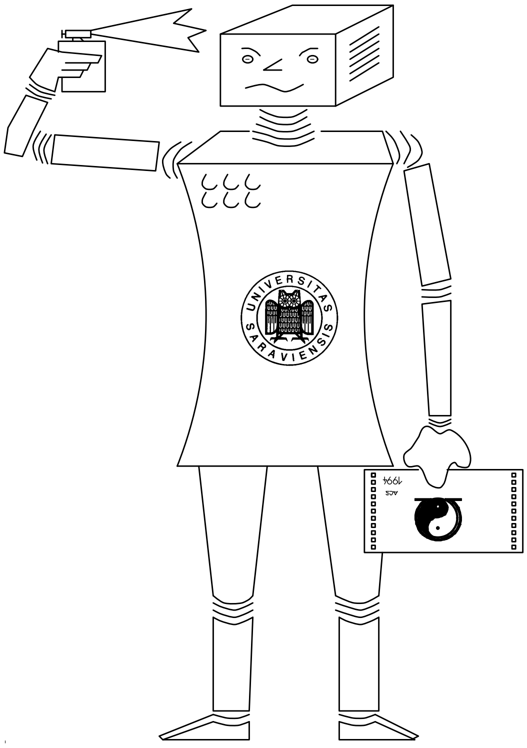
UNIVERSITÄT DES SAARLANDES
FACHBEREICH INFORMATIK
D-66041 SAARBRÜCKEN
GERMANY
WWW: http://js-sfbsun.cs.uni-sb.de/pub/www/



An Approach of Formalizing Mathematics by Reformulations

– A Proposal for QED –

Manfred Kerber

Published as: In electronic journal *Mathesis Universalis – A Neo-Leibnizian Forum for Mind-and-Matter*, 4. Spring 1995, No. 1: On Automated Reasoning, URL: gopher://plearn.edu.pl/11/MU/menu

An Approach of Formalizing Mathematics by Reformulations
– A Proposal for QED –

Manfred Kerber
Fachbereich Informatik, Universität des Saarlandes
Im Stadtwald 15, D-66041 Saarbrücken, Germany
e-mail: kerber@cs.uni-sb.de Tel.: (+49) 681-302-4628

March 17, 1995

1 Introduction

It has been one of the goals of the whole field of mechanized reasoning to put into work Leibniz's old ideas of having a lingua characteristica, in which each mathematical fact can be encoded, so that each dispute whether a statement is true of not can be settled by simply calculating. This idea has been a guideline for early reasoning systems like Automath or newer ones like Nuprl, Getfol, IMPS, Mizar, or our Ω-MKRP. For different reasons, however, unlike to computer algebra systems none of these systems has gained a broader acceptance among mathematicians.

One of the most important problems with each of the above systems is the more or less fixed object logic, a user has to employ in order to encode and to prove theorems. In the following we propose a *liberal* approach to the question of this language. This approach is essentially based on the idea to stress a meta-system of reformulations between different object languages which allow each user to use his own object language, but nevertheless to be able to communicate the axioms, the definitions, the theorems, and the proofs. Thereby we propose an approach to QED [1], in which not just one further system in the list above is built but we concentrate on a very important point for the communication of proofs.

In such a liberal approach there are essentially two ways in order to guarantee correctness: either each proof must be reformulated into each other system, or there is a meta-proof that the proofs correspond to one another. In the following we propose a *constructive* meta-logic which satisfies both of these requirements: If you prove on a meta-level the correctness of the formal system one for the formal system two, then you can construct from each proof in system one a proof in system two.

2 Motivation

The objections of Gérard Huet against the feasibility of QED presented in the QED discussion at CADE-94 should be taken very serious. His main argument has been that there will be neither any consent on the logical framework (type theory, set theory, constructive logic, classical logic, generic logic, ...) nor on the implementation language. I think that this is true, but in the same discussion Andrzej Trybulec has been right too, when claiming that QED has already started in systems like IMPS, HOL, and Mizar.

As a matter of fact, mathematical foundation is opaque and there are different competing approaches, while the mathematical everyday language is quite standardized and foundational questions only play a minor role. The question which of these formalizations should be taken is not so easy to answer, since each of them has its own advantages and drawbacks. Even more severe for one single problem it might be that different formulations are appropriate, for instance, an explicit one in which a user can easily represent and recognize a theorem, and a more implicit one which is more suitable for a fully mechanical or a machine-supported proof (cp. [7]).

Let us shortly consider the advantages of different representations by means of sorts. In many situation a sorted formalization is more adequate for the formalization and the proof process (automatic as well as interactive) then an unsorted one. In some cases, however, it is not possible to use a (standard) sorted formalization, for instance, when you want to make a statement that a certain term has *not* a particular sort. The same situation holds for type theory compared to set theory. Certain statements cannot not be made in type theory, which are possible to make in set theory, nevertheless in most cases the types provide useful information which can be used during a proof search and they often allow a more compact representation and a shorter proof.

The main part of any formalization of QED has to rely on the basic language for formalizing mathematics and the notion of proof. If it should not be a mere description of static history, it has to take into account the ongoing development of the notion of a proof (compare for instance the graphically represented information in the Hyperproof system [2]), as well as the ongoing development of the mathematical technical language. Furthermore it should not depend on the correctness or even adequacy of a single axiomatization, since mathematics went on in spite of disastrous setbacks (the discovery of the irrational numbers in the ancient world or of the Russell paradoxes in the beginning of our century). Since semantics is hard to fix, QED should start with a proof theory and let the semantics open.

The experience in our project Ω-MKRP [6] has been that a big deal of our efforts has been made in order to improve the mathematical input language (from unsorted first-order logic, sorted first-order logic, typed but unsorted higher-order logic to a sorted higher-order logic, currently we are trying to operationalize a higher-order logic with dependent types. Furthermore there has been some discussions how to represent partiality, e.g. by Kleene's strong three-valued logic). The technical language of mathematics has shown to be so rich that many concepts seem not to be adequately represented in one standard formalism (I don't speak of the expressivity in principle). A similar experience seems to have been made in the Mizar project: "Freezing the changes in the input language and in the Mizar processor has been a goal for quite a while, yet it seems to move away like the horizon when you try to approach it." [9, p.9].

3 Approach

As a consequence of the problems with one fixed language, QED should not only support one single formal language, in which all statements have to be made, but a (not too big) variety including the standard approaches (like set theory or type theory). In order to be able to transfer proofs from one format to another, it is necessary to have an exchange format for different syntactic objects, namely terms, formulae, assertions (axioms, definitions, theorems with their proof status), and proofs. This exchange format should satisfy in particular the requirements that it is easy to parse (prefix notation). Therefore it should be different from a user-friendly high-level format that can easily be read by humans.

In order to avoid a variety of unrelated languages, they must have provable relationships. Ideally there is one *constructive meta-logic*, for instance, the Nuprl logic [3], in which the relationships between different formalizations can be formally proved. The main advantage of a constructive meta-logic consists in the possibility to extract from each meta-level proof an algorithm, which translates proof from one system to another.

In order to distinguish object level and meta-level, the notions of quoting "." and unquoting $\langle . \rangle$ as well as the general definitions of [4, Chapter10] can be used, where primitive predicates like Constant, Variable as well as functions like a function subst are defined.

Abstractly logical meta-level characterizations of all admitted logical systems are necessary. These characterization would make use of meta-formulae of the kind $\mathsf{formula}^j(``\varphi^j")$ which stands for the fact that $\varphi^j$ is a formula in the formal system $S^j$. Correspondingly predicates $\Pi^j(``\Delta \vdash^j \Gamma", \pi^j)$, standing for π^j is a proof for $\Delta \vdash^j \Gamma$ in the formal system S^j, can be employed. Of course, the meta-language has to be rich enough that the notions of formula and proof can be defined in it. For instance, if we have some derivability relations like

$$\frac{\Delta_1 \vdash^j \Gamma_1; \ldots ; \Delta_n \vdash^j \Gamma_n}{\Delta \vdash^j \Gamma}\,\mathsf{RULE}^j_k \qquad \text{and} \qquad \frac{\emptyset}{\Delta \vdash^j \Gamma}\,\mathsf{AXIOM}^j_k$$

we have the following formulae in the meta-language in order to axiomatize the notion of proof

$$\Pi^j(\text{“}\langle\Delta_1\rangle \vdash^j \langle\Gamma_1\rangle\text{”}, \pi_1) \wedge \ldots \wedge \Pi^j(\text{“}\langle\Delta_n\rangle \vdash^j \langle\Gamma_n\rangle\text{”}, \pi_n) \rightarrow$$
$$\Pi^j(\text{“}\langle\Delta\rangle \vdash^j \langle\Gamma\rangle\text{”}, \mathsf{RULE}^j_k(\pi_1, \ldots, \pi_n))$$
$$\Pi^j(\text{“}\langle\Delta\rangle \vdash^j \langle\Gamma\rangle\text{”}, \mathsf{AXIOM}^j_k)$$

where all variables Δ_j, Γ_j, and π_j are implicitly universally quantified.

Let $\mathcal{S}^1$ and $\mathcal{S}^2$ be two formal systems. A reformulation Θ is a partial mapping between $\mathcal{S}^1$ and $\mathcal{S}^2$, that is, a functor which maps formula sets to formula sets and proofs to proofs, with elements U_f for an undefined formula and U_π for an undefined proof, such that for all formula sets Γ of $\mathcal{S}^1$, $\Theta(\Gamma)$ is either U^2_f or a formula set of $\mathcal{S}^2$. Analogously for all proofs π in $\mathcal{S}^1$, $\Theta(\pi)$ is equal to U^2_π or a proof in $\mathcal{S}^2$. Furthermore we assume $\Theta(\mathsf{U}^1_f) = \mathsf{U}^2_f$ and $\Theta(\mathsf{U}^1_\pi) = \mathsf{U}^2_\pi$. We need the elements U since certain expressions in one formal system may have no correspondence in another one.

Then the interesting properties of reformulations like totality, soundness, and completeness are easy to define (for an example look at the end of the next section).

4 Example

In a meta-logic like the one described in the last section, it is not too hard to define for instance, the Natural Deduction (ND) calculus (cp. [5]) for unsorted and sorted first-order logic. Furthermore by proving on a constructive meta-level it is possible to show that to each sorted proof there is an unsorted proof for the relativized theorem.

In an ND calculus for sorted as for unsorted first-order logic you have the axioms (with $j \in \{s, u\}$, s standing for “sorted” and u standing for “unsorted”):

$$\varphi \vdash^j \varphi \text{ (Assumption)} \qquad \text{and} \qquad \vdash^j \varphi \vee \neg\varphi \text{ (Tertium non datur)}$$

For each connective and quantifier, there are introduction and elimination rules. For the connectives they are the same, e.g.

$$\frac{\Gamma \vdash^j \varphi \qquad \Gamma \vdash^j \varphi \rightarrow \psi}{\Gamma \vdash^j \psi} \;\rightarrow\!\mathrm{E} \qquad\qquad \frac{\Gamma \vdash^j \varphi \qquad \Gamma \vdash^j \psi}{\Gamma \vdash^j \varphi \wedge \psi} \;\wedge\mathrm{I}$$

For the quantifiers the rules differ. For instance, in the sorted case we have:

$$\frac{\Gamma \vdash^s \varphi[x/a]}{\Gamma \vdash^s \forall x \in S_{\blacksquare}\varphi} \;\forall\mathrm{I} \text{ with eigenvariable condition and } a \text{ has sort } S \qquad \frac{\Gamma \vdash^s \forall x \in S_{\blacksquare}\varphi}{\Gamma \vdash^s \varphi[x/t]} \;\forall\mathrm{E} \text{ if sort of } t \text{ compatible to sort of } x$$

The corresponding unsorted rules are:

$$\frac{\Gamma \vdash^u \varphi[x/a]}{\Gamma \vdash^u \forall x_{\blacksquare}\varphi} \;\forall\mathrm{I} \text{ with eigenvariable condition} \qquad \frac{\Gamma \vdash^u \forall x_{\blacksquare}\varphi}{\Gamma \vdash^u \varphi[x/t]} \;\forall\mathrm{E}$$

These rules can be expressed in the constructive meta-logic. In the following, we employ a sorted first-order variant of constructive logic, where each term has two sorts, one for indexing the formal system $\mathcal{S}^j$ the term belongs to (here again $j = s, u$

for sorted or unsorted) and one for abbreviating unary meta-predicates like variable, term, or formula. For instance the sorted version of the $\forall$E rule becomes:

$$\begin{aligned}&\forall\Gamma^s_{\mathsf{formula\_set}}\text{▪}\forall x^s_{\mathsf{variable}}\text{▪}\forall S^s_{\mathsf{sort}}\text{▪}\forall\varphi^s_{\mathsf{formula}}\text{▪}\forall t^s_{\mathsf{term}}\text{▪}\forall\pi^s_{\mathsf{proof}}\text{▪}\\ &\qquad\Pi^s(\text{“}\langle\Gamma\rangle\vdash^s\forall\langle x\rangle\in\langle S\rangle\text{▪}\langle\varphi\rangle\text{”},\pi)\wedge\mathsf{sort\_compatible}(x,t)\rightarrow\\ &\qquad\Pi^s(\text{“}\langle\Gamma\rangle\vdash^s\langle\mathsf{subst}(\varphi,x,t)\rangle\text{”},\forall\mathrm{E}(t)\circ\pi)\end{aligned}$$

In the unsorted case we have:

$$\begin{aligned}&\forall\Gamma^u_{\mathsf{formula\_set}}\text{▪}\forall x^u_{\mathsf{variable}}\text{▪}\forall\varphi^u_{\mathsf{formula}}\text{▪}\forall t^u_{\mathsf{term}}\text{▪}\forall\pi^u_{\mathsf{proof}}\text{▪}\\ &\qquad\Pi^u(\text{“}\langle\Gamma\rangle\vdash^u\forall\langle x\rangle\text{▪}\langle\varphi\rangle\text{”},\pi)\rightarrow\Pi^u(\text{“}\langle\Gamma\rangle\vdash^u\langle\mathsf{subst}(\varphi,x,t)\rangle\text{”},\forall\mathrm{E}(t)\circ\pi)\end{aligned}$$

In order to prove the sort theorem, we have to show

$$\forall\varphi^s_{\mathsf{formula}}\text{▪}\forall\pi^s_{\mathsf{proof}}\text{▪}\Pi^s(\text{“}\emptyset\vdash^s\langle\varphi\rangle\text{”},\pi)\rightarrow\exists\mu^u_{\mathsf{proof}}\text{▪}\Pi^u(\text{“}\emptyset\vdash^u\langle\Re(\varphi)\rangle\text{”},\mu).$$

This proof can be given recursively by a case analysis on the applied rules, its computational content constructs from each sorted proof an unsorted one of the relativized theorem.

5 Conclusion

The question of the concrete object language of QED is of course only one among others, but nevertheless a very important one. Since a fixed language seems to be too restrictive, we want to propose not to fix such a language once and forever, but to take a liberal approach where different languages can be linked together. For this purpose we propose a constructive meta-language, in which the formal relationships between the different languages used in QED can be stated and proved. By these constructive proofs, idealist can directly employ the theorems of another theory just by translating the theorem in his own language, while nominalists[1] translate the theorems *and* the proofs in their own formalism. The proposed framework allows easily to integrate existing systems in the QED system.

References

[1] Anonymous. The QED manifesto. In Alan Bundy, editor, *Proceedings of the 12th CADE*, pages 238–251, Nancy, France, 1994. Springer Verlag, Berlin, Germany. LNAI 814.

[2] Jon Barwise and John Etchemendy. *Hyperproof.* CSLI Lecture Notes. Chicago University Press, Chicago, USA, 1994.

[1] For the notions idealist and nominalist in the context of automated theorem proving see [8].

[3] R.L. Constable et al. *Implementing Mathematics with the Nuprl Proof Development System*. Prentice Hall, Englewood Cliffs, New Jersey, USA, 1986.

[4] Michael R. Genesereth and Nils J. Nilsson. *Logical Foundations of Artificial Intelligence*. Morgan Kaufmann, San Mateo, California, USA, 1987.

[5] Gerhard Gentzen. Untersuchungen über das logische Schließen I & II. *Mathematische Zeitschrift*, **39**:176–210, 572–595, 1935.

[6] Xiaorong Huang, Manfred Kerber, Michael Kohlhase, Erica Melis, Dan Nesmith, Jörn Richts, and Jörg Siekmann. Ω-MKRP: A proof development environment. In Alan Bundy, editor, *Proceedings of the 12th CADE*, pages 788–792, Nancy, 1994. Springer Verlag, Berlin, Germany. LNAI 814.

[7] Manfred Kerber and Axel Präcklein. Tactics for the improvement of problem formulation in resolution-based theorem proving. SEKI Report SR-92-09, Fachbereich Informatik, Universität des Saarlandes, Im Stadtwald, Saarbrücken, Germany, 1992.

[8] Francis Jeffry Pelletier. The philosophy of automated theorem proving. In John Mylopoulos and Ray Reiter, editors, *Proceedings of the 12th IJCAI*, pages 538–543, Sydney, 1991. Morgan Kaufmann, San Mateo, California, USA.

[9] Piotr Rudnicki. An overview of the Mizar project. Bastaad, Sweden, 1992.