Estratto da

C. Bernardi e P. Pagli (a cura di), *Atti degli incontri di logica matematica* Volume 2, Siena 5-8 gennaio 1983, 6-9 aprile 1983, 9-12 gennaio 1984, 25-28 aprile 1984.

Disponibile in rete su http://www.ailalogica.it

HIGH-LEVEL PROGRAMS AND TYPED VS.UNTYPED CONSTRUCTIVE FOUNDATIONS

Solomon Feferman (Stanford University)

Outline of the talk. The computational significance of constructive mathematics has been present from its inception in the ideas of Kronecker and the work of Brouwer, at least in principle. The advent and wide-spread utility of high-speed computers has made conceivable the step to actual implementation of the algorithms supposedly provided by constructive proofs. Already in 1967 Bishop expressed the hope that "each constructive result can be realized as a computer program requiring minimal preparation and supervision by the operator of the computer" (cf. Bishop 1967, p.355 and also pp. 354-357, as well as Bishop 1970). Despite the subsequent great development of computer science and of powerful "user-friendly" programming languages, this hope has yet to be realized.

Some efforts to move in this direction are provided by Martin-Löf 1979, Hayashi 1983, Beeson 1984a, and Constable and Zlatin 1984, but in each case in a rather preliminary way. The talk was devoted to a discussion of how logical work on constructive formalisms may contribute to these efforts, and what further work is needed. As to the latter, it was emphasized that it is time now to move beyond theory to a number of actual and substantial case studies.

The general problems in this direction are considered to be the following:

1) In what sense do constructive proofs provide high-level programs?
2) Can these be actually implemented?
3) Are the resulting programs feasible?
4) What is the value of doing this?

Actually 4) should be considered first: some of the reasons given were (a) to look at establishing the correctness of programs via the high-level (intelligible) correctness of proofs, rather than the low-level step-by-step "logic" of programs, thence (b) to allow mechanical checking of programs-from-actual proofs, and (c) to establish a hierarchy of answers to users of programs who want reasons for computational conclusions.

The main role of logic here is to give a precise answer to problem 1): logic explains in formal terms just what constitutes a constructive proof. However, the answer is not unique, since it depends on the informal constructive approach which is analyzed, and even then there may be several quite different candidates for formalization. For example, one may take the schools of Brouwer, of Markov and Sanin, of Bishop or others as the point of departure. We have decided to concentrate on Bishop's approach to constructive mathematics (BCM) since it has gone the farthest mathematically and is closest to classical practice while remaining constructively justified.

Remark. The formal systems for Brouwerian mathematics depend for their constructivity on restriction to intuitionistic logic; the extraction of programs from proofs is then generally guaranteed (in principle only) by realizability methods or proof theory. See Feferman 1984b for extensive discussion of this point.

Having settled on BCM as the kind of constructive mathematics to be analyzed logically, there are still alternative formal systems for it to be considered. Among those which have been proposed are (i) a subsystem of intuitionistic ZF due to Myhill 1975 (cf. also Friedman 1977), (ii) a type-free system to due to Feferman 1975 (cf. also his 1979), and (iii) the typed systems ML of Martin-Löf 1975 (cf. also his 1979) [(1)]. Arguments were given against following (i) on fundamental grounds, particularly that the assumption of extensionality in Myhill's system is in conflict with the basic philosophical viewpoint of BCM (cf. the discussion in Feferman 1979,

(1) The system ML was set up on independent grounds, not specifically to formalize BCM-but it has been considered for that purpose.

the extensive discussion in the forthcoming Beeson 1984b and also Cantini 1983).

The talk then turned to a brief description of the systems ML and T_0. The main arguments given against following ML is that it is syntactically cumbersome (because of the aim there to reduce logic to a "logic-free" theory) and more specially because it does not allow the direct formation of sub-types $\{x \in A / \emptyset(x)\}$ in the usual sense. The latter is readily done in the type-free system T_0. Furthermore, T_0 has a subsystem $EM_0 \upharpoonright$ in which all of BCM can be formalized, except for Bishop's theory of Borel sets (subsequently dropped by Bishop in his alternative treatment of measure theory), and is such that $EM_0 \upharpoonright$ is a conservative extension of Peano Arithmetic (PA). Using an abstract form of the recursion theorem for indexed partial functions, it is very simple to derive all needed recursions in $EM_0 \upharpoonright$. However, this requires self-application $\lambda x.(x(x))$, which does not have a direct classical interpretation. Nevertheless, T_0 has both classical models and a model in which very (partial) function is (partial) recursive. The latter is again a guarantee in principle that proofs in T_0 of statements of the form $\exists f \in (\mathbb{N} \rightarrow \mathbb{N}).\emptyset(f)$ establish the existence of computable f.

In the final part of the talk, an alternative intermediate approach via a theory VT of variable types from Feferman 1984a was sketched. This has a direct classical reading, but since self-application is no longer syntactically available, recursion must be built by special axioms. The most general way to do this is still being sought [(2)].

Assuming some reasonable formal analysis of BCM, it seems plausible that the answer to 2) above is positive. Still, this part

(2) VT is also conservative over PA.

remains to be carried out; here is where case studies are sorely needed. (Plans for a project of this sort are being made at Stanford for 1985). However, it is almost certain that the resulting programs are not feasible as they stand. For example, one must constantly apply sorting to find the maximum or minimum of a finite sequence of rational values. The algorithm provided for this (in principle) by Bishop is certainly not as good as that developed by computer scientists. The matter is discussed further in Feferman 1984b, from which the following quotation is taken: "I conjecture that there are a small finite number of 'black boxes' for which reasonably efficient programs have been obtained as a result of work in computer science, and which must be incorporated in the programs extracted directly from Bishop-style proofs in order to make them efficient as a whole. What those black boxes are will emerge only after a sufficient number of case studies".

For further information about type-free systems like T_0, (besides my papers) see Chapters 1,2 of Buchholz, Feferman, Pohlers and Sieg 1983, Cantini 1983 and the forthcoming Beeson 1984b.

References

M.BEESON

1984a "Proving programs and programming proofs", Proc. 7th.Int.Cong. Logic, Methodology, Phil.Sci. (Salzburg 1983) (To appear).

1984b *Foundations of constructive mathematics. Metamathematical studies* (Springer Ergebnisse series) (to appear).

E.BISHOP

1967 *Foundations of constructive analysis* (McGraw-Hill, N.Y.).

1970 "Mathematics as a numerical language". In *Intuitionism and Proof Theory* (North Holland, Amsterdam), 53-71.

W.BUCHHOLZ, S.FEFERMAN, W.POHLERS and W.SIEG

1983 *Iterated inductive definitions and subsystems of analysis. Recent proof-theoretic studies*. Lecture Notes in Maths. v.897 (Springer).

A.CANTINI

1983 *Proprietà e operazioni. Teorie non estensionali delle classi.* (Bibliopolis, Napoli).

R.L.CONSTABLE and D.R.ZLATIN

1984 "The type theory of PL/CV3", *ACM Trans.Prog.Langs. and Systems*, v.6, 94-117.

S.FEFERMAN

1975 "A language and axioms for explicit mathematics" in *Algebra and Logic*, Lecture Notes in Maths. v.450 (Springer) 87-139.

1979 "Constructive theories of functions and classes" in *Logic Colloquium '78* (North Holland, Amsterdam) 159-224.

1984a "A theory of variable types", Proc.6th Latin American Symposium on Logic (Bogota 1981) (To appear).

1984b "Between constructive and classical mathematics", Proc. Logic Colloquium '83 (Aachen) (to appear).

H.FRIEDMAN

1977 "Set-theoretic foundations for constructive analysis" *Annals of Maths.* v.105, 1-28.

S.HAYASHI

1983 "Extracting LISP programs from constructive proofs" *Pub.Research Inst.Math.Sciences, Kyoto Univ.*, #19, 167-191.

P.MARTIN-LÖF

1975 "An intuitionistic theory of types. Predicative part", in *Logic Colloquium* (North Holland, Amsterdam) 73-118.

1979 "Constructive mathematics and computer programming", Proc.6th Int. Cong.Logic Methodology and Phil.of Science (Hannover).

J.MYHILL

1975 "Constructive set theory", *J.Symbolic Logic* v.40, 347-382.