Mental Images and the Architecture of Concepts

J.A. Makowsky[1,2,3]

[1]Department of Computer Science
Technion - Israel Institute of Technology
Haifa 32000, Israel
e-mail: janos@cs.technion.ac.il,

[2] Department of Mathematics
University of Lausanne
Lausanne - Dorigny, Switzerland

[3] Mental Images G.m.b.H.
Berlin, West Berlin

Appeared in: *The Universal Turing Machine. A Half–Century Survey*, R. Herken ed., Oxford University Press, 1988, pp. 421-432.

Abstract. We discuss the role of format of data , communication protocols and notational systems in computing. We use the term mental images for the internal data structure of human thinking and formal concept for an externalized data structure associated with it. Format, communication protocol and notation together are called architecture of the concept. It is argued that our current model of computability abstracts too much from the issue of architecture of concepts to provide us with a workable theory of interactive computing and data transfer. Such a future theory would have to take into account aspects of data base theory, cryptographic protocols, probabilistic complexity theory and a theory of learning which extends the statistical theory of estimation of dependencies based on empirical data. We also draw attention to anthropological studies concerning the evolution of mathematical concepts which show that such evolutions are inherently slow. This last aspect serves to dampen our hopes of a quick break through in the evolution of intelligent computing.

In memory of my mother
Marika Erzsebeth Makowsky-Deutsch
14.10.1923- 9.7.1987

1 Introduction

Some fifty years ago a major discussion in the foundations of mathematics seemed settled. The intuitive notion of computability was made precise. It was equated with the mathematical notions of Turing Machine computability, recursive functions and *lambda*-definability (Church-Turing Thesis). By now many other models of computation have been suggested and all of them have been shown to be extensionally equivalent or weaker than any of the above. This is generally taken as evidence that we now understand the notion of computability. But this should not be taken as evidence that we understand the notion of computing as an interactive and distributed activity. Turing based his model of computability on mechanistic aspects of numeric computations and string manipulations of one computing device, be it man or machine. Some of these early models (Turing machines, register machines) were also used for the design of the first hardware. Especially von Neumann advocated the use of universal Turing machines as concepts underlying the physical realization of computers. It was suggested by Trakhtenbrot [Tr1988] in this volume that other early models of computability such as recursive functions or *lambda*-calculus should be viewed as software oriented models. If we push this thought a bit further, then the Church-Turing thesis also stipulates the equivalence and interchangeability of the two notions computable by hardware and computable by software by a device which has the computational power of a universal Turing machine. From Turing+s point of view it is fair to say that most issues of computability are today fairly well understood.

However, all these models of computability capture only aspects of computability which reflect the use of string manipulation machines in batch mode. They are inadequate to describe the manipulation and transfer of concepts without explicit reference to the coding of these concepts. They are inadequate to describe various modes of interaction between man and machine and between machines and they are especially inadequate to describe the influence of the choice of particular data structures and data representations on the complexity of computations. The abstraction from these aspects of computability accounts at the same time for the versatility of these models of computability as well as for their uselessness. The present level of computing activities with various user interfaces, virtual machines, networks and the search for new architectures justifies a new look at the basic issues.

The purpose of this paper is to discuss three recent developments in theoretical computer science and to sketch how they possibly contribute to a better understanding of computing as an activity in an interactive context. The three areas discussed are relational data base theory, learnability of concept classes and the analysis of interaction protocols when concepts are passed from one actor (man or machine) to another.

A system of actors, each equipped with a concept class, is called a cultural system. Such systems are subject of study in anthroplogy and such an anthropological view of the mathematical activity was initiated by R.Wilder in his book RMathematics as cultural systemR [W1981]. In a last section we shall briefly outline how the study of man- machine interaction as a cultural system can further our understanding of computing. The paper is meant to address a wider audience. Technicalities are avoided as much as possible. However, the reader interested in the more technical aspects should be able to fill in all the technical details by consulting the cited references.

2 A guiding example

We are all familiar with the game of guessing sets or sequences of natural numbers from a given finite set of examples. Given the numbers $2, 4, 6, ...$ we are usually expecting a continuation $8, 10, ...$; given $2, 4, 8, ...$ we expect $16, 32, ...$ etc. Underlying such expectations, however, are concepts such as arithmetical or exponential progression, and it is obviously very easy to find other continuations with very reasonable explanations. The sequence $2, 4, 8, ..$ e.g. can be continued with $15, ..$ by referring to the three dimensional euclidean space and the maximal number of regions one gets by cutting it with $1, 2, 3, ...$ two dimensional planes. However, a student who will propose $15, ..$ as a continuation of $2, 4, 8, ...$ in an Intelligence Test would probably fail, because the author of the test did not include the possibility of explaining the answer. On the other side we would be inclined to consider $15, ...$ together with its explanation as a more sophisticated if not more intelligent answer than $16, ...$

When we speak of concepts in our daily usage of the word we might have several things in our mind. What matters here is that having a concept of something means having a set of instances of the concept. In the first example above the instances are $2, 4, 6, ...$ and the concept is the set of natural numbers of the form $2n$ for $n > 0$; In the second example the instances are $2, 4, 8, ...$ and the concept is the set of numbers of the form 2^n for $n > 0$ or the set of numbers $f(n)$, where $f(n)$ is the maximal number of regions one gets by cutting the three dimensional euclidean space with n two dimensional planes. The expression $2n$ represents the concept of even numbers but itself it is again an instance of something. More such instances can be given, such as $3n$, $4n$,.... and now the reader would be inclined to continue with $5n$, $6n$,... which can be represented by the expression an, where a and n play different roles. At this stage we might look at the sequence $3, 5, 7, ...$ which most people would continue with $9, 11, ...$ which cannot be represented by an expression of the form an and we would be lead to expressions of the form $an + b$.

In a similar vein we can also look at pairs of natural numbers, such as $(2, 4), (6, 8), (10, 12)$...... represented by an expression of the form $(2+4n, 4+4n)$ for $k >= 0$ or $2 + n - m = 0$ and $m = 4k$ for some $k > 0$. Usually the former type of a solution is considered more desirable for its explicitness. In the case of triples as indicated by $(3, 4, 5,), (5, 12, 13)\dots$ and represented by all triples such that $a^2 + b^2 = c^2$ finding an explicit representation was considered a major success of mathematics in its period.

The history of mathematics teaches us several lessons concerning the evolution of the concepts of a set of natural numbers and a real function.. For one the concept class "sets of natural numbers" and "real functions" was frequently enlarged. Each such enlargement gave rise to a crisis in mathematics and fierce debates among mathematicians till the new concept class was universally (or almost universally) accepted. For many mathematicians the set of all natural numbers which are Goedel numbers of true formulae of their favorite model of set theory is not a well defined set of natural numbers. If sets of natural numbers have to be computable this feeling is justified by Goedel's theorem. But still we tend to think that we have a mental image of such a set (at least I do) even if the structure (architecture) of the underlying concept is not clear. I would be hard pressed if I had to describe my favorite model of set theory in an unambiguous way.

3 Notational systems

In the above examples the sequences of numbers are usually given names such as $S_1, S_2, ...$ which are treated as atomic names and are not informative at all in the sense that they do not help to think about these sequences. However, it is the naming of such sequences which mark considerable progress in our understanding them. Let S_1 be the sequence $2, 4, 6, 8, ...$ If instead we name it $2, +2$ we may indicate by this that the sequence starts with 2 and each consecutive element is obtained from the previous one by adding 2. Alternatively we could name it $2n$ indicating by this notation that the elements of the sequence are all the even numbers, but now we have to decide whether 0 is an admissible value for n. Notation clarifies our understanding, but we cannot conclude from the absence of notation a lack of understanding. In Diophant's work no notation for zero and the negative numbers is present. Philologists concluded from this that the Greek of Diophant's time did not have the concepts of zero and the negative numbers. But mathematicians reading Diophant's work with paper and pencil concluded that he must have been aware of zero and the negative numbers because otherwise the gaps in his arguments could not been filled. Notation thus makes our reasoning more explicit and, more important, notation is a carrier of thought. The Babylonians wrote $(3 + 5)^2 = 3^2 + 2 * 3 * 5 + 5^2$ and were completely aware of the generality of this statement which would not be clear had they written $(2 + 2)^2 = 2 * 2 + 2 * 2 * 2 + 2 * 2$ since 2 plays three different roles in the latter.

Notation plays a fundamental role in our thinking and in programming machines as well. Notation has inherent limitations which discipline our thinking, but it also has its own temptations which make us go beyond the intended. Very often creativity consists in yielding to this temptations and stumbling blocks in understanding come from not trusting notation. It took the physicists of the beginning of this centure almost twenty years to believe that matrix calculus really furthers the understanding of quantum mechanics, even if not every step has an obvious physical interpretation.

4 Computing vs. learning

Computing deals with the manipulation of strings within a fixed notational systems. A class of objects is computable if there exists a notational system such that a Turing machine can generate their names. Computing does not deal with the creation of notational systems. Neither does it deal with the mechanism of how we (or machines) assimilate the meaning of a notational system. This rather trivial remark seems to be fundamental when we want to speak of the mechanization of intelligence. When Kepler discovered his laws of planetary movements he had availbale a host of data. He tried to format these data by making several fundamental decisions. For one the planetary movement was to be a function of time, the initial position of a planet, its speed and mass. It was not to be dependent on local conditions on earth such as weather or politics, though both weather and politics were considered in his time to be influenced by the position of planets. Once he had a hypothesis concerning the format of the data, he still needed a notational system to represent the planetary movements in form of equations. Contrary to their implicit claim, P. Langley and H. Simon's project "BACON" of mechanization of physical and chemical discoveries [LBS1983 in MCM1983] only deals with the last stage of discovery, given the data format and the notational system, how could Kepler determine the actual equations of planetary movements. This last stage is called in the literature learning by example, but I think this name is misleading. It is more precisely described by calling it identification of concepts of a given format by examples. Research in this area was popular in the East Block countries already decades ago, as documented in [HH1978, V1982].

5 Mental Images and the architecture of concepts

We might use the term "mental images" as describing the data structure of our mental reasoning. We do not have access to this data structure before we are able to externalize it. We do not know its exact format, as we do not even know the format of the data we perceive. I do have a mental image of what a beautiful woman is (or of what artificial intelligence should be) to the extent that I can recognize her, Rachel, (I can recognize a project as not being in AI, such as chess playing programs) though I can not define my concept. I want to introduce here the term mental image for concepts as internal data structures of our mind and (formal) concept as the externalized version of the mental image. For a concept to be formal it is not enough that we can recognize its instances. A formal concept has a format and a notational system which displays this format. The format, though, does not necessarly display all the aspects of a formal concept, it may on purpose hide some of them. A formal concept is an abstraction of a mental image, it is usually a finite dimensional projection of an infinite dimensional image. We shall use the term architecture of a concept to mean format and notational system together. Incidentally, this may also put the rivalry between intuitionism and formalism into a new perspective: the intuitionists insist on displaying all the manipulative aspects of a concept, such that existential statements become constructive, the formalists allow hiding them to an extent that only those aspects needed for a particular argument are on display. In this perspective the axiom of choice is just an encapsulation of a possibly complicated argument. The above discussion can easily be made more concrete in several brief examples.

6 The role of format and notation in music

Most people have a rather clear mental image of some kind of music relevant to their culture, even without having had explicit musical training. But very few people are aware, even among musicians, to what extent their music is dependent on the existence or absence of notational systems for it. An anecdote attributed to the great linguist R.Jacobson may illustrate the issue. It is said that once in Prague Jacobson set up the following experiment. An excellent flautist of the renowned orchestra, who was particularly known for his capacity to play after hearing, was invited to join an African expert in the playing of some tribal wind instrument resembling the flute in many respects. The two were supposed to proceed as follows. The African was to play a short piece of music on his instrument, then the European was to play on his instrument what he thought he had heard. If the African agreed, the session would stop. If not, the African had to play the same piece again, so the European would have a further chance. No verbal communication was to be used, except to express assent or dissent. Here is what supposedly happened. The African played and so did the European. The audience was excited about how well he had repeated the piece, but the African suggested that the European's version was not even similar. He then played, what he thought was the same piece, but it was unrecognizable for the European audience, and so they had to continue for many more rounds. Ultimately the two musicians found agreement. It turned out that for the African neither pitch nor rythm were part of the format which determines the equivalence class of different performances of the same piece, but the only relevant property in common of all the performances he had given was in the quality of the attack of each individual sound. Such music could not have been written in our conventional notational system. But to our ears even the format was wrong. Our ear learns to hear and overhear by a format which is culturally created. No culturally innocent person would consider a Wagner opera played by an orchestra or its piano excerpt version the "same". This very notion of piano excerpt has very much to do with the fact that the piano keyboard has become the "universal Turing machine" of classic and romantic music of western Europe. There are no piano excerpts of Gamelan music ! And many new tendencies in twentieth century music have very much to do with the extension of the format of sounds, though the corresponding developments of notation lag behind. In other words the mental image of this new music has not yet been completely converted into a formal concept.

7 Relational data bases

The above discussion can easily be made more concrete by the example of relational data bases. Relational data bases define a user interface of data in which the data format is well defined and various query languages (sc. notational systems for extracting information from the data bases) can be precisely defined. A very good exposition of the theory of relational data bases may be found in [U1983, Mai1983, CH1980 and the review thereof Mak1987]. The relational approach to data bases might suggest that one can model by a formal concept our mental image of a data base in the sense that exactly the information which can be explicitly extracted is the information we can really obtain. This may be wrong both on the level of the formal concept, inasmuch as implicit information need not be equivalent to explicit information even in the mathematical sense of the notions. A.Zvieli and I have proved, cf. [Mak1984], that the many-sorted implicit queries in the notational system of first order queries are exactly the computable queries in the sense of Chandra and Harel [CH1980]. But also informally, in the sense that the computable queries over the disjoint union of two data bases form in general a larger class of queries than the union of the computable queries over two data bases. The latter remark is important for the notion of data security and data privacy, e.g. even if we can prove that the data basis of the police and the data basis of the tax authorities each do not disclose certain information, it may be that software being able to use both may disclose it. More generally, what we can learn by our brain having simultaneous access to several mental images or even formal concepts is different from what we can learn from each alone. This may give us one clue to the different learning capabilities of different humans in otherwise very similar situations.

8 Communicating concepts and communication protocols

In contrast to human communication machine communication allows the copying of data in an unambiguous way, if we abstract from the down to earth intricacies of the communication channels and communication networks. One can copy data and formal concepts from machine to machine but not mental images. In contrast to machine communication humans can communicate mental images by a mechanism which is not yet completely understood: Communication by example. The advantage of this is clear, a finite set of examples is presented to the other and in return the other will form a mental image, which will more or less capture the mental image of the first. The "more or less" is important here, severe misunderstandings are quite frequent in daily life and possibly less frequent in well defined areas of interest such as legal disputes or mathematics. Another aspect is important here, which was described in the section above on playing music of different cultures: The communication protocol. A riddle familiar among mathematicians illustrates this further: A mathematician A meets, after a long time his colleague B. After exchanging formalities A reveals to B that he has three daughters and asks B to guess their age (integers). As a first hint he says that the product of their ages is 36. B thinks of all the ways of decomposing 36 into three factors and complains to A that there are too many possibilities. A then gives as a next clue that the sum of the ages was equal to B's house number. Now B does some calculations and after a while reports that the ages were still not uniquely determined, whereupon A, as a last clue says, that the eldest very much liked bananas. The example clearly gives the distinction between computation and hinting by example. The observer C of that story can also guess the ages, even without knowing the house number, but only once he understands that he has to do all the computations to understand the significance of the last clue. B himself also might have modelled the problem first in the integers with multiplication, then also with addition and only at last equipped with a linear ordering. If he did so, then he understood, that each clue not only revealed new information on the ages, but also about the format needed to reason about them. Reasoning is not an activity by an individual (man or machine), but a two or rather a many actor game between mind and the external world. It should be noted that the reasoning minds form one or many cultural systems and their functioning cannot be separated from this. Reasoning consists of introspection and reading of external data. The metaphor of the sole reasoning mind derives from the special case, where the mind assumes that he can simulate all possible reading of external data. This metaphor is similar to the idea of a universal turing machine. A mathematician can reason about the natural numbers, because the reading of the "empirical" data about them is simulated internally or can be computed on virtual internal workspace, i.e. on paper by pencil. Turing, when proposing his model of computation had exactly this in mind. But reasoning about external data is different from computing. We can query the external world by measuring certain aspects (not necessarily numerically), and for this we have to invent methods of measurement . The point I would like to make here is that methods of measurement involve not only determining data about the object to be measured but also a protocol of how these data are to be obtained. In more complex situations this protocol is interactive like in questioning a witness in court [ES1979] or in verification of digital signatures [G1988]. This protocol also presupposes something about the computational capacities of the measured object. Measuring the specific weight of the king's crown by measuring the volume involves immersing the crown in water and assuming that the water can "compute" the volume of the crown. Inventing methods of measurement, inventing the right question, again involves format, notation and computation within this notational system. That something is computable means that there is a notational system in which it can be computed. This hidden existential quantifier in the definition of computability reminds us of the difference between deterministic and non-deterministic computing e.g. in polynomial time. Computing relates to thinking like P to NP.

But this analogy is misleading, because in the former we can convince ourselfs that the hierarchy is proper and unbounded. If we wanted to equate reasoning with first order reasoning, then this hidden existential quantifier can be made explicit by passing to second order logic. The problem which remains, is that the domain of this hidden quantifier remains always vague. We do not have, so far, a general definition of format, notation, protocol, and worse, had we one, it could be easily transcended by some kind of a diagonal argument.

9 Cultural systems

The evolution of notational systems for number systems was studied from an anthropological point of view by R. Wilder [W1950, W1968, W1981]. His studies deserve attention especially when one has in mind the evolution and development of programming languages, operating systems, user interfaces and other paradigms of computing. His studies clearly show several phenomena: that the evolution of concepts to widely accepted norms of practice takes much longer and need more than just the availability of such concepts; that the evolution of concepts is not due to individuals but is embedded in a (or several competing) cultural systems which are themselfs embedded in host cultural systems; that nevertheless the fame and prestige of the protagonists of science and scientific progress do play an important, possibly also counterproductive role; that cultural stress and cultural lag play a crucial role in the evolution of concepts; that periods of turmoil are followed by periods of consolidation after which concepts will stabilize; that diffusion between different fields usually will lead to new concepts and accelerated growth of science; that environmental stresses created by the host culture and its subcultures will ellicit observable response from the scientific culture in question; and, finally , revolutions may occur in the metaphysics, symbolism and methodology of computing science, but not in the core of computing itself. Wilder has developed in [W1981] a general theory of "Laws" governing the evolution of mathematics, from which I have adapted the above statements. It remains a vast research project to assimilate Wilder's theory to our context, but it is an undispensable project, if we want to adjust our expectation of progress in computing science to realistic hopes. Wilder's work also sheds some light into the real problems underlying the so called "software crisis": The cultural lag of programming practice behind computing science and the absence of various cultural stresses may account for the abundancy of programming paradigms without the evolution of rigorous standards of conceptual specifications.

10 Computing vs. intelligence.

Our generally accepted model of computability is rather robust, but still fixes format, notation and protocol. Format and notation are abstracted from by coding into strings (numbers, etc.) and the protocol reduces to putting data into registers (writing it on tape), performing the fixed operations and reading the data. It allows us to analyze the resources needed for a computation and to prove negative results. It allows us to analyze what is computable within fixed formal concepts. It also can lead us in extending the notion of computability to other data types, as in [Sh1988, CH1980, DM1986]. But it does not give us a means to discuss the creation of data, the passage from mental images to formal concepts, i.e. what Turing [T1939, cf. F1988] called intuition and ingenuity. The last twenty years of research in computing (rather than computer) science seem to indicate growing awareness of the role of the choice of data structures and different protocols of data creation and data exchange between machines and between man and machine. Especially research in mathematical cryptography, complexity theory and learning theory [G1988, Val1984, BEHW1986, BI1988] all invoke statistical concepts to model aspects of computing which are abstracted from in Turing's model. An anthropological look at the history of mathematics and the evolution of mathematical concepts teaches us caution in our expectation of a break through in our expectations of intelligent machine performance. It may well be that within a couple of hundred years the cumulative experience of man-machine interaction will lead to an integration of man- machine intelligence. It may well be that programming by example will evolve as the ultimate mode of man-machine communication. But as much as children need ten years of constant exposure to language and writing to reach minor literacy and our cultural system needed several hundred years to reach it,the computer population will need a similar stretch of time to overcome the problems of its infancy. And it is safe to say, that the conceptual (both practical and theoretical) groundwork for intelligent man-machine interaction has just started. I hope this paper will stimulate research which will integrate these various lines of thought into a global theory of conceptual vs. computational programming.

Acknowledgments:

I wish to thank J.Tal for all he has taught me in music during our joint research on notational systems for electro-acoustic music, R.Herken, J.B.Neidhart and O.Wiener, for their continuous challenge making me think about these problems, and to R.Herken, J.B.Neidhart and R.Zucker for encouraging me to write that paper and come to an end with it, though I am fully aware that much more could and should be said.

References

[BEHW1986] A.Blumer, A.Ehrenfeucht, D.Haussler and M.Warmuth, Classifying learnable geometric concepts with the Vapnik-Chervonenkis dimension, Proceedings of ACM-STOCS '1986, pp. 273-282.

[BI1988] G.Benedek and A.Itai, Nonuniform Learnability, to appear in the proceedings of ICALP '88, Springer LNCS 1988.

[CH1980] A.Chandra and D.Harel, Computable queries for relational data bases, JCSS 21 (2) 1980, pp. 156-178.

[C1988] A.Chandra, Theory of data base queries, to appear in the proceedings of PODS '88.

[DM1986] E.Dahlhaus, Computable directory queries, Proceedings of CAAP '86, Springer LNCS vol. 214, pp. 254-265.

[ES1979] K.A. Ericsson and H.A. Simon, Theory of protocols, Unpublished Manuscript, Carnegie-Mellon University, 1979 (appeared in 1993 with MIT Press as *Theory of protocols: Verbal reports as data*)

[F1988] S.Feferman, this volume

[G1988] O. Goldreich, this volume

[HH1978] P. Hajek and T. Havranek, Mechanizing hypothesis formation (Mathematical foundations for a formal theory), Universitexts, Springer Verlag, 1978.

[LBS1983] P. Langley, G.Bradshaw and H.Simon, Rediscovering chemistry with the Bacon system, in [MCM1983], pp.303-329.

[Mai1983] D.Maier, The theory of relational databases, Computer Science Press, 1983

[Mak1984] J.A.Makowsky, Model theoretic issues in theoretical computer science, part I: Relational data bases and abstract data types, in Logic Colloqium '82 (G. Lolli, G.Longo and A.Marcja eds), North Holland 1984, pp. 303-343.

[Mak1987] J.A.Makowsky, Review of [U1983, Mai1983 and CH1980], JSL 51 (4), 1987, pp. 1079-1084.

[MCM1983] R.Michalski, J.Carbonell and T .Mitchel eds., Machine learning, Tioga Publishing 1983.

[Sh1988] J.Shepherdson, this volume

[Tr1988] B. Trakhtenbrot, this volume

[Tu1939] A. Turing, System of logic based on ordinals, Proc.London Math. Soc. (2) 45, pp. 161-228.

[U1983] J.Ullman, Principles of database systems, 2nd ed., Computer Science Press 1983

[Va1984] L.Valiant, A theory of the learnable, Comm. ACM, 27 (11) 1984, pp.1134-1142.

[V1982] V. Vapnik, Estimation of dependencies based on empirical data, Springer 1982.

[W1950] R.Wilder, The cultural basis of mathematics, Proceedings of the International Congress of Mathematicians, vol. 1, 1950, pp.258-271.

[W1968] R.Wilder, Evolution of mathematical concepts, An elementary study, John Wiley 1968.

[W1981] R.L. Wilder, Mathematics as a cultural system, Pergamon Press 1981