

AFTERTHOUGHTS ON ANALOGICAL REPRESENTATIONS¹

Aaron Sloman
Cognitive Studies Programme²
School of Social Sciences
University of Sussex

In 1971 I wrote a paper attempting to relate some old philosophical issues about representation and reasoning to problems in Artificial Intelligence. A major theme of the paper was the importance of distinguishing “analogical” from “Fregean” representations. I still think the distinction is important, though perhaps not as important for current problems in A.I. as I used to think. In this paper I’ll try to explain why.

Throughout I’ll use the term “representation” to refer to a more or less complex structure which has addressable and significant parts, and which as a whole is used to denote or refer to something else. Thus maps, sentences, and phrases like “The paternal grandfather of the present mayor of Brighton” are representations. There is much that’s puzzling and complex about the concept of using something to “denote” or “refer to” something else, but for the present I’ll dodge that issue and rely on our intuitive understanding thereof.

The analogical/Fregean distinction is not new: people have been discovering and re-discovering it for a long time, though they rarely manage to say clearly and precisely what it is, despite agreement on (most) examples: e.g. maps, photographs and family trees are analogical representations whereas many sentences, referring phrases, and most logical and mathematical formulae are Fregean. I use the word “Fregean” because it was Gottlob Frege who first clearly saw that a great deal of natural-language syntax and semantics could be analysed in terms of the application of functions to arguments, and that this analysis was far superior to previous attempts to understand the structure of sentences. For instance, it enabled him to invent the logic of quantifiers and develop a notation which provided some of the essential ideas of Church’s lambda-calculus, and thereby some of the goodies in programming languages like LISP, ALGOL and POP-2. I use the word “Fregean” not only to honour Frege but also because there is no unambiguous alternative. The most popular rivals - “symbolic” and “verbal” - are used in too many different ill-defined ways, and in addition the first seems too general, the second too narrow. People seem to have a lot of trouble seeing clearly what the distinction is, so I’ll list and comment on some of the more common misrepresentations of what I wrote in the 1971 paper.

Misrepresentations

1. “Analogical representations are continuous, Fregean representations discrete”. Comment: I gave examples of discrete analogical representations, e.g. a list whose elements are ordered according to the order of what they represent.
2. “Analogical representations are 2-dimensional, Fregean representations 1-dimensional.” Comment: I gave examples of 1-d analogical representations (e.g. the list example). Much

¹Originally published as A. Sloman, (1975) Afterthoughts on Analogical Representation, in *Theoretical Issues in Natural Language Processing* (TINLAP-1), Eds. R. Schank & B. Nash-Webber, pp. 431–439, MIT, Reprinted in *Readings in knowledge representation*, Eds. R.J. Brachman & H.J. Levesque, Morgan Kaufmann, 1985

²Author is now at School of Computer Science, The University of Birmingham. Several related papers developing these themes are now available at <http://www.cs.bham.ac.uk/research/cogaff/>

mathematical notation is 2-dimensional *and* Fregean (e.g. integral or summation symbols, the normal representation of fractions).

3. “Analogical representations are isomorphic with what they represent.” Comment: I discussed 2-d pictures which are *not* isomorphic with the 3-d scenes they represent analogically.
4. “Fregean representations are symbolic, analogical representations non-symbolic.” Comment: I find this notion unintelligible. The only sense of “symbolic” which I can understand clearly includes both maps and sentences. People who arrive at this misinterpretation seem to be guilty of using “symbolic” in a sloppy, ill-defined sense, to contrast with some equally ill-defined alternative. Their excuse may be that this is frequently done (e.g. by Minsky and Papert in their *Progress Report*, and by Minsky in his more recent paper on frames - 1974.)
5. “Sentences in a natural Language are all Fregean.” Comment: I pointed out that *some* English sentences function in a partly analogical way, as is illustrated by the difference in meaning of “She shot him and kissed him” and “She kissed him and shot him”. Compare “Tom, Dick and Harry stood in that order”. Contrast “She shot him after she kissed him”, where a relation is explicitly named, and the semantics is Fregean.
6. “Analogical representations are complete: whatever is not represented in a picture or map is thereby represented as not existing. By contrast Fregean representations may abstract from as many or as few features of a situation as desired: if I say ‘Tom stood between Dick and Harry’, then nothing is implied about whether anyone else was there or not.” Comment: there may be an important distinction between descriptions or representations which are complete (relative to the resources of a language) and those which are incomplete, but this has nothing to do with the analogical/Fregean distinction. E.g. a map showing only some of the towns and roads of Britain is still an analogical representation. We are free to specify for some pictures or maps that they are to be interpreted as complete, and for others that they depict relations between some but not all parts of a situation or object. Similarly a LISP list might contain items representing events in the order in which the events occurred, yet be incomplete in that new items are added as new knowledge about the time-order of events is acquired.
7. “Fregean representations have a grammar, analogical representations do not.” Comment: it is easy to define a grammar for lists and trees, frequently used as analogical representations in computing. One can also define a grammar for a class of line-drawings which includes pictures of polyhedral scenes.
8. “Although digital computers can use Fregean representations, only analog computers can handle analogical representations.” Comment: see (1) and (2) above.

Explanation of the Distinction

What then is the distinction? Both Fregean and analogical representations are complex, i.e. they have parts and relations between parts, and therefore a syntax. They may both be used to represent, refer to, or denote, things which are complex, i.e. have parts and relations between parts. The difference is that in the case of analogical representations *both* must be complex (i.e. representation and thing) and there *must* be some correspondence between their structure, whereas in the case of Fregean representations there need be no correspondence. Roughly, in a complex

Fregean symbol the structure of the symbol corresponds not to the structure of the thing denoted, but to the structure of the procedure by which that thing is identified, or computed.

We can be a bit more precise about analogical representations. If R is an analogical representation of T , then (a) there must be parts of R representing parts of T , as dots and squiggles on a map represent towns and rivers in a country, or lines and regions in a picture represent edges and faces in a scene, and (b) it must be possible to specify some sort of correspondence, possibly context-dependent, between properties or relations of parts of R and properties or relations of parts of T , e.g. size, shape, direction and distance of marks on a map may represent size, shape, direction and distance of towns, and different 2-d relationships of lines meeting at a Junction in a picture may represent (possibly ambiguously) 3-d configurations of edges and surfaces in a scene. The relationship between R and T need not be an isomorphism, for instance when a relation between parts of R (such as direction or distance) represents different relations between parts of T in different contexts. In a perspective drawing there is no simple, context-independent, rule for translating angles between lines into angles between edges or surfaces in the scene depicted. In such cases the task of interpreting R , i.e. working out what T is, may involve solving quite complex problems in order to find a globally consistent interpretation. (See Clowes 1971.) From (a) and (b) it follows that in analogical representations, relationships within T do not need to be explicitly *named* in R , i.e. there need not be a *part* of R corresponding to relations like “above”, “behind”, “intersects” in T . The conditions (a) and (b) do not hold for Fregean representations.

A Fregean formula may be very complex, with many parts and relationships, but none of the parts or relationships need correspond to parts or relations within the thing denoted. The phrase “the city 53 miles north of Brighton” contains the symbol “Brighton” as a part, but the thing denoted does not contain the town Brighton as a part. The thing denoted, London, has a complex structure of its own, which bears no relation whatsoever to the structure of the phrase. Similarly “the father of Fred”, “63-24”, “(CAR(CDR(CDR(CONS A (CONS B (CONS C NIL))))))” have structures which need bear no relationship to the structures of what they denote. In these, and the examples discussed by Frege, it is possible to analyse symbolic complexity as arising only from the application of functions to arguments. Predicate calculus, apparently invented independently by Frege and C.S. Peirce, is the consequence of this idea. A full explanation would require an exposition of Frege’s distinctions between first-level and higher-level functions: e.g. he analysed “all”, “exists”, and the integral sign used by mathematicians, as names for second-level functions which take first-level functions as arguments.

Comments on the Distinction

The analysis proposed by Frege fails to account for the full richness and complexity of natural language, just as it fails to account for all the important features of programming languages. For instance, Frege apparently required every function to have a definite “level”, determined by the levels of its arguments. In languages like POP2 and LISP, where variables need not have types, functions like MAPLIST and APPLY take functional arguments of any type, and therefore do not themselves have types. APPLY can even be applied to itself. In POP2, APPLY(4,SQRT,APPLY); has the same result as SQRT(4); for instance. Since one can use English to explain how such functions work, English also cannot be restricted to Fregean semantics. There are many other ways in which a Fregean analysis of English breaks down: e.g. some adverbial phrases don’t fit easily, and many linguists would quarrel with the analysis of “the capital of France” as an application of a function to France, according to the decomposition “the capital of (France)”. (I am inclined to think the linguists are wrong, however.)

So I am not claiming that every symbolism, or representational system, must be analysed as being either analogical or Fregean. The distinction is not a dichotomy, though there may be some generalisation which is. Wittgenstein, in his *Tractatus Logico Philosophicus* outlined an all-embracing “picture” theory of meaning which attempted to subsume Fregean representations under analogical (e.g. by describing a structure such as $R(a,b)$ as one in which the relation between the names “a” and “b” analogically represented the relation between their denotations corresponding to R. In his later writings he acknowledged that this attempt to eliminate the distinction was unsuccessful. Conversely, I once explored the possibility of interpreting all analogical representations as being composed of function-signs and argument-signs and decided that it could not be done, since an essential feature of Fregean symbolisms is that argument signs can be replaced by others without altering anything else, and in general this cannot be done with pictures: replace the representation of a man in a picture with the representation of an elephant, and there will often be some other syntactic change, since different outlines leave different things visible. Compare “My uncle is in the garden” and “My elephant is in the garden”. Here any difference in the implications depends not on structural differences between the sentences but on background premises about the differences between men and elephants. I conclude that attempts to obliterate the distinction cannot succeed, although, as remarked previously, a Fregean formula may analogically represent a procedure evaluation or identification.

Linguists are not usually interested in the role of language in reasoning. Many philosophers, logicians, and mathematicians are under the mistaken impression that only proofs using a Fregean symbolism can be rigorous, or even valid. The suggestion is that the old proofs used in Euclidean geometry, which relied on diagrams, were not proofs at all. Real proofs were found only when axioms and inference rules were formulated which made the diagrams redundant. I believe that this denigration of analogical representations is connected with some of the worst features of mathematics teaching, especially at Universities. Excessive concern with too restricted a range of analogical representations is probably just as bad.

In the 1971 paper I tried to show that the concept of a rigorous valid inference could be applied to problem-solving using analogical representations just as it is normally applied to proofs in a Fregean language (e.g. predicate calculus). I tried to show that in some cases (some readers thought I meant *all* cases!) analogical representations combined rigour and validity with greater heuristic power than Fregean. I went on to suggest that program-writers and theoreticians in A.I. should pay more attention to analogical representations, and hinted that this would lead to programs which could solve problems much more easily and intelligently, since the possible manipulations of an analogical representation would be much more tightly restricted by its structure than the possible manipulations of Fregean representations. E.g. when relations are explicitly named, then any relation-name can be replaced by any other relation-name of the same syntactic category, whereas when relations are represented by relations (e.g. distance or order in a map) then the structure of the representing medium may constrain possible variations, thus usefully restricting search space.

The distinction between analogical and Fregean representations is real and important, though not exhaustive (as I pointed out in the paper). However, some of the things said or implied in my paper were erroneous, and should be withdrawn, which is what I am now about to do.

Valid Criticisms of my 1971 Raper

First of all I suggested that people in A.I. were not making use of analogical representations (except for the analogical relation between programs and processes). This was just wrong: any

intelligent programmer will use ordering and other relationships within data-structures to correspond to real relationships when this is useful. For example, Raphael's S.I.R. program did this. So did vision programs which used graph-like data-structures to represent the topology of pictures. Even PLANNER, with its apparently Fregean assertions, and procedure-invoking patterns, can be interpreted as a mechanism in which problems about actions or deductions are solved by *simulating* these actions or deductions: the simulation process then constitutes an analogical representation. However this was one of the major defects of PLANNER as a problem solver: often it is much more sensible to examine and describe possibilities for action than to execute them or simulate them, if one wishes to solve some problem about them. For a trivial case, consider the question "If I start in room A and then move back and forth between room A and room B, which room will I be in after exactly 377 moves?" The best way to solve this is not to *simulate* the moves but to form a generalisation about the effects of N moves where N is an odd number and where N is an even number. What we need are not my vague general exhortations to make more use of analogical representations, but detailed analysis of the differences between problems where it is and where it is not helpful to solve problems with the aid of some kind of partial simulation. The chess-board and domino problem is a good illustration of how an analogical representation can get in the way. Often one does better to manipulate descriptions of relationships than the relationships themselves.

Secondly I wrote as though anyone using a Fregean language, like predicate calculus, would not be interested in *organised* the sets of assertions describing some world or problem. (Minsky and Papert make the same mistake.) However, intelligent programmers do not devise theorem-provers which blindly store all axioms in whatever order they are read in, and always have to search the whole lot in order to find assertions relevant to any particular sub-problem or sub-goal.

If the set of stored assertions is large it will obviously pay to have some kind of indexing scheme, or to store assertions in a network such that each one is associated with pointers to others which might possibly be relevant. In fact, Bob Kowalski has shown that one can intimately combine the indexing system with a "resolution" inference system so that making inferences by resolution becomes a process of modifying the index to the data-base of axioms. However, no resolution theorem-prover, to my knowledge, gives the user sufficient access to the data-base handling mechanisms so that he can use a domain-specific indexing scheme. The same complaint can be made about PLANNER and CONNIVER. Once a set of Fregean formulae is stored in a structured network or graph the organisation of the network may itself have many properties of a non-Fregean, analogical representation. A trivial example would be the storage of a set of assertions of the form

$$R(a,b),R(b,c),R(c,d),R(d,e),R(a,d)$$

where R is a transitive asymmetric relation (e.g. "taller than" or "north of"). If each of the above assertions is stored in association with pointers to other assertions mentioning the same individuals, then the resulting structure can be used as an analogical representation of the order of the individuals, just as storing the names in a list like (A B C D E) can. The full equivalence could be obtained only if redundant assertions were pointed to in a different way, or perhaps removed, e.g. $R(a,d)$. This might not be useful for all problem-domains (e.g. it could be less useful where R defines only a partial ordering).

Embedding one's analogical representations in a Fregean symbolism like this makes it easier to switch flexibly between different representational systems according to the needs of the problem. Of course, the mere presence in a computer of a data-structure which *we* can describe as an analogical representation is not enough: the program must embody procedures which *make use of* the analogical features of the representation. In the case of a predicate calculus theorem-prover,

this means that there must be ways of controlling the order in which assertions or inference steps are tried, so as to correspond to the structure of the problem. E.g. if you wish to know whether c comes between a and e in the order defined by R , work through the set of assertions from a (or from e) in one direction at a time.

A more complex illustration of all these points can be constructed by devising a scheme for storing predicate calculus assertions about family relationships with links which enable the data-base to be used like the usual kind of family tree, instead of an arbitrarily ordered list of facts. So questions like “Who were all X’s cousins?” or “Was X the grandfather of Y?” can be answered with little or no searching, using the analogical properties of the data-base (i.e. relations represent relations).

More generally, it is possible (and maybe even fruitful) to think of all computation as being concerned with updating and accessing information explicitly or implicitly stored in a data base. The code for arithmetical functions implicitly represents, in a limited space, answers to a potentially infinite set of problems, for example. An important aspect of the intelligence of programmers, and programs, is the construction, manipulation, and use of indexes, so as to find relevant answers to questions quickly. However, an index is just as much a store of information as anything else, and problems of arbitrary complexity may be involved in finding a relevant index entry. (e.g. months of archaeological research may be needed in order to decide which entry in a library or museum catalogue to follow up). So the distinction between index and data-base disappears: any item or procedure may play a role in tracking down some other information. The data-base is its own index.

From this viewpoint one can assess the role of analogical representations as indexes, and note that relations of ordering and nearness, and distinctive substructures within analogical representations may define important access routes by which mutually relevant items of information may be linked. But, as Pat Hayes has pointed out, it doesn’t matter how this is implemented so long as it works. Thus, storing visual input in a 2-d array enables the neighbourhood relationships between pairs of numbers (array subscripts) to be used as an analogical representation of neighbourhood relationships within the original input and to some extent within the scene represented. And one can make good use of this analogical representation even if the array is stored as a set of Fregean assertions about what value is located at co-ordinates n and m for all relevant integers n and m , provided that there is a good index to these assertions.

So I should have acknowledged that all the benefits of analogical representations can be gotten from Fregean representations, suitably organised and interpreted. *However, this does not imply that analogical representations are not needed, only that they can sometimes be implemented using Fregean ones.* Similarly, it could be argued, at a still lower level, all Fregean formalisms used in a computer are ultimately represented analogically, even in a digital computer. But the matter is of little importance.

Finally I wrote as if it was going to be fairly straightforward to get programs to do things in the ways which people find easy. E.g. people often find it much easier to solve problems when they can see a picture of the problem-situation than when they are presented only with a set of assertions about it. However, doing this requires very complex visual abilities, which, although they *feel* easy to use, are probably the result of a very long and hard learning process extending back over millions of years of evolution (species-learning) and several years of individual learning.

I do not believe anyone has very good ideas yet on how to give a computer the same kind of sophisticated grasp of two-dimensional structure as we use when we look at pictures, maps, mazes, diagrams, etc. It seems to be a mixture of a large store of general principles about topology and geometry, intricately combined with a large store of specific knowledge about particular shapes,

shape-classes, and possible patterns of change (translation, rotation, stretching, fitting together, going-through apertures, etc.).

And somehow all this knowledge is indexed for rapid access when relevant, though by no means infallibly indexed. I am sure that many of the problems in explaining how this is possible are common to both vision and natural language processing. (The links are especially clear in the case of reading poor handwriting.)

Thus my suggestion that A.I. workers interested in problem solving should design machines to solve problems by looking at diagrams, maps or other spatial structures may be many years premature. Even the best vision programs presently recognise and use only very few aspects of the 2-dimensional structure of the pictures (or TV-inputs) which they attempt to interpret.

The upshot of all this is that I now realise that although it may be interesting and important from a philosophical or psychological standpoint to analyse the analogical/Fregean distinction, and to explore the relative merits of the two sorts of representations, such theoretical discussions don't necessarily help anyone engaged in the task of designing intelligent programs. The really hard work is finding out what factual and procedural knowledge is required for intelligent performance in each domain. The most one can achieve by the philosophical analysis is the removal of prejudices. But was anyone in A.I. ever really prejudiced against analogical representations, properly defined?

Bibliography

Clowes, M.B. (1971) "On seeing things", in *Artificial Intelligence*, Vol. 2, 1971.

Frege, G. (1952) *Philosophical Writings* (translated by P. Geach M. Black), Blackwell, Oxford.

Hayes, P.J. (1974) "Some Problems and non-problems in representation theory", *Proceedings AISB Summer Conference*, Sussex University.

Kowalski, R. (1974) "A proof procedure using connections graphs". Memo No.74, Department of Computational Logic, Edinburgh University.

Minsky, M. Papert, S. (1972) *Progress Report on Artificial Intelligence*, A.I. Memo 252, MIT Artificial Intelligence Laboratory.

Minsky, M. (1974) A Framework for Representing Knowledge, *A.I. Memo 306*, MIT Artificial Intelligence Laboratory.

Sloman, A. (1971) "Interactions between philosophy and A.I. - the role of intuition and non-logical reasoning in intelligence". *Proceedings 2nd IJCAI*, London, reprinted in *Artificial Intelligence*, vol. 2, 1971.