Possible extensions

The limitation of coefficients to integer values is unimportant for many purposes, since the case of arbitrary rational values may often be dealt with by multiplying a polynomial through by the lowest common denominator of its coefficients, leaving roots and other relevant properties unchanged. That this would sometimes lead to rather large integer coefficients is not a reflection on the method itself so much as on the class of problem that is being dealt with, since it is only in simple contexts that truncation or rounding-off can be guaranteed not to lead to intolerable error.

Certain generalizations which at first seem to offer useful facilities turn out, on examination, to involve theoretical problems. One of these, for example, is the introduction of non-integral powers; this immediately destroys the uniqueness of any system of representation, since it is not an elementary matter to decide whether, for example, a given polynomial has a square root.

It would be a relatively simple matter to accommodate the general case of rational coefficients by letting all coefficients be quotients of integers, preserving uniqueness, if desired, by cancelling common factors from numerator and denominator. A generalization of a similar kind would deal with the "field" of quotients of polynomials, rather than with the "ring" of polynomials. The operation of finding the common factor of two polynomials is, however, a relatively time-consuming one, and in many cases it would be wasteful to attempt to secure uniqueness of representation by this means.

Certain generalizations of another kind—namely, to the use of complex numbers and vectors—are relatively straightforward and may be introduced as applications demand.

The wealth of applications awaiting these developments justifies some attention to methods of securing the highest possible speed of operation. The provision of some special hardware should not be ruled out. In this connection the algorithm described might prove especially suitable as a basis.

References

Brown, W. S., Hyde, J. P., and Tague, B. A. (1962–3). "The ALPAK system for non-numerical algebra on a digital computer", *B.S.T.J.*, Vol. 42, pp. 2081–2119; Vol. 43, pp. 785–804; Vol. 53, pp. 1547–1562.

Collins, G. E. (1966*a*). "PM, A System for Polynomial Manipulation", *Comm. ACM*, Vol. 9, No. 8.

Collins, G. E. (1966*b*). "Subresultants and Reduced Polynomial Remainder Sequences", *IBM Research Paper No. RC*-1595.

Hamblin, C. L. (1962). "Translation to and from Polish notation", *The Computer Journal*, Vol. 5, pp. 210–13.

Seidenberg, A. (1954). "A New Decision Method for Elementary Algebra", *Ann. Math.*, Vol. 60, pp. 365–74.

Tarski, A. (1948). *A Decision Method for Elementary Algebra and Geometry*, Rand Corporation; 2nd edn., Berkeley and Los Angeles, 1951.

Williams, Leland H. (1962). "Algebra of Polynomials in Several Variables for a Digital Computer", *J. ACM*, Vol. 9, pp. 29–40.

Automata Theory, edited by E. R. Caianiello, 1966; 342 pages. (New York: *Academic Press Inc.*, 112s.)

This volume consists of some thirty papers presented at a NATO Summer School in 1964 at Ravello. Although it is hardly, as its title might suggest, that systematic survey of a broad field which we are still lacking, it undoubtedly contains much that is worth the attention of the computer scientist, whatever his line of country. Büchi and Rabin, for instance, provide clear expositions of finite automata theory, mainly from the abstract algebraic standpoint. McCulloch and Harth speculate on the latest brain models based on "a more realistic neuron". A few admirable pages by Martin Davis clarify in simple terms what recursive function theory is about and how it concerns automata. At the other end of the road, some of the more bizarre propositions of that theory, such as that any program may be indefinitely accelerated (as long as you are prepared to ignore a finite subset of the domain), are here extended to logics by Michael Arbib. Problems of formal linguistics, also inseparable from automata theory, are the subject of authoritative contributions by Schützenberger and others from the *Institut Blaise Pascal*; it is perhaps a pity these were left untranslated. In a long and interesting paper, Böhm and Gross introduce their description language CUCH, being an amalgam of CUrry combinatory logic and CHurch lambda-calculus. CUCH is a powerful language of wide potential application, as is also the system of Generalized Normal Algorithms described by Caracciolo di Forino.

Something has gone very wrong with pages 119 to 121 in Caracciolo's otherwise lucid paper; at this price should the reader have to work quite so hard to restore the original text?

M. Bell