings 30th Colloquium on Automata, Languages and Programming—ICALP'03, Eindhoven, Lecture Notes in Computer Science, Springer-Verlag, 2003. To appear.

[23] B. Bloom, S. Istrail, and A. Meyer, *Bisimulation can't be traced*, J. Assoc. Comput. Mach., 42 (1995), pp. 232–268.

[24] S. L. Bloom and Z. Ésik, *Iteration Theories: The Equational Logic of Iterative Processes*, EATCS Monographs on Theoretical Computer Science (W. Brauer, G. Rozenberg and A. Salomaa eds.), Springer-Verlag, 1993.

[25] D. J. B. Bosscher, *Grammars Modulo Bisimulation*, PhD thesis, CWI, Amsterdam, 1997.

[26] G. Boudol and I. Castellani, *Concurrency and atomicity*, Theoretical Comput. Sci., 59 (1988), pp. 25–84.

[27] O. Burkart, D. Caucal, F. Moller, and B. Steffen, *Verification on infinite structures*, in Handbook of process algebra, North-Holland, Amsterdam, 2001, pp. 545–623.

[28] J. H. Conway, *Regular Algebra and Finite Machines*, Mathematics Series (R. Brown and J. De Wet eds.), Chapman and Hall, London, United Kingdom, 1971.

[29] F. Corradini, R. De Nicola, and A. Labella, *A finite axiomatization of nondeterministic regular expressions*, Theor. Inform. Appl., 33 (1999), pp. 447–465. Fixed points in computer science (Brno, 1998).

[30] ——, *An equational axiomatization of bisimulation over regular expressions*, J. Logic Comput., 12 (2002), pp. 301–320. Fixed points in computer science, 2000 (Paris).

[31] P. Darondeau and P. Degano, *Causal trees*, Publication Interne No. 442, IRISA, Rennes Cedex, France, 1988. An extended abstract appeared in: Proceedings ICALP 89, Stresa (G. Ausiello, M. Dezani-Ciancaglini & S. Ronchi Della Rocca), LNCS 372, Springer-Verlag, pp. 234–248.

[32] R. De Nicola and A. Labella, *Nondeterministic regular expressions as solutions of equational systems.* Proceedings of FICS 2000 (Paris), 2000.

[33] R. De Nicola and F. W. Vaandrager, *Three logics for branching bisimulation*, J. Assoc. Comput. Mach., 32 (1995), pp. 458–487.

[34] W. Fokkink, *On the completeness of the equations for the Kleene star in bisimulation*, in Algebraic Methodology and Software Technology: 5th International Conference, AMAST '96, Munich, Germany, M. Wirsing and M. Nivat, eds., vol. 1101 of Lecture Notes in Computer Science, Springer-Verlag, July 1996, pp. 180–194.

[35] W. Fokkink and H. Zantema, *Basic process algebra with iteration: Completeness of its equational axioms*, Computer Journal, 37 (1994), pp. 259–267.

[36] J. L. Gischer, *The equational theory of pomsets*, Theoretical Comput. Sci., 61 (1988), pp. 199–224.

[37] R. van Glabbeek, *A complete axiomatization for branching bisimulation congruence of finite-state behaviours*, in Mathematical Foundations of Computer Science 1993, Gdansk, Poland, A. Borzyszkowski and S. Sokołowski, eds., vol. 711 of Lecture Notes in Computer Science, Springer-Verlag, 1993, pp. 473–484.

[38] ———, *On the expressiveness of ACP*, in Proceedings 1st Workshop on the Algebra of Communicating Processes, Utrecht, The Netherlands, A. Ponse, C. Verhoef, and B. van Vlijmen, eds., Workshops in Computing, Springer-Verlag, 1995, pp. 188–217.

[39] ———, *The linear time–branching time spectrum. I. The semantics of concrete, sequential processes*, in Handbook of Process Algebra, North-Holland, Amsterdam, 2001, pp. 3–99.

[40] R. van Glabbeek and F. Vaandrager, *Petri net models for algebraic theories of concurrency*, in Proceedings PARLE conference, Eindhoven, Vol. II (Parallel Languages), J. d. Bakker, A. Nijman, and P. Treleaven, eds., vol. 259 of Lecture Notes in Computer Science, Springer-Verlag, 1987, pp. 224–242.

[41] J. F. Groote, *Transition system specifications with negative premises*, Theoretical Comput. Sci., 118 (1993), pp. 263–299.

[42] J. F. Groote and F. W. Vaandrager, *Structured operational semantics and bisimulation as a congruence*, Information and Computation, 100 (1992), pp. 202–260.

[43] G. H. Hardy, *A Mathematician's Apology*, Canto, Cambridge University Press, Cambridge, 1992. With a foreword by C. P. Snow, Reprint of the 1967 edition.

[44] M. Hennessy, *Axiomatising finite concurrent processes*, SIAM J. Comput., 17 (1988), pp. 997–1017.

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

[46] Y. Hirshfeld and M. Jerrum, *Bisimulation equivalence is decidable for normed process algebra (extended abstract)*, in Automata, languages and programming (Prague, 1999), vol. 1644 of Lecture Notes in Comput. Sci., Springer, Berlin, 1999, pp. 412–421.

[47] Y. Hirshfeld, M. Jerrum, and F. Moller, *A polynomial algorithm for deciding bisimilarity of normed context-free processes*, Theoretical Comput. Sci., 158 (1996), pp. 143–159.

[48] J. Hopcroft and J. Ullman, *Introduction to Automata Theory, Languages and Computation*, Addison-Wesley, 1979.

[49] A. Ingólfsdóttir, J. C. Godskesen, and M. Zeeberg, *Fra Hennessy-Milner logik til CCS-processer*, Master's thesis, Department of Computer Science, Aalborg University, 1987. In Danish.

[50] A. Jeffrey, *CSP is completely expressive*, Computer Science Technical Report 92:02, School of Cognitive and Computing Sciences, University of Sussex, 1992.

[51] A. J. Korenjak and J. E. Hopcroft, *Simple deterministic languages*, in Conference Record of Seventh Annual Symposium on Switching and Automata Theory, Berkeley, California, 26–28 October 1966, IEEE, 1966, pp. 36–46.

[52] H. Korver, *Computing distinguishing formulas for branching bisimulation*, in Proceedings of the Third Workshop on Computer Aided Verification, Aalborg, Denmark, July 1991, K. Larsen and A. Skou, eds., vol. 575 of Lecture Notes in Computer Science, Springer-Verlag, 1992, pp. 13–23.

[53] K. G. Larsen and A. Skou, *Bisimulation through probabilistic testing*, Information and Computation, 94 (1991), pp. 1–28.

[54] K. G. Larsen and L. Xinxin, *Compositionality through an operational semantics of contexts*, Journal of Logic and Computation, 1 (1991), pp. 761–795.

[55] T. Margaria and B. Steffen, *Distinguishing formulas for free*, in Proc. EDAC–EUROASIC'93: IEEE European Design Automation Conference, Paris (France), IEEE Computer Society Press, February 1993.

[56] R. Mayr, *Process rewrite systems*, Information and Computation, 156 (2000), pp. 264–286.

[57] R. McKenzie, *Tarski's finite basis problem is undecidable*, Internat. J. Algebra Comput., 6 (1996), pp. 49–104.

[58] R. Milner, *A modal characterisation of observable machine behaviour*, in CAAP '81: Trees in Algebra and Programming, 6th Colloquium, E. Astesiano and C. Böhm, eds., vol. 112 of Lecture Notes in Computer Science, Springer-Verlag, 1981, pp. 25–34.

[59] ——, *A complete inference system for a class of regular behaviours*, J. Comput. System Sci., 28 (1984), pp. 439–466.

[60] ——, *Communication and Concurrency*, Prentice-Hall International, Englewood Cliffs, 1989.

[61] ——, *A complete axiomatisation for observational congruence of finite-state behaviors*, Information and Computation, 81 (1989), pp. 227–247.

[62] F. Moller, *The importance of the left merge operator in process algebras*, in Proceedings 17^{th} ICALP, Warwick, M. Paterson, ed., vol. 443 of Lecture Notes in Computer Science, Springer-Verlag, July 1990, pp. 752–764.

[63] ——, *The nonexistence of finite axiomatisations for CCS congruences*, in Proceedings 5^{th} Annual Symposium on Logic in Computer Science, Philadelphia, USA, IEEE Computer Society Press, 1990, pp. 142–153.

[64] ——, *Infinite results*, in Proceedings of CONCUR '96, Concurrency Theory, 7th International Conference, Pisa, Italy, August 26–29, 1996, U. Montanari and V. Sassone, eds., vol. 1119 of Lecture Notes in Computer Science, Springer-Verlag, 1996, pp. 195–216.

[65] C. H. Papadimitriou, *Database metatheory: Asking the big queries*, in Proceedings of the Fourteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, May 22-25, 1995, San Jose, California, ACM Press, 1995, pp. 1–10.

[66] D. Park, *Concurrency and automata on infinite sequences*, in 5^{th} GI Conference, Karlsruhe, Germany, P. Deussen, ed., vol. 104 of Lecture Notes in Computer Science, Springer-Verlag, 1981, pp. 167–183.

[67] J. Parrow, *The expressive power of parallelism*, Future Generation Computer Systems, 6 (1990), pp. 271–285.

[68] G. Plotkin, *A structural approach to operational semantics*, Report DAIMI FN-19, Computer Science Department, Aarhus University, 1981.

[69] V. Pratt, *Modeling concurrency with partial orders*, International Journal of Parallel Programming, 15 (1986), pp. 33–71.

[70] A. M. Rabinovich, *A complete axiomatisation for trace congruence of finite state behaviors*, in Mathematical Foundations of Programming Semantics, 9th International Conference, New Orleans, LA, USA, April 7-10, 1993, Proceedings, S. D. Brookes, M. G. Main, A. Melton, M. W. Mislove, and D. A. Schmidt, eds., vol. 802 of Lecture Notes in Computer Science, Springer-Verlag, 1994, pp. 530–543.

[71] G.-C. Rota, *Ten rules for the survival of a mathematics department*, in Indiscrete Thoughts, F. Palombi, ed., Birkhäuser Boston, Inc., Boston, MA, 1996. With forewords by Reuben Hersh and Robert Sokolowski. A version of this article is available from http://www.rota.org/hotair/rules.html.

[72] A. Salomaa, *Two complete axiom systems for the algebra of regular events*, J. Assoc. Comput. Mach., 13 (1966), pp. 158–169.

[73] G. Sénizergues, *$L(A) = L(B)$? decidability results from complete formal systems*, Theoretical Comput. Sci., 251 (2001), pp. 1–166.

[74] P. Sewell, *Nonaxiomatisability of equivalences over finite state processes*, Annals of Pure and Applied Logic, 90 (1997), pp. 163–191.

[75] R. d. Simone, *Higher-level synchronising devices in* Meije–*SCCS*, Theoretical Comput. Sci., 37 (1985), pp. 245–267.

[76] J. Srba, *Roadmap of infinite results*, December 2002. Available from `http://www.brics.dk/~srba/roadmap/roadmap.pdf`.

[77] B. Steffen and A. Ingólfsdóttir, *Characteristic formulae for processes with divergence*, Information and Computation, 110 (1994), pp. 149–163.

[78] C. Stirling, *Modal logics for communicating systems*, Theoretical Comput. Sci., 49 (1987), pp. 311–347.

[79] ——, *Decidability of DPDA equivalence*, Theoretical Comput. Sci., 255 (2001), pp. 1–31.

[80] ——, *Deciding DPDA equivalence is primitive recursive*, in Proceedings of the 29th International Colloquium on Automata, Languages and Programming, ICALP 2002, Malaga, Spain, July 8-13, 2002, P. Widmayer, F. T. Ruiz, R. Morales, M. Hennessy, S. Eidenbenz, and R. Conejo, eds., vol. 2380 of Lecture Notes in Computer Science, Springer-Verlag, 2002, pp. 821–832.

[81] H. Straubing, *Finite Automata, Formal Logic, and Circuit Complexity*, vol. 11 of Progress in Theoretical Computer Science, Birkäuser, 1994.

[82] S. T. Tschantz, *Languages under concatenation and shuffling*, Mathematical Structures in Computer Science, 4 (1994), pp. 505–511.

[83] F. Vaandrager, *Expressiveness results for process algebras*, in Proceedings REX Workshop on Semantics: Foundations and Applications, Beekbergen, The Netherlands, June 1992, J. de Bakker, W. d. Roever, and G. Rozenberg, eds., vol. 666 of Lecture Notes in Computer Science, Springer-Verlag, 1993, pp. 609–638.

[84] F. Vaandrager and N. Lynch, *Action transducers and timed automata*, in Proceedings CONCUR 92, Stony Brook, NY, USA, W. R. Cleaveland, ed., vol. 630 of Lecture Notes in Computer Science, Springer-Verlag, 1992, pp. 436–455.

[85] C. Verhoef, *A congruence theorem for structured operational semantics with predicates and negative premises*, Nordic Journal of Computing, 2 (1995), pp. 274–302.

[86] D. Walker, *Bisimulation and divergence*, Information and Computation, 85 (1990), pp. 202–241.