International Journal of Theoretical Physics, Vol. 2, No. 3 (1969), pp. 297–299.

Computers and Physical Axiomatics

J. YATES

Chemistry Department, University of Salford, England

Received: 31 *March* 1969

Algorithmic techniques of various kinds are almost universally used in mathematical physics. These are frequently supplemented, in the case of fundamental physics, by a few of the elementary techniques of formal logic. In most cases, the semiotic aspect of the subject is not considered explicitly. This may be justifiable in principle if one has arrived at a short set of mathematical axioms from which the results of the theory under consideration may be deduced mathematically in a direct manner, though any interpretations of terms and connectives left without semantic comment until such axioms are obtained are in general likely to be unrewarding from the point of view of a directional semantic discussion of physical concepts. For instance, typical summaries of the lattice-theoretic approach to quantum mechanics assume that the sign '$\supset$' means 'implies'. Reference to the literature of formal logic gives a large number of different ways 'implication' may be mathematically formalised (Curry, 1957), none of them (of course) entirely satisfactory as formalisations of the U-language term. There is no immediate reason why the term 'implication' should be forced into the description of a physically complex theory like quantum mechanics. One can quote Kleene (1962); '"Implication" is a handy name for '$\supset$'. In using it, we follow the practice common in mathematics of employing the same designation for analogous notions arising in related technical theories. An example is the many different kinds of "addition" and "multiplication" in mathematics'. The easiest way out is to regard 'implies' as a label for '$\supset$', and to endow that label with a U-language significance whenever one feels inclined to do so.

However, if semiotic problems are ignored in this way, there are problems involved if one wishes to deduce a set of mechanical rules which will allow the axioms of a theory to be deduced from given experimental data. As well as the obvious difficulty that it is hard to define connectives properly if one does not know what they are intended to mean in the U-language, there is also the difficulty that until the metasemiosis problem has been considered a primitive frame with a clear U-language significance is unlikely to be obtained. In particular, since connectives '$\vee$', '$\wedge$', and '$\supset$' appear to be basic connectors in many schemes of deduction, they may also arise as adjunctors in the axioms if a distinction (Curry, 1957) is not made between connectors and adjunctors. The doing of this is a specific piece of formalism, but the reasons arise through metasemiosis.

Further, in the case of a set of axioms obtained by mechanical rules, it may not be possible to assign any semantic meaning, however flimsy, to the axioms generated if one awaits their existence before assigning meanings to their components and to the generating rules. Indeed, to gloss over the semiotics implies largely ignoring the U-language infrastructure which has so far been a chief means of linking one scientific theory with another in interdisciplinary fields.

The approach to computer axiomatics presented here therefore uses Curry's (1957) primitive frame, with the following additions.

111. *Theorems*

C. *Order of application of rules*

These rules specify that the rules must be applied in such an order that, given certain terms in the system which represent experimental data, the rules 111A arise. Once this has been done, the rules 111B are deristricted as to order of application.

D. *Rules of computer operation*

These in turn delimit 111C.

Thus, when specifications 1, 11, and 111 are made explicit, one has a represented formal system. To begin with, one defines the morphology in such a way that the experimental data can be represented. In the present exercise, all elementary propositions were of the form '— = —'. For a given automaton, the definition of 111D is implicit already, and one defines 111B and 111C in such a way as to be useful for any usual morphology. There is plenty of choice in the logical literature for 111B, and the formalism of the Gentzen L-system rules, including negation and quantifiers, was adopted. 111C is of course more difficult to determine, but the general approach is to generalise specific expressions by quantification, to check the truth or falsity of the axioms thus obtained by generation of all expressions which may be generated from them, and to try several routes to a set of axioms which will generate all the given data, and to choose the set of axioms which contains the minimum number of terms. Some of the difficulties involved are discussed and a more detailed resume is given elsewhere (Yates, 1969).

It is interesting that falsifiability is virtually essential, in practice, if one is to avoid inconsistency, but that this falsifiability is not immediately equivalent to the lattice-theoretic notion of complementarity, and occurs in 111B as

$$\frac{A \qquad \neg A}{F}$$

where A is a proposition, and F is a minimum refutable proposition. It would appear that formalisms which in effect split 111B so that parts of it are included in the U-language and parts of it in 111A are likely to run into difficulties on this point if used creatively in the development of axioms.

A computer program has been developed using the methods outlined above. It allows the development of axioms containing up to ten variables and twenty quantifiers to be deduced, using a possible total of eighty variables (Yates, 1969). It can give short, plausible results for simple problems, such as the motion of a particle under a constant applied force. Numerical distance (d_i) and time (t_i) data, fitted into a morphology conceived only for the ends of formalisation and not specifically toward certain axioms, produce the following axiom

$$\forall(d_i)\, d_i = \lambda t_i^2$$

(d_i, t_i) are the (distance, time) pairs and λ is a specific numerical constant.

These simple trial calculations suggest that a substantial escalation of the amount of computer time required does not appear to be an inevitable concomitant of more advanced work along these lines.

References

Curry, H. B. (1957). *A Theory of Formal Deducibility*. University of Notre Dame.
Kleene, S. C. (1962). *Introduction to Metamathematics*, p. 139. North-Holland.
Yates, J. (1969). Lithographed notes.