PHILOSOPHY AND THE COMPUTER

L. APOSTEL
Gewoon hoogleraar R.U.G.

1. *INTRODUCTION*

In the well known periodical " Computers and the Humanities" (1) we find, anno 1968, pp. 245-46 under the heading "philosophy" a list of a series of important undertakings : the "Index Thomisticus" (by Roberto Busa, perhaps the first work in the field); the concordance to Plato by Leonard Brandwood, the concordance to Kierkegaard and Wittgenstein undertaken by Alisdair Mac Kinnon.

Use is made of the computer to list key words of the more or less voluminous writings of the mentioned authors and to enumerate their contexts. Given the intensity of German work on Kant it is not astonishing that Gottfried Martin started recently a similar work with reference to Kant.

At Ohio State University, we see the beginning in 1969 of an automatised philosophy information retrieval Center, indexing philosophical journals and serving the philosophical profession as a whole. Lineback's venture is listed in 'Computers and the Humanities 1969'.

All these efforts are similar to the ones present in other fields of the humanities. Only the peculiarities of philosophical style could be of interest and could distinguish these efforts from similar concordances and automatic indexing efforts in other humanistic sciences.

As far as we know, no new results for computer science have been acquired through these undertakings, and no sensational discoveries in the history of philosophy have been made by these means. This is no argument against them : the work is only beginning and, it seems obvious that whenever a large amount of text is available that repays intensive study, automatic concordances should be prepared for it. We expect that this will happen for all the major philosophers of history.

In this article it is however our aim to suggest that the cooperation of the computer sciences and philosophy could be made more intimate : *it is possible to study certain philosophical problems by means of computer simulation.*

We shall try to suggest possibilities that are very close to present day work and also possibilities that are far away. It is in theory of knowledge that the more immediately interesting applications of computer simulation could be found and for this reason the bulk of the paper will concentrate on this field. But we

Extrait de la Revue (R.E.L.O.)
VII, 1 à 4, 1971. C.I.P.L. - Université de Liège

do not wish to give the impression that other fields of philosophy (e.g. metaphysics and ethics) cannot be fruitfully studied by means of this tool. We were somewhat hesitant to present these more remote approaches, but finally it seemed more important to open up some new roads, than to appear cautious and respectable.

Before coming to our real topic that implies rather *a different type of cooperation of computing science and philosophy than the one present in the litterature,* we want to stress that the concordances now automatically prepared could be made use of in order to help solving two major systematic problems :

a) what are the distinctive features of philosophy ? how does philosophical writing differ from scientific or litterary writing ? A sampling technique could be applied to the concordances, and programs could be sought that would be able not only to list the contexts of terms but also the contexts of sentences, in order to analyse the forms of argumentation in Thomas, or Plato or Kant. These forms of reasoning should then be compared with selected texts of scientists or litterary writers, working in the same period. Various tasks of interest, difficult and important, would have to be undertaken a) programs able to analyse the structure of sets of sentences would have to be developed. The tradition of computational linguistics could help here b) sampling procedures within texts and of texts within periods would have to be developed c) comparability of philosophical and non-philosophical texts should be established, and comparability relations should be defined.

We realise that this type of undertaking, using the automatic indexing methods to solve the main problem of *meta-philosophy,* is something radically different from the useful but restricted concordances that are now being prepared. Still we want to suggest that these tasks can be undertaken, and that they would contribute, on an objective basis, to the famous demarcation problem that can only be solved by means of an objective analysis of representative samples of philosophical texts.

The methodological problems are very large. We must ask :

1. How to delimit our unit of analysis : where does an argumentation begin, and where does it stop ? Can we use only lexical criteria (if so, which ones ?) or also syntactical criteria (if so, which ones ?) ?
2. How to handle ambiguity, extremely important in philosophical texts ?
3. How to handle the problem of hidden premises (can we find out by comparison that a certain proposition is tacitly presupposed, if so, how can we execute this comparison ?) ?
4. How to summarize our findings ? Two strategies could be followed on first sight : we could take a 20th century logic textbook; observing that the logical constants "and", "or", "if then", "not", "all", "some" are crucially important we would try to develop an *empirical logic* by concentrating on the pre and post contexts (of what length ?) of these words and by comparing the rules of our text books, with the contexts of our constants. This procedure, not without interesting difficulties itself, offers however the danger of anachronism. The second strategy defended by Dr. Tombeur, consists in analysing a text by means of a logic textbook of the same period (here the danger of anachronism is not

as large, but we must be aware that the often purely semantic definition of "subject" or "predicate" in classical logic texts precludes operational use of it).

The usefulness of computer analysis of the properties of linguistic equivalents for logical constants lies first in the necessity to write the program.

For instance, when will a text *apply* the rule "If p and q then p" ? Should the program verify 1. If p has been twice present in a sentence of a given type (indicative), 2. If p was first followed by q, and in the second occurrence not, 3. If between the two occurrences no intermediaries of a certain type occurred (for instance the withdrawing or refutation of a set of earlier assertions including p, or a sentence of the form "as r, and r implies p); 4. If such a configuration occurs not only once for given types of sentences, but repeatedly (how many times ?) for sentences as with varied lexical content ?

This is one proposal, only to be verified with difficulty (the operation 3 is obviously *very* incompletely defined). Another proposal would be that the rule occurs explicitly as mentioned and that also some other assertion is justified (at least once ? or many times ?) by means of it (linguistic criteria for "justification" must be given).

As it is not our main purpose here to analyse the application of the ordinator to philosophical texts but to ask ourselves how we can use it to solve philosophical problems, we shall not pursue this topic further. Let us only repeat

a. only by means of the ordinator can we hope to build an *empirical logic* (we need too many data to do it otherwise).
b. even the construction of the programs for an empirical logic analyser will already be of fundamental interest.

History of philosophy also, if it wants to be more than a collection of summaries and of unsystematic comparisons of various systems, chronologically ordered, should try to find laws of development : this task demands the study of philosophers in order of filiation, of masters, students and students of students, trying to find the laws of transformation. It can be undertaken by simply applying comparative concordances, diachronically ordered, in such a way that the most frequent key terms are followed in the works of philosophers influencing each other.

It is our considered opinion that the methodological study of these problems and the testing out of some tentative program for the demarcation problem, for the historical problem and for an empirical logic of philosophic reasoning is more advisable than the multiplication of unconnected concordances.

But the decision to be taken here is perhaps a decision of scientific policy. We realise that the programs to be developed would be of another order of complexity than the ones in use in the preparation of a concordance (without underestimating the intricacies present even there).

The aim of our paper however is not, as we said, to develop in some sense the programs already under study at the present moment, but to suggest the possibility to undertake the working out of philosophical problems by means of the computer.

2. *THEORY OF KNOWLEDGE AND COMPUTING SCIENCE*

One of the main controversies in the history of philosophy is the controversy between rationalists and empiricists.

Dr. F. Vandamme suggested to us that we should ask ourselves if the computer could not help us to analyse this controversy. In discussion with Dr. De Mey, we saw clearly that we had to consider an infinity of types of empiricism and rationalism; that we had to avoid the more extreme types and had to study the properties of theories of knowledge that are mixtures of empiricism and rationalism.

We saw Piaget's work as one of these intermediaries between rationalism and empiricism and we thought that it would be possible to simulate his main hypotheses on a computer and to test out their efficiency by means of this simulation. It appeared that this could be done in many different ways and the fruitfulness of the simulation seemed to be seen already in this preliminary stage. But the confrontation of rationalism and empiricism is not the only epistemological problem that we can undertake by means of the computer.

The conviction of the best known rationalist of our period, Noam Chomsky, is that an innate language competence exists, explaining language learning, but that other tasks (perception learning, motor learning asf.) need other, perhaps equally innate, competences. This "regionalism" of Chomsky is akin to the old hypothesis of pluralisms of scientific method, while to this pluralisms we could oppose a monism.

It is our opinion that these opposite views also could be simulated and tested out on the computer. Marvin Minsky has explained in a well-known article, that the dualism between introspection and extrospection can be modelled in the programming of a computer able to develop a model of his own structure. So the mind-body problem could equally be simulated.

Finally, the age-old contrast between realism and idealism can also be approached, when concept-formation tasks are given that either can build up input classification by starting from program or sub-program classification, or can build up program classification from input classification. This idea, when worked out would give us a comparison between a "realist" and an "idealist" program.

We ask the reader to have patience, if these short initial remarks fail to convince him that a system with inputs could yield a model for an idealistic theory of knowledge, or that a mechanical system can be used to simulate the controversy between spiritualism and materialism.

It is precisely the main point of this paper that the use of the computer is neutral with reference to the four oppositions mentioned : rationalism-empirism, pluralism-monism, spiritualism-materialism, realism-idealism.

We shall try to make this point in the following paragraphs, but before doing this we have to clarify one main distinction between simulation in the theory of knowledge and simulation in the psychology of intelligence.

In computer epistemology, we want to study the set of *all* programs able to solve so called "intelligence" problems. In computer psychology we want to study the *specific* programs that solve "intelligence" problems in the *human mind.*

Computer epistemology stands to computer psychology as mathematics to physics. To speak in Kantian terms, in the first discipline we study the necessary or sufficient conditions of all knowledge or of all types of growth of knowledge, in the second discipline we study the specific properties of humanknowledge. The affinities are as clear and certain as are the differences.

3. *THE COMPUTER SIMULATION OF THE OPPOSITION BETWEEN RATIONALISM AND EMPIRICISM*

Classical empiricism considers the human mind as a "tabula rasa". All information present in it is introduced from the outside by means of perception and organised by a few very simple learning methods : association by contiguity, by contrast or by similarity. In such a way relations are introduced between the elementary impressions that are the inputs to the system.

"Nihil est in intellectu quod non prius fuerit in sensu" (2)

Classical rationalism to the contrary, claims that much information is stored in the intellect (at the limit that even all information is stored in the human mind). The only function of experience is to make us aware of this information.

One could think that a computer is, by definition the embodiment of rationalistic principles for the following reasons

a) it has given predetermined input facilities,
b) it has a given predetermined memory space,

c) it can perform only a certain number of actions, commanded by the instructions of a program that is determined from the outside,
d) finally it can only yield certain specific types of output.

Yet we think that we can show convincingly that on all these points we can contrast an "empiricist" computer with a "rationalist" computer.

A. As to the *input,* let us introduce the fact that by means of pattern recognition, the inputs are stored in the memory as members of given form classes. It is obvious that the pattern recognition subroutine can either be strictly determined (similar for instance to the well-known program proposed by MacCullough for neuron networks : n operations of a given group are performed upon all inputs and whenever by means of one of these operations two inputs are mapped upon each other they are stored as identical), or, quite to the contrary, that the inputs are stored as such (first case) or that learning strategies are present that either determine in function of certain events, what operations are to be performed, or even, what new operations, function of the earlier ones, are to be tried out.
The empirist opponents of the MacCullough pattern recognition program are the perceptrons. The "Perceptrons" were studied most intensely by Rosenblatt. A perceptron is a network of nodes with variable strength of connections. The strength of connections grows when the nodes are simultaneously excited and when it is strong enough, the connected nodes are all excited when one is excited and they produce the same output. The empiricist character of the model is clearly visible when the initial connections are *random* ("tabula rasa" ! !) (3).

Another version of the rationalist pattern-recognition idea is the Selfridge-Neisser "Pandemonium" program (4) that analyses given inputs into elements that can be binary coded. In parallel processing a form is analysed into a multiplicity of fixed features. No new features can be engendered.

It would seem to us that complete rationalism is not realisable in a computer (it would have no input), but that at least a strong approximation to strict rationalism is realisable, for the input data, and that the degree of approximation to rationalism depends upon a) the unconditionality and b) the invariability of the pattern recognition (or if one prefers, classification) procedures.

All possible degrees of unconditionality and unvariability are available. It is important to realize that certain compromises between empiricism and rationalism are embodied in a pattern recognition model that is widely recognised as an improvement upon both the Pandemonium-MacCullough, or the Perceptron strategy. We shall do 2 things.

1. Describe the program and point out the compromises between rationalism and empirism present in it and
2. Show the utility of philosophy for computer science, and of computer science for philosophy by pointing out a large family of pattern-recognition systems of the same type, that constitute also

compromises between rationalism and empirism.

The program for pattern recognition we want to present here as one of the many compromises between empiricism and rationalism is due to Uhr and Vossler (5). The input to be analysed (in their case always a written letter) is subdivided into cells. Groups of cells are operated upon by operators, an operator being a I-0 matrix representing certain significant features (for instance an intersection of vertical or horizontal, or a line segment from right to left). The degree of coincidence of the cell block with the operator is stored according to a predetermined set of indices. Until here the rigidity is perfect but now two learning strategies are added, the one an empiristic one, the other a rationalist one : when a given operator has not been useful to distinguish or identify inputs for a given time interval, it is discarded; when it is very useful it is either amplified, or small random variations of it are produced as new operators. The rationalist operator on operators implies that combinations and analyses of operations are produced, and when applied are preserved if they have sufficient discriminatory utility.

The compromise between rationalism and empiricism is here obtained by means of the following mechanisms : a) the initial operators are given (innate-rationalism) b) but they can be modified (variability-empirism) c) in two ways : one by frequency of success and random variation (empirism) and d) by systematic variation and frequency of success (rationalism *and* empiricism).

We want now to point out that we could have started with random divisions of the input and with random operations (perceptron like) in order to obtain by means of the same learning mechanism another compromise between rationalism and empirism.

Moreover, just as we could modify the operators, we could also modify the way in which we modify the operators, and iterate this process (thus repeating our synthesis of the opposite view-points on successive levels). It could moreover be arranged that the predominance of the systemic or of the empiric variations would be changed : more systemic or more empiric predominance on higher levels.

The construction of this sketch of a hierarchy is proof of what we wanted to defend : that it is usefull to recognise the general structure of the rationalism-empiricism opposition in the situation we are concerned with, because it allows us to discover its motivation and moreover the large number of possible generalizations.

B. Let us now consider another important feature : the *memory structure.* When the inputs are classified they must be stored for future use. How is the organisation of the memory ? We can consider the memory as a static system, hierarchical in nature, in the parts of which the information units are stored. Or, we can have a memory that is very loosely structured, in which the information units are stored as nodes of relations referring not in one but in many ways to other information units. In both cases there is some predetermination : the distinction between the ways of linking the units together that is present in the second case is an a priori feature and the whole of hierarchy is equally such. We can mention concrete examples of such memory structurations : the Quillian associative memory as an example of the loose and undirected structure; the memory as a list of lists in the first case We would consider the degree of approximation of the memory organisation to a tree like graph as a measure of its approximation to rationalism (6).

A second time we must direct the attention of the reader to the fact that the two extremes cannot be realized : one must trade stability and hierarchization of the memory for strength and variability of the search subprogram. Indeed : the more doors to the memory we have, the less directed and systematized it is, the less complexity must the information retrieval operation have; the more systematic hierarchy there exists in the memory, the stronger, more adaptable and more complex search routine we must provide for.

C. This brings us to the following point : the "empiricist" and "rationalist" computer can be contrasted with each other also in the structure of their programs. Here it is obvious that rationalism always wins on some level : some superprogram must always remain invariant; but this superlevel can be very high : if this is the case, one approaches empiricism; it can be very low : in that case one approaches rationalism. We call here "height" the number of levels on which modification of a program as a result of its execution may occur.

D. It is certainly possible to show the same oppositions also in the organization of the output format, and in the organization of the immediate memory, the working tape. But we think that after pointing out the opposition on the level of input reading, memory structure and program modification we can consider out point as evident, namely a) the classical opposition between rationalism and empiricism becomes a gradual opposition (an undeterminate number of intermediary positions can be distinguished) and b) the classical opposition becomes a multidimensional opposition (the remark made with reference to the input, the memory and the program is a different remark; and it seems very important to us that we could at least once come to the conclusion that we have to buy stability and order in one subsystem by allowing variability in another subsystem. This situation could, according to us, be generalized).

E. We have now to show that the age old discussion between Plato and Aristotle, between Stoics and Empiricists, between Leibniz and Locke can be taken up again by means of simulation.

Our point of view will be essentially pragmatic : we have to test more or less rationalist or empiricist computers on a given set of problems, and if the problems are better or faster solved by one type of computer than by the other, then we obtain partial confirmation for one of our points of view.

Two important difficulties appear : a) what types of tasks should the computers be presented with ? b) and against what type of practical organization of the empiricist or rationalist computer should they be matched ?

F. The first question should be answered as follows : the tasks should be problem solving tasks in which combinatorial exhaustion of possible solutions is not possible; at least four types of tasks should be present : theorem proving tasks; finding or constructing tasks; perceptive classification and ordination tasks; and language decoding tasks. The difficulty will be to find a computer that will be able to receive programs and inputs for these different types; they can however be reduced to similar presentations (and then we have to watch that their difference will not disappear).

G. The second question should be answered as follows : it is obvious that we have to give both to the empirist computer and to the rationalist computer some input, memory and program routines. It should not however be the case that the routines we give them are already sufficient to decide the competition in their favor or against them. This is really the main difficulty that we did not yet see the possibility to overcome : a perceptron-like automaton will certainly not be able to play checkers as well as Samuels very specialized program (7). But this is by no means a decision in the controversy we are presently engaged in : the one is constructed for perception like tasks of small magnitude and has minimal structuration, while the other is an extremely specialized mechanism of high structure. Certainly the one is of the empiricist type, the other is of the rationalist type but we should be able a) either to combine the two possibilities in one program b) or to diminish them and bring them closer together so as to compare them in the middle range c) or to apply the one to the tasks of the other.

Our difficulty here is rather the following one : we can give the same task to two programs only if they have the same input facilities; but the one should have a strong classification, a rigid storing and an invariable program while the other has plastic classification, less organized storage and modifiable program. Both should have programs and memories of such a type a) that the task can be performed b) but *that the type of rigidity or of variability is not determined by the features of the tasks furnished.* The experiment is conceptually possible; but in view of its intellectual difficulty and in view of the infinity of intermediary degrees between the two extreme positions, we should only try out significant intermediary positions.

Here we have one guide : one of the systems intermediary between rationalism and empirism has been empirically developed : Jean Piaget's system.

4. *JEAN PIAGET'S "GENETICAL EPISTEMOLOGY" AND ITS COMPUTER SIMULATION*

We saw in the earlier parts of this paper that systems intermediary between classical empiricism and rationalism had to be investigated.

The examples mentioned indicate that development should be built in : the ways according to which innate patterns and external informations combine should be transformable by learning processes.

One example of such a system is to be found in the developmental theory of the intellect, presented by Jean Piaget. This theory is the experimental implementation of philosophical views coming from Brunschvicg and Baldwin by means of experimental techniques inspired by Claparède and Janet. We are thus still remaining within the limit of philosophy and are not confusing philosophy and psychology.

It is interesting to see that in Apter's book "The Computer Simulation of Behaviour", we read the following sentence "it would be worthwhile to attempt to simulate all Piaget's stages of cognitive development in terms of a program consistent with his own theory" (p. 81) (8).

In the following paragraph we want to show a few possibilities in that direction.

The rationalist in Piaget is to be seen when he states that the organism has, in the beginning stages of its development, a series of schemes of action, schemes that are applied to all stimuli, irrespective of their properties.

The concept of *"scheme"* is a difficult one; we propose to translate it by means of the concept of *"subprogram"*. A sequence of subprograms are stored and named as units in the initial memory and these subprograms are applied whenever a stimulus is present, whenever an input is given.

The empiricist in Piaget comes to the foreground however when it appears that these schemes will either assimilate the external stimuli to themselves, or will accomodate themselves in order to adapt to the intrinsic nature of the stimulus.

These concepts are very difficult ones. The following transcription may be proposed.

Let us consider that a given subprogram has associated to it a series of inputs to which it can be applied and let *assimilation* be the wiping out of the real input symbol, and the replacement of it by one of the associated inputs to the subprogram, the one that is, according to a given similarity measure, closest to the input given. The subprogram is then applied to this transformed input.

Accomodation to the contrary means the following : the input remains what it is. If it is not present in the associated set of any of the stored subprograms, then an analogy perceiving subprogram is applied, producing a subprogram description that stands in the same relation to the input in question as a stored subprogram to at least one of the members of associated set.

Evans' analogy perceiver can here be applied, not to input pairs but to pairs of input and subprogram descriptions. The subprogram generated is then applied to the input in question (9).

In the initial situation Piaget states that there is no *"equilibrium"* between assimilation and accomodation. This general statement has to receive a specific implementation if programming is to be possible (10).

We select that following one : in the initial state either at random or in regular oscillation pure application of subprograms (schemes) in isolation (one subprogram applied to one input) is mixed with accomodation and assimilation. In the memory are stored the consequences of pure application, of assimilation and of accomodation. A memory searching program is then introduced that must register by pattern recognition applied to the series stored, if the series is in equilibrium or not.

The fruitfulness of programming an epistemology appear very clearly when we see how many different definitions of equilibrium we could apply. We can say that a series of n acts (the n has to be chosen) is in equilibrium_1 if there is an equal number of the three types of behaviour; we can say that such a series is in equilibrium_2 when the same input has been repeated in such a way that it has been submitted an equal number of times to each of the three types of subprograms; finally (and this is presumably closest to Piaget's intention) we can say that there is equilibrium_3 if the series of n acts is such that after an initial segment of length r (the length has to be chosen) all the inputs appearing afterwards belong to the associated sets of the subprograms stored at that moment and the accomodation and assimilation programs have not to be applied any more.

In the theory as expressed in Piaget it was not clear that the selection of these very important parameters was needed in order to make the program work. The utility of simulation demonstrates itself by showing that we have to specify much more than we did before.

If however development is to occur, then the equilibrium must be disrupted. How is this going to occur ?

We propose the following scheme that seems to us close to the author's intentions but that is certainly neither the unique possibility nor the complete specification.

In the memory the inputs that have occurred must be stored as inputs upon which a given sequence of subprograms has been executed. When equilibrium of a temporary kind is obtained we must admit that after a given length of time t (t to be fixed if the program is to work) whenever this input appears the stored memory image is evoked and new accomodation or assimilation measures become necessary because this complex description is no longer already contained in the associated sets of the subprograms stored.

Remembering that Piagetian development is always coordination of actions that had initially occurred uncontrolled by each other we could propose the following instruction : when an input appears accompanied by the descriptions of the series of subprograms that have been applied to it, then all these subprograms are evoked simultaneously. The ordinator must then introduce some type of parallel processing of all the subprograms, or if it remains sequential, must apply segments of the subprograms followed by segments of other subprograms. By means of one of these two stipulations the equilibrium will be ruptured and coordination will be reached, if new subprograms are generated (a subprogram to do this must be present in the computer and must be made to work at this stage) that are combinations of instructions of earlier subprograms (the execution of the instructions of on SP made dependent upon the execution of another one).

As a final essential step we must program Piaget's concept of *"interiorisation".* An interiorisation of a subprogram is the transformation of this subprogram into another one of the same type that is however not applied to the inputs, but to a representation of sets or series of inputs or of subprograms stored at a given place in the memory by means of a symbolic representation. There are thus very many degrees and types of interiorisation. (As many degrees as there are levels in the control tree of the program).

The simulation program must be written in such a way that *the progress of coordination and of interiorisation are linked to each other.*

We want to test the simulation in a *pragmatic manner* : for what types of initial subprograms, for what types of selection of the parameters, for what types of definitions of coordination, of equilibrium and of interiorisation, will given types of tasks be well and fastly learned by a P-program (a Piaget simulator program) ?

It will already be very clear that Piaget's work is not a theory, but is, quite to the contrary a set of theories. This becomes clear in a constructive, and not in a destructive way, even in the preparatory stage of computer simulation that we meet in the sketch presented here.

Let us now mention that this theory of development by means of coordination, interiorisation and equilibration is one member of an infinite set of theories of development.

It would be essential to study the whole set; the computer can here be the essential tool.
It is by no means our intention to say that the only intermediaries between rationalism and empiricism worth studying are P-type theories.

We might consider C-type theories in which the inputs are not stored as targets of given subprograms that are being applied to them but as objects to which certain names are attached. Here the communication process would take the place of the action process as basic source of descriptors.

We do not wish however to do more here than point out this possibility.

The basic choices to be made in the simulation are a) the selection of the initial schemes b) the definition of the operator that will disrupt the equilibria c) the definition of the final stages, if there are any.

We might perhaps at the end of this paragraph express our views on the relationship between a theory and a simulation program.

An empirical theory is in general a set of formal theories that could be obtained by precisation of the concepts left undetermined in the empirical theory. For any formal theory moreover, there is at least one Turing machine that will yield as output all the theorems of the theory (and in general there will be many such machines). We call a computer program "computer equivalent" to a theory if it is a program of a Turing machine generating the theorems of the theory. The proposals contained in this paper go out from the following conviction : when an empirical theory is given, it is a good heuristic strategy to try to simulate it upon a computer. The program obtained will then be a good starting point for the formalization and precisation of the theory in question.

This is the first reason for our recommending computer simulation of epistemological theories. The second reason is that a large variety of such theories have to be tested out as to their success and behaviour, in the long range (i.e. : in view of their behaviour at the end of longer periods). This can only be done with the help of mechanical tools.

5. *THE MIND-BODY PROBLEM AS SIMULATED IN THE COMPUTER*

Repeating an assertion made by Marvin Minsky (11) we consider a hypothetical computer having the ability to represent general features of its input until a certain moment. We shall call this ability the world picture of the computer. Let us equally suppose that an image of the computer itself, of his structure (possible operations, control section, input and output facilities and memory) and of his function (history of the content of all these subsystems) can be stored in the memory. The Mind-Body dualism refers to the relation between the image of the computer and the image of the world, both constructed in the computer. Let us call these images IW and IC.

The introspective image would be an analysis of the structure and functioning of the computer from the point of view of the highest control and evaluation operations of the system (means ends analysis decision process and overall evaluation of success or failure of plans would be prominent there : as they are indeed in any type of introspection). The extrospective language would also contain an analysis of the computer but seen from the point of view of his lower level functioning : his calculating processes and his input-output relations.

Various hypotheses are possible : if IW contains an IC on low level and if moreover the C contains an IC on high level then 1. It might not be necessary or useful to relate these two pictures to each other or 2. it is necessary and useful to translate the two images into each other for some types of tasks (or for all types of tasks).

We are in complete disagreement with Gilbert Ryle in his "Concept of Mind" when he claims that the distinction between introspective and extrospective language cannot be upheld. Quite to the contrary even, it seems to us a problem to be studied by computer simulation in *how far the two types of self-representation are to be translated into each other.* Our hypothesis would be that we need at least two different types of dualism 1) the dualism between a structural and a functional description and 2) the dualism between the low level and high level description. The two types of dualism are not necessarily related to each other in all, or even in most tasks.

It seems to us that we can think of many tasks that need one self image rather than the other and that would be more disturbed than helped if the various self images would intervene. For this reason we agree quite completely with Marvin Minsky when he claims in the article in question, that operational dualism cannot be overcome and that computers, for very unmentalistic reasons will have to behave as if a basic difference between various self representations were never to be overcome. We only deviate from Minsky's opinion when we assert that a multiplicity of divergent dualisms is needed in the self representation of the computer, and when we would propose for experimental study the search for tasks in which after all combinations and translations (partial or complete) of such self images would be more useful than continued separation.

6. *PLURALISM AND MONISM IN THE COMPUTER*

Noam Chomsky's rationalism (12) is in a sense a typical : the tendency towards unification, so strongly present in classical rationalism, is less prominent in his. Indeed he accepts the presence in the human mind of an innate competence for language learning but he rejects the idea that a general innate competence for problem solving or thinking would be present, the application of which to language would yield the languages we see before our eyes.

It is our claim that if we have no specialized mechanisms but, to the contrary "general robots" that have to solve a variety of intelligence tasks, then programs that would allow the interaction and transfer of subprograms coming from widely divergent tasks would have much larger probability for success. To be more specific : if innate capacities, representing information about the universe are present in the nervous system of a living being it is probable that these adaptation patterns reflect general features of the environment (like for instance : the propensity to look for bundles of objects, for stable configurations, for simple symmetries) and not very specific and highly specialized skills (like the structure of human languages). We would propose to test out on the "general robot" (13) Chomsky's implicit claim that it is better to give a series of specialized skills to the program than generalized capacity of ordering the inputs of any problem according to predetermined schemes, applicable in any region.

7. *REALISM AND IDEALISM IN THE COMPUTER*

Again it might seem that the computer idea implied realism : without input it is unable to function. The reader will perhaps be reminded of a similar situation when it appeared to be the case that the computer could not function as an empiricist. We saw very soon that this was not true.

We think we can propose various simulations of the opposition under discussion here :
1. Classical idealism represents the world (here the set of inputs) as a development of consciousness.
2. Classical realism represents the world to the contrary as independent from consciousness.

An "idealist" program would have the following feature : the instructions would be to develop first a representation of the computer itself. This task could be done in very many ways : a pattern recognition analysis of the memory or a syntactical analysis of the central controlling program would be performed. Whenever an input would then be introduced from outside, it would only be taken account of as representing a certain stage (or feature of a stage) of the developing self-representation of the program.

A "realist" program to the contrary would have the following features : (here we must be very careful not to identify realism with empirism; we are going to mould our definition in such a way that the realist program is quite to the contrary, rather strongly rationalist) : it would not start with developing a self representation, but would start with developing a representation of fragments of the input tape. Then it would develop a self representation, in function of the features of the input tape in such a way that the differences between the eventual mechanism that would have generated the input tape and the program that is now analysed would be maximized (all due account taken of the hard data however). This divergence-maximalization could have been built in however in the idealist program as well (the Fichtean program could be as well simulated as the other idealist systems : we have not yet forgotten a significant remark made by Seymour Papert a few years ago, asserting that it is more useful for computer science to read the German Idealists than to read Hume !).

We hope that the computer scientist will not think that we have the illusion that the projects for research described in the earlier paragraphs are sufficiently clearly stated to be undertaken without much more work. We only hope that he will see that significant oppositions coming from age old discussions can be fruitfully exploited in present day research.

We want to conclude the epistemological part of our paper with the following two remarks :

a) the spirit in which this paper is written implies a rejection of the point of view, sometimes present in analytical philosophy or in logical empiricism and according to which the history of philosophy can be discarded : quite to the contrary, we think that the basic controversies of the past are significant enough to be taken up again, with vigour
b) but our claim that they can and should be studied by means of the construction of computer programs is the introduction into philosophy of an experimental philolophy. *Philosophy should be as much an experimental science, as logic should be an empirical science.* Experimentation however is not to be limited to the observation of the spontaneously given intelligent systems, but can be broadened in a very significant fashion, by means of the new tool at our disposal : the ordinator program.

Let us finally stress that this experimental spirit implies that it can not and ought not to be the case that the computer simulation of epistemological problems implies already a decision in one or other direction. To be very explicit : that the computer is a material system is of no theoretical but only of practical importance : it is clearly given and controllable. That is why it should be used. The simulation of epistemological problems depends essentially on the structure of the programs; it is the testing out of structures. The way in which these structures are realized is of no fundamental importance.

The author of this paper would identify himself as a) realist b) materialist c) monist d) hybrid of rationalist and empirist with strong rationalistic tendencies and e) historicist. But it is his claim that these options are by no means implied by the research program sketched before, research program that could be important and useful for people calling themselves spiritualists or idealists.

8. *THE SIMULATION OF ETHICAL PROBLEMS IN THE COMPUTER*

Next to theory of knowledge, ethics is a traditional part of philosophy. The topics of this science are difficult to define. Very naively one could say that it is the science studying what should be (ought to be) done. But the operational meaning of "ought to be" is not clear. One thing is evident : human beings are all goal pursuing systems, having goals of wide generality and also of great specificity and ethical discourse functions as an instrument to coordinate goal pursuing behaviour within the individual or within the group.

Functionally we could say that ethics is an attempt to discover methods to overcome conflicts between goals. The computer should be used to simulate various types of conflicts and various types of solutions for them to say a) how stable the solution is b) how much transfer is possible from one solution method to another, in the individual and in the group. Rodney M. Coe in his article "Conflict, Interference and aggression: computer simulation of a social process" gives a first example of the programming of conflict (14).

In order to show the reader who is interested in ethics that ethically important experiments can be done by way of computer simulation, we want to use John C. Loehlin's work "Computer Models of Personality" (15).

The model is well known : objects are represented as series of digits. Emotional attitudes are equally represented, in various intensities : love, fear and anger are three basic attitudes chosen. Three types of actions, represented by the features of outputs, are possible in concordance with the three basic attitudes approach, attack, flight. Actions have consequences that again provoke emotions. The emotional reactions produced in the past have an influence on the action and moreover the attitudes towards an object modify, but weakly, the attitudes towards the classes to which this object belongs.

Loehlin does show us that, by changing the numerical values of some of the program's parameters, ethically significant differences can be introduced a) an important ethical parameter is : influence of general conceptions and far away memories in comparison to the influence of the immediate situation : one can manipulate this variable in the Loehlin model. b) Another important ethical parameter is the amount of love or of aggression present in an individual : power ethics and agape ethics can be contrasted by means of this distinction. It is possible to program the computer in such a way that it tends towards the maximization of the love value (positive eudemonism) or towards the minimization of the fear and aggression value (negative eudemonism) or towards the maximization of the aggression value (power maximization). It is also possible to program the computer in such a way that it tends towards a maximization of a function of all three basic inbuilt emotions.
All this is not ethics and Loehlin's work does indeed hardly go far enough to reach our domain.

The only original proposal we have to make here is *to combine the simulation of problem solving with the simulation of personality.* We would then give various problem solving tasks to the computer and we would arrange things so as to give various emotional values to the informations to be used in the problem solving tasks.

Again our pragmatic criterion would have to be used to evaluate general rules of behaviour : are the tasks faster and better solved if one general ethical rule is given to the emotional computer or if another ethical rule is given ?

This field of philosophy is not the domain in which the present writer did his main work and so he cannot speak with as much conviction as elsewhere, but it seems to us that the same transformation of speculative philosophy into experimental philosophy that can be introduced into epistemology can also by these means (we repeat : by the, not yet realized and here proposed combination of the simulation of problem solving with that of emotional behaviour) be introduced into ethics.

9. *THE SIMULATION OF METAPHYSICAL PROBLEMS IN THE COMPUTER* .

We wish to come to the conclusion of this paper by making a very paradoxical remark : it is in the field of metaphysics that for obvious and very elementary reasons, the computer has to make its main contribution.

Indeed, what is metaphysics ? It is the attempt to bring all we know about the universe together into one system, and to introduce as many connections into this system as we possibly can.

Why has metaphysics fallen in disrepute ? Because no living person is able to collect the diversified information needed to construct a metaphysical system. Why is the computer used in engineering or in economics ? Because it can do what no living person can do : solve large sets of equations with many unknowns. The reason why it is introduced in these complex sciences is exactly the reason why it has to be introduced in metaphysics : only the computer can handle the enormous amount of data needed to develop a unified world view.

What could then be done in this field ? For those who know that even the large memories at our disposal are much to small for certain very specific tasks our proposal will perhaps seems preposterous. Yet it is not : we should at least begin by building a syntactical and semantical analyser and comparator of formal and semi formal systems. This is a major task, but not a task in principle different from those undertaken by computational linguistics. Then we should feed axioms and central theorems of different sciences into the computer, and let the program discover as many as possible formal or semantical analogies and deductive or inductive relations. Iterating this process is to our mind the only means to overcome the present underdevelopment of metaphysical thought.

We could here perhaps try to overcome the astonishment of the reader by pointing towards the often proposed necessity of countering the information explosion of our time by means of a general data bank. The proposal has been made for the behavioral sciences but it can be generalized, and will probably be implemented within certain fast growing strongly connected domains (nuclear physics).
No longer will libraries hold the dispersed information about a given field, but a general memory will be constructed, by means of interconnected computers, in which the organized information will be stored. Any new information coming and accepted after using certain criteria will have to be stored as addition

or modification of the general information network.

For reasons that are by no means speculative, the general aim of the unification of science that is the core itself of metaphysics, is present, though in disguise, in the center of practical deliberations.

10. *CONCLUSION*

This is a programmatic paper. We know very well that one is ill advised when making ambitious proposals. But we have taken pains to introduce into all parts of the paper specific suggestions that can be judged on their merits, and that can, in the hands of better technicians than we are, lead to the aim we pursue : an experimental philosophy, that is however the direct continuation of the secular tradition. Peirce is our great example here.

(1) *Computers and the Humanities,* Queens College, City University of New York, 1966 sqq.

(2) For a classical discussion between empirism and rationalism, see G.W.F. LEIBNIZ, *Nouveaux Essais sur l'Entendement Humain.* This polemic with Locke, the empirist defends the rationalist point of view. The best known expression of radical empiricism will be D. HUME's *Treatise on Human Nature.*

(3) F. ROSENBLATT, *Principles of Neurodynamics,* Washington D.C., Spartan books, 1962. Also F. ROSENBLATT, *The Perceptron : a probabilistic model for information storage and organisation in man.* Psychological review, November 1958.
In Marvin MINSKY and Seymour PAPERT's recent book *The Perceptron,* the formal study of the perceptron is pushed forward.

(4) O.G. SELFRIDGE and U. NEISSER, *Pattern Recognition by Machine.* Scientific American, August 1960, 203, 2, pp. 60-68.

(5) Leonard UHR and Charles VOSSLER, *A pattern recognition program that generates evaluates and adjusts its own operators,* in : FEIGENBAUM and FELDMAN, *Computers and Thought,* New York, MacGraw Hill, 1963.

(6) *Semantic Information Processing,* M.I.T. Press, 1968, M. MINSKY ed. – The two types of memory are present in the work; a more rationalistic one in Bertram RAPHAEL : *SIR semantic Information Retrieval* (pp. 33-134).
The more empiristic one in M. Ross QUILLIAN, *Semantic Memory,* pp. 216-270.

(7) A.L. SAMUEL, *Some studies in machine learning using the game of checkers* in FEIGENBAUM and FELDMAN, op. cit.

(8) M.J. APTER, *The computer simulation of behaviour,* London, Hutghinson University Library, 1970.

(9) Thomas G. EVANS, *A Program for the solution of Geometric Analogy Intelligence Test Questions,* in : MINSKY, op. cit., pp. 271-353.

(10) J. PIAGET has expressed his views in many publications. As a first introduction, the recent *Biologie et Connaissance* might be used, but one should consult also his papers in the series *Publications of the Center for Genetic Epistemology.*

(11) Marvin L. MINSKY, *Matter, Mind and Models,* in : MINSKY, op. cit., pp. 425-432.

(12) N. CHOMSKY, *Cartesian Linguistics,* London, Harper & Row, 1966.
N. CHOMSKY, *Language and the Mind,* New York, Harcourt, Brace and World, 1968.

(13) For the idea of the "General Robot", consult RAPHAEL, *An integrated robot system,* in : A. MORRELL, ed., *Information Processing 68,* Proceedings IFIP Congress, 1968, vol. 2, Amsterdam, North Holland Publ. C°, 1969.

(14) Behavioral science, April 1964, 9, 2, pp. 186-197.

(15) John C. LOEHLIN, *Computer Models of Personality,* New York, Random House, 1968, p. 40.