A nopreprint on Algebraic Algorithmics: Paraconsistency as an Afterthought

Jan A. Bergstra
Minstroom Research,* Utrecht

January 22, 2015

Abstract

Algebraic Algorithmics, a phase taken from G.E. Tseitlin [31], is given a specific interpretation for the line of work in the tradition of the program algebra of [7] and thread algebra [8]. An application to algebraic algorithmics of preservationist paraconsistent reasoning in the style of "chunk and permeate" is suggested and discussed.

In the first appendix "nopreprint" is coined as a tag for new a publication category, and a rationale for its use is given. In a second appendix some rationale is provided for the affiliation from which the paper is written and `viXra`-ed.

Contents

*Minstroom Research BV, Utrecht, The Netherlands (hereafter called MRbv), KvK nr. 59560347. Author's email address: `info@minstroomreserch.org,janaldertb@gmail.com`. In Appendix A extensive and detailed statements are included concerning copyright protection of this document and about its formal status, and in addition some general information is provided about its objectives.

1 Introduction

I will first baptise a certain programming style, which admittedly may merely be useful for limited research activities of a fairly theoretical nature, as EXIP (exclusively imperative programming). The notation of PGA program algebra is supposedly well-suited for EXIP style programming. EXIP as well as its metatheory which has been developed under the name "program algebra" is subsumed under algebraic algorithmics.

The core of the paper consists of an attempt to clarify the explanation of the notion of an instruction sequence in the context of program algebra by means of an application of paraconsistent reasoning. There is little novelty in this explanation because it merely exploits a similarity between what is well-known about fractions, and what may be stated about instruction sequences.

In the case of fractions the use of paraconsistency was established in [3], and a comparable use may be made in the context of instruction sequences.

1.1 Exclusively imperative programming (EXIP)

Program algebra, understood as a theory about algebras of instruction sequences, provides a theoretical framework for what I propose to call exclusively imperative programming (EXIP). The idea is that the sequential order of events during the effectuation of an instruction sequence is completely fixed by the structure of that instruction sequences in combination with the replies that the computing environment returns during effectuation.

In addition each effectuation of an instruction has at most two objectives (which may occur in combination) at the same time: either to transform a state outside the instruction sequence or to modify the control state (program counter) within the instruction sequence.

EXIP provides no abstraction mechanisms that allow to define functions in any other way than by computation from elementary operational building blocks, the so-called primitive instructions. EXIP programs in program algebra style are notoriously unreadable while at the same time such programs admit semantic understanding and theoretical analysis more easily than programs written in more advanced program notations.

1.2 Algebraic algorithmics

Instruction sequences in program algebra syntax provide a meaningful program notation for EXIP programming. The instruction sequence notation is useful for denoting simple algorithms, and for that reason its algebraic metatheory may be subsumed under the theme of algebraic algorithmics, thus following the original use of that phrase by Tseitlin in [31].

A significant difference with algebraic algorithmics as discussed in [31] and e.g. in [22] lies in the fact that those papers provide an algebraic formulation of classical schematology where a conditional statement is a three place operator whereas in program algebra a conditional statement is represented by a single instruction which incorporates the test and has two exits. Although, as a consequence of this design choice, the algebra of instruction sequences may considered to be rather simple in comparison to the algebras discussed in [31], I firmly believe that the program algebra approach is algebraic in nature as well.

Program algebra and instruction sequence theory have a behavioural chapter named thread algebra which I consider to be subsumed under algebraic algorithmics as well. Some papers that follow an algebraic approach in the setting of program algebra and thread algebra are: [4, 5, 13, 14, 15]. Further papers on thread algebra are [8, 9]. For a recent account of the connection between instruction sequences and threads I refer to [11].

One might object that algebraic algorithmics takes inspiration from universal algebra rather than from algebra. In my view universal algebra is a part of algebra rather than the other way around. These matters are far from clear,

however. Several authors have worked towards an embedding of the algebra of fields in a setting of universal algebra. I may mention: [18, 27, 20, 21, 25, 16, 12]. Even if one assumes that universal algebra is subsumed under algebra and that these works succeed in closing gaps between the theory of fields and universal algebra, it is possible that one judges these works as moving the subject matter further away from the core of algebraic approaches. It is not clear to me to what extent these judgements are matters of taste.

Assuming that work in algebraic algorithmics leads to the development of new algebras of programs which are suitable for the representation and analysis of algorithms it must be kept in mind that developing logics of algorithms on the basis of such developments still may involve design choices with ample degrees of freedom. The idea that algebras have their own logic so to say is somewhat surprising, but at the same time it is reassuring and intriguing, because it allows a flexible use of an algebra. That vision of flexibility has recently been put forward in [28].

2 Two views on instruction sequences

Assuming a repertoire R_I of instructions, an instruction sequence (given that repertoire) is a sequence of elements of R_I. I will write IS for the set of finite and infinite instruction sequences.

We assume that there is an equality predicate on R_I, cast as a two place function to $\mathbb{B}_a$, that is Booleans extended with a third value named a. The sequence may be finite or infinite. Instruction sequence concatenation is denoted with a two place infix operator written "$-;-$", thus following the famous semicolon which occurs in many imperative program notations.[1]

Assuming that $-;-$ is associative brackets can (but need not) be deleted and finite instruction sequences can be written as $u_0;..;u_{n-1}$ for $u_i \in R_I$. Infinite sequences may be repeating in which case there exists a finitary representation $u_0;..;u_{n-1};(u_n;..;u_{n+k-1})^\omega$ for $u_i \in R_I$ (using the notation of [7]).

2.1 A contradiction?

In [7] it was taken as a definition of a computer program that it is a sequence of instructions. In [13, 14] that definition has been worked out in more detail for the simplest classes of programs. However, one may hold that rather than constituting an instruction sequence $u_0;..;u_{n-1};(u_n;..;u_{n+k-1})^\omega$ is an expression denoting an instruction sequence. It unavoidably follows from defining a program as an instruction sequence that, viewed as an expression rather than as its meaning, $u_0;..;u_{n-1};(u_n;..;u_{n+k-1})^\omega$ would not be a program. I will write ISE for the set of closed instruction sequence expressions.

In [7] it was proposed to consider the (infinite) instruction sequence obtained by unfolding the expression $u_0;..;u_{n-1};(u_n;..;u_{n+k-1})^\omega$ and to baptize

[1] Following [7] on that matter, $-;-$ can be referred to as text sequential composition.

that entity a program object. That is a mathematical object representing the program qua instruction sequence.

But at the same time the common intuition implies that a person can write a program. Writing that program expression is always possible, while writing the program object is in general not (unless it happens to be finite, the case of so-called straight line programs). In addition the program (instruction sequence) expression can be assigned a size (for instance $n + k$ in this case), while the program object cannot be assigned a size other than it being infinite.

2.2 A remedy: repetition instructions

In [14] a remedy to these difficulties was proposed by introducing a so-called repetition instruction that produces the same effect as the repetition operator. In $u_0; .. ; u_{n-1}; u_n; .. ; u_{n+k-1}; \backslash\backslash\#k + 1$ the effect of $\backslash\backslash\#k + 1$ is as if the last $k+1$ instructions including $\backslash\backslash\#k+1$ are placed at the tail of the program thus replacing $\backslash\backslash\#k + 1$. This is expressed in the following semantic equations:

$$| X^2 |=| X; X |$$

and

$$| u_0; .. ; u_{n-1}; u_n; .. ; u_{n+k-1}; \backslash\backslash\#k{+}1 |=| u_0; .. ; u_{n-1}; (u_n; .. ; u_{n+k-1})^2; \backslash\backslash\#k{+}1 | .$$

This solution is consistent with a program being an instruction sequence, but it gives rise to the following issue: Is $Y = u_0; u_1; u_2; \backslash\backslash\#5$ a program? It clearly cannot be unfolded by means of the given semantic equations. Now for any instruction v in R_I, $v; Y$ is a program. This observation suggests that $\lambda v_{\in R_I}(v; Y)$ might be (or might denote) the meaning of Y.

2.3 Typechecking as an alternative to lambda abstraction

Alternatively one might introduce type checking and claim that type checking Y produces a negative outcome, whence not every instruction sequences qualifies as a program. A somewhat unfortunate consequence of that view is that subsequences of programs need not constitute programs.

3 Paraconsistency: an afterthought that solves the dilemma?

The dilemma sketched above is that on the one hand a program is an instruction sequence, that is a possibly infinite program object, while on the other hand a program has to be finite which points in the direction of an instruction sequence expression rather than of an instruction sequence proper. In addition to this dilemma five observations can be made:

1. Program objects are more abstract than program expressions.

2. For that reason program objects are closer to the mathematical meaning of a program, irrespective of how that meaning has been defined.

3. Manipulation and transformation during construction and effectuation (see [1]) of programs takes place at the level of program expressions.

4. At the expression level (alternatively: syntactic level) various notions of size are available (the number of instructions of the repeating part ($\mathtt{size}_{\mathtt{rp}}(-)$), and the number of instructions of the non repeating part ($\mathtt{size}_{\mathtt{nrp}}(-)$)), which are not (or only indirectly) available at the level of (unfolded) execution sequences.[2]

5. The same program object may be denoted by different instruction sequence expressions, a unique one of those being the simplest, however.

6. Equating programs with equal program objects, in the presence of size functions leads to an inconsistency, for instance:
$2 = \mathtt{size}_{\mathtt{rp}}(a;(b;c)^\omega) = \mathtt{size}_{\mathtt{rp}}(a;(b;c;b;c)^\omega) = 4$.

3.1 Concept image patterns

Conceptual states of affairs that might be inconsistent have been described by Tall and Vinner in [30] and have been termed concept images. Concept images are held by subjects about themes. For an inconsistent concept image it is a matter of maturation for the subject involved to resolve the problems posed by an inconsistency. Hopefully that maturation is promoted by well structured educational activities with the effect of mentally redesigning inconsistent concept images into consistent ones. Consistency is achieved by choosing preferred options.

Instead of resolving or refining an inconsistent concept image into a consistent one, it may be useful to recognise a pattern within the inconsistent concept image and to use paraconsistent logic to deal with the matter without making any premature choices.

3.1.1 The fraction pattern

A similar state of affairs has been found in [3] concerning the seemingly unproblematic notion of a fraction. The five observation made above match what I will call the fraction pattern:

- Rational numbers (objects denoted by fraction expressions) are more abstract than fraction expressions.
- Rational numbers are closer to mathematics than fraction expressionss.
- Calculation involves the manipulation and transformation of fraction expressions.

[2] The length of the repeating part of $a;(b;c;b;c)^\omega$ is 4, a number that cannot be derived from its unfolding to an infinite instruction sequence.

- From a fraction expression one may extract numerator (`num(-)`) and denominator (`denom(-)`), both can be viewed as a measurement of size or complexity of the faction.
- Different fractions (fraction expressions) denote the same rational number (e.g. $1/2 = 2/4$), fraction expressions can be simplified in a unique manner.
- Equating fraction expressions with the same meaning in the presence of `num(-)` leads to an inconsistency: $1 = \texttt{num}(1/2) = \texttt{num}(2/4) = 2$.

The fraction pattern arises in cases where one and the same kind of object is perceived at two levels of abstraction "at the same time". The fraction pattern indicates a setting where one has permanent difficulties to find a stable meaning for the concept of a fraction: is it an expression or is it the rational number denoted by that expression. In [3] it is proposed to solve this dilemma by viewing it as an example of paraconsistent reasoning, in particular of an application of a reasoning pattern termed "chunck and permeate" (C&P) in [19].

The fraction pattern is an instance of a concept image pattern. Solving the fraction pattern by way of paraconsistency allows one to switch back and forth between both views. One may imagine a person who oscillates between fraction as term and fraction as rational on a regular basis, say 10 times per second. When communicating with another person it will always be necessary to determine from what state the other person constructed his/her words. Both persons know that for each of their concept image patterns which have been solved by means of C&P style paraconsistency there will be a need to ensure synchronisation with another agent. This is something a person can do.

3.1.2 Paraconsistency needed?

Of course one may choose and try to fix a meaning for the term fraction for once and for all. The problem with that is that both meanings go hand in hand. One works with fraction expressions because one wants to deal with rational numbers. The term fraction denotes in an ambiguous manner both fraction expression (1/2 as a term) and fraction meaning (1/2 as a rational number). Combining both meanings is inconsistent, choosing between both meanings is difficult.[3] C&P style reasoning provides for a logical bookkeeping that avoids the occurrence of inconsistencies like $1 = \texttt{num}(1/2) = \texttt{num}(2/4) = 2$. A consequence of this approach is no less than that it must be considered plausible that some form of paraconsistent reasoning lies at the basis of school arithmetic.

3.2 The instruction sequence pattern

The notion of an instruction sequence features a similar pattern to that of a fraction, with instruction sequence proper constituting an abstract perspective on

[3] In [26] an attempt is made to be systematic about choosing rational number as the dominant meaning of fraction. In [24] a wide but inconsistent concept image for fractions is sketched and the topic is said to be complex and no disambiguating choice is made or suggested.

instruction sequence expression. The situation is so similar to that of fractions that a similar conclusion may be drawn. Rather than to choose one of both meanings and to restrict what may be validly written or said, it is plausible to accept that "instruction sequence" oscillates between at least two meanings at different levels of abstraction and to accept in addition that C&P style paraconsistent reasoning appears to provide a pragmatic way out of otherwise unpleasant inconsistencies.

3.2.1 Source, permeate, and target

Following the C&P format of paraconsistent reasoning a source theory is determined first. Then a part of that theory is determined which will be permeated to the target, for which some independent axioms have been fixed in advance. Paraconsistent reasoning may combine the permeated information with the additional information for the target context. Adding all logical known facts about the source setting to the additional axioms for the target setting leads to an inconsistency, which is prevented by filtering out from the theory of the source all non-permeated facts.

In [3] a result of combined reasoning is presented which produces a conclusion in two stages, the first stage consisting of reasoning in the source theory, with conclusions that are permeated (filtered) to the source level where reasoning proceeds from a somewhat different point of departure. Source and target alone don't suffice to obtain this conclusion. The challenge is to identify a source and a target and to find such a plausible result of combined reasoning in the case of instruction sequences. I claim that this observation is straightforward.

3.2.2 Source setting

The source structure of instruction sequence expressions must be augmented with a structure $\mathbb{B}_a$ for a three-valued logic with constants named t, f, and a. Further predicates on instruction sequence expressions may be reprersented by functions into these booleans. At this stage we need the notion of a canonical form for an instruction sequence. That is an instruction sequence expression with at most one occurrence of iteration, and with a shortest possible non-repeating part and a shortest possible repeating part (for expressing the same instruction sequence, that is finite or infinite sequence of instructions).

Further we introduce a function CF which produces t on closed canonical forms forms, and f on all other closed terms. Moreover we introduce a function $\mathsf{EQ}_{\mathsf{syn}}$: $\mathsf{ISE} \times \mathsf{ISE} \to B_a$ which will work as follows: on equal terms it produces t, on syntactically different terms it produces f. We assume the existence of an algebraic specification of ISE equipped with these additional functions. This is doable by means of a finite algebraic specification with the help of auxiliary functions. Following [17] such a specification can be found in such a way that a complete term rewriting system is obtained.

In addition the source contains a function EQ: $\mathsf{ISE} \times \mathsf{ISE} \to B_a$ for which an

axiom scheme is as follows: with r and s ranging over closed ISE terms.

$$\mathsf{CF}(r) = t \to \mathsf{EQ}(r, r) = t$$

$$\mathsf{CF}(r) = t \wedge \mathsf{CF}(s) = t \wedge \mathsf{EQ}_{\mathsf{syn}}(r, s) = f \to \mathsf{EQ}(r, s) = f.$$

$$\mathsf{CF}(r) = t \wedge \mathsf{CF}(s) = t \wedge \mathsf{EQ}_{\mathsf{syn}}(r, s) = t \to \mathsf{EQ}(r, s) = t.$$

$$\mathsf{CF}(r) = f \vee \mathsf{CF}(s) = f \to \mathsf{EQ}(r, s) = a.$$

Together with the mentioned specification these axioms constitute the source for the C&P approach in this case.

We may thus assume that the source theory proves all valid equations of the form $\mathsf{EQ}(r, s) = t$ and $\mathsf{EQ}(r, s) = r$ for closed terms in canonical form $r, s \in \mathsf{ISE}$.

3.2.3 Permeate

Permeate, the set of assertions exported from the source into the target consists of all equations $\mathsf{EQ}(r, s) = t$ and $\mathsf{EQ}(r, s) = f$ for closed r and s that are provable in the source. Permeate forgets a, CF, and $\mathsf{EQ}_{\mathsf{syn}}$ as well as any equations that make use of these constants and functions.

3.2.4 Target setting

The target theory consists of the equations of PGA from [7] which are known to be sufficiently strong to prove each closed term equal to a canonical form.

3.2.5 Result of combined reasoning

Together with permeated equations the target is now sufficiently strong for the following form of completeness: for all closed expressions r and s, $\mathsf{EQ}(r, s) = t \vee \mathsf{EQ}(r, s) = f$ can be shown.

This fact requires a combination of transformational reasoning which is perfectly supported in IS in combination with reasoning about canonical forms and syntactic equality which is perfectly supported in the datatype build around ISE which is inconsistent with the abstraction level of IS. These assertions for closed r and s constitute reasonable examples of results of combined but paraconsistent reasoning in this setting.[4]

4 Mathematical triviality of these considerations

In mathematical terms what has been described above is entirely trivial. For instance in the theory of formal languages there seems to be no confusion between grammar and language. A grammar is a syntactical object of which a language presents some form of abstraction. By always separating these notions

[4] Admittedly the fraction example from [6] is much simpler and for that reason more convincing. But this case arises in every day programming practice, of course with different program notations, and I believe that it is equally fundamental.

much confusion is avoided. Mathematically what has been described is a trivial observation concerning a subset of the regular languages and special forms of grammars for such languages. The problem at hand lies not in finding or proving the mathematical facts. It lies in the impact such facts have on the terminology and on its use. These facts are supposed to support an ambiguous notion of instruction sequence which is vulnerable to being portrayed as being paraconsistent at best.

In the case of process theory matters are already far harder than in the case of grammars and languages. The notion of a process notoriously oscillates between expression and transition system and no definite choice between those is convincing in the case of processes so it seems. The concept of a process resides between program (syntax) and behaviour (not syntax) an seems to need the mental flexibility of its "users" to allow a range of meanings. It follows that in process theory the occurrence of areas where paraconsistent reasoning can be applied must be expected.

In the case of fractions again no reliable choice between a syntactic perspective (involving numerator and denominator) and a more abstract semantic perspective (explaining fractions terms of rational numbers) seems to be feasible.

The problem posed in this paper arises once instruction sequences are contemplated as the basis for a theory of computer programs. Like process and fraction, instruction sequence defeats the introduction of a definite default interpretation, either as an expression or as a proper sequence in a set theoretic sense. The inability to find a convincing definition of the concept of a program led to the work on program algebra as reported in [7]. Some 15 years later that inability is still in place, although the proposed projection semantics provides a reasonable definition of what is a program in terms of instruction sequences, a definition which I still consider to be convincing.

Only recently I have come to believe that some form of seemingly unsolvable conceptual problems concerning the precise definition of an instruction sequence (as well as concerning the use of that definition in building a theory of programs and programming) mimic the so-called fraction pattern and for that reason call for the introduction and use of paraconsistent logics and in particular call for the application of paraconsistent reasoning based on the C&P paradigm.

5 Concluding remarks

It appears that C&P style paraconsistent reasoning applies in even simpler conditions than fractions. I believe that in each direction that one can choose when detailing the foundations of of mathematics initial stages can be found where different levels of abstraction are not dealt with by means of separating terminology, thus giving rise to possible occurrences of the fraction pattern or of other similar concept image patterns. That the fraction pattern emerges when considering instruction sequences comes as no surprise, in hindsight.

5.1 Polyadic instruction sequences and paraconsistency

The terminology of instruction sequences has presented other difficulties for which paraconsistent reasoning may provide a solution. In [10] the notion of a polyadic instruction sequence was introduced. A polyadic instruction sequence is a family of instruction sequences rather than a single one. But intuitively one seems to assume that the concept of an instruction sequence has the flexibility of modularisation, that is of different parts constituting a whole. Bridging this gap seems to call for paraconsistent reasoning.

It seems that one may consider the setting with single instruction sequences as a source theory. In the source theory one finds a function $\texttt{getins} : \texttt{INSEQ} \times \mathbb{N} \to R_I$. That function must be deleted when permeating knowledge to the target theory level in which instruction sequences are polyadic by default. There is some deviation from the C&P paradigm in that occurrences of `getins` must be replaced by similar occurrences which provide in addition a rank number for the instruction sequences from which the instruction must be fetched. In this way the flexibility is created to make instruction sequence refer to polyadic instruction sequence by default, thus preventing the jargon to become unpleasantly pedantic[5]

5.2 Acknowledgements

I acknowledge Andrea Haker for enduring long discussions on how to proceed with writing papers given the limitations that I experience with writing scholarly publications in a conventional style. My wife Ineke Loots has been (and still is) helpful in getting the burocratic details of Minstroom Research sorted out, which is for the time being a definitely unrewarding task. I acknowledge Thomas Cool for pointing out to me the existence of `viXra` as a useful alternative for `arXiv` which at this stage I consider to be preferable over `arXiv` as a platform for publishing certain specific kinds of e-prints.

References

[1] J.A. Bergstra. Putting instruction sequences into effect. `arXiv:1110.1866 [cs.PL]`, (2011).

[2] J.A. Bergstra. Decision taking avoiding agency. `viXra:1501.0088`, (2015).

[3] J.A. Bergstra and I. Bethke. Note on paraconsistency and the logic of fractions. `arXiv:1410.8692[cs.LO]`, (2014).

[5] The matter is specific for determination of an appropriate extension of the phrase instruction sequences. When thinking in terms of programs it should be noted that the projection semantics (see [7] for that notion) for polyadic instruction sequences (as given in [10]) already creates the flexibility that the notion of program includes the case of polyadic instruction sequences.

[4] J.A. Bergstra, I. Bethke and A. Ponse. Thread algebra and risk assessment services. *Proc. Logic Colloquium 05, Ed. C. Dimitracopoulos, L. Newelski, D. Normann, and J. Steel. Lecture Notes in Logic, CUP/ASL*, 1–17, (2005).

[5] J.A. Bergstra, I. Bethke and A. Ponse. Decision problems for pushdown threads. *Acta Informatica*, 44: 75-90, (2007).

[6] J.A. Bergstra and M. Burgess. Promise theory, principles and applications χ`tAtxis Press, ISBN 9781495437779`, (2014).

[7] J.A. Bergstra and M.E. Loots. Program algebra for sequential code. *Journal of Logic and Algebraic Programming*, 51 (2): 125-156, (2002).

[8] J.A. Bergstra and C.A. Middelburg. Thread algebra for strategic interleaving. *Formal Aspects of Computing*, 19 (4) pp. 445–474, (2007).

[9] J.A. Bergstra and C.A. Middelburg. Distributed strategic interleaving with load balancing. *Future Generation Computer Systems*, 24 (6), pp. 530-548, (2008).

[10] J.A. Bergstra and C.A. Middelburg. Thread extraction for polyadic instruction sequences. *Scientific Annals of Computer Science*, 21 (2): 283–310, (2011).

[11] J.A. Bergstra and C.A. Middelburg. Instruction sequence processing operators. *Acta Informatica*, 49, 139-172, (2012).

[12] J.A. Bergstra and C.A. Middelburg. Division by zero in non-involutive meadows. *Journal of Applied Logic*, 13(1): 1–12 (2015).

[13] J.A. Bergstra and A. Ponse. Projection semantics for rigid loops. `arXiv:0707.1059v1 [cs.PL]`, (2007).

[14] J.A. Bergstra and A. Ponse. Periodic single pass instruction sequences. `arXiv:0810.1151v2 [cs.PL]`, (2013).

[15] J.A. Bergstra and A. Ponse. An instruction sequence semigroup with involute anti-automorphisms. *Scientific Annals of Computer Science*, 19: 57–92, (2009).

[16] J.A. Bergstra and A. Ponse. Division by zero in common meadows. `arXiv:1406.6878 [math.RA]`, (2014).

[17] J.A. Bergstra, and J.V. Tucker. Equational specifications, complete term rewriting systems, and computable and semicomputable algebras. *Journal of the ACM,* 42(6):1194–1230, (1995).

[18] J.A. Bergstra and J.V. Tucker. The rational numbers as an abstract data type. *Journal of the ACM*, 54 (2), Article 7 (2007).

[19] B. Brown and G. Priest. Chunk and Permeate, a paraconsistent inference strategy. Part I: the infinitesimal calculus. *Journal of Philosophical Logic*, 33:379 ?388 (2004).

[20] J. Carlström. Wheels–on division by zero. *Math. Structures in Computer Science*, 14 (1) pp. 143–184, (2004).

[21] J. Carlström. Partiality and choice, foundational contributions. *PhD. Thesis, Stockholm University*, `http://www.diva-portal.org/smash/get/diva2:194366/FULLTEXT01.pdf`, (2005).

[22] A. Doroshenko, G. Tseitlin, O. Yatsenko and L. Zacharyia. A theory of clones and formalised design of programs. *in: Proc. Int. Conf. "Concurrency, Specification, and Programming" (CS&P) 2006*, 27–29, (2006).

[23] K. Kimbler and L.G. Bouma (Eds.) Feature interactions in telecommunications and software systems V. IOS Press, (1998).

[24] F. Padberg. *Didaktik der Bruchrechnung* (4_{th} edition). Series: Mathematik, Primar- und Sekundarstufe, Springer-Spektrum. (2012).

[25] T.S. dos Reis and J.A.D.W. Anderson. Construction of the transcomplex numbers from the complex numbers. *Proc. WCECS 2014*, `http://www.iaeng.org/publication/WCECS2014/WCECS2014_pp97-102.pdf` (accessed 29-12-2-14), (2014)

[26] S. Rollnik. *Das pragmatische Konzept für den Bruchrechenunterricht*. PhD thesis, University of Flensburg, Germany, (2009).

[27] A. Setzer. Wheels (draft). `http://www.cs.swan.ac.uk/~csetzer/articles/wheel.pdf`, (1997).

[28] S. Shapiro. Structures and logics: a case for (a) relativism. *Erkenntniss*, 79, 309-329 (2014).

[29] M. Slors. Conscious intending as self-programming. *Philosophical Psychology*, DOI: 10.1080/09515089.2013.803922 (2013).

[30] D. Tall and S. Vinner, S. Concept image and concept definition in mathematics, with special reference to limits and continuity. *Educational Studies in Mathematics*, 12, 151?169, (1981)

[31] G.E. Tseitlin. Algebraic algorithmics: theory and applications. *Cybernetics and System Analysis*, 39 (1), 1–15, (2014).

A Formalities and policy statements I: about nopreprints

This Section deals with a range of topics which arise if one publishes research paper-like work in a somewhat unconventional manner. Two aspects constitute a deviation from ordinary publishing for someone with an academic affiliation: (i) the work is performed and posted from a private affiliation (in this case MRbv), and (ii) the work is categorized as a so-called nopreprint.

In my view the nopreprint status and the use of the MRbv affiliation are independent through not entirely unrelated matters. Both aspects require an explananation and to some extent a justification.

I must apologise in advance for the boring length of these considerations, as well as for the fact that in its kind this text is still in an early stage of maturity. I will include similar texts in further documents (either having nopreprint status or written from my MRbv affiliation) so that some evolution to a stage of mature stability can take place.

A.1 Nopreprints, a (new ?) publication category

The repository `viXra.org` publishes so-called e-prints. I consider e-print to be an indication of the technical format, papers on `arXiv` are e-prints as well. In addition to being e-prints documents `arXiv` unless already scholarly published (see below for a definition of this notion) have the status of preprints. However, given the wide accessibility of `viXra` postings I assume that e-prints thus posted qualify as "publications" given of course that fairly general requirements on the document are met. In this section "author" will include the case of a team of multiple authors.

The notion of a publication in a scientific context has the connotation of it having been peer reviewed and its distribution being performed by an outlet which requires compliance with the terms and conditions of the selective peer reviewing mechanism as entertained by that outlet. Typical outlets are research journals and the proceedings of the occurrence unique conferences with education organization or of a well-organized conference series which is held and organised under the responsibility of a scholarly society.

The meaning of publication just mentioned deviates from more liberal and more common definitions which focus on form, objective, and availability, rather than on the presence of generally recognised quality control mechanisms. Derived from this (science context) interpretation of a publication, is the notion of a pre-publication or a preprint. Nowadays preprints in electronic form (that is e-printed preprints) can easily be distributed as widely as their "printed" realizations (successors). If printing is performed within a pay wall the preprint may even turn out to be far more easily and cheaply accessible for a general audience.

Printing increasingly means no more than (i) having been positively assessed by a selective peer reviewing system of known reputation operating from an

equally reputed organization, (ii) having been adapted to requirements imposed by that reviewing mechanism, and (iii) having been posted through the technical facilities (website, ebooks etc.) of that particular organzation. Such works I will refer to as having been scholarly published.

Preprints typically are documents that its authors intend to be promoted sooner or later to the published status (just specified as "having been printed" or having been scholarly published). Therefore, although a preprint placed on `arXiv` may never be published (in the sense of being "printed" as just outlined), it has the preprint status on the basis of its author's intentions. A preprint has not been scholarly published by definition, at least not on the date of its appearance as a preprint.

A.1.1 Nopreprint: a preprint-like e-print, which is not a preprint

In the absence of intentions towards scholarly publication posting an e-print on `arXiv` is less plausible given the objectives of `arXiv`. The notion of an archive suggests that documents which have already obtained some form of status are preserved in archival mode. Archiving as such does not, by itself, confer that form of status. Now a nopreprint is an e-print (or if one so wishes a paper document that is sent around to an interested readershipp), which intentionally is not equipped with the connotation of a preprint, that is of a document waiting to be (somewhat adapted) and published in a selectively peer reviewed outlet which is under the control of a reputable body.

The classical notion of a technical report has the flexibility to include nopreprints but it fails to exclude preprints. For that reason nopreprint is a different concept. In Academic practice technical reports more often than not have the status of preprints. Another related notion is that of a postprint, a copy (perhaps differing in very minor ways) of a scholarly published paper is `arXiv`-ed around the date of scholarly appearance, carrying the relevant information about the official publication, nowadays often preceding appearance on physical paper. Postprints and nopreprints are remote relatives only. A postprint has obtained the scholarly published status that a nopreprint will probably never acquire.

A.1.2 A rationale for writing and publishing nopreprints

Peer reviewed publications go with the claim that science is made up of such works, and that works which in hindsight fail to comply with scientific requirements will eventually be withdrawn. Not every document about a research theme merits that status in the perception of its authors. There is a remarkable focus in science (publicly funded research) on so-called high quality work. Evidently high quality work can only exist in a context giving room for works of lower quality just as well. I will assume that, seen from the perspective of formal science and research, a nopreprint in general (that is by default) will not even potentially contain a high quality work which could have passed all relevant screening just as well. Thus nopreprints are a class of non-high quality

works (or at at least non-"top quality" works).

Now one might suggest that nopreprints should be submitted to less pretentious peer reviewed outlets. But this may not comply with author objectives. Obviously the line of argument is risky. If even low quality journals won't publish a paper, or if you don't want it having been published in such a journal why write (and publish as an e-print with nopreprint status) it at all. Many different viewpoints are possible on this matter. I feel that one may (i) wish to see one's "true" (that is scholarly published) research output embedded in (that is to exist in a context of) a volume of works (blogs, news items, scattered comments) of a secondary status, (ii) that one may wish to contribute to that volume of secondary status items oneself, and (iii) that one may wish to do so while paying attention to the working ethics of ordinary scholarly research. For instance nopreprint status provides no justification for plagiarism of any kind (where self-plagiarism must be defined and dealt with in a careful manner), no justification for the misuse of copyright owned by other parties, no justification for defective references to prior art, and no justification for making scientific claims without proper proof or investigation.

A.1.3 Options for nopreprint content

Here are some examples of content kinds from which may plausibly make up the content of a nopreprint.

- Popular descriptions of content selected from one or more scholarly published works.
- Explanations of content of existing published work for a non-specialist (though research aware) audience.
- Providing additional details for the justification and explanation of existing scholarly work.
- Opinions about existing and forthcoming scholarly work.
- Listings of challenges, problems, puzzles.
- Examples of general theoretical results.
- Informative but not innovative applications of theory from one area to another area.
- Results that are considered (by the author) too simple for scholarly publication but which may nevertheless be considered informative for a wider audience as an illustration of known principles.

A.1.4 Nopreprint form versus nopreprint content

Claims concerning the validity of research outcomes which are in any sense risky, that is the author can imagine that readers may dispute such claims because

there is more at stake than a mere difference of opinion, must be submitted to peer review on the long run. This is a critical point. Nopreprint status may be a matter of document form, that is non-compliance with ordinary scholarly rules of the game. But it must not be a coverup for "publishing" results without proper checks and balances. It follows from this perspective that nopreprints must be harmless to some extent.

On `viXra` there is room for other works than nopreprints. Nopreprint status is a kind of disclaimer: this work contains, to the best of its author's knowledge, no conclusions that (on the long run) ought to be peer reviewed instead of merely be included in a nopreprint. In other words, a nopreprint is not intentionally unpublished (in the scholarly sense involving peer review) because its author experiences a lack of appropriate publication outlets but because the author sees no justification (or reason, or need) to have it peer reviewed. That is not a purely subjective matter, and a nopreprint author must be open for debate concerning the question if the document must be, as a whole or in part, (in contradiction with the author's original views) be transformed to preprint status, and submitted for scholarly publication thereafter

A.1.5 Nopreprint publication, a matter of paraconsistency?

As just defined an e-printed nopreprint is a publication and a non-publication at the same time. Publication status is probably undisputed outside the scientific context, while publication status will probably be disputed within a scientific context where scholarly publication is the default understanding of publication. Dealing with inconsistencies without getting these out of the way is the subject of paraconsistent logic and reasoning.

Is paraconsistent reasoning needed to understand the concept of a publication? Inside and outside the scientific context different default settings govern the interpretation of the concept of publication. Outside the scientific context an instance of publication implies neither the presence nor the absence of the application of a reliable quality control mechanism. Inside a scientific context it is the other way around. This matches with paraconsistent reasoning in accordance with the so-called chunck and permeate paradigm as proposed in [19]. This paradigm suggests to think in terms of at leas two chuncks of knowledge, source and target.

In particular it is useful to consider a theory of "what is a preprint" (in a scientific setting) as the source theory (including the assertion that an e-printed preprint is a publication, though not necessarily a scholarly one). The target theory results by removing the concept of intended submission to peer review as a condition (for being a publication) and by replacing it by a constraint about content that involves peer review differently, i.e. by assuming that peer review is immaterial for the document or for any part of it.

Now the chunck and permeate strategy allows selective transfer of facts from the source context to the target context. In the case at hand this selective transfer allows one to infer rules and requirements on nopreprints while not being logically "silenced" by the apparent contradiction if source and target

theories are simply combined.

A.1.6 Slippery slope risks

Assuming that neither the risk of rejection, nor the absence of a peer reviewed outlet appropriate for submitting a paper (for which a choice between preprint and nopreprint yet has to be made) convincingly justifies nopreprint status, an author might be inclined to favor the nopreprint publication category for the simple reason that this allows working according to a well-prepared plan without the need to "do something about the paper" after it has been published (as an e-printed nopreprint).

Once the writing of a sequence of papers acquires momentum it may become seemingly practical to downgrade potential preprints in such a way that nopreprint status becomes defensible given the paper. The latter says nothing about the tolerance of an author's professional environment about nopreprint publication. Now an author may slowly do away with the objections against nopreprint publishing to the extent that fragments of papers emerge in nopreprints which at least in principle (that is intentionally) should have been submitted for peer review. This is a risk of a slippery slope nature.

In other words: writing nopreprints is (or should be) neither explicitly nor implicitly an expression of criticism on the existing publication outlets. To the extent that `arXiv` policy discourages the publication (on `arXiv`) of what I have defined as nopreprints, I consider that policy to be both useful and justified. There is no need for a preprint repository to accept nopreprints, on the contrary.

A.1.7 Linking nopreprint publishing with a private affiliation

It seems unproblematic to publish a nopreprint from an academic affiliation, and it seems equally unproblematic to publish a preprint (as an e-print) from a private (that is non-academic or non-institutional) affiliation such as MRbv.[6]

I have chosen for the time being, and until convincing arguments against this choice surface, that I will personally write nopreprints from the MRbv affiliation only, for the simple reason that I prefer not to use an academic affiliation for a kind of activity which it may not wish to endorse. To what extent this separation of concerns is feasible (and useful) on the long run remains to be seen.

A.2 Defensive novelty analysis for this particular paper

By classifying the work on algebraic algorithmics above as a nopreprint I become committed to the claim that no part of it should have been prepared for peer review. This work I will refer to as defensive novelty analysis. This analysis must underpin the claim that no part of the nopreprint would be rejected on

[6] I consider it problematic to use different affiliations for posting papers on `arXiv` at the same time, and I do not think that `arXiv` must have that flexibility either. I feel no such problem with `viXra` although I am not posting papers on `viXra` because of dissatisfaction with `arXiv`.

grounds of lacking soundness or lacking representation of and/or reference to existing work by a competent reviewer. One may question the very possibility of checking this criterion in advance, but in my view this is doable and in the case of a nopreprint written by an author with a background in academic research it is a necessity. There are two stages in this activity: filtering out this fragments that might be considered to run the risk of containing claims or statements sensitive to peer review, and secondly arguing for each of these fragments that there is no problem.[7]

Here is a listing of fragments in the paper that might catch the attention of a critical reviewer (when embedded in other work that might be in scope of a scholarly publication outlet), together with an argument as to why the risk can be taken.

1. Introducing and naming the notion of exclusively imperative programming (EXIP). (EXIP as a phrase has not been claimed by any other research group on software engineering, and instruction sequences are exclusively imperative programs. Therefore I see no risk in the introduction of this phrase. Whether there is any virtue in doing so is another matter and judging is not part of this defensive justification exercise.)

   However a reader might expect that an introduction of a name for a programming methodology is carried out within the context of a more substantial work. Acknowledging that point of view, I don't try to claim priority for the introduction for this phrase (which is in accordance with its defining occurrence in a nopreprint.)

2. Classifying program algebra style instruction sequence theory under algebraic algorithmics. (I assert this as a subjective viewpoint. It is consistent with the (scholarly published) descriptions of algebraic algorithmics which I have found.)

3. Making reference to [31] as an original source of the phrase "algebraic algorithmics". (I take the risk that the same author has earlier works using the phrase. I have no experience of reviewers being very well-informed about such matters.)

4. Introduction of the notion of a pattern in the context of paraconsitent reasoning. (I cannot imagine a grounded objection against that idea. I see a similarlity with software patterns.)

[7] Paradoxically it seems that, at least in principle, an author of a forthcoming nopreprint would prefer or even need to have the effectuation of these two steps peer reviewed in an official manner. It remains to be seen if such mechanisms will emerge on the long run. In any case the claim of the author of a nopreprint would be that these steps can and have been carried out to the satisfaction (at least in theory) of a reasonable sample of peer review teams.

Setting up a nopreprint peer review system which validates the author provided defensive novelty analysis of a nopreprint after its (non-scholarly) publication, or even of a prospective nopreprint in advance of its publication, may become relevant if this mechanism proves attractive for sufficiently many other authors besidesr myself.

5. Extraction of a fraction pattern from [3]. (I consider this step to be straightforward and its naming to be more or less without alternative.)

6. Asserting convincing similarity between the fraction pattern and the instruction sequence pattern. (I see no risk that this assertion is considered invalid.)

7. Inferring the relevance of C&P style paraconsistent reasoning in the context of instruction sequences. This is a subjective matter, I intend to go no further than to state that in my view C&P style paraconsistent reasoning enters the picture of instruction sequences in the specified manner.

8. Classification in a specific `viXra` category. (The `viXra` category "Data Structures and Algorithms" (DS&A) is chosen because the theme of the paper may be subsumed under software generation (which `viXra` subsumes under DS&A).)

B Formalities and policy statements 2: working from a private and insignificant affiliation

This document has been writing in the setting of a private affiliation that constitutes an insignificant small enterprise (MRbv). That aspect requires justification as well as clarification. Here is the justification that I consider applicable in this particular case.

B.1 Using a private platform as an affiliation

Using MRbv as a platform for work that might have been performed from an academic affiliation as well has these grounds:

- The objective to develop (and thereafter use) a platform (affiliation) from which a wider scope of topics can be discussed than from a specialised academic affiliation where one should preferably not tresspass on subjects that have been "occupied" by colleagues from the same institution.

- Moving out of the academic competition, that being only one form of ambitious behaviour, which one is expected to repeat over and over again.

- The wish to develop a platform from which a wider range of options (with regard to method and form) for the expression of results is available.

- The wish to define impact and visibility of the output of work by means of self-chosen metrics.

- The wish to make use of other sources of funding, including revenues of consulting, revenues of the production and delivery of commercial courses and courseware, and revenues from the exploitation of IP obtained by research-like work.

- The wish to experiment with the use of a private platform and to understand in that manner much better what an academic affiliation implicitly offers and requires.

- The wish to give expression to the viewpoint that academic institutions have captured control of intellectual space to an unreasonable extent, and that open access to science at large requires that independent affiliations can more easily be developed and used whenever appropriate.

At the same time it is important to live up to codes of conduct that govern academic work, to the extent that this is of relevance for various activities. For nopreprint-style work originating from MRbv `viXra` is chosen as the preferred outlet.

This document is given MRbv document classification category A. The meaning of that labeling is explained below.

B.2 Justification of the MRbv DCS category A classification of this particular paper

The MRbv category A classification is chosen for this paper on the following grounds:

1. There is no immediate or even intended vision of application or valorisation of the content of this nopreprint. (This rules out categories C and D).

2. The views in this work (with the exception of the statements concerning the nopreprint category) primarily have a commentary status and are unlikely to be held as MRbv views with methodological consequences in any future context. (This rules out category B).

3. The nopreprint status is intentional, submission to a (selectively) peer reviewed publication outlet is not intended. (This indicates MRbv as an appropriate affiliation bringing with the need for classification in A B, C, or D).

4. Subsequent academic research on the basis of the content of this work is not foreseen by the author. Subsequent non-academic research, however, is not implausible.

5. The results of the paper are purely conceptual (which is indicative of category A).

B.3 MRbv document classification scheme (DCS)

This Section provides a slightly improved version (featuring considerable overlap) of the corresponding Section of [2].

The MRbv document classification scheme (MRbv-DCS) for publicly accessible documents and content originating from MRbv has four categories named

A, B, C, and D. MRbv-DCS classification is of relevance only for documents with MRbv as the affiliation of at least one of the authors. Classification primarily depends on content and form of a document, but it may also depend on the objectives of work that is reported about in the document. The four document categories are defined as follows:

A: MRbv is used as a preferred affiliation on grounds related to the quality, the style, the objectives, or the form (or any combination of these) of the work. The work has not been carried out with future use within MRbv as a primary objective, however the possibility of such future use is not excluded unless a statement to that extent is included (in which case replacements of the document may be classified under another category).

B: Work aimed at the development of conceptual schemes and viewpoints with the following requirements: (i) these are MRbv viewpoints and must be (intended to be) as stable as ordinary research outputs by the same author(s), (ii) not necessarily leading to, or contributing to, the development of products or services to be offered by MRbv, (iii) but having the potential for being developed into products or services that may be offered by MRbv.

C: Work meant for future use or for development towards future use within MRbv.

D: Work that is directly linked to MRbv practice, e.g. cases, projects, courses, and books or other content which will only be made available against compensation.)

B.4 IP policies and dissemination policies

IP policies and dissemination policies are features which are specifically configured for each document.

1. LICENCE: this work is licensed under Creative Commons 4.0 (BY) `http://creativecommons.org/licenses/by/4.0/`. In as far as consistent with this licence the following rules apply in addition:

    - Reference can be made by providing author, title, url on `viXra.org` and year (in this case 2015).
    - Referencing this work is always permitted.
    - Although making appropriate reference to this work is appreciated, referencing this work is in no circumstance required, requested, or expected (by the author or by anyone representing MRbv) as a sign of intellectual debt, or as an acknowledgement of priority concerning certain ideas or results.
    - However, readers must be aware that copying or incorporating parts of this work in other works without proper referencing may be construed as some form of plagiarism (or otherwise as a violation of CC

4.0 BY) by agents not under of control of MRbv. MRbv reserves the right to agree in public with such claims when made by other parties, in cases such judgements are requested by mentioned parties, but MRbv will not base any claims or complaints on such states of affairs.

2. DEFINITIVE FORM. This work is not meant for publication in any other medium that claims to exert quality control of whichever form. In particular the work has not been and will not be posted on `arXiv.org` in this form or in a more or less similar form. This is a promise in the sense of [6].

3. The paper will not be withdrawn from `viXra.org` but it may be replaced when a newer version is available.

4. AMBITION. The work will lead to other works from MRbv that are in part based on this work. These works in combination may evolve to a stage from which documents can be extracted, by selecting and combining suitable fragments that are ready for scholarly publication.

5. The work is `viXra`-ed for reference purposes and for easy and durable accessibility.