Analogical reasoning and modeling in the sciences

ANALOGICAL REASONING AND MODELING IN THE SCIENCES[1]

Prof. Paulo Abrantes
Department of Philosophy
University of Brasília,
Brasília, DF
Brazil
70744-070
Telephone: + 61 2721932
Fax: + 61 2733810
E-mail: abrantes@unb.br

In this century there has been a long trend of interest in analogical reasoning as a basis for scientific modeling. The idea that 'analogies' provide guidelines for the construction and development of theories dates back to Campbell (1920). The thesis that an analogy is constitutive of scientific theories had its revival in the 1950's as part of the emerging criticism against the logical-empiricist 'semantic explication' of scientific models. Critics like Toulmin and Hesse pointed out that this explication didn't capture one of the roles models play in science: that of providing guidelines for the development of theories. In the 1960's, we find many other philosophers arguing the same point: Achinstein, Harré, McMullin, Swanson, among others.

The critiques of the 'semantic explication' of 'model' were often voiced in terms of a 'surplus meaning' or 'surplus content' associated with models, which introduces an asymmetry in the relationship between model and theory, contrary to the symmetry presupposed by the 'formal' explication of this relationship (as in Braithwaite's or Hempel's accounts). It was argued that for a model to play the role of 'pointer' for the development of the theory, it has to share with the theory more than a common syntactical structure. This is the fundamental motivation of the distinction Hesse makes between 'formal' and 'material' analogies (that we also find in other philosophers).

There has recently been a great interest in analogical reasoning in the fields of Cognitive Science (CS) and Artificial Intelligence (AI). Various simulations of analogical reasoning as a human cognitive process have been proposed as well as applications of analogical reasoning to AI, as problem-solving heuristics. Despite this prolific activity, there is still much controversy concerning the nature of analogical reasoning, the role played by analogical reasoning in cognition, the fundamental processes involved in analogy-making and their computational implementation.

This debate, by itself, raises many interesting philosophical questions. In this paper I will make explicit some of these issues and use this debate as a framework to re-address

1A first version of this paper has been written during my stay as a Visiting Fellow at the Center for Philosophy of Science of the University of Pittsburgh. It was presented in a lunchtime talk at the Center in January, 1996. I am grateful also to the Brazilian Research Council (CNPq) for a postdoctoral grant that made possible this research.

discussions, among philosophers of science, about the heuristic role of analogy in scientific modeling. There are, certainly, difficulties in blending approaches concerned, on the one side, with the role analogical reasoning plays in scientific methodology and, on the other side, with its role in cognition and in computational systems. Nonetheless, there is a common concern with analogy as supporting reliable problem-solving heuristics, and as underlying creativity, conceptual dynamics and learning.[2]

This paper has two parts. First, I will present three simulations of analogical reasoning proposed in Cognitive Science: Gentner's Structure Matching Engine (henceforth SME), Mitchel's and Hofstadter's COPYCAT and the Analogical Constraint Mapping Engine (henceforth ACME), a simulation proposed by Holyoak and Thagard. The differences in these simulations disclose explicitly many of the controversial points that are being discussed in this area.

In the second part I'll suggest that these issues also arise in the context of philosophical accounts of formal and material analogies as a basis for scientific modeling.

-------- x --------

In the AI and Cognitive Science literature, the prevailing consensus holds that analogical reasoning can be subdivided into (at least) the following fundamental sub-processes :

a) <u>retrieval</u> (or selection) of an analog (among a number of possibilities available in a knowledge-base), given a (partial) representation of the target;

b) <u>mapping</u> between the representations of the retrieved source analog and the target.

c) <u>transfer</u> to the target of surplus structure and semantic content from the source.[3]

In a problem-solving context the initial mapping is the basis for the transfer to the target of a hypothetical solution for the problem. In science we typically face 'empirical problems': we have phenomena, in a certain domain, that we want to understand or to explain. This normally presupposes a theoretical language that is set up through the exploration, extension, and modification of available symbolic and conceptual resources, in general developed for or associated with other domains of phenomena.

The scheme: retrieval, mapping, etc. doesn't have anything specifically tied to analogy. It can just be seen as a general scheme of the processes involved in the exploration of previous knowledge: given a certain problem, <u>retrieve</u> adequate available knowledge to solve it; <u>manipulate</u> the source in order to adapt it to the (target) problem; <u>apply</u> the resulting

---

2A recent example of this convergence of interests is Dartnall (1994).

3Actually, two more sub-processes are also discussed in the literature: d) <u>evaluation</u> of the mapping and the inferences based upon it; e) <u>consolidation</u> of the analogical process through the abstraction of a reasoning schema or abstract structure for a future use. For a detailed overview of the AI research concerning each sub-process, see Hall (1989). See also Kedar-Cabelli (1988); Holyoak et al. (1989).

representation to solve the target problem. A theory of analogy has to characterize what is specifically 'analogical' in these sub-processes by making explicit the particular constraints that apply to each of them.

The computational implementations that we will discuss are purportedly psychologically realistic, that is, they have been proposed as simulations of analogical reasoning as a (human) cognitive process. Their proponents are, therefore, committed to ground them in empirical evidence about how people make analogies. I am not interested here in the issue of how good these simulations are with respect to psychological evidence, but rather in highlighting their presuppositions about the nature of analogy and their implications for the analysis and implementation of the processes involved in setting up analogies.

1. GENTNER: Analogy as structural mapping

Several computational implementations presuppose that analogical reasoning is essentially a kind of mapping based on the structure shared by the representations of the source and the target domains.

Gentner's SME is, as it were, a 'pure' implementation of this intuition. She claims that analogy is a kind of mapping that disregards 'superficial' similarities between the source and target representations, that is, any commonalities at the level of the attributes of objects. Analogical mappings should be based solely on the relational information encoded by the representations.

First, it is important to give an idea of the representation format used by Gentner. There is a basic presupposition that a 'domain' can be represented in terms of its objects, their properties, relations between them, and functions that map one object onto another object or constant.

In SME, knowledge is represented by 'propositional networks' of nodes and predicates. The nodes represent objects (or, if you like, their concepts), and predicates applied to these nodes generate propositions (Gentner, 1982, p. 108; Gentner et al., 1983, p. 102). The properties of objects are represented by one argument predicates, and relations are represented by two or more argument predicates.

The order of a relation plays an important role in Gentner's theory. First-order relations are represented by predicates that take two or more objects as arguments. Second and higher-order relations are represented by predicates that take propositions, other predicates or functions as arguments. An example of a higher-order relation is the causal relation.

Gentner's main contribution was to distinguish analogical from other types of mapping that can be made between the source and the target domains. Each mapping is based on specific kinds of similarities that are implicated in this sub-process of analogical processing. She places these different kinds of similarity in a 'similarity space' having as coordinates the attributes (one place predicates) and the relations (two or more place predicates) shared by the source and the target.

a) Analogy is defined as a kind of mapping in which relational information is transferred from the source to the target, independently of any superficial similarities between the source and target objects at the level of their attributes. The similarities between objects' properties (superficial/ 'perceptual' qualities) are disregarded and the mapping is based on similarity (or identity) of relations: "Entities and functions are placed in correspondence with other entities and functions on the basis of the surrounding relational structures" (1989, p. 209). The example she develops in many papers is the analogy between the hydrogen atom and the solar system.[4]

b) The second kind is a literal similarity mapping, in which (similar or identical) attributes as well as relations are mapped. Not surprisingly, the examples given are that of mappings in which the source and the target are from the same domain, e.g. "the helium atom is like the neon atom" (1982, p. 110).

c) The mapping Gentner calls a 'mere-appearance match' is based on object attributes (percepts). An example she gives is: "the glass tabletop gleamed like water" (1989, p. 206).

d) Abstraction is a kind of mapping in which the source is an abstract structure (a mathematical theory, for instance) that is applied to a particular domain. In this case, since the source is abstract "no concrete properties of objects are left behind in the mapping" (1989, p. 208). The example given is: "Heat flow is a through-variable" (1989, p. 206).

e) Metaphor superposes in one extreme with 'analogy' and in the other extreme with 'mere appearance' matches.

The main constraints on analogical mapping are, in Gentner's theory, consistency, target validity, and systematicity.

Structural consistency specifies that each object in the source is assigned to one and only one object in the target.[5] Target validity refers to the property of an analogy that minimizes contradictions between relations in the source domain and the corresponding ones in the target domain.[6] Systematicity specifies that the mapping be of whole relational

4Holyoak and Thagard argue that this analogy had no historical importance in the discovery of the Rutherford-Bohr 'model' of the atom. And apparently that is also the case in another example we find in the literature, the analogy between the molecular structure of gases and billiard balls (1995, p. 189; cf. Campbell, 1920).

5There is much disagreement in the literature regarding the structural constraint. Many theoreticians of analogy think that, as stated, it is too strict and would, if applied, eliminate the possibility of many analogies that can prove to be of important heuristic value. Owen (1990) and Bartha (1995) substantiate this claim with examples in the mathematical field. Holyoak and Thagard's position is, on this issue, not clearcut (1995, p. 258; cf. p. 5-6).

6Cf. Palmer (1989, p. 336-7).

hierarchies, "governed by higher-order relations with inferential import, rather than isolated predicates" (Gentner, 1989, p. 201).[7]

Now, Gentner defines analogy as a mapping of identical relations in the source and the target domains. Objects from the source and the target are mapped based on the (common) relations that they possess, and not on their common attributes. Dissimilar objects in the source and target domains (that is, objects that don't share any properties) are mapped to each other if they are arguments in identical relations. If you like, they are mapped if they play the same 'roles' in their respective systems. Analogical mappings have, therefore, the following form:

M: [R(bi, bj)] -> [R(ti,tj)] ,

where bi and bj represent the source objects, ti and tj the target objects, and R the identity relation (Gentner et al., 1983, p. 104) .

Furthermore, in Gentner's theory, an analogical mapping should give preference to systems of connected relations, that is to higher-order relations. The main constraint on analogical mapping is 'systematicity' (Gentner, 1989). Formally we have:

M: [R'(R1(bi, bj), R2(bk,bl))] -> [R'(R1(ti, tj), R2(tk,tl))] ,
(Gentner et al., 1983, p. 104).

The systematicity constraint 'filters' just the structural information about the source and target domains. In a typical situation, we have more structure represented in the source domain than in the target. A mapping of a (partial) common structure between both domains is supposed to give plausibility to the 'carry-over', that is, to the transfer of this surplus structure (higher-order relational predicates) from the source to the target (1989, p. 202).

The basic intuition behind Gentner's theory of analogical mapping is that 'attributes' tend to be domain-specific, while 'structure' isn't. So, if we want to transfer knowledge across domains, we have to bet on the existence of a common structure, represented by an interconnected web of relations.

1.1 Discussion of Gentner's theory

Gentner's theory and its implementation present many problems.

a) The problem of sensitivity to a particular representation

SME is a syntactical engine: the mapping between the source and the target is made independently of any semantic or pragmatic considerations. The symbols (that represent

7These constraints function as selection mechanisms, where the available knowledge about the target domain plays a fundamental role in finding out the adequate mapping. (We are excluding the extreme case of 'pure carry-over' where no knowledge about the target is available initially). See Gentner (1989, p. 202).

predicates, relations, functions) don't 'mean' anything. They seem meaningful to us, in the examples given, because the words used are taken from a natural language! Only their form (and not their content) is relevant for the mapping.

The syntactical character of SME can be seen as an advantage, since the constraints that apply to the mapping don't presuppose any general or domain-specific knowledge. These constraints can be implemented as general procedures that can be applied to any domain, once we've come up with the representations.

But, for the same reason, the theory has a fundamental shortcoming: its inherent sensitivity to the particular way the source and the target domains are represented (by the programmer).[8]

This criticism is sometimes made even more emphatically: in Gentner's simulation the representations of the source and the target are especially 'tailored' to bring forth the 'right' mapping. The mapping is, so to speak, built by the programmer into the representations themselves.[9] If we didn't already know the relevant predicates or the insightful mapping in advance, the particular way the source and the target domains are represented would seem completely arbitrary. If we change the representation, different mappings (or no mapping at all) are possible (given just the syntactic constraints proposed by Gentner).

b) The problem of the syntactical identity of predicates

Another problem with Gentner's implementation is that it wouldn't map predicates that are not syntactically identical, which is too restrictive. If we are interested in making analogies between objects and processes in two different domains, we should expect to map different descriptions, using different predicates.[10]

This shortcoming of SME can be seen, again, as a special case of the general problem that merely syntactical criteria are obviously sensitive to the particular way the domains are represented. One solution to this problem would be to ground the mapping not in syntax, but in semantics: different predicates might be matched if they represent the same or similar concepts. We will see below how this idea is implemented by other simulations of analogy-making.

c) The problem of the construction of representations

The two problems described above can be seen as arising from the fact that Gentner's theory of analogy bypasses completely the issue of the construction of representations. The representations of the source and the target are given as inputs to the matching engine and don't change during the mapping process. Gentner doesn't see this as a

8Palmer (1989, p. 337-9) makes also this point.
9 See Dierbach et al. (1991); Bartha (1994, p. 58,78); Palmer (1989); Hofstadter (1995, p. 182).
10Different predicates can, actually, have the same meaning (be two different representations of the same concept) or the same predicate can be used, in different domains, to represent different concepts.

problem since she believes that the issue of how representations are constructed, albeit an important one, is not related intrinsically to the problem of defining the nature of analogical reasoning. A complete computational architecture for analogy should keep separate the modules that deal with these two processes (representing and mapping).

It has been argued, however, that the construction of representations cannot be separated from the mapping in analogical reasoning. The real problem for a theory of analogy would be to give an account of the construction of representations and the mapping as intertwined processes. In this view, a theory of analogy cannot be based purely on syntactical constraints.[11] This is the main presupposition behind Mitchell's program of 'analogy-making', COPYCAT. Before presenting it, I would like to refer briefly to another simulation of analogical reasoning that uses semantic knowledge to solve some of the problems that arise for theories of analogy based on purely syntactical constraints.

2. Holyoak and Thagard's simulation

The intuition that analogy is a mapping based on the shared structure between two representations is also central to another computational simulation of analogical reasoning, ACME, proposed by Holyoak and Thagard. They distinguish three kinds of mapping between objects: a) attribute mappings; b) relational mappings; and c) system mappings.

A very simple attribute mapping is that in which two objects are mapped because they share the same properties (that is, because they are represented as arguments in the same one-place predicates). This kind of mapping is not, in general, considered as insightful as relational mappings. In the latter, objects in different domains are matched because they are related 'in the same way' (because they play the same roles in their respective structures). In the same spirit of Gentner's 'systematicity' constraint, system mappings are considered the most insightful kind of mapping because they are based on higher-order relations. In system-mappings, the mapping of whole structures takes priority over that of isolated relations.[12]

11It can, however, be argued that Gentner was not actually interested in giving an account of how we construct representations when we 'perceive' an analogy, given different 'pressures', but rather in how we understand an analogy that has already been made between two domains. Morrison (1994) argues that the clash between Gentner's and Mitchell's theories stems from a fundamental misunderstanding concerning the aims of each theory. Actually, they address different issues. Gentner's is a (normative) theory of what should count as an analogical mapping (and the implementation she proposes reflect this theory of the nature of an analogy; see 1989, p. 223). This explains why, in her implementation of analogical mapping, the representations of the source and the target are given and remain unchanged during the processing. Hofstadter, instead, is concerned with how the representations (of the source and the target) are generated in the first place, and subsequently transformed. If Morrison's interpretation is correct, Gentner's theory has nothing to contribute to devise a constructive strategy for extending our knowledge and concepts to account for new domains of experience.
12Holyoak and Thagard claim that the ability to represent knowledge explicitly at various levels of abstraction is a necessary condition for analogical thinking: "The capacity for explicit propositional thought is an essential requirement for the kind of abstraction needed for analogical thinking" (Holyoak et al. 1995, p. 25). They argue that, without a language, a cognitive system cannot represent a domain adequately for making the kinds of mappings required for analogical reasoning (Holyoak et al. 1995, ch. 3). I'll not be concerned in this paper with this strong symbolic stance in accounting for analogical reasoning.

Actually, Holyoak[13] claims that analogical reasoning involves multiple constraints. 'Structure' is considered as an 'internal' constraint on analogical reasoning. What distinguishes Holyoak's implementation from Gentner's is the role played by two other 'external' constraints: 'similarity' and 'purpose'.

The mapping engine in Holyoak's implementation, ACME, is able to map different kinds of predicates on the basis of the semantic similarity of the concepts they represent. This presupposes that the program/cognizer has access to (or is provided with) a network of concepts (stored in long-term memory).[14]

The similarity of two concepts is determined by their closeness in the semantic network in which they are embedded. Mere syntactical theories of analogy, like Gentner's, cannot deal with cases like this (since judgements of similarity presuppose world-knowledge represented, for instance, by a semantic network or some type hierarchy).

Another difference between Holyoak's implementation and Gentner's is the role played by the goals of the agent in the processes of analogical reasoning. If the analogical agent has as her aim to come up with an explanation for given phenomena, then we expect that causal relations should play a privileged role in the mapping. If the agent has 'practical' aims, like problem-solving, planning, decision-making, other kinds of relations will be favored, and the cognizer will come up with other mappings. Analogical reasoning in the mathematical field, for instance, would attach a greater importance to relations such as 'implies' than to causal relations.[15]

Holyoak's ACME is just a mapping engine (like Gentner's SME). However, Holyoak addressed also the 'retrieval' sub-process of analogical reasoning with a retrieval program, ARCS (for Analog Retrieval by Constraint Satisfaction). They provided the computer with a huge electronic thesaurus of english words, Wordnet, which organises these words (allegedly) in the way their related concepts are organised in human memory. The long-term memory of the computer is, therefore, provided with a semantic network of concepts. The retrieval (or selection) of a source is made through the similarity of concepts and also through structure matching.

Gentner's implementation of analogical reasoning and Holyoak's have, however, a major common point: the fact that the representations of the source and the target are handcoded by the programmer and don't emerge as a constituent part of analogy-making

---

13Henceforth I will mention just the first of the two authors of the book Mental Leaps.
14It can be argued that categorization is based on attribute mappings. However, there is an interplay between concepts and similarity (that can strike some people as a 'chicken-and-egg' puzzle). A central claim of a particular theory of concepts is that they are formed on the basis of similarity judgements. But concepts and the way they are represented and stored in memory is a fundamental basis for perceiving similarities between objects. Holyoak et al. (1995, p. 22) state the problem in the following way: "The hard problem for cognitive science is to explain why particular similarities are used to form concepts, and how concepts in turn alter judgements of similarity". See also Rips (1989, p. 22 and p. 55, note 1).
15 See Holyoak et al. (1995, p. 271).

itself. The next simulation amounts to a radical departure from a conception of analogy that separates the mapping from the construction of representations.[16]

3. Mitchell's COPYCAT

Mitchell and Hofstadter developed a program, called COPYCAT, that they claim, "is able to discover insightful analogies" in an idealized "world", that of letter-strings. This is a major difference if we constrast this program with the simulations presented previously, which are not limited to a micro-domain and which pretend to simulate analogical reasoning in 'real-world' situations.

COPYCAT makes use of knowledge about a particular world in forming analogies between a source and a target. This knowledge is incorporated in a conceptual network with several biases that constrain the application of concepts in making analogies in the letter-world.

At first glance this might seem a very limited undertaking, but the authors are attempting to simulate something that they take to be common to various high-level cognitive processes: the 'fluidity of concepts'. They claim that 'high-level perception' is involved in one of the processes where the 'fluidity of concepts' is especially important: the construction of representations The encoding of raw-data through perception is 'flexible' because it is sensitive to the context and to 'top-down' influences from the conceptual schemata of the cognizer (1995, p. 172).

Mitchell claims that, in this respect, there is a resemblance between perception and "even the most abstract and sophisticated mental acts" (1995, p. 210), like analogical reasoning (1995, p. 179). She certainly has in mind the shortcomings of Gentner's simulation of analogical reasoning, since the latter doesn't address the intrinsic association between analogical mapping and the construction of the source and target representations.

Hofstadter makes this point in the context of a criticism of computational implementations of cognitive processes in general. He argues against the "objectivist view of perception and of the representations of objects, situations, and categories", prevalent in the AI community. In general, these implementations take for granted the assumption "... that it is possible to model high-level cognitive processes independently of perceptual processes" (Hofstadter 1995, p. 176, 181-5, 282).[17]

16What I am calling the 'construction of a representation' should not be conflated with the retrieval sub-process of analogical reasoning. I'll not have much to say in this paper about the retrieval of an analog. I should just mention here that Gentner (like Holyoak) proposed also computational mechanisms for the retrieval. We saw that in Holyoak's solution (ARCS), but not in Gentner's, 'semantic similarity' plays a central role in the retrieval of an adequate source, given a target (Thagard, 1992, p. 24). In any case, the next simulation of analogical reasoning does not address the retrieval sub-process either.
17Cf. Bartha (1994, p. 58,78); Dierbach et al. (1991). Concerning the flexibility of conceptual thought, see Clark (1993, ch. 4).

I can give here just a flavor of how COPYCAT works. It's not my intention to enter into the details of the implementation (and even the description of the architecture, I admit, will be sketchy). My aim is to make explicit the theory of analogy that is implemented by the architecture.

Let's analyse a typical example of analogy-making in the domain of letter-strings. COPYCAT typically solves the following kind of problem: "Suppose the letter-string **aabc** were changed to **aabd**; how would you change the letter-string **ijkk** in 'the same way'?"(1995, p. 206).

The fundamental point Mitchell wants to make (henceforth Hofstadter's central contribution to the development of COPYCAT will not be mentioned explicitly) is that this problem admits a variety of answers. It depends on the way the strings are 'perceived' (conceptually encoded or represented), given different 'pressures'.

One answer to the problem could be the string: **ijkl**. Here, the rule 'replace the rightmost letter by its alphabetic successor', that describes the transformation of the first string in the source into the second, is applied 'literally' (that is, without modification or 'slippages') to the target in order to obtain the string that is missing. This answer is not completely satisfactory because it does not take into account one feature of the context, namely, the duplication of the letter K in the target string. Once this 'pressure' is perceived, a better answer would be **ijll**. Now, the group KK is perceived as a single object (a 'sameness group') and mapped to the letter C in the source string. Based on this mapping, the previous rule can be modified by allowing the concept 'the rightmost letter' to <u>slip</u> into the concept 'the rightmost group'.

However, another class of answers can be reached if the duplication of the letter 'A' in the source string is perceived as highlighted by the duplication of the letter 'K' in the target string. This contextual 'pressure' can direct our attention to the fact that the group **aa** plays a role in the source string similar to the role played by the group **kk** in the target string. These two 'sameness groups' can be mapped on the basis of this <u>similarity of roles</u>: leftmost <--> rightmost. Given this 'perception', another answer could then be **jjkk,** by translating the original rule by the rule 'replace the leftmost letter by its alphabetic successor'. Here the concept 'righmost letter' slips into the concept 'leftmost letter'. Another answer might also be given by allowing a slippage of the concept 'successor' into the concept 'predecessor' (given a different perception of the symmetries involved in the source): **hjkk**.

So, different solutions are possible in different contexts and also through the perception of similarities between the source and the target <u>at different levels of abstraction</u>. These are different 'pressures' constraining the solutions. 'Bottom-up' pressures arise from the 'context', that is, particular objects or structures[18] in a string are highlighted by the objects or structures perceived in other strings. If we change one of the strings, this will change the probability of building particular objects and making particular mappings.

18A 'structure' here is an object built by codelets (see below) through bonding of elemental letters.

For instance, if in the given example the target string were just **ijk**, then in this context **ijl** would be a perfectly good answer to the problem. The mapping of the letter K and the letter A in the source string (based on a slippage 'rightmost letter'---------> 'leftmost letter') would not be perceived as compelling!

Besides 'bottom-up' pressures, there are 'top-down' pressures that presuppose stored knowledge that the analogical-cognizer has about this micro-world of letter-strings. These 'top-down' biases include not only the availability of particular concepts, their degree of activation, but also the topology of the semantic network in which they are embedded. We will see that these links allow or favor certain slippages in order to adapt the situation depicted in the source to a different situation/context in the target. An important 'top-down' bias is a privilege granted to the instantiation of 'deeper', that is, more abstract concepts. This bias will favor the perception of similarities at higher levels of abstraction.

For instance, in the same example, it would be considered a 'bad' answer to apply the rule 'replace the rightmost letter by the letter D' to the target string, to generate the string **ijkd**. This answer would be considered too 'literal' since it instantiates the concept of an object, 'D', instead of the relational concept, 'alphabetic successor', that is more abstract.

3.1 COPYCAT's architecture

COPYCAT's architecture involves three components: the Slipnet, the Workspace and the Coderack.[19]

1. The <u>Slipnet</u> is a 'platonic' network of concepts. Its nodes are associated with 'types' of objects and relations. It comprises the knowledge that the analogy-maker has of the letter-world.

2. The <u>Workspace</u> is the "locus of *perceptual activity*", where explorations of the strings take place, objects are built and mappings (bridges) between strings are made as a consequence of contextual (bottom-up) and conceptual (top-down) pressures.

3. The <u>Coderack</u> stores a population of agents (the codelets) that carry out several tasks, including the identification and building of objects in the Workspace, as well as the building of bridges. Codelets also come up with the rules that apply to the source and the target transformations, respectively.

3.1.1 Slipnet

The Slipnet incorporates the 'knowledge' (about the letter-world) available to the system. This comprises basically concepts of objects and concepts of relations (as well as functions). The Slipnet contains, for instance, nodes corresponding to the concepts of the 26

19The central aspects of this architecture were already implemented in a former program developed by Hofstadter's group: JUMBO, a program for making anagrams (see Hofstadter, 1995).

letters of the alphabet and to the concepts of basic relations, like 'successor' and 'predecessor'. Concepts of higher-order relations, like 'opposite', link several other nodes, some of them corresponding to basic relations, as the two I've just mentioned.

The Slipnet is 'platonic' in two senses: a) there is a fixed and permanent stock of concepts (no new concept is created; there is no 'learning' in this sense); b) the concepts are hierarchically organized and linked, forming a net whose basic topology is the same at the beginning of each run of the program.

However, the network is dynamical: the 'distances' between concepts change dynamically with changes in the 'perception' of the objects in the strings.The mechanism is the following. There are several types of links, and some of them are labeled. These labels are also concepts. When a label-concept is activated, given the perception of some object or bridge in the Workspace, all the links labeled by this concept 'shrink', increasing the probability of slippage of the node-concepts linked.

An example would be the label-concept 'opposite' (a second-order concept), that links several pairs of nodes, like (left, right) or (successor, predecessor). When the concept 'opposite' is activated, the probability of slippage for all these node-pairs increases.

In COPYCAT, a concept, although centered on a node, 'spreads' like a cloud around the node. Each concept is, therefore, a 'fuzzy' unity. When a link between two concepts shrinks, their 'clouds' overlap increasing the probability of conceptual slippage. The 'closeness' of two concepts in the Slipnet is a measure of their similarity.

The form of this semantic network is changing all the time, as concepts associated with links increase or decrease their activations in response to contextual pressures (the perception and building of objects in the Workspace). These changes are effective only during the same run of the program.

Concepts have also different 'depths'. The depth of a concept is a pre-determined static feature that stands for its generality and abstractness: the greater the difficulty of directly perceiving structures that instantiate a concept, the greater its depth. The concept of a particular object has, in general, less depth than relational concepts (there are exceptions, though, like the concepts 'last' and 'first', for positions in the alphabet). The concept 'opposite' has a great depth since it links many other relational concepts favoring slippages between them (for instance, the mapping of 'leftmost letter' and 'rightmost letter' in one of the answers to the problem exemplified before).

COPYCAT has, therefore, a bias towards favoring analogies based on deeper concepts. This corresponds to an *a priori* propensity to allow more easily slippages of shallower concepts than of deeper ones. This is an implementation of the intuition that if there is a common 'essence' between two different objects or situations, it can often be 'perceived' only at an abstract level. Conceptual slippage is, therefore, a way to 'adapt' the encoding of one object or situation to fit a different one that is 'perceived' as 'the same' (type of) object or situation at a higher level of abstraction. To sum up: two factors facilitate slippage, the shallowness of the concepts involved and their proximity (overlap of 'halos') in the Slipnet (Hofstadter, 1995, p. 219).

3.1.2 Workspace

At the beginning of a run, the letter-strings in the Workspace are only "unconnected raw data". At this stage there is merely an association between each letter-token and its conceptual type in the Slipnet; the letters on the edges of the strings are also identified as 'leftmost' and 'rightmost' letters.

Then the strings begin to be scanned by the codelets and, as a result, many structures are detected and objects are built at various levels. The lower-level groups are based just on the instantiation of 'bond-types' between adjacent letters. These bond-types are concepts in the Slipnet, like 'sameness', 'successor', 'leftmost', etc.

In our example, objects like **aa** or **kk** can be readily built by a 'sameness' bond. The groups that are formed by these bonds can be considered as higher-level objects and new bonds can be established between them, and so on. For instance, **ijkk** can be seen as a 'successor group' formed by the component objects, **i, j,** and the sameness group **kk**.

'Bridges' (or mappings) can then be made between the objects detected/ constructed in different strings, based on identity or similarity relationships (these objects can be parts of a string or eventually the whole string).

Different answers correspond to mappings based on two <u>kinds of similarity</u>: similarity of <u>objects</u> (intrinsic or 'literal' similarity) and similarity of <u>roles</u> played by the objects in the strings. Insightfull anwers are those based on the similarity of roles played by objects in their respective strings. So, referring once more to our example, the group **aa**, in the source, can be mapped to **kk** in the target on the basis of the roles they play in their respective strings: they are perceived as instantiating the concepts 'leftmost' and 'rightmost'. These concepts are linked in the Slipnet by an 'opposite' link. Once the concept 'opposite' is activated, it favors the mapping of the letters **c** and **i** on the basis of the slippage 'rightmost letter' -----> 'leftmost letter'. The probability of a slippage 'successor' --------> 'predecessor' also increases, since these concepts are linked by the same relation 'opposite'.

To sum up: the objects that are built and the mappings that are made are the result of Workspace pressures and conceptual pressures. Mitchell describes this as a "*feedback loop* between perceptual activity and conceptual activity" (Hofstadter, 1995, p. 223), between top-down and bottom-up pressures.

A final remark: objects and bridges can be built or destroyed in the Workspace. Two factors determine the 'strength' of an object or a bridge: a context-independent one (the depth of the concept that is instantiated by the object or structure) and a context-dependent one (how this object or bridge is consistent with other objects or bridges built in the

Workspace). This consistency constraint corresponds to a drive towards what Mitchell calls a 'viewpoint', that is, a global fit between objects built in the Workspace.

3.1.3 Coderack: (population of Codelets)

The exploration of the Workspace is made by a population of agents called 'codelets'. It's not necessary, for our purposes, to discuss this in detail here.[20] The codelets' functions are to 'notice' a promising structure in the Workspace and eventually to build or destroy an object or a bridge between objects in different strings. Each codelet acts in a semi-autonomous way, but the collective action of all the codelets is subject to global biases and urgencies.

The urgency of a particular task, and the consequent release and 'reproduction' of a particular codelet, is defined by the pattern of activation of the concepts in the Slipnet, and the objects that are already built in the Workspace (we saw that they are actually inter-related). If a particular object is built (and the corresponding concept is activated), for instance, the search for similar/identical structures becomes more urgent.

The processes of scanning, bonding, bridging, etc. go on in parallel in the Workspace. There are bottom-up codelets ('noticers') and top-down codelets ('seekers'), among other kinds of codelets. The noticer's task is 'unfocused', but the top-down codelets are looking in the Workspace for structures of a particular kind ('defined' in the Slipnet).

At the beginning, codelets are sent to do their tasks in a random way. This corresponds to an initial 'open-mindedness' of the program that is still exploring and trying to build promising objects. This initial open-mindedness evolves into a closed-mindedness that corresponds to a focus on some promising paths, that is, those hypothetically leading to a coherent 'view'. There is a measure of the relative disorder in the Workspace: temperature. It is an (inverse) measure of the degree of perceived order in the Workspace.

3.2 General comments

I understand Mitchell's COPYCAT as an implementation of the basic intuition that two systems are analogous if they can be perceived (encoded or represented) as similar at some level of abstraction, where their common 'essence' is defined. Analogy-making consists in the construction of representations of the source and the target letter-strings and in mapping them so that their structures can be perceived as instances of 'similar' concepts. We saw how this 'construction' of similarity is affected by the context and by the configuration of the conceptual network. In COPYCAT, analogy-making involves abstraction: deep concepts are activated and shalow concepts 'slip' to allow mappings between the source and the target strings. 'Slippages' can be understood as cases of 'conceptual' or 'meaning' change at a lower-level, aiming at the construction of descriptions

---

20This element of COPYCAT's architecture is probably the most original, not only from a computational point of view (especially the parallelism and non-determinism of the activity that goes on in the Workspace) but also as a model for perception.

of the source and the target domains that can be interpreted as instantiations of the same higher-level conceptual ‘scheme’.

Abstraction and instantiation are the basic processes involved in the way COPYCAT makes analogies. Another program of analogy-making, TABLETOP, based on the same architecture and principles as COPYCAT, is presented, explicitly, as an implementation of these two fundamental procedures (French, 1995, p. 4).

As a summary, we compare in the following section the insights concerning the nature of analogy implemented in COPYCAT, with those implemented in SME and in ACME.

3.3 COPYCAT versus SME

3.3.1 Differences

Gentner’s SME and COPYCAT implement different conceptions of the nature of analogical reasoning. For Gentner, ‘analogy’ is just a particular kind of mapping between two representations. For Mitchell, analogy involves also representation building, that cannot be separated from the mapping.

In COPYCAT the mappings are based on the instantiation of concepts and not on syntactical criteria. Since the concepts are linked in a network, and since they have different degrees of overlap that change dynamically, mappings of different objects can be made based on ‘similarity’ (that is, based on the closeness, in the Slipnet, of the concepts they are instantiations of). In SME, only identical predicates can be mapped: there is no place for similarity.

COPYCAT is a program that is supposed to apply just to a particular micro-world (that of letter-strings) and not to any domain, like Gentner’s SME. This is a consequence of the fact that in COPYCAT analogical mappings are based on meaning, that is, based on the instantiation of concepts that encode the system’s knowledge of the domain.

3.3.2 Agreements

COPYCAT and SME converge, nonetheless, in implementing the same basic judgement concerning what distinguishes a ‘good’ from a ‘bad’ analogy: good analogies correspond to the perception of commonalities at an abstract level; bad analogies are based on the perception of common ‘immediate’ features.

However, Gentner and Mitchell implement different criteria for evaluating what is ‘abstract’. In SME, the order of a relation- that is, a mere syntactical feature- is a ‘measure’ of ‘abstraction’. In Mitchell, abstraction is related to ‘conceptual depth’, a parameter that is based on the available knowledge of the variety of objects, relations and symmetries in a particular world.[21]

Neither SME nor COPYCAT address the selection sub-process. In COPYCAT the source strings are given.

3.4 COPYCAT versus ACME

3.4.1 Differences

I pointed before that in ACME, as well as in SME, the representations of the source and the target are handcoded by the programmer and don't emerge as a constituent part of analogy-making itself. In other terms, the representations are not constructed (or re-constructed) as a result of different kinds of 'pressures' as in COPYCAT.

3.4.2 Agreements

Mitchell's and Holyoak's implementations concur in basing analogical reasoning on semantics and not just on syntactics as Gentner does. Semantic similarity plays a central role in both COPYCAT and ACME. However, the latter implements a static semantic network. What distinguishes COPYCAT is the flexibility of the Slipnet: its topology changes in response to context (to 'bottom-up' pressures).

There are, therefore, deep differences between ACME's (and ARCS') implementation of a semantic network and COPYCAT's. In the SLIPNET, the concepts have a core (that corresponds to a node) but they have also a 'fuzzy' cloud that allows intepenetration as the links shrink.

However, the flexibility or adaptability of Slipnet is a limited one since there is no mechanism in COPYCAT for the creation of new concepts or of new links between concepts. We can see this as a major limitation of the program, since our intuition is that analogy-making in humans often involves these processes.[22]

21However, we saw that Gentner's purely syntactical criteria of 'abstractness' does not always agree with conceptual depthness. In COPYCAT, an attribute can eventually have more depth than a relation (see Hofstadter, 1995, p. 213). It can be argued, however, that these differences are only superficial ones, since COPYCAT's semantics is a very poor and limited one. A program with a more complex semantics, and based on the same architecture, is TABLETOP (see French, 1995). Cf. Indurkhya (1992, p. 387).

22Another criticism has been leveled at COPYCAT's world for being too idealized and idiosyncratic (the Slipnet applies to a very narrow and idealized domain). The authors counter-reply that, in fact, this idealization favors a better understanding of the fundamental processes involved in analogical reasoning. The idealization also avoids misunderstandings concerning what the analogical computer really knows about the domains that are being compared and mapped. For instance, since Gentner's representations use predicates formulated in natural language, the reader may be fooled into presupposing that the computer is capable of knowing what these predicates <u>mean</u>. In a micro-world, the knowledge that is, in fact, involved in analogy-making, can be made explicit (as in COPYCAT's SLIPNET).

4. Analogical Modeling in the Sciences

Now, does the debate concerning the processes and constraints involved in the computational implementation of analogy-making have any bearing on the issue of scientific modeling, and vice-versa?

First of all we have to distinguish different meanings of the term 'model' as used by scientists and philosophers, associated with different roles 'models' play in science. What follows applies just to a particular kind of modeling methodology, based on an analogy between representations of two domains of phenomena or two systems.

Even after making this restriction, however, we should still distinguish two kinds of 'analogical models', whether they are based on 'formal' or on 'material' analogies.

I include in the class of models based on formal analogies the following: a) 'models' as understood by 'syntactic' and 'semantic' reconstructions of the structure of scientific theories[23]; and b) accounts of 'modeling' in terms of 'nomic isomorphism' or, more generally, 'structural isomorphism' (that are presupposed in 'simulation' techniques).[24]

Material analogies are supposed to underlie another kind of modeling heuristics that provides explanations and not just simulations. Modeling based on a material analogy aims to play a role in the construction of theories by suggesting a theoretical language and explanatory hypotheses for new domains of phenomena.

For instance, 'mechanics' was the source of many models for different areas of physics in the eighteenth and nineteenth-centuries. An example was the use of bouncing balls as a model for explaining the behaviour of gases in terms of molecules speeding inside a container. Maxwell's models for the electromagnetic ether can also be seen as based on material analogies.

In the following I want to re-address this distinction between formal and material analogies in the context of scientific modeling through the perspective of the discussions presented in the previous sections. There are obvious differences in the roles played by analogical reasoning in scientific methodology, on the one side, and in cognition or in AI, on the other side. Despite these differences, I think some insights can be obtained by adopting a computational point of view, if we aim at having a detailed account of the various processes involved in setting up models based on material analogies.

4.1 The construction of similarity

23For models in the syntactic conception of the structure of scientific theories, see Braithwaite (1955). For models in the semantic conception, see Giere (1988) and Van Fraassen (1987). Despite the differences between the so called syntactic and the semantic views, they are both not concerned with heuristics. The syntactical and the semantic views address the questions 'what is a theory?' and 'how does a theory relate to the world?', and neither with the methodological question 'how is a theory constructed?' .

24On nomic isomorphisms, see Hempel (1965, p. 436). On structural isomorphisms, see Kroes (1989, p. 148).

We saw that a main divergence between computational simulations of analogy-making is whether the construction of representations for the source and the target might be considered an integral part of analogy-making or not. Are representing and mapping autonomous or interdependent processes? A related question is whether the mapping should be based on mere syntactical considerations or whether it presupposes semantic knowledge.

It seems to me that these issues also arise in distinguishing formal from material analogies as underlying different kinds of modeling heuristics in the sciences.

In formal analogies representations of the source and the target are given and we just elaborate a mapping between them. We are not concerned with how these representations were generated in the first place.

Furthermore, in formal analogies only syntactical considerations are the basis for the 'analogical' mapping. The model and the modeled theories don't have to be 'similar' in any non-formal sense of 'similarity'.[25] The interpretations of the terms in each domain play no role in the mapping. In formal analogies corresponding terms (associated through the mapping) just play the same logical or mathematical role inside the overall structure of both theories. The terms or predicates mapped may have completely different interpretations in their respective domains. We just rely on some kind of isomorphism between the source and the target representations.

I won't have much more to say about formal analogies, from this point of view, besides the fact that they have a mere syntactical basis, like the mappings in Gentner's SME. For instance, in the logical-empiricist construal of the notion of 'model', the model and the theory have in common a syntactical structure (a calculus). Palmer showed, indeed, that Gentner's definition of analogical mapping and some of the constraints she proposes can be redescribed in terms of Tarski's model theory (1989, p. 336). Therefore, we can expect that in those cases where formal analogies are supposed to play a heuristic role, the sensitivity to the way the source and target domains are represented, which is a problem in Gentner's implementation, might also arise in the methodological context of scientific discovery.

In the case of material analogies, we usually start with a partial representation of the target phenomena, usually involving just observational predicates. The construction of a full-fledged representation of the target domain, given the available representation of the source domain, is what modeling is all about.[26] Modeling based on material analogies can be understood as a process of representing the model and the modeled systems as 'similar', that is, as the same type of system at some level of abstraction. This is a necessary condition for the extension of the source representation to encompass the target domain, providing an adequate conceptual framework and terminology for describing the latter.

25Many philosophers would refrain from employing the term 'similarity' in the context of mere formal analogies.
26We will see below that often the source domain has to be re-represented to fit the knowledge we have of the target domain.

A simple example may help making this point. Let's suppose that we have a theory that accounts for what happens when we throw a stone in a quiet pond. This theory might involve, for instance, concepts like that of 'wave'. 'Modeling' sound phenomena on this theory would involve a procedure for creating a new and more abstract concept of wave that can be instantiated in both domains of phenomena.

A sketchy account of the modeling procedure in this case would include the following steps. We have to distinguish what features of both domains of phenomena have to be ignored and what are the relevant ones. For instance, superficial features of water waves like 'wetness' and, say, 'color', have to be ignored. Furthermore we have to know how to map the predicates used to represent the relevant features. The problem is that these predicates are often not identical and we have to allow certain 'slippages' (cf. COPYCAT) to make the mapping . The 'height' of a water wave above the surface of the pond should be mapped to the 'loudness' of sound. These different concepts become 'similar' by linking them to a more abstract concept, say 'amplitude', used to characterize an abstract type of wave. Other concepts associated with this abstract concept of wave, like 'wave-length', and 'frequency', allow us to see as similar other concrete features of the two phenomena, like the distance between two successive water crests, the 'pitch' of a sound, etc. These similar features are mapped allowing other 'slippages'. If we try to extend this description to include light phenomena, the commonalities between these realms of phenomena will have to be made at an even higher level of abstraction.

As a result of the creation of new and more abstract concepts, these realms of phenomena or systems are 'seen' (or represented) as similar in certain 'essential' respects. The source and the target systems are then described as instances of the same 'kind' of (abstract) system.

This modeling procedure is very much like analogy-making as understood by Mitchell and Hofstadter, and implemented in COPYCAT. This comparison between cases of scientific modeling and COPYCAT's analogy-making highlights, however, the limitations of this implementation. The depth of a concept is fixed beforehand by the programmers and can't change. It is certainly an interesting feature of COPYCAT that the SLIPNET modifies its shape with the emergence of an analogy, but these changes are still very limited. Furthermore, COPYCAT makes analogies in a very restricted domain, despite the amazing diversity of structures the letter-strings can display. In real-world analogy-making, we obviously deal with much more complex domains that are, often, very far away in terms of their structures and processes (at least as we can judge based on our immediate experience). An inter-domain analogy presupposes often deep abstractions involving the creation of new concepts and sometimes radical modifications in our conceptual schemes.

Furthermore, while COPYCAT at the beginning of a run starts again in the same 'state', in real-world situations a cognitive system is expected to learn with experience and to 'start' a task with previous representations and conceptual links that were constructed in previous 'runs'.

Keeping in mind these qualifications, I will examine whether the constraints that apply to the processes that underlie COPYCAT's analogy-making might also underly modeling based on material analogies.

4.2 Bottom-up and top-down constraints in analogical modeling.

We saw that in COPYCAT there is an interplay between 'bottom-up' and 'top-down' pressures (or constraints). Different representations of the source and the target strings are built in different contexts. Biases incorporated in the way the concepts are linked in the SLIPNET play also a central role in the construction of representations and in the mapping. These 'pressures' can be compared with 'bottom-up' and 'top-down' constraints proposed by philosophers of science to account for modeling based on material analogies.

Aronson et al. (1995) characterize 'bottom-up' accounts of material analogies as those that consider 'similarity' as a primitive notion. Their criticism of these accounts is based on an argument that is widely accepted in the philosophical and psychological literature: any two objects or situations can always be perceived as 'similar' in some respect or another.[27] A theory of analogy should acknowledge that 'similarity' is a derivative notion, make explicit the basis of judgements of similarity and propose adequacy criteria for these judgements. Their contribution is to highlight that similarity judgements presuppose knowledge about the world (an 'ontology') and to suggest that this knowledge can be conveniently represented by type-hierarchies (an AI representation format). This type-hierarchy provides 'top-down' constraints on similarity judgements and on the mapping of the source model and the target. They claim that similarity-based and logic-based accounts of modeling have in common the fact that they are both "... insensitive to the ontological constraints on the selection of models in real science"(1995, p. 49).

Mary Hesse, in her account of material analogies (1966) argues, however, that material analogies should be based on 'pre-theoretical' judgements of similarity. She believes that comparisons between the observational predicates in the representations of the source and the target allow us to find out the similarities and the dissimilarities between the domains (in her terms, the 'positive' and the 'negative' analogies, respectively). Modeling should be triggered by similarities perceived at an observational level between a target domain and some available source.[28]

What Hesse calls the 'neutral analogy' has a fundamental importance for the constructive role she assigns to scientific modeling. The 'neutral analogy' encompasses

---

27See, for instance, Goodman (1976, p. 77) and Black (1962).

28Hesse's 'horizontal constraint' is in sharp contrast with Gentner's characterization of an analogical mapping as one that disregards similarities at the level of attributes of objects. However, Hesse's theory of analogy is 'semantic', and not syntactic as Gentner's is. Hesse, in a recent paper, re-states her 'horizontal' constraint for material analogies, explicitly allowing now that the similarities between objects' properties be "either directly perceived" or a consequence of "conceptual resemblances of ideas or pictures derived from preconceived cultural forms". She interprets this move as that of "drawing attention ... from superficial similarities to complex contextual examples" (1988, p. 324-5). We can retrospectively see the references to metaphors in her 1966 book as anticipations of this kind of 'semantic' approach to analogical reasoning. Cf. Indurkhya (1992, p. 54-5).

those entities, properties or relations in the 'source' that we don't know, in a certain phase of the investigation, whether they fall into the 'positive' or the 'negative' analogies.

The existence of a 'neutral' analogy gives a fundamental dynamical and heuristic character to Hesse's conception of a model, that distinguishes it from models based on 'formal analogies'.

Theories are, for Hesse, dynamic entities and the 'neutral analogy' between a (source) model and a (target) theory constitutes its "growing points" (1966, p. 10).[29] By exploring the neutral analogy between the source and the target, models can play a heuristic role in the construction of theories, by suggesting a theoretical language and by providing explanatory hypotheses for the phenomena in the target domain.

If we adopt Aronson's analysis, Hesse's should be considered as a similarity-based account of modeling, in which 'similarity' plays the role of a 'bottom-up' constraint (in Hesse's terms, a 'horizontal' constraint).[30]

If 'similarity' is considered the only constraint in analogy-making (cf. Holyoak and Thagard's constraints), then we will have problems in distinguishing the relevant properties (in each domain) from those that should be ignored in the mapping.[31] How can we guess the right or insightful mapping? This is an especially big problem when the predicates that are matched are not identical, which is often the case, as we highlighted in our discussion of Gentner's theory of analogy.[32]

This problem can be illustrated by one example given by Hesse: that of setting up a material analogy between sound and light phenomena. In elaborating this analogy, how can we guess that the predicate 'brightness', in the representation of the target domain, should

29Strictly, we should perhaps speak not of a single theory, but of a 'research program' in the sense of Lakatos, or of 'theory-families' in the sense of Harré (1987), in which the neutral analogy with the model would provide the 'positive heuristic'.

30Nagel also distinguishes 'formal' from what he calls 'substantial' analogies, in which the relationship between the target theory and its models is one of analogy or of similarity: "... apprehensions of even vague similarities between the old and the new are often starting points for important advances in knowledge" (Nagel, 1961, p. 107).

31Hesse proposed also a 'vertical' constraint on analogy-making: the knowledge we have of the source domain, especially the knowledge of causal relations, provides criteria for determining what are the relevant properties to be mapped. Hesse's second materiality constraint, as I understand it, addresses the problem of the filtering of the relevant properties. The emphasis on causal relations has a pragmatic justification (refer to the role of the 'purpose' constraint in Holyoak and Thagard's account of analogical reasoning) . The basic aim of analogical modeling in science is to provide explanations for the target phenomena. Therefore, the relevant 'vertical' relations that should constrain the selection of properties, the mapping and the analogical inferences (transfer) are causal relations. See also McLaughlin (1982, p. 90).

32Cf. Sellars' distinction between first order and second order similarity between particulars. In the latter, the predicates are not identical but similar. He argues for a notion of 'similarity' based on 'second order' properties shared by particulars, in a criticism to Hesse's similarity at the level of first-order (that is, observational) properties. The problem I see with Sellar's account is that, in his examples, second-order properties are always of a formal character (1965, p. 178-84). See also Brown (1986, p. 295), Cartwright (1993).

be mapped to 'loudness'? Or 'color' to 'pitch'? Furthermore, are these the relevant properties to be mapped?

In a top-down account, 'similarities' are not the starting point but the upshot of analogical modeling. In the approach proposed by Aronson et al., a type-hierarchy 'filters' the relevant features and provides guidelines for the mapping: two systems are 'similar' if they can be represented as subtypes of an abstract type of system. As an illustration, they propose a reconstruction of the analogy between the solar system and Bohr's atom as involving an abstraction to a common supertype: 'central force field'. The selection of the relevant properties in both domains and the way they should be mapped are a consequence of picking up a supertype in the type-hierarchy at an adequate level of abstraction.[33] The similarity between the two systems is, therefore, 'perceived' at the level of the common supertype. The encoding of the target and the transfer of semantic knowledge from the source to the target is mediated by the type-hierarchy, through procedures of abstraction and inheritance involving the subtypes (that is, the source and the target systems) and the (proposed) supertype.

The way we represent (or re-represent) the source and the target domains depends on how we place them in a global 'ontology', or in the present state of our knowledge of the world.[34] This constitutes a top-down pressure in analogy-making (guiding the construction and the modification of the representations as well as the mapping between them). We can expect that this knowledge will eventually change as a consequence of analogy-making itself (old links between concepts are destroyed, new supertypes and links are created, etc.) .

In the first part of this paper, we emphasized that abstraction and instantiation are also processes involved in the way COPYCAT and TABLETOP make analogies.[35]

Hesse's arguments for a 'horizontal' constraint cannot be dismissed, though. We can make her point using the very same framework proposed by Aronson. In a typical situation, the common supertype is unknown or doesn't even exist in the type-hierarchy: it has to be generated. This will involve an interplay between contextual ('bottom-up' or perceptual) and conceptual ('top-down') pressures. The way the source and the target are initially

---

33They criticise Gentner because in her account of this analogy the features of the source and the target are compared 'at the same level', that is, without presupposing an abstraction procedure guided by a type-hierarchy (see Way, 1991, p. 140-6).
34See also McLaughlin (1982, p. 94)
35French claims that abstraction and inheritance both presuppose 'slippages'. What Mitchell and Hofstadter called 'conceptual slippage' takes place in the instantiation or inheritance of concepts from the supertype (or what he calls an 'abstract schema'), when another 'slippage' of those abstract concepts that don't apply to the target is often necessary. He argues that the latter slippage is often necessary given the level of abstraction chosen initially. French makes explicit the problem of how to find out the right level of abstraction, avoiding overgeneralization or little generalization. How do we know, for instance, in the analogy between the atom and the solar systems, that 'central force field' is at the right level of abstraction (and not, say, 'dynamical system' or 'planetary system', as cases of overgeneralization or of little generalization, respectively)? Aronson et al. don't give any clue about this, but French suggests that the supertype should be located at Rosch's 'basic level' (French, 1995, p. 4). The level of abstraction determines the conceptual slippages necessary to instantiate the abstract schema in the target domain.

represented constitutes an important contextual pressure in coming-up with an analogy between them. Nonetheless, the source has often to be ‘re-represented’ to be able to become a model for the target domain (in COPYCAT this corresponds to the destruction and re-building of structures and bonds in the Workspace). Given a particular context, some features in both domains are highlighted and others down-played (and eventually ignored). The source and the target representations ‘interact’ in many ways in model-building.[36]

It can also be argued, against purely ‘top-down’ approaches to analogy-making, that if we know beforehand how the source and the target are located in the type-hierarchy, as well as the common supertype the source and the target systems are subtypes of, then the representation of the source system is entirely redundant. In this case, we wouldn’t have to rely on ‘analogical’ reasoning: it would be enough to instantiate directly the knowledge we have of the supertype in the target domain (see Dejong, 1989).

We are far from having a general theory of analogical modeling in science. Simulations of analogy-making in simple and idealized worlds, like that of COPYCAT, can help figure out the basic processes and constraints that should be investigated. I hope these general considerations can at least stimulate an approximation of theoretical developments in fields that still largely ignore each other.[37]

References

P. Achinstein: 1968, Concepts of science. Johns Hopkins University Press, Baltimore.

J. L. Aronson, R. Harré and E. C. Way: 1995, Realism rescued: how scientific progress is possible. Open Court, Chicago.

P. Bartha: 1994, Analogical reasoning and plausibility in the sciences. Doctoral dissertation, Department of Philosophy, University of Pittsburgh.

M. Black: 1962, Models and metaphors. Cornell University Press, Ithaca.

_______: 1979, More about metaphor. In A. Ortony (ed): Metaphor and thought. Cambridge University Press, Cambridge, pp. 10--43.

R. B. Braithwaite: 1953, Scientific explanation. Cambridge University Press, Cambridge.

J. R. Brown: 1986, Sellars, concepts and conceptual change. Synthese 68: pp. 275--307.

36This interaction is central to Black’s theory of metaphor, that is accepted by many accounts of modeling, like those proposed by Hesse and by Aronson et al. See also Holyoak et al. (1995, p. 193). An interaction between source and target is also central in Indurkhya’s characterization of ‘similarity-creating metaphors’ (1992). Nersessian (1992) showed the importance of the ‘manipulation’ of the source in the modeling procedures underlying Maxwell’s construction of his electromagnetic theory.

37I am grateful for the remarks and corrections suggested by the referees.

N. R. Campbell: 1920, Physics: the elements. Cambridge University Press, Cambridge.

N. Cartwright: 1993, How we relate theory to observation. In P. Horwich (ed): World changes. The MIT Press, Cambridge, pp. 259--273.

A. Clark: 1993, Associative engines: connectionism, concepts and representational change. The MIT Press, Cambridge.

J. Cushing: 1982, Models and methodologies in current theoretical high-energy physics. Synthese 50: pp. 5--101.

T. Dartnall (ed.): 1994, Artificial intelligence and creativity: an interdisciplinary approach. Kluwer, Dordrecht.

G. Dejong: 1989, The role of explanation in analogy; or, The curse of an alluring name. In S. Vosniadou and A. Ortony (eds): Similarity and analogical reasoning. Cambridge University Press, Cambridge, pp. 346--365.

C. Dierbach and D. Chester: 1991, A formal basis for analogical reasoning. In J.A. Allen, R. Fikes and E. Sandewall (eds): Principles of knowledge representation and reasoning: proceedings of the second international conference. Morgan Kaufmann Publishers, San Mateo.

R. M. French: 1995, The subtlety of sameness: a theory and computer model of analogy-making. The MIT Press, Cambridge.

D. Gentner: 1982, Are scientific analogies metaphors? In: D. S. Miall (ed): Metaphor: problems and perspectives. Harvester Press, Brighton, pp. 106--32.

_________: 1983, Structure-mapping: a theoretical framework for analogy. Cognitive Science 7, pp. 155--170.

_________: 1989, The mechanisms of analogical learning. In S. Vosniadou and A. Ortony (eds): Similarity and analogical reasoning. Cambridge University Press, Cambridge, pp. 199--241.

D. Gentner and D. Gentner: 1983, Flowing waters or teeming crowds: mental models of elecricity. In D. Gentner and A.L. Stevens (eds): Mental models. Lawrence Erlbaum, Hillsdale, pp. 99--129.

R. Giere: 1988, Explaining science: a cognitive approach. University of Chicago Press, Chicago.

N. Goodman: 1976, Languages of art. Hackett Publishing Co., Indianapolis.

R. P. Hall: 1989, Computational approaches to analogical reasoning: a comparative analysis. Artificial Intelligence 39: pp. 39--120.

R. Harré: 1970, The principles of scientific thinking. University of Chicago Press, Chicago.

______: 1987, Varieties of realism. Blackwell, Oxford.

______: 1988, Where models and analogies really count. International studies in the philosophy of science 2: pp. 118--133.

D. H. Helman (ed): 1988, Analogical reasoning. Kluwer, Dordrecht.

C. G. Hempel: 1965, Aspects of scientific explanation. Free Press, New York.

M. Hesse: 1954, Models in physics. British Journal for the Philosophy of Science 4: pp. 198--214.

________: 1961, Forces and fields. Littlefield, Adams & Co., Totowa.

________: 1966, Models and analogies in science. University of Notre Dame Press, Notre Dame.

________: 1967, Models and analogy in science. In P. Edwards (ed): Encyclopedia of Philosophy 5: pp. 354--359.

________: 1988, Theories, family resemblances and analogy. In D. H. Helman (ed): Analogical Reasoning. Kluwer, Dordrecht, pp. 317--340.

D. Hofstadter et al.: 1995, Fluid concepts and creative analogies. Basic Books, New York.

K. Holyoak and P. Thagard: 1989, A computational model of analogical problem-solving. In S. Vosniadou and A. Ortony (eds): Similarity and analogical reasoning. Cambridge University Press, Cambridge, pp. 242--266.

__________: 1995, Mental leaps. The MIT Press, Cambridge.

B. Indurkhya: 1992, Metaphor and cognition. Kluwer, Dordrecht.

S. Kedar-Cabelli: 1988, Analogy- from a unified perspective. In D. H. Helman (ed): Analogical Reasoning. Kluwer, Dordrecht, pp. 65--103.

P. Kroes: 1989, Structural analogies between physical systems. British Journal for the Philosophy of Science 40: pp. 145--54.

R. McLaughlin: 1982, Invention and appraisal. In R. McLaughlin (ed): What? Where? When? Why?, Reidel, Dordrecht, pp. 69--100.

M. Mitchell: 1989, Analogy-making as perception. Cambridge University Press, Cambridge.

C. T. Morrison: 1994, Analogy & constructive representation. Master thesis, State University of New York at Binghamton.

E. Nagel: 1961, The structure of science. Routledge & Kegan Paul, London.

N. Nersessian: 1987, A cognitive-historical approach to meaning in scientific theories. In N. Nersessian (ed): The process of science: contemporary philosophical approaches to understanding scientific practice. Nijhoff, Dordrech, pp.161--178.

__________: 1992, How do scientists think? Capturing the dynamics of conceptual change in science. In: R. Giere (ed): Cognitive models of science. University of Minnesota Press, Minneapolis, pp. 3--44.

S. Owen: 1990, Analogy for automated reasoning. Academic Press, Boston.

S. E. Palmer: 1989, Levels of description in information-processing theories of analogy. In S. Vosniadou and A. Ortony (eds): Similarity and analogical reasoning. Cambridge University Press, Cambridge, pp.332--345.

M. Redhead: 1980, Models in physics. British Journal for the Philosophy of Science 31: pp. 145--163.

L. J. Rips: 1989. Similarity, typicality, and categorization. In S. Vosniadou and A. Ortony (eds): Similarity and analogical reasoning. Cambridge University Press, Cambridge, pp. 21--59.

W. Sellars: 1965, Scientific realism or irenic instrumentalism. In R. S. Cohen and M. W. Wartofsky (eds): Boston Studies in the Philosophy of Science 2: pp. 171--204.

M. Steiner: 1989, The application of mathematics to natural science. Journal of Philosophy 86: pp. 449--480.

P. Thagard: 1962, Conceptual revolutions. Princeton University Press, Princeton.

B. van Fraassen: 1987, The semantic approach to scientific theories. In N. Nersessian (ed): The process of science. Nijhoff, Dordrecht, pp. 105--124.

S. Vosniadou: 1989, Analogical reasoning as a mechanism in knowledge acquisition: a developmental perspective. In S. Vosniadou and A. Ortony (eds): Similarity and analogical reasoning. Cambridge University Press, Cambridge, pp. 413--437.

S. Vosniadou and A. Ortony (eds): 1989, Similarity and analogical reasoning. Cambridge University Press, Cambridge.

E.C. Way: 1991, Knowledge representation and metaphor. Kluwer, Dordrecht.