Manuscript draft, Febr.13, 2015

# A Forgotten Theory of Proofs ?

*E.Engeler, ETH Zurich*

The Hilbert Program in Göttingen was winding down in the early 1930s. By then it was mostly in the hands of Paul Bernays who was writing the first volume of *Grundlagen der Mathematik*. Hermann Weyl had succeeded David Hilbert. There were three outstanding doctoral students in logic: Haskell B.Curry, Saunders MacLane and Gerhard Gentzen.[1] These three students are at the beginning of three threads in mathematical logic: Combinatory Logic (Curry), Proof Theory (Gentzen) and Algebra of Proofs (MacLane), the last one essentially forgotten, except perhaps for some technical results useful in computer algebra (cf. Newman's Lemma).

The present author, also a student of Bernays, when looking up this mathematical ancestry, was fascinated by the contrast between MacLane's enthusiasm about the ideas in his thesis as expressed in 1934 diary quotations, and the almost complete absence of any mathematical follow-up. Is it possible that mathematical development has passed something by, just because MacLane did not find resonance for this work and, back in the U.S., was soon successful, as the strong mathematician he was, in other fields. His work on the conceptual structure of mathematics, category theory and its pervading influence throughout mathematics, is well known.[2]

In this essay we only attempt to revive the idea of an algebra of proofs and place MacLane's

[1] As Bernays was only Dozent (and was soon to be dismissed as foreign, Swiss and of jewish ancestry), Weyl was official thesis advisors, who of course took personal interest until he also left (for Princeton, Bernays for ETH.)

[2] C.McLarty, The Last Mathematician from Hilbert's Göttingen: Saunders MacLane as a Philosopher of Mathematics, Brit.J.Phil.Sci., 58 (2007), 77–112.

thesis work and its vision in a new framework.

## 1. Prolegomena to an algebra of mathematical thoughts

Let us first talk about thinking. Thinking means to apply thoughts to thoughts, thoughts being things like concepts, impressions, memories, activities, projects – anything that you can think about, including mathematics. And, of course, the results of applying a thought to a thought. Thinking is free, all combinations of thoughts are admitted into the universe of thoughts. As a mathematician I perceive here the structure of an algebra: Thoughts are the elements of the algebra and applying a thought $X$ to a thought $Y$ is a binary operation which results in the element $X \cdot Y$, again a thought.

Mathematical thoughts are about sets of *definitions, problems, theorems, proofs and proof-strategies*. In the present context, to do mathematics means to make a selection of such sets, states of knowledge and proof procedures as it were, and apply these sets to each other. To mathematize this idea, we need to represent states of mathematical knowledge and the pursuit of its development in a form that permits an application operation between them. Let us first experiment with formalized mathematics and its states of knowledge.

Mathematical logic aims to represent mathematics by a system based on a formal language. Formal mathematical thoughts thereby consist of sets of statements (axioms, theorems) and proof-trees. Take propositional logic. Let $A$ be a the set of propositional formulas $a, b, c, \ldots$ composed from some atomic propositions by some connectives such as $\wedge, \vee, \supset, \neg$. A formal proof has the form of a tree such as

$$\dfrac{\dfrac{\dfrac{a \quad b}{c} \quad d}{g} \quad \dfrac{e \quad f}{h}}{k}$$

In an obvious notation, this tree would be rendered as

$$\{\{\{a, b\} \to c, d\} \to g, \{e, f\} \to h\} \to i.$$

Such a proof can be *parsed* differently in order to reflect the conceptual structure of the proof – which in fact originally may have progressed through the development or employment of various auxiliary theorems and general lemmas. For example, $g$ may be a lemma and the proof of $i$ starts with this lemma and $e$ and $f$:

$$\{\{\{a,b\} \to c,d\} \to g\} \to (\{\{e,f\} \to h\} \to i).$$

Another parsing would be:

$$\{\{\{a,b\} \to c\} \to (\{d\} \to g)\} \to (\{\{e,f\} \to h\} \to i).$$

Neither denote trees. They are what will be called *proof-expressions* and denoted by lower-case letters such as $x$, $y$, $z$ from the latter part of the alphabet. The set $P$ of proof-expressions is built up recursively from $A$:

$$P_0 = A, P_{n+1} = P_n \cup \{\alpha \to x : x \in P_n, \alpha \subseteq P_n finite\}, P = \bigcup_n P_n.$$

Of course, these "proof-expressions" represent formal proofs only in the case that the arrows correspond to legal steps in a formal proof (here of propositional logic); of this later. The result of the proof denoted by a proof expression $x$ is $x$ itself if it is a propositional formula, an element of $A$; otherwise, if $x$ is composite $\alpha \to y$, it is $y$. We denote the result of a proof $x$ by $x^{\vdash}$, it is a propositional formula.

Sets of proof-expressions are denoted by capital letters $X$, $Y$, ... or by special symbols introduced as cases arise. Such sets represent "mathematical thoughts" in the sense of the introduction to this section, here restricted to the realm of formal propositions.– To complete the picture there, it remains to specify the operation of application, $X \cdot Y$ as follows:

$$X \cdot Y = \{x : \exists \alpha \subseteq Y, \alpha \to x \in X\}.$$

This definition is best understood if $X$ is considered as a sort of graph of a (partial and many-valued) function, each of its elements $\alpha \to x$ associating an argument(-set) $\alpha$ to a value $x$. By this operation the set of subsets of $P$, i.e. the set of mathematical thoughts, becomes an algebraic structure, the algebra $\mathcal{P}$ of propositional thoughts.

*Modus Ponens* is the thought which applied to the set of formulas $\{a \supset b, a\}$ produces $b$. Correspondingly, $[modusponens]$ as an element of the algebra $\mathcal{P}$ contains at least the one element $\{a \supset b, a\} \to b$; we posit that it consist of all elements of that form. Thus, if $X$ is a set of propositional formulas, $[modusponens] \cdot X$ is the set of all propositional

formulas provable from the set of propositional statements in $X$ in one step. Compare this with the usual notation

$$\frac{a \supset b \qquad b}{b}[modusponens],$$

specifying the proof-rule on the right.

More to the point, Modus Ponens can also function as a *proof-constructor*. The corresponding element of $\mathcal{P}$ is

$$[MP] = \{\{x, y\} \to b : \exists a \exists b \in A \quad such \quad that \quad x^{\vdash} = a \supset b, \quad y^{\vdash} = a\}.$$

$[MP] \cdot X$ combines *proofs* of formulas $a \supset b$ and $a$ to a proof of $b$. Thus, the iteration of $[MP]$ produces the propositional theory of $X$ restricted to the one proof-rule.
And so on, to develop propositional logic as the algebraic theory of $\mathcal{P}$, see below.

Instead, we take another elementary example, finitely presented groups:
Let $A$ be the set of terms $u, v, \ldots$ built up from variables and constants ("generators") denoting some elements of a group $G$ by the operations of multiplication, inverse and the unit element. Finite sets of constant terms, called relations, constitute a group-presentation. Based on $A$ we construct a *calculus of reductions* $\mathcal{R}$ starting from the set $R$ of reduction-expressions $x, y, \ldots$ analogously to $P$ above, (most of which of course would not denote valid reductions). Valid reductions are based on laws such as associativity and on the relations given by the presentation:
The associative law, when applied to a reduction-expression $x$, replaces a sub-term of the final term $x^{\vdash}$, assuming it has the form $u(vw)$, by $(uv)w$, or $(uv)w$ by $u(vw)$. Let $[ASS]$ denote this element of $\mathcal{R}$, hence $[ASS]$ is the set of all $\{x\} \to t$, where $t$ results from $x^{\vdash}$ by substituting some sub-term $u(vw)$ or $(uv)w$ of $x^{\vdash}$ by 1. Similarly for inverse law: $[INV]$ replaces sub-terms $uu^{-1}$ or $u^{-1}u$ of $x^{\vdash}$ by 1. The identity law is realized as an operation $[ID]$ on reductions, using replacements of $u1$ or $1u$ by $u$.
Relations $r_1, \ldots r_n$ of the presentation give rise to reduction laws and therefore to reduction-constructors $[r_i[$. For example, if $r_1 = g_1 g_2^{-1} g_1$ with generators $g_1, g_2$, then $[r_1]$ is the set of all $\{x\} \to t$, where $t$ results from $x^{\vdash}$ by substituting some sub-term $g_1 g_2^{-1} g_1$ by 1 .
Example: To construct a reduction (by "normalization") of the term $(st^{-1})t = 1$ we start

with the set $X$, consisting of this term, and use the three operators in succession, resulting in a linear reduction-tree $x$ with $x^{\vdash}$ equal to 1:

$$[ID] \cdot ([INV] \cdot ([ASS] \cdot \{(st^{-1})t\}))^{\vdash} = 1.$$

Taking the closure of $[ASS] \cup [INV] \cup [ID] \cup [r-1] \cdots \cup [rn]$ under iteration as above, we obtain an object $[ALG]$ of $\mathcal{R}$ which, applied to $X$ gives its normalization, $[ALG] \cdot X$ in this finitely generated group.

## 2. MacLane's Thesis and its Vision, Revisited

The above example is from MacLane's thesis *"Abgekürzte Beweise im Logikkalkul"*.[3] It is "abgekürzt", shortened – but more importantly it is a proof-template, a formal object in a proof-manipulating system for elementary group theory, a "Reduktionsbeweis". In the original, it reads:[4]

Anfang Th, Sub (4), Sub (2), Ende (3).

Admittedly, this result of the formalization of deduction processes does not look very impressive. Today; but to actually complete the project, there were tedious and occasionally delicate technical details of substitution, replacement etc. to be handled. In fact, what MacLane did was at the start of a mathematics of symbol manipulations systems which later became computer algebra and computational logic, (cf. normal forms, confluence, etc.). Later in life, MacLane was aware of this[5]

The Logikkalkul of MacLane takes its examples is from the formal logic of *Principia Mathematica*. [6]

The main technical development in the thesis shows how MacLane convinces himself that

[3]Göttingen, Huber & Co.1934. Reprinted in I.Kaplansky (ed.), Saunders MacLane Selected Papers, New York, Springer-Verlag 1979, pp.1 – 62.

[4]in the order of applications, reversed from the operational notation above; "Th" denotes the equation to be derived, "Sub (4)" denotes the application of associativity [ASS], etc.

[5]S.MacLane, A Late Return to a Thesis in Logic, in: I.Kaplansky (ed.), l.c. pp.63 – 66.

[6]A.N.Whitehead and B.Russell, Cambridge University Press 1913. cf. reprinted edition (to *56), ibid. 1962).

his approach how proof theory suffices to treat all of mathematical logic, whose main corpus at that time was *Principia*.[7] But the aim of the development was broader, it was to study all formal and informal proof activities as a mathematical subject. Some of these are mentioned in the thesis, in particular the beautiful lectures of Weyl for transparent non-formal proofs, a book by E.H.Moore, and intuitionism.

The basic insight is that proofs are built up from individual proof operations by composition. In the above example this is Sub (4), Sub (2), Ende (3).These are in a way algebraic expressions in proof-steps; in the notation of section 1 above, this is $[ID]\cdot([INV]\cdot[ASS])$. Correspondingly, short descriptions of logic-proofs use operators corresponding to the introduction or elimination of logical connectives familiar from contemporary proof theory.[8]

By introducing names for proofs of auxiliary theorems MacLane enriches the totality of proof-operators by names for proof-plans. It is clear that he develops the rudiments of a calculus of such expressions for proof-operators.

But now I'm puzzled.

1. Puzzle: Relation to Curry.

Proofs, including proof plans, as algebraic objects with an operation of composition form an algebraic structure which is in fact a *combinatory algebra*, a model of Curry's combinatory logic. Moreover, the formation of arbitrary proofs by combination of proofs corresponds to the basic axiom scheme of combinatory logic. Curry was a "Kommilitone" ( roughly: a fellow-student) of MacLane, whom he remembered in his autobiography as "a good friend of mine from Göttingen". He is not mentioned in the thesis. Had MacLane presented his "Logikkalkul" in the form proposed above, he would have found a model, and therefore a consistency proof, of combinatory logic just by looking at the structure $\mathcal{P}$, the elementary algebraic proof system of section 1. Whether he would have constructed the combinators $\boldsymbol{S}$ and $\boldsymbol{K}$ as explicit objects in $\mathcal{P}$ is questionable. This had to wait almost fifty years to this authors construction,[9] which bases the Plotkin-Scott model of the Lambda Calculus on arbitrary set (useful for applications to non-numeric modeling inter-

[7] as it was for Gödel three years before, (he strangely is not mentioned in the thesis).

[8] H.Schwichtenberg and S.S.Wainer, Proofs and computations, Cambridge U.P. 2012.

[9] E.Engeler, Algebras and Combinators, Algebra Universalis,13 (1981), 389 – 392

active systems, logic programming, [10] and more recently to neural science.[11] $\boldsymbol{K}$ and $\boldsymbol{S}$ are examples of a general algorithm that compiles expressions $\phi(X_1, \ldots X_n)$ to yield a "combinator" $[phi]$ with $((\ldots([phi] \cdot X_1) \cdot X_2) \cdot \cdots X_n) = \phi(X_1, \ldots X_n)$:

$$\boldsymbol{K} = \{\{y\} \to (\emptyset \to y) : y \in P\},$$

$$\boldsymbol{S} = \{\{\tau \to (\{r_1, \ldots, r_n\} \to s)\} \to (\sigma_1 \to r_1, \sigma_n \to r_n\} \to (\sigma \to s)) :$$

$$n \geq 0, r_1, \ldots r_n \in P, \tau \cup_i \sigma_i = \sigma \subseteq P, \sigma \quad finite\}.$$

2. Puzzle: Relation to Gentzen.

Gerhard Gentzen was more than just MacLane's contemporary at Göttingen; indeed he translated MacLane's thesis from his English into the required German, at least in part.[12] And he worked on his own famous thesis at just this time, in which he does also treat of normalization and operations on proofs. What did MacLane know or foresee and communicate on all this with Gentzen, (who is not mentioned in the thesis)? Why did he not work out the algebraic and (computational) aspect of this in something like $\mathcal{P}$ which would give a perhaps convenient framework for such proof-manipulations, e.g. cut-elimination?

3. Puzzle: Gödel ?

Of course, this author is simply hiding a proposed research project on algebraic proof they behind this puzzling. Another obvious topic in sight is the relation of proofs to computations (e.g. the Curry-Howard correspondence) because combinatory algebras (of proofs, for example) are excellent models of computation. We mention Gödel here to signal that the limits of computation (and therefore of proofs) are in view.

[10] E.Engeler, Cumulative Logic and Modeling, Logic Colloquium '86, North-Holland 1988, 83 – 93

[11] E.Engeler, Neural Algebra and Consciousness, Algebraic Biology, Lecture Notes in Computer Science, 5147, Springer 2008, 96 –109

[12] letter of MacLane to Menzler, Febr.1988, printed in: E.Menzler-Trott, Logic's Lost Genius, the Life of Gerhard Gentzen, Amer.Math.Soc. 2007, p.27.