Journal of Logical and Algebraic Methods in Programming ••• (••••) •••–•••

Contents lists available at ScienceDirect

Journal of Logical and Algebraic Methods in Programming

www.elsevier.com/locate/jlamp

Why mathematics needs engineering

Raymond Boute

INTEC, Ghent University, Belgium

ARTICLE INFO

Article history:
Received 2 June 2015
Received in revised form 4 January 2016
Accepted 4 January 2016
Available online xxxx

Keywords:
Category theory
Codomain
Engineering
Function
Mathematics
Relation

ABSTRACT

Engineering needs mathematics, but the converse is also increasingly evident. Indeed, mathematics is still recovering from the drawbacks of several "reforms". Encouraging is the revived interest in proofs indicated by various recent *introduction to proof*-type textbooks. Yet, many of these texts defeat their own purpose by self-conflicting definitions. Most affected are fundamental concepts such as relations and functions, despite flawless accounts 50 years ago. We take the viewpoint that definitions and theorems are tools for capturing, analyzing and understanding mathematical concepts and hence, like any tools, require diligent engineering. This is illustrated for relations and functions, their algebraic properties and their relation to category theory, with the *Halmos principle* for definitions and the *Arnold principle* for axiomatization as design guidelines.

1. Introduction: mathematics and engineering

Mathematics has been intertwined with engineering since antiquity [11,49].

Kline notes that "*More than anything else mathematics is a method*" [30]. Arguably, the primary purpose of this method is *effective reasoning*. This view best explains what Wigner calls *the unreasonable effectiveness of mathematics* [61], in particular its practical usefulness far beyond the originally intended application areas. From this perspective, the dichotomy between Platonism and formalism dissolves: mathematical objects *do* exist, albeit in an abstract universe. Formalism, definitions and theorems are the tools to study them.

Tools, being artifacts, deserve careful design, borrowing criteria and guidelines from engineering. Some of these also been discussed by José Oliveira [41] in another context. Here we focus on using engineering principles in mathematics.

Foremost is enhancing the effectiveness in reasoning. Symbolic notation properly designed and used yields extra guidance via the shape of the expressions. It should function like well-meshed gears in a Swiss precision clockwork.

Aptness and economy in capturing the abstract objects of interest ensures conceptual malleability, generality and practical usefulness. Human factors are influential here, and it is often overlooked that this is a highly individual matter of temperament and background. Even so, everyone benefits from clear conceptualization and reasoning. For instance, *separation of concerns* avoids the common misconceptions caused by intellectual noise and conceptual tangling.

In classical mathematics, methods and notations were often thought-out carefully. In algebra, for instance, symbolic notation started with Diophantus and evolved into its current form via Viète and Descartes [5,11], rarely violating good design practices, thus making symbolic calculation today's norm. In comparison, notations from "modern mathematics" as used in everyday practice are substandard, hampering symbolic reasoning and thus making it unpopular.

The cause of this stagnation is largely historical. When introducing so-called "modern mathematics", forgetting its roots caused serious educational mistakes, denounced in rather strong terms by Arnold [3]. In a severe overreaction, the view

E-mail address: raymond.boute@pandora.be.

http://dx.doi.org/10.1016/j.jlamp.2016.01.001
2352-2208/

Please cite this article in press as: R. Boute, Why mathematics needs engineering, J. Log. Algebraic Methods Program. (2016), http://dx.doi.org/10.1016/j.jlamp.2016.01.001

of mathematics as a method was sacrificed in favor of mathematics as a bag of tricks and attempting to elicit motivation by so-called "real-life" examples no more realistic than the *farmer-sells-potatoes*-type problems in grade school – and in PISA tests! The well-proven structure *definition–examples–theorems* was frowned upon, and mathematical exposition had to become a "narrative".

As a result, classics like Rudin's *Principles of Mathematical Analysis* [48] are, as Krantz observes, "*often no longer suitable, or appear to be inaccessible, to the present crop of students*" [32]. Here the blame does not fall on the students.

Narratives lack the punctuation provided by headings like "Definition" and "Theorem", which help novices to distinguish between, say, statements that can be deduced from earlier ones and statements introducing new elements.

If *definition–examples–theorems* expositions often deserve criticism, it is not for the usual reasons (take your pick), but because definitions are usually presented as "given", or as arbitrary points of departure for a game of logic. In fact, definitions are the result of *design decisions*. They also determine the flavor of the theorems (and proofs) derived from them. Hence it is crucial for understanding that these decisions are explained and justified.

In mathematics texts, this is all too rarely done. One of the few exceptions is Halmos's *Naive Set Theory* [24] which, if only for this reason (yet also for other reasons!), should be required reading for all beginning students — and many mathematicians as well. Halmos not only explains the design decisions and their shortcomings for most conventions, but also does not shrink back from calling some poor practices "unacceptable but generally accepted". Quine [45] even designates lesser offenses as "glaring perversity", which seems an apt characterization of mathematicians acting against better judgment.

Indeed, perceived "general acceptance" is often taken as a licence to perpetuate junk conventions. Users of inept designs typically defend them by feigning confusion at proper alternatives, calling them "nonstandard" even if they have been around for a long time and are routinely used by plenty of other authors.

If an engineer is sloppy, his design may fail, even catastrophically. Mathematicians often condone sloppiness, even if it sets bad examples and abuses confidence. Discerning students will be dissatisfied by the discrepancy between the reputation of mathematics as being precise and actual practice. Others may even get confused if insecure teachers insist on "doing things as in the book".

Yet, the engineering literature is not blameless either. Years ago Lee and Varaiya [37] corrected many inept mathematical practices in signal processing.

Playing down such issues as "just a matter of notation" is misleading. Poor notation prevents the shape of expressions from giving guidance in reasoning. It also reflects poor understanding, according to Boileau's aphorism "*Ce que l'on conçoit bien s'énonce clairement – Et les mots pour le dire arrivent aisément*". If authors misunderstand their own definitions, what about their students?

"If it ain't broke, don't fix it" is another engineering maxim. Yet, as we shall see, even basic concepts that worked fine 50 years ago somehow got "broke".

This paper addresses the issue in the title by presenting a design view on various concepts from the literature, but it is *not* some linear, complete proposal.

Often references include page numbers to make them truly useful for the reader. For brevity, co-authors are omitted when mentioning names in the text.

2. Case study A – relations: two logically equivalent definitions

2.1. Simple and safe formulations

The simplest "modern" definition of a *relation* is typical in older texts such as Bourbaki [8, p. 71], Suppes [57, p. 57], Tarski [58, p. 3], but only in a few current books, such as Jech [29, p. 10], Scheinerman [50, p. 73] and Zakon [62, p. 8].

Definition 1 *(Relation).* A relation is a set of ordered pairs.
Equivalently, in symbols [8,57]: $R \text{ isrel} \equiv \forall z \,.\, z \in R \Rightarrow \exists x \,.\, \exists y \,.\, z = (x, y)$.

Taking *set* and *ordered pair* colloquially, and with 'nonmathematical' examples, the word statement of Definition 1 is even accessible at grade school level.

In this paper, when saying just "pair", we always mean "ordered pair".

Some notational design issues arise here. First, an ordered pair is commonly written (x, y). Some authors use $\langle x, y\rangle$, a waste of symbols. In fact, one can even write x, y and reserve parentheses for emphasis or disambiguation, which also covers n-tuples like (x, y, z) and trees like $((x, y), z)$. Identifying (x, y, z) with $((x, y), z)$ as in Bourbaki [8, p. 70] is clearly a bad design decision.

Second, the literature diverges about writing (x, y) or (y, x) and $x\,R\,y$ or $y\,R\,x$. Quine [45, p. 24] offers many good reasons for following Peano and Gödel in using the *natural order* from spoken language, writing "a is the father of b" as $a\,F\,b$, and "a is smaller than b" as $a < b$. Similar reasons would favor writing, for instance, "velocity versus time" as (v, t). However, mathematicians used to writing the "independent variable" first might feel disoriented—unlike novices! Tradition can be reconciled with reason by writing $(x, y) \in R$ iff $y\,R\,x$. For human engineering reasons, such conventions should be stated conspicuously.

Please cite this article in press as: R. Boute, Why mathematics needs engineering, J. Log. Algebraic Methods Program. (2016), http://dx.doi.org/10.1016/j.jlamp.2016.01.001