PHILOSOPHY RESEARCH ARCHIVES
VOL. XI, MARCH, 1986

33. STRUCTURES AND PROCEDURES: CARNAP'S CONSTRUCTION IN THE *AUFBAU*

WILLIAM M. GOODMAN
DURHAM COLLEGE

ABSTRACT. This paper takes up the challenge which Carnap poses in his *Aufbau*: to make of it a basis for continued epistemological research. I try to close some gaps in Carnap's original presentation and to make at least the first few steps of his constructional outline more accessible to the modern reader. Particularly emphasized is Carnap's implicit recognition that, to be effective, "structural" models of epistemology (using logical symbols) must be complemented with "procedural" models (his "fictitious operations"). The paper shows how a procedural model, a computer program, can "bypass" Nelson Goodman's counter example to Carnap's logical construction of "similarity circles".

This paper is in part about some logical constructions found in Rudolf Carnap's *Der Logische Aufbau der Welt* (*The Logical Structure of the World*). According to the original preface for that work, its subject matter concerns primarily "questions of epistemology, that is . . . questions of the reduction of cognitions to one another".[1] Carnap does not attempt to offer a final solution to these questions in the *Aufbau*, only a *foundational outline*. Hence, the system forms and relations he introduces are, by his own admission, only "tentative"; and the *Aufbau's* purpose is served if it has illustrated a scientific method for pursuing epistemological inquiry and has encouraged others to fill in or correct its details (106).

Although, as Nelson Goodman once wrote, the *Aufbau* has become "anathema" to nearly everyone[2] (even Carnap himself later rejected the system and its mode of strictly extensional definition[3]), it would be a mistake, I believe, to overlook its significance for the general advancement of epistemological study. For in attempting to attain new levels of precision in its argument, it actually presages a *procedural* interpretation of constructivism which is even now still in its infancy in current "artificial intelligence" research. Therefore, this paper has a second purpose: to use Carnap's text as a "case study" to illustrate the crucial importance of employing such procedural modeling techniques in epistemological investigations. Included in this paper are details of a computer program I wrote for actually implementing a procedural counterpart to a Carnapian class construction.

It is suggested here that the crucial innovation of the *Aufbau* lies with its *method*, rather than with its particular constructivist premisses.

That the raw data for epistemology consist of uninterpreted and non-intentional experiences was a view developed previously by the sensationists. Again, the insight that when we refer to "objects" in our cognitions, we have already performed a constructive *synthesis* on the "manifold of our intuitions" had been stated before by Kant--among others.[4] But what makes Carnap's effort unique is that it moves beyond the arguments which claim merely *that* such construction takes place, and it systematically tries to *model* this process in a rigorous and formal way. Moreover, it sketches outlines both for modeling the *structure* of what is created by this synthesis and for representing a set of dynamic *procedures* by which this synthesis might occur.

In the next few paragraphs I shall endeavor to explain what I mean by this distinction between "structure" and "procedure". Aaron Sloman, a philosopher and researcher in artificial intelligence (AI) at the University of Sussex, elaborates on some motivations for philosophically oriented research in his field:

> By trying to turn our explanations and theories into designs for working systems, we soon discover their poverty. The computer, unlike academic colleagues, in not convinced by fine prose, impressive looking diagrams or jargon, or even mathematical equations. If your theory doesn't work then the *behaviour* of the system you have designed will soon reveal the need for improvement. Books don't behave. We have long needed a medium for expressing theories about behaving systems. Now we have one . . .. Progress in philosophy (and psychology) will now come from those who take seriously the attempt to *design a person*.[5]

Carnap, in his *Aufbau*, shares intentions very similar to those of Sloman. In effect, he says with this work: "Let us discard pointless discourse on metaphysics, etc., in favour of the designing of actual, sample constructions which can be *tested* for their effectiveness".

But there are fundamentally different ways of carrying out, or at any rate describing, such a "design". One of these is the logistic approach, which represents each "construction" or design in the form of a logical representation of a class; where the logical expression for that class is *derivable* by standard rules of logic from the primitive "givens"--possibly via intermediate derivations. I call this the "structural" approach because it focuses on the logical *structure* of the class which is to be constructed. Carnap prefers to rely primarily on such structural representations in his system, since the language of logic provides (in virtue of its formal rules and proof-procedures) a seemingly objective basis for the desired independent testing of one's constructions. All other ways of looking at the system, "languages" as Carnap would have it, appear to him as useful only for aiding one's intuitive understanding of the concepts involved; and so Carnap treats them as merely convenient auxiliaries to the logic.

Nonetheless, Carnap does take some time to develop alternative models for the classes he constructs. He calls these "fictitious operations". In these models, which I would call "procedural", one focuses not on the structure of the derived classes as such, but rather on the procedures by which one could generate these classes given only the primitive data plus all classes already constructed. Ironically, though Carnap

objects that these sorts of models are not testable in the manner that the logic is, it appears that he never attempts to engage in this testing in earnest. That is, often in his text, if one were actually to consider a sample set of objects in a class and mentally perform the operations upon them which Carnal describes, one would quickly discover that, in fact, they fail to result in classes of the sort allegedly being derived.

To illustrate the distinction between "structural" and "procedural" representations of constructions, consider the twin representations of an assembled radio or other device which can be found typically in hobbyist magazines: (a) a schematic representation of the fully assembled object vs. (b) a set of step-by-step instructions for building it. Because Carnap believes that the first approach alone is sufficient, a number of the fictitious operations he describes take on a curious form: "A identifies those classes which satisfy such-and-such logical conditions". Sometimes this sort of expression could describe a practicable method (as in "A identifies all classes satisfying the condition of having more than 5 members"), but most often the "operation" as stated is meaningless. Imagine being told that the procedure for building a radio is simply to "identify those sets of components which satisfy the structure of the schematic diagram". Clearly, *no* set of components is likely to start out by satisfying those conditions; and what the schematic leaves out is precisely how to cause it to occur that some of them *take on* that structure.

No map or model is complete in itself, but if these two complementary guides (of "schematic" and logical description and of "building instructions" or procedural model) are used in conjunction, they can often correct each other's weaknesses. If, for instance, the phrasing of an instruction is ambiguous, one can look at the schematic to see what is the intended result of the operation. If, on the other hand, the ambiguity lies with the schematic itself, one can look to the corresponding steps in the instructions to find out how one is expected to generate the structure which is illustrated. Just this two-pronged method is implicit in Carnap's foundational outline of the *Aufbau*. This paper will endeavor to make that method explicit.

As I have suggested, Carnap does in fact represent most of his constructions both in the "*symbolic language of logistics*" and also in the "*language of fictitious constructive operations*" (95). Yet, because he holds that only the "symbolic language of logistics [especially as developed in Whitehead's and Russell's *Principia Mathematica*] gives the proper and precise expression for the constructions", he numbers the fictitious operations among the three other approaches which "serve only as more comprehensible auxiliary languages" (95). These "other" languages consist of translations of each logical class description (1) into "a simple *paraphrase in word language*", (2) "into the *realistic language* customary in the empirical sciences", and (3) into the procedural "fictitious operations" of present concern. The latter representations (3) he views merely as devices for facilitating "the intuitive recognition of the *formal* correctness of the construction (i.e., the testing of whether each constructional definition is . . . not ambiguous, not empty, and purely extensional)". The paraphrases (1) are geared simply to help the reader follow the logical derivations. Finally, the realistic language (2) is helpful for "testing the correctness of the *content* of the [constructions] (i.e., whether or not the constructional definition actually refers to the familiar object to which it purports to refer)" (95).

In effect, then, this paper takes up the challenge which Carnap poses in his *Aufbau*--namely, to make of it a basis for continued epistemological research. In an attempt to revitalize Carnap's effort, the paper tries to "update" his own presentation, to fill in some of his text's more obvious gaps, and generally to make at least the first few steps of its constructional outline more accessible to the interested modern reader. For continuity, I retain Carnap's use of the above-mentioned "four languages": and where one of these appears incomplete or ambiguous, I attempt to reconstruct what is missing, and to append needed sections of "Comments", "Notes", or "Explanations".

Moreover, when the crucial step of constructing "similarity circles" is reached, the important contributions of Nelson Goodman to Carnap study are carefully considered. I refer particularly to his "counter example" to Carnap's method in that place, since this is often considered very damaging to Carnap's *Aufbau* system. It is in that discussion that I describe my working computer program for constructing similarity circles procedurally; and, in the process, an algorithm which can apparently "sidestep" the import of Nelson Goodman's counter example is examined.

A possible advantage of this exercise is that, hopefully, it can counterbalance the view of AI thinkers such as Sloman who believe that all attempts at reductionism are "refuted" by AI work. Since, for example, a high level program such as an accounting package can run on *any* brand of hardware, Sloman concludes by analogy that high level cognitive processes too need not be "reduced" to some low level, primitive human faculties. In fact, those same high level processes could be conceived and understood at their own level, even if they were somehow able to run on alternative hardware.[6]

What Sloman neglects to consider, however, is that even though high level programs can be conceivably run on "any" hardware, they can only run in practice on *some* hardware or other. That is, just as *compilers* are needed in computers to reduce the language of the high level package to the particular machine language capabilities which are available, so too is work such as Carnap's necessary to show that human cognitive processes can be "compiled" down to the primitive capabilities of the human "hardware". The *Aufbau* primitives model the effects of operation of what Pylyshyn might call the "functional architecture" of the human machine,[7] while Carnap's constructional system represents a possible "compiler" which can reduce full-blown human epistemology to that primitive "machine language" via constructions by extensional means. What I hope this paper can contribute is an understanding that such "constructions", to be workable and testable, must be crafted from a structural *and* a procedural point of view.

THE SYSTEM BASIS

There are two "primitives" in Carnap's constructional system: the basic relation and the basic elements. But the "basic elements"--*qua* primitive (i.e., non-derived)--are only the arguments which appear in the basic relation. As soon as these elements are grouped into an identifiable class of their own, a constructional step has already been performed. Thus, although from one point of view there are indeed two

primitives in the *Aufbau* system, from another viewpoint we see that it contains *only one primitive step*: the introduction of "Rs", the "Recollection of Similarity" relation.

By means of the Rs-relation one can order one's experience. Neither Rs without experiential content, nor this "raw experience" without its ordering by Rs, has any meaning or interpretation in this system. Hence section 108 of the *Aufbau*, which introduces Rs, is in fact concerned with *both* the system "primitives" named above. Once Rs is applied over a set of arguments, however, the true process of construction begins.

Section 108, as mentioned, treats of the Rs relation and its empirical properties. Section 109 constructs a class of objects, elex, whose members are the primitive, basic elements. In section 110, a more general relation than Rs is introduced, Part Similarity, which will in turn be highly useful for generating further objects and relations.

The Basic Relation (Rs) [108]

Basic Relation: Rs

Relation Form: Rs(x,y) [or: x Rs y]

Paraphrase: Recollection of similarity holds between x and y. (78)

Realistic State of Affairs: A representation of an elementary experience x is recalled and found to be part similar to another elementary experience y (i.e., "x and y are found to agree approximately in a constituent"). (108)

Fictitious Operation: It is not initially supposed that the constructing individual--call her A--knows anything about the structure of the Rs relation. All that is given is a *basic relation list*, which is the exhaustive set of all pairs of basic elements which have served for A as arguments in the Rs field: that is, this list, as a whole, comprises the relation extension of Rs in the experience of A. On this basis the following operation can be performed:

1. Suppose that A has amassed the following basic relation list:

   {<1,5><2,5><4,5><2,10><4,10><5,6><5,8><5,9><5,10>}

2. Each number in the above list is simply an "arbitrary but determinate" token (108) that designates one argument which has appeared in the Rs-extension during the course of A's experience. This *inventory list*--consisting solely of the various numbered elements which have been ordered by the Rs relation--is called the *basic relation list*. It contains the only material which A has available for synthesis.

3. Even at this low level, however, some synthesis is already possible. A is now in a position to begin formulating *object descriptions*. With the exception of the basic elements (i.e., the arguments of the Rs relation), all "objects" in this system are

in fact "quasi-objects"--i.e., they are *classes* which have been *constructed*. But for all objects, constructed or primitive, one can build up "object descriptions" by adding *supplemental entries* corresponding to each member of the given object/class. At present A has the tools at hand--namely, the basic relation list which has been ordered by Rs--to build up object descriptions for the basic elements, by specifying the following supplemental entries for each member of the Rs inventory:

a) What other members of the basic relation list stand directly in the Rs relation to this particular member?

b) To what other members does this element, itself, stand in the Rs relation?

4. Example 1: Given the pair-list data shown in (1), the following supplemental entries can be made for the element denoted as "5". In realistic language, these entries tell us information "about the object" which "5" denotes.

   a) Rs (x,5) holds when x is a member of {1,2,4}

   b) Rs (5,y) holds when y is a member of {6,8,9,10}

5. Example 2: For the same pair list, only one supplemental entry can be given for element 4; since

   a) Rs (4,y) holds when y is a member of {5,10}

   b) Rs (x,4) does not occur.

6. As the pair list expands, the object descriptions can be appropriately revised. For instance, when the pairs <5,11><4,12> are added to A's inventory, A can then revise the supplemental entries for 4 and 5 accordingly:

   a) Rs (5,y) holds when y is a member of {6,8,9,10,11}

   b) Rs (4,y) holds when y is a member of {5,10,12}

Theorem: TH 1 Rs ϵ as (empirical)

Paraphrase: The relation Rs is a member of the class of asymmetrical relations; i.e., Rs is an asymmetrical relation.

Fictitious operation:

1. A has built up an inventory list for the basic relation. For example,

   {<1,2><1,3><1,5><2,6><1,7><2,7><3,7><4,8><1,10><2,10><5,10> ...}

2. On inspecting this list, A observes *empirically* that, given any ordered pair of distinct arguments <x,y> which appears in that list, there is *never* also in the list a corresponding pair comprised of those same elements in reverse order <y,x>. Thus A ascertains the asymmetry of the ordering relation Rs.

3. We call the theorem empirical because A discovers its truth solely by an actual inspection of her basic relation list data. These data reflect the ordering which Rs has imposed on A's *experience*. That is, A learns the structure of Rs only by observing the inventory class which it in fact creates.

The Basic Elements (elex) [109]

Carnap's Construction: elex $=_{df}$ C' Rs

Paraphrase: The class elex is the class of elements which have comprised (for A) the field of the Rs relation.

Explanation of Symbols: Many of Carnap's *Principia*-style symbols (such as "C' ") are no longer widely used or familiar. Hence each of Carnap's derivations will be translated into a "revised notation". These revisions will be given in the form of contextual definitions, so that it becomes more obvious that what Carnap really intends are definitions *in use* (i.e., definitions which explain what one means to say *about* α when one uses the constructed term "α" in a sentence) (39,40). In this instance, the revised version of Carnap's elex definition can be styled as follows:

Revised Notation:

$$\alpha \in \text{elex} =_{df} (\exists x)\,[Rs(\alpha,x) \vee Rs(x,\alpha)]$$

Paraphrase: Both members of any pair list which is formed by applying Rs are called *elementary experiences*. The class which contains *all* members which appear in ordered pairs of the Rs inventory is the class elex of (all) elementary experiences.

Realistic State of Affairs: The basic relation is Rs, the recollection of similarity. The arguments for this relation are taken from the lowest-level *object sphere* (cf. 29);, i.e., its member-objects cannot be quasi-objects (which are themselves already constructions from some previous stage of the system). Therefore, its arguments must be considered the *basic elements* of the system; and the experiences they represent can be deemed the *elementary experiences*.**Fictitious Operation:**

1. A has formed a basic relation list, e.g.:

   {<1,2><3,4><3,5><4,6><1,7><2,7><3,7><7,8>}

2. From these data the inventory list for the class elex can now be constructed. Namely, it is the class which contains, as its elements, each unique member of the Rs pairlist.

   Thus, from (1), we derive:

   elex = {1,2,3,4,5,6,7,8}

3. It was shown earlier how to begin an object description based solely on the data in the Rs pair list. Given the inventory

which is illustrated in (1), for example, A could now enter these three supplemental entries for element 7:

a) Rs(x,7) holds when x is a member of {1,2,3}

b) Rs (7,y) holds when y is a member of {8}

c) 7 ∈ elex

Part Similarity (Ps)
[110]

Relational Form: Ps(x,y) [or: x Ps Y]

Carnap's Construction: Ps $=_{df}$ Rs ∪ $\breve{R}$s ∪ Rs°

Paraphrase: The extension of the relation Ps is constructed as the union of the extensions for:

a) the relation Rs itself;

b) the converse relation of Rs (i.e., the relation which holds between the Rs arguments, but in the opposite order); and

c) the relation which holds between identical arguments in the Rs field.

Fictitious Operation (for Carnap's Version):

1. A has already constructed an inventory list for the Rs relation, e.g.,

   {<1,2><1,3><2,4><3,5><4,5><1,5>}

2. From this, the inventory for $\breve{R}$s can be constructed also:

   {<2,1><3,1><4,2><5,3><5,4><5,1>}

3. Further, A can identify the individual members of the field of Rs, namely, 1,2,3,4,5, and construct the relation Rs° which is the identity relation as it applies to each member of the field: thus

   {<1,1><2,2><3,3><4,4><5,5>}

4. The extension of the relation Ps is therefore defined as the union of the sets which have just been enumerated in steps 1.-3., i.e. (in this instance),

   {<1,2><1,3><2,4><3,5><4,5><1,5><2,1><3,1><4,2><5,3><5,4><5,1><1,1><2,2><3,3><4,4><5,5>}

Revised Notation:

Ps(x,y) $=_{df}$ (x ∈ elex) & (y ∈ elex) & [Rs(x,y) ∨ Rs(x,y) ∨ (x=y)]

Paraphrase: Two elementary experiences, x and y, are *part similar* if either (1) the Rs relation holds between x and y, or (2) it holds between y and x, or (3) if "x" and "y" name the same elementary experience.

Fictitious Operation (for the construction in Revised Notation): A employs the contextual definition to generate the extension for the part-similarity relation. Only elementary experiences are contained in the fields of Ps. Thus, given any two members, x and y, of the class elex, A adds them to the list of Ps members, provided either Rs(x,y) or Rs(y,x) holds between them. Also, given any identical member of the elex inventory, x=y), this pair also (i.e., $\langle x,y\rangle = \langle x,x\rangle$) is added to the Ps list.

Example:

1) A has constructed an Rs pair list:

   {<1,2><1,3><2,5><3,5><5,6>}

2) The corresponding elex inventory is

   {1,2,3,5,6}

3) Next, the exhaustive set of possible <x,y> pairs that satisfy [(x ∈ elex) & (y ∈ elex)] is constructed:

   {<1,2><1,3><1,5><1,6><2,3><2,5><2,6><3,5><3,6><5,6><2,1><3,1><5,1><6,1><3,2><5,2><6,2><5,3><6,3><6,5><1,1><2,2><3,3><5,5><6,6>}

4) Of these, the pairs which satisfy Rs(x,y) or Rs(y,x) or (x=y) (and so correspond to Ps) are identified:

   {<1,2><1,3><2,5><3,5><5,6><2,1><3,1><5,2><5,3><6,5><1,1><2,2><3,3><5,5><6,6>}

Realistic State of Affairs: If two elementary experiences, x and y, are related by part similarity, then it follows that some part of the experience x is similar to some part of the experience y; and a part of y is likewise similar to a part of x.

Fictitious Operation (Continued): Whichever of the above two versions of "fictitious operation" we assume A has used, A will in any case have now constructed her inventory for the extension of the Ps relation. These data are now used to create additional supplemental entries for the object descriptions of each Rs-member. Below, I illustrate the three types of supplemental entry which have so far been introduced:

Given the Basic Relation List

{<1,4><2,5><4,5><4,6><5,6><5,8><4,10><5,10>}

the supplemental entries to be entered for member 5 are:

1) Rs(x,5) holds when x is a member of {2,4}
2) Rs(5,y) holds when y is a member of {6,8,10}
3) 5 ∈ elex
4) Ps(z,5) holds when z is a member of {2,4,5,6,8,10}

Theorem: **Th. 2** Ps $\in$ sym (analytic)

Paraphrase: The relation Ps is a member of the class of symmetrical relations; i.e., Ps is a symmetrical relation.

Theorem: **Th. 3** Ps $\in$ refl (analytic)

Paraphrase: The relation Ps is a member of the class of reflexive relations; i.e., Ps is a reflexive relation.

Note: These theorems are considered *analytic* since they follow as a direct consequence of how Ps is *constructed.* That is, no appeal to actual pair-list data is required to confirm them.

That this is so in the case of Theorem 2 becomes especially obvious on inspection of Carnap's construction for Ps which is provided above: If any ordered pair $\langle x,y \rangle$ is in the extension of Ps (where x is not equal to y), then the pair must also be in either the extension of Rs or of $\breve{R}s$. If in the former, then, by the construction, both $\langle x,y \rangle$ itself and its converse $\langle y,x \rangle$ are included by union in the Ps field. If in the latter (i.e., if $\langle x,y \rangle$ appears not in the Rs-list but in the extension of its converse relation, $\breve{R}s$), then it follows that $\langle y,x \rangle$ must have occurred as a member-pair in the original Rs inventory. Once again, *both* orderings of the $\langle x,y \rangle$ pair will be included in the Ps list by union. In either case, we see that a symmetrical structure for the Ps list is insured.

That Ps is reflexive is likewise analytic, based on the Ps-constructions. Both versions alike simply *declare*, in their respective notations, that, given any member x of the elex class, its identical pair, $\langle x,x \rangle$, is included in the Ps inventory. This makes the latter reflexive in structure.

THE STRUCTURE OF SIMILARITY CIRCLES

In Carnap's system, the elementary experiences are non-constructed objects at the lowest possible level; for they are merely the primitive arguments for the basic relation. Once "elex" has been constructed as a class or concept, however, each individual elex (i.e., each elex member) can next be assigned a number of properties (such as membership in elex and part similarity to other specific elex's) by means of supplemental entries.

In the remainder of this paper, we will explore in some detail the construction of the next, *higher level* set of objects in Carnap's system, the Similarity Circles. Although a constructed class, a "similcirc" (i.e., a similarity circle) is still very primitive as compared with, say, a constructed phenomenal quality such as "red-in-color" or "rough-in-texture". Indeed, the nature of similarity circles can be scarcely comprehended by our (usual) intuitions.

To imagine what sort of entity a similarity circle is, one might attempt the following experiment: Open and shut your eyes very quickly and recall to yourself everything you saw, felt, heard, smelled, and tasted in that instant when your eyes were open. (Try, as much as possible, to forget your names for things and to attend just to the impressions of color, texture, and so on.) Next, group together, with this expe-

rience, all your images for *every* other such experience in which *something* was similar to that last one (whether in taste, smell, touch, sound, or appearance)--and include *all* aspects of those images (i.e., also include the parts which are not similar to the image being compared). If you can imagine the collectivity of all those impressions--which is, in practice, unlikely--you are roughly picturing to yourself a single "similarity circle".

In this section, we take a close look at the structural description of a constructed similarity circle. Nelson Goodman's classic counter example to Carnap's construction will be introduced, as well an alternative derivation which appears to bypass that objection. Discussion of a procedural complement to these descriptive models will be reserved for the section following.

Similarity Circles (similcirc) [111]

Carnap's Construction: similcirc $=_{df}$ Simil' Ps

Paraphrase: Similcirc is a class of classes, whose member-classes are just those "similarity circles" which can be derived by quasi-analysis based on the Part-Similarity relation.

Explanation: Earlier in his text, Carnap has elaborated a concept of "quasi-analysis" in some detail. (See, especially, his sections 70-72 and 80.) He has hoped to make clear how by using a symmetrical and reflexive relation--such as Ps--the classes called "similarity circles" can be constructed. Hence, similcirc is now simply *defined* as being that class which is comprised of *all* similarity circles which can be generated by such quasi-analysis based on Ps.

Although Carnap is not explicit in this section on how his construction "Simil' Ps" should be interpreted, his guidelines from those earlier-mentioned discussions can be used to get a clear picture of his intentions. My own interpretation follows this note, under the heading "Revised Notation". In keeping with the preference of this paper, it is presented as a contextual definition.[8]

Revised Notation: $\alpha \in$ similcirc $=_{df}$

$$(e)[(e \in \alpha) \rightarrow (e \in \text{elex})] \;\&$$
$$(u)[(v)[(v \in \alpha) \rightarrow \text{Ps}(u,v)] \rightarrow (u \in \alpha)] \;\&$$
$$(w)(x)[((w \in \alpha) \;\&\; (x \in \alpha)) \rightarrow \text{Ps}(w,x)]$$

Paraphrase: Similcirc is a class of classes. For any given class, α, to meet the conditions for membership in similcirc, the following must hold true of it:

1) All its members must be elementary experiences; and
2) Any elementary experience, u, which is such that all members of α are part-similar to that experience (u) must be itself a member of α; and
3) All elex-pairs (distinct or identical) which are members of α must be part-similar to each other.

Notes:

1. According to this construction, similcirc is precisely defined in terms of previously constructed objects, the basic elements, and the traditional symbols and operations of modern logic.

2. The construction puts forward three main conditions for membership of a class α in the class of all similarity circles; and these requirements closely correspond to Carnap's own natural-language definition for similcircs, which he provides in his section 80 (bracketed numbers added):

   > Thus, by 'similarity circles' we understand those classes of elementary experiences which have the following two properties: (1) any two elementary experiences of such a class are part similar to one another (Ps); (2) if an elementary experience is part similar to all elementary experiences in such a class, then it belongs itself to that class.

   Compare the paraphrase for the present construction:

   1) Carnap's condition (1) appears as the present condition (3);
   2) Carnap's condition (2) corresponds to the present condition (2); and
   3) The present condition (1) is *presupposed* when Carnap states that his own two conditions are applied to classes of *elementary experiences*.

3. Historically, one of the most influential criticisms of Carnap's *Aufbau* was published by Nelson Goodman in his own *Structure of Appearance*. Goodman's objections to Carnap's construction of Similarity Circles struck at the very foundation of Carnap's system; yet they clearly assumed an interpretation of Carnap's similcirc construction such as that presented in my "revised notation". If it is possible that similcircs could be constructed *differently*, then their derivation might conceivably escape the force of Goodman's arguments.

In brief, Goodman's objection to the construction hinges primarily on the problem of what he calls "imperfect community". This is the situation which is said to apply when elex's can be joined into *apparent* similarity circles (based on the revised construction), and yet, "in fact", there is no single respect in which all alike are similar. This becomes possible, it is said, just when every given pair of elements in the faulty circle is similar--with respect to *some* quasi-constituent, but yet the constituent with respect to which this similarity occurs is *different* for different pairs of elements. Thus, when such "imperfect community" obtains between elements, the process which is thought to be a construction of true similarity circles is, in fact, a misleading grouping into one class of experiences which share no one common similarity.

It is to illustrate this problem that Goodman introduces his well known counter example to Carnap's construction of similcirc. Following Carnap''s own lead (70), he simplifies the presentation of his example by the use of an analogy: Instead of speaking of "elementary experiences"

having "quasi-constituents" with respect to which the elex's might be "part similar", the analogy substitutes "objects" which have "colors" which can be "the same". Hence, Goodman's example:

1. bg 2. rg 3. br 4. r 5.b 6. g

(The numbers indicate objects (elex's), and the letters refer to their respective colors (quasi-constituents).)

Such a set of objects forms the counter example to Carnap, according to Goodman, for this reason: If the objects 1, 2, and 3 are grouped into a single class, the class thus formed *appears* to satisfy the conditions for similcirc class-membership. It does so, he claims, precisely in virtue of the "imperfect community" of its members. That is, each *pair* within {1,2,3} does indeed share *some* common color--yet *no one color* is in fact shared commonly by *all* the objects. Clearly this defeats the purpose of the construction.

It is obvious that Carnap intended his "Simil' Ps" construction of similcirc to be taken as logically equivalent to what I presented previously in my revised notation. If this is the case, then his system is indeed vulnerable to the weight of Goodman's counter example. For instance, let a be the set of elex's {1,2,3} corresponding to Goodman's counter example (where the constructional terminology is used in place of the color/object analogy). Since all its members are defined as elex's, condition (1) of the derivation is met. Since any elex which is part-similar to all the class members is itself a class member, condition (2) is also satisfied. Finally, condition (3) is fulfilled, since all a's member-pairs are part-similar. Thus a set whose members share only an "imperfect community" seems able, nonetheless, to pass the logical tests given by Carnap to comprise a similarity circle.

However, it is possible to construct alternative derivations for similcirc. The alternative presented below remains faithful, I believe, to Carnap's primary intentions for "Simil' ", but yet avoids at least some of the difficulties which Goodman points out.

Alternative Construction: $\alpha \in \text{similcirc} =_{df}$
$(e)[(e \in \alpha) \rightarrow (e \in \text{elex})]$ &
$(u)[(v)[v \in \alpha) \rightarrow \text{Ps}(u,v)] \rightarrow (u \in \alpha)]$ &
$(w)(x)[((w \in \alpha)$ & $(x \in \alpha)) \rightarrow \text{Ps}(w,x)]$ &
$(\exists y)[(y \in \alpha)$ & $(z)\{\text{Ps}(z,y) \rightarrow (z \in \alpha)]$

Paraphrase: Similcirc is a class of classes. For any given class, a, to meet the conditions for membership in similcirc, the following must hold true of it:

1) All its members must be elementary experiences;
2) Any elementary experience, u, which is such that all members of α are part-similar to that experience (u) must be itself a member of α;
3) All elex-pairs (distinct or identical) which are members of α must be part-similar to each other; and
4) There must be at least one elex, y, which is a member of α, and for which it holds that *any* elex at all which is part-similar to y is *also* an element of α.

Notes: At the very least, this version *does* escape the trap posed by Goodman's specific counter example. It does so by adding a fourth criterion for a class's membership in similcirc. Unfortunately, there exist still other cases where even this third construction is insufficient.

In the next section of this paper we will see how our design of a procedural model for similcirc construction can guide our selection of an appropriate logical derivation for the class. But for now we will rest our analysis on some sample cases, and consider only structural criteria. The object/elex analogy will be maintained. For each case below, let us see what classes each of the two structural similcirc definitions last given would accept or reject as forming true similcircs. (Let "D1" = Carnap's similcirc construction (as per my revised notation); and "D2" = the alternative version.)

Example 1 (This is Nelson Goodman's counter example):

1. bg	2. rg	3. br
4. r	5. b	6.g

similcirc classes constructed by ...

D1:	vs.	D2:
{2,3,4}	[1]	{2,3,4}
{1,3,5}	[2]	{1,3,5}
{1,2,6}	[3]	{1,2,6}
{1,2,3}	[4]	

Note on Example 1: The alternative construction (D2) has successfully eliminated class [4] from the similcirc list. Carnap's Version (D1) had allowed it to enter similcirc based on the "imperfect community" of its members; and this had formed the basis for Goodman's complaint. D2 is able (correctly) to reject it from similcirc on account of its added fourth condition.

Note that *each* member of D1's class [4] shares a color with (i.e., is part-similar to) at least one non-member of the class. That is, there is not at least one member which does not form the Ps-relation to any non-class-member. Thus that class does not conform to the requirements of D2.

Example 2:

1. bg	2. rgy	3. brw
4. ry	5. bw	6. gy

similcirc classes constructed by ....

D1:	vs.	D2:
{1,3,5}	[1]	{1,3,5}
{2,3,4}	[2]	
{1,2,6}	[3]	
{2,4,6}	[4]	
{1,2,3}	[5]	

Note on Example 2: This example clearly illustrates the *conservative* nature of construction D2 as compared with D1. In fact, this author contends, D2 will never accept a class as being a valid similcirc which D1 would *rightly* reject. But the price which is paid for this added tightness of construction is that D2 will often reject valid--as well as invalid--classes from similcirc which D1 would have accepted. This is the case in the present example.

Speaking in what Carnap would call the "realistic language", we might say that class [1] in this instance corresponds to "blue" (i.e., its membership consists of all and only objects containing blue), class [2] corresponds to "red", class [3] corresponds to "green", and class [4] corresponds to "yellow"; while only class [5] is improperly formed on the basis of an imperfect community. (That is, there is no quasi-constituent in virtue of which all members of {1,2,3} are actually part-similar.) Nonetheless, all 5 of these classes are affirmed by D1 to form valid similcircs. The alternative construction, D2, has correctly rejected class [5] from similcirc; but, in the process, it has also passed over the legitimate classes [2] to [4], because none of these satisfy its fourth condition.

Example 3: 1. br 2. bg 3. rg 4. y

similcircs constructed by...

D1:	vs.	D2:
{1,2,3}	[1]	{1,2,3}
{4}	[2]	{4}

Note on Example 3: This last example shows that construction D2, though conservative, can nonetheless also generate false similcircs under certain circumstances. Once again, this can be accounted for by an "imperfect community" among the pseudo-class's members. (This could also occur on account of a constant companionship of dissimilar qualities in the same objects--a contingency which Carnap himself acknowledges (70).) Still, it is significant that whenever such instances occur, construction D1 would also generate the *same* improper similcircs. Hence, if one's aim is to *minimize* the construction of false classes (while allowing for the risk that, pending the acquisition of further data, some legitimate classes might be rejected), D2 is preferable. Yet D1 has the reverse advantage of accepting true similcircs more readily--but at the greater risk of falsely accepting improper sets.

Realistic State of Affairs (This is intended to apply regardless of which derivation for similcirc is ultimately accepted): All elementary experiences contain a wide array of qualities. If a quality which appears in one such elex is sufficiently similar to a quality appearing in another, then the two elex's involved can be said to be part-similar. The construction of similcirc-classes is the attempt to identify those quality-clusters with respect to which the various elex's might be deemed to be similar.

Several possible constructions have been offered for these similarity-groupings. Their common intent is to insure that:

a) no elex is excluded from a similcirc if it contains a quality from just that quality-cluster which the derived similcirc is designed to represent; and
b) no elex is included in a similcirc if it fails to contain a quality from just that quality-cluster which the similcirc is designed to represent.

Each of the above constructions must ultimately be judged by how well it can insure that these two intended conditions (in the realistic language) will obtain.

FICTITIOUS OPERATIONS FOR SIMILCIRC CONSTRUCTION

Similarity circles, I have said, are the first constructed objects in Carnap's system beyond the class of elementary experiences. All subsequent objects depend--directly or indirectly--on these for their derivation. Hence the details of this similcirc construction are especially important for testing the vitality and overall viability of the proposed constructional system.

Following Carnap's indications in his text, I have attempted to expand upon his "Simil' Ps" construction of similcirc; that is, I have tried to make explicit, in my "revised notation", the logical specifications which Carnap intends for his constructed similcirc classes. But does the formulation of these *logical rules* for class definition constitute, at the same time, a *procedural basis* for that class's construction? Carnap clearly seems to believe that it does. While he acknowledges the heuristic and didactic advantages of appending the fictitious operations to the formal constructions--to help ascertain "that the (constructions are) purely extensional"--he firmly believes "*that the constructional system itself has nothing to do with these fictions*" (99).

If I am correct, however, in the view expressed in this paper's introduction, than a defensible construction for human cognitive processes and faculties must be able to stand on *both* logical and procedural grounds. Acceptance of this position demands that one be vigilant to insure that both approaches remain always in harmony. By studying below some "fictitious operations" for Carnap's similcirc construction, I hope to show both the importance and the difficulty involved in such vigilance.

CARNAP'S "FIRST PASS": THE ANALOGY OF PROPER ANALYSIS

In effect, Carnap provides a first approach at a procedural representation of similcirc construction in Section 70 of his text. Here he describes the possibility for an analysis, properly so called, of a set of colored objects into classes of those objects which are similarly colored. This could be done, he says, without reference to the objects' intensional properties of being red or green or blue, etc. Instead, provided only one knows which pairs of these objects are "color akin" (i.e., share a common color), one could group them by extensional methods into classes corresponding to their shared colors.

Moreover, he suggests that these same extensional procedures which are used in proper analysis, just described, can be applied as

well for the process of similcirc construction; for the latter process can be viewed as a "quasi-analysis" of elementary experiences into classes corresponding to their quasi-constituent "clusters" of similar qualities. In short, the imagery of proper analysis is employed as a model, or "fictitious" operation, for the procedures of similcirc construction.

So far, then, the imagery Carnap employs in Section 70 appears potentially quite helpful. However, in presenting his model, he makes a naive assumption, which I take to express his general view: He treats the *procedure* (in this case, of proper analysis) as merely involving the successive application--in some undefined manner-- of his formal, *logical rules*. But how are these rules to be employed, and in what order is this succession of applications to occur? These questions are passed over as if of no consequence. That is, having presented the formal defining characteristics of a constructed similarity circle (elaborated in this text in Note 2 for the construction in revised notation)[9], Carnap merely *states* that "the classes formed in this way will be the [similarity circles]". In actuality no "way" (in the sense of a genuine method or procedure) has really been expressed at all. All he does is refer the reader back to the "two [logical] properties" said to characterize a properly formed similarity circle (70).

CARNAP'S "SECOND PASS": A FICTITIOUS OPERATION

Carnap's "first pass" procedure for constructing similcirc appears in the preliminary sections of his *Aufbau*. In Section 111, where the similcirc derivation enters the formal constructional outline, Carnap introduces a more specific Fictitious Operation for its generation. The proposed algorithm, unfortunately, is deficient not only because it yields classes that are subject to Nelson Goodman's objections,[10] but also because the algorithm is rather ambiguous as to the intended mechanism for *terminating* the construction. In short, it is difficult to know exactly what the procedure generates and whether these results are satisfactory.

To conserve space, I will not elaborate this second algorithm in full detail. It is sufficient, I believe, to characterize it as a method for transforming the constructor's initial inventory of part similar elex-pairs into ever larger classes of mutually part similar members. This much insures that all resultant classes meet Carnap's first condition for similcirc construction. By "erasing" from this set of (tentatively) constructed classes all subsumed subclasses, Carnap further appears to satisfy the condition that the resultant classes by "as large as possible".

Yet it is open to question how exactly Carnap's procedure should be terminated. If only the *largest* classes are retained, the probabilities for accepting false classes (such as in Goodman's counter example) turn out to be considerably reduced (though many valid classes may also be rejected in the process). If, on the other hand, one follows the more likely reading of Carnap's text, i.e., that *all* non-"erased" classes should be retained, then Carnap clearly does open himself to Goodman's objections; since his method would readily permit construction of Goodman-type false classes. Either way, it seems that Carnap did not seriously work through his own recommended procedures, nor did he recognize the disparity between (1) the set of classes these procedures generate,

vs. (2) the set of classes which would (independently) happen to meet the structural conditions imposed by his logic.

NELSON GOODMAN'S EXTRAPOLATIONS:

In Nelson Goodman's commentary on the *Aufbau* (in the *Structure of Appearance*) he provides a service to Carnap readers by actually trying to work out some consequences of attempting to generate Carnap-style classes. For instance, he offers a procedural algorithm to accompany Carnap's analogy of proper analysis for similcirc generation. Although Goodman does not acknowledge this explicitly, the algorithm he offers is clearly of his own invention. It differs from both of Carnap's alternatives which I have set out above. (Goodman describes himself as "amplifying" Carnap's exposition, rather than as adding to its substance.[11]) Unfortunately, like Carnap, Goodman offers only an incomplete description of the procedures he envisions.

Following Carnap's image of proper analysis involving colored objects, Goodman gives an example of six such objects, shown below.

1. br	4. g
2. b	5. r
3. bg	6. bgr

TABLE I

The color-kinship pair list for this example's object set would thus consist of the following (symmetrical) pairs:

(1,1)(1,2)(1,3)(1,5)(1,6)
(2,2)(2,3)(2,6)(3,3)(3,4)(3,6)
(4,4)(4,6)(5,5)(5,6)(6,6)

TABLE II

The problem which is posed for the constructional algorithm is this: How can A work from only the pair list of Table II to derive the original object set (of Table I) as output? Below is quoted Goodman's attempted solution. In his view, the method which follows would accord with Carnap's own directives.[12]

> The rules for accomplishing the [analysis] are: (A) Each class must be such that every pair of members of the class is listed in [Table II]. (B) Each class must be such that no thing that is not a member is paired [by that table] with all the members; that is, each class must be a greatest possible class satisfying rule A.
>
> Proceeding to apply these proposed rules, we may commence by listing all the things paired with thing 1 by [Table II, i.e., by color kinship]. These are 1,2,3,5,6. But the class {1 2 3 5 6} breaks rule A since pairs 2:5 and 3:5 are

> not listed in [Table II]. Hence we must drop either 5 alone or both 2 and 3 if we are to have a color class. Dropping 5, we have left the class {1 2 3 6}. This satisfies the two requirements: (A) every pair in it is listed in [Table II]; (B) nothing excluded (i.e., 4 or 5) is paired by the table with every member of the class, since pairs 3:5 and 1:4, for example, are not on the list. The class {1 2 3 6} we therefore label "k_1".
>
> To construct a second class, we may drop from the preliminary class first suggested ({1 2 3 5 6}) numbers 2 and 3 instead of number 5. This leaves us with the class {1 5 6}, which satisfies both rules and which we may call "k_2".
>
> A third color class may be found by listing, say, all the things paired with 3. . . .
>
> Further investigation will reveal no fourth class answering the two requirements. [k_3 is {3 4 6}.]
>
> Now . . . let us glance back at [Table I]. We see that k_1 includes all and only b-things; that k_2 includes all and only r-things; and k_3 all and only g-things. Thus our rules have enabled us to discover these true color classes on the basis of a list which told us merely what pairs comprised two things having *some* unit in common.

One's first impression on inspecting all this detail is that Goodman must have indeed set out a complete procedure. A begins by constructing a preliminary class for object 1; and then refines this by cutting out just so many members as will permit the resultant class to satisfy the rules for similarity circles. Since this can be performed in two distinct ways, there are two distinct circles which result from the analysis of the first preliminary class.

So far, Goodman's method seems quite effective. Yet why does he skip immediately from consideration of preliminary class 1 to the preliminary class corresponding to the object 3? No doubt, he has realized that object 2 belongs to no true color class beyond that which has already been generated as k_1. But A (the constructor) cannot know this prior to completing the derivation. Surely, a proper algorithm must permit a preliminary class to be constructed for *every* object which appears in a color akin pair. If the resultant similarity circles formed are duplications of already constructed classes, the procedures must also be able to recognize this, and to assign a common "tag" to these redundantly generated classes.

More revealing of Goodman's perspective is his almost cavalier suggestion that a third class may be found "by listing, say, all the things paired with 3". His use of the term "say" surely implies that, for him, it is a matter of indifference in what order the preliminary classes are taken. Now the reader may at first object that this is no criticism of Goodman's method. After all, it might be argued, *any* method which calls for testing all possible subsets of the experienced object set against the similarity circle formation rules would *eventually* accept or reject the same classes as validated similcircs. Therefore, *why should it not* be a matter of indifference (as it is for Goodman) in just what order the

testing is performed? The response of this paper is that there are *alternative* procedures in which *not* all possible subsets of the total list of objects would be considered for further testing. In short, it is imperative that one specify exactly just what sorts of preliminary classes will be evaluated, since not all conceivable methods will in fact generate the identical sets of output classes.

This last point reoccurs in regard to Goodman's counter example to Carnap's definition. As detailed previously, Goodman posits a set of six objects to be constructed. Then he asks us to "suppose" that ". . . someone suggests that things 1, 2, 3 constitute a color class".[13] The case is taken to refute Carnap's derivation since the specified set meets all formal conditions for similcirc membership, and yet does so only because of its members' so-called "imperfect community".

But why should we presume that the false class {1 2 3} could be deemed "constructed" simply because "someone suggests" as much? Obviously, if Goodman thinks this is viable, it is because he assumes that, inevitably, all the logical possibilities will be considered anyway. In the algorithm which follows, however, that very premiss is overturned. That is, while only classes which satisfy Carnap's formal conditions are generated by the method, *not all* possible permutations of object lists are considered. In fact, Goodman's set of objects stands as an example--not as a counter example--if the upcoming method is employed.

AN ALTERNATIVE CONSTRUCTIONAL ALGORITHM (WITH A COMPUTER IMPLEMENTATION):

Historically, the algorithm presented here was developed as part of a project to design a computer implementation of Carnap's similcirc construction.[14] Once again, Carnap's analogy of similcirc construction with proper analysis of colored objects is employed. I shall first present the algorithm itself in five steps.

The Algorithm

1. Input of Pair List Data

For both Carnap and Goodman, the input data for similcirc construction consist of a pair list extension. For true similcirc generation the pairs would be those elex's which are related by part similarity. In the analogy with proper analysis, we can speak of the pairs of objects related by sharing some common color (i.e., by "color kinship").

2. Determination of the TCLASSes:

What are here called "tentative color classes" or "TCLASSes" correspond to what Goodman denotes as "preliminary classes" in his own extrapolation (see above). These are the classes which are to be further tested with regard to their possible eligibility as true color classes (or similcircs). Just as Goodman implies (though he does not specify) that there is to be a preliminary class corresponding to each given object, the present method will posit a TCLASS for each object.

The basis for determining the TCLASSes can be formalized, as follows:

1) Given n objects, there should be n TCLASSes formed.

2) Each $TCLASS_i$ consists of the object i together with all other objects which have "kinship" with i.

This part of the algorithm is designed to satisfy Carnap's second criterion for similarity circle membership. Since each object is postulated to belong to every possible TCLASS for which it has kinship with the first member, this ensures that no object could have kinship with every object in a color class and yet fail to belong to that class.

Proof:

1) Suppose the i'th TCLASS happens to be a true color class which corresponds to some commonly held color c.

2) Suppose further that object o, which also contains color c, is color akin to all members of $TCLASS_i$.

3) But if o is akin to *all* $TCLASS_i$ members, then it must of course be akin as well to the first member, object i.

4) By the present procedure, therefore, o's membership in $TCLASS_i$ is insured; because all objects which are akin to object i will be included in $TCLASS_i$.

5) Hence, it has been shown that any object o which has kinship to all the objects in a true color class will be included (as per Carnap's directive) in that class.

3. Determination of Non-Akin Pairs:

The next broad task is to determine which TCLASSes are true color classes. The non-akin pairs just referred to are those couplets of objects which do *not* share color kinship. Accordingly, since all members of a true color class are color akin, these non-akin pairs are couplets which *will not be found* among the members of a true color class. That is, by first determining the set of such non-akin pairs, a basis for rejecting TCLASSes will be made available.

4. Determination of True Color Classes:

If all TCLASSes which contain non-akin pairs are rejected, then those which remain are the true color classes (or similcircs). This preserves Carnap's first condition for similarity circle membership, according to which all pairs of members of a true color class must be color akin.

5. Constructing the Analyzed Object Set:

The model is complete once the set of true color classes is used to construct the corresponding set of objects whose color constituents have been identified.

Justifying the Algorithm

It is clear that, if this algorithm were used in place of Nelson Goodman's, Goodman's counter example would lose its force. The only TCLASSes which would contain all three of objects {1, 2, 3} would be $TCLASS_1$ ({1 2 3 5 6}), $TCLASS_2$ ({2 1 3 4 6}), and $TCLASS_3$ ({3 1 2 4 5}). No class {1 2 3} would ever be formed using my algorithm.

The question remains, however, whether this algorithm can be identified with what Carnap himself intended by similcirc "construction". This author believes that it can be; for Carnap's system represents a dynamic and selective *process* of class generation, rather than a mere analysis of all logically possible sets.

From this perspective, one must look at a detail of Carnap's own presentation which seems to be easily overlooked. In his own words, the second criterion for color class construction is this: The color "classes are the largest possible classes all of whose members are color-akin (i.e., there is no thing outside a color class which stands in the relation of color kinship to all the things in the class" (70). The second, parenthetical part of this definition, it is true, is expressed negatively, and does not provide sufficient grounds for rejecting spurious TCLASSes along the lines of Goodman's counter example. But considered as a whole, the criterion provides a strong sense of how these TCLASSes might be *constructed.* Namely,

1) From the point of view of any object, it should not be left out of any TCLASS to which it could possibly belong.

2) From the point of view of any TCLASS, it should not exclude any object that could possibly belong to it.

The algorithm of this section, it is felt, accords with this clear sense from Carnap's text. Since the emphasis of Carnap's Section 70 is indisputably on the *generation* of the "largest possible classes . . .", there is no obvious necessity to generate and test *every* logically possible combination of objects (such as Goodman's problematic class {1 2 3}) and to subject these to further scrutiny. It is sufficient if the procedures for deriving the "largest possible classes . . ." can insure that *the resulting classes* meet Carnap's logical criteria.

THE COMPUTER PROGRAM

The algorithm described above has been embodied in a computer program, written in the Logo programming language as implemented by Terrapin for the Apple II computer. "Logo" names a high level language that was originally developed for use in an educational context. It is a highly structured language and designed to further the construction and recursive employment of function-like routines called "procedures".

There is, unfortunately, not the space in this article to present the full listing of my program,[15] or to detail all its specific features. Nonetheless, the guiding outline for the program has already been given in terms of the six-point algorithm which I have described.

The only input for the program which has theoretical significance consists of a Carnap-style inventory (called the PAIRLIST) of all known pairs of color-akin objects (or, if applied to quasi-, rather than to proper, analysis, of all parts of Part Similar elex's). In principle, since each object in the PAIRLIST is indicated there by a numerical token, the computer itself could have been programmed to calculate automatically the number of distinct objects to be constructed from the pairing found in the input inventory. Also, a random sequence of arbitrary class names could be computer-generated, if needed, to identify all TCLASSes being constructed by the procedure. However, to simplify the programming task and to conserve system memory, I in fact wrote the program to accept as secondary inputs (1) the number of distinct objects represented in the PAIRLIST, and (2) a corresponding list of TCLASS names (such as "T1", T2", and so on).

After these data have been input, the program executes the five-step algorithm. Part I of the procedure runs until it has printed (1) the initial PAIRLIST data; (2) the exhaustive list of all TCLASSes which can be formed from that data; (3) the list of all "non-akin pairs"[16] of objects; and, finally, (4) the "token" numbers of all objects whose TCLASSes have turned out to be valid color classes (or similcircs) according to the algorithm.

The second stage of the program[17] uses the outputs just described to construct the object set into which the original input data pair list can be "analyzed". First, for convenience, the computer operator is asked to provide appropriate names for the colors (similcircs) which have just been identified. Then the program concludes by connecting each object with the names of all those validated TCLASSes to which that object belongs. That is, in the "realistic language", the program analyzes each object into its constituent colors.

I illustrate below a crucial "run" of the program which has just been outlined. For input, I use just that pair list inventory which corresponds to the data set for Nelson Goodman's counter example to Carnap's similcirc construction. Observe how the program is able *correctly* to analyze the data given, that is, to "construct" from these date precisely--and only--those "objects" which are presupposed by the example case. (Goodman, of course, had said that this could not be achieved in this instance.) Recall that the object set represented by the pair list (i.e., which must be *constructed* on the basis of that list) is as shown:

1. bg	2. rg	3. br
4. r	5. b	6. g

A SIMULATION OF THE ACTUAL PRINTOUT WHICH THE CARNAP PROGRAM GENERATES

THESE PAIR LIST DATA HAVE BEEN INPUT
(READ THE DATA IN COUPLETS)

1 2 1 3 1 5 1 6 2 3 2 4 2 6 3 4 3 5 DO NE

```
THE PROGRAM WILL NOW DETERMINE THE
TENTATIVE COLOUR CLASSES[18]
(TCLASSES), CORRESPONDING TO EACH OBJECT.

PLEASE WAIT...

THE TCLASS FOR OBJECT 1 IS...
1 2 3 5 6

THE TCLASS FOR OBJECT 2 IS...
2 1 3 4 6

THE TCLASS FOR OBJECT 3 IS...
3 1 2 4 5

THE TCLASS FOR OBJECT 4 IS...
4 2 3

THE TCLASS FOR OBJECT 5 IS...
5 1 3

THE TCLASS FOR OBJECT 6 IS...
6 1 2

NOW SEARCHING TO IDENTIFY NON-AKIN PAIRS...

THE FOLLOWING LIST SHOWS THE
NON-AKIN PAIRS OF OBJECTS
(READ IN COUPLETS)

1 4 2 5 3 6 4 1 4 5 4 6 5 2 5 4 5 6 6 3 6 4 6 5 DO NE

NOW SEARCHING TO IDENTIFY WHICH
TCLASSES ARE TRUE COLOUR CLASSES.

OF THE SET OF TCLASSES SHOWN ABOVE
ONLY THE FOLLOWING HAVE BEEN SHOWN
TO REPRESENT GENUINE COLOUR CLASSES:[19]

4 5 6

NEXT, THE PROGRAM TRIES TO DETERMINE
WHICH COLOURS BELONG TO WHICH OBJECTS
(I.E. OF WHAT SETS EACH OBJECT IS A MEMBER...

AS SHOWN ABOVE, TCLASS 4
WAS A TRUE COLOUR CLASS.
ITS ELEMENTS ARE..4 2 3

PLEASE INDICATE A DESIRED ONE-WORD
COLOUR NAME TO CORRESPOND WITH THE
LIST:... RED

AS SHOWN ABOVE, TCLASS 5
WAS A TRUE COLOUR CLASS.
ITS ELEMENTS ARE ... 5 1 3
```

```
PLEASE INDICATE A DESIRED ONE-WORD
COLOUR NAME TO CORRESPOND WITH THE
LIST:...BLUE

AS SHOWN ABOVE, TCLASS 6
WAS A TRUE COLOUR CLASS.
ITS ELEMENTS ARE.. 6 1 2

PLEASE INDICATE A DESIRED ONE-WORD
COLOUR NAME TO CORRESPOND WITH THE
LIST: ... GREEN

THESE ARE THE COLOUR ASSIGNMENTS FOR EACH OBJECT:

OBJECT NUMBER...1
BLUE GREEN

OBJECT NUMBER...2
RED GREEN

OBJECT NUMBER...3
RED BLUE

OBJECT NUMBER...4
RED

OBJECT NUMBER...5
BLUE

OBJECT NUMBER...6
GREEN
```

CONCLUSIONS

The immediate contention of this paper is that the "fictitious operation" just given for similcirc construction (in the form of a programmable algorithm) is more dependable than that which is provided by either Carnap or Nelson Goodman. That is, the above algorithm will more often yield resultant classes which conform both to Carnap's logical criteria *and* to our own intuitions. The program run for Goodman's "problem case" is a strong argument in favor of this claim; since the procedures have *correctly* analyzed the presumed object set.

But it does not follow that a successful algorithm cannot be represented both logically *and* structurally. To the contrary, the "alternative" construction for similcirc which I proposed earlier includes an additional logical requirement for similcirc class membership--beyond the provisos already required by Carnap. This condition (that at least one member of a true color class must have kinship *only* with fellow members of that class) was added upon my *discovery*, while running the above program, that all acceptable classes which the program generates inevitably contain this added structural feature.

Indeed, it can easily be proved that this new logical condition must apply to all classes generated by the computer program. Every confirmed color class (or similcirc) must be--according to the procedural rules--a TCLASS for (at least) one of the objects. Suppose that one of these confirmed color classes is $TCLASS_i$. According to the rules, $TCLASS_i$ must contain *all* other objects which have kinship with the specific object i. That is, there is no thing outside of TCLASSi which has kinship with thing i. Therefore, for that color whose extension is $TCLASS_i$, object i must be that object which satisfies the added proviso in the construction--i.e., the object which has kinship *only* with fellow class members.

In effect, then, I have reversed the sequence of procedure elaboration and class definition used apparently by Carnap and Goodman. These thinkers hold that the logical structure of the desired output classes can be used also as a testing mechanism in the fictitious operation. That is, one generates all possible object sets (in some indifferent order) and merely tests which of these accord with the desired class's formal definition. In the preceding section, however, the emphasis was on making the "fictitious" operations subject to practical testing--via a computer simulation. In this process it was discovered that it was not necessary--indeed, it was desirable *not* to--construct all possible sets of objects, but rather to generate an ordered sequence of sets. Then, once these procedures were in place, I later reconsidered the logical structure of the actual output classes, with a view to discovering in these some unexpected new properties--conditions to be added to a refined descriptive model.

It would be equally mistaken, however, to advance the opposite thesis, that procedural modeling must always precede structural modeling in epistemological inquiry. Rather, I believe that both modeling techniques must go hand in hand, with the real test being whether one can construct a consistent, complementary pair of models--one procedural and one descriptive--which jointly comprise a proposed construction. I contend that Carnap's employment of the "fictitious operation" device represents a first step towards realizing the importance of joint descriptive/procedural modeling. In this paper, I have used Carnap's work essentially as a case study, in order to clarify the relation between these two modes of modeling, as well as to emphasize the importance of taking both seriously. It is hoped that this effort can contribute both to an increased understanding of Carnap's writings and to a greater insight on the potential use of complementary modeling techniques in epistemological (and other) research.

ENDNOTES

[1] Rudolf Carnap, *The Logical Structure of the World* and *Pseudoproblems in Philosophy*, trans. Rolf A. George (Berkeley and Los Angeles: University of California Press, 1969), p. xvi. All subsequent references to this text will be given in the form "(70)"--where the bracketed number indicates the section in the *Aufbau* which is being cited.

[2] Nelson Goodman, "The Significance of *Der logische Aufbau der Welt*", in *The Philosophy of Rudolf Carnap*, ed. Paul A. Schilpp (London: Cambridge University Press, 1963), 545.

[3] *Ibid.*, 556

[4] Rolf George, "Kant's Sensationism", *Synthese* 47 (1981), 242.

[5] Aaron Sloman, *The Computer Revolution in Philosophy: Philosophy, Science, and Models of Mind* (Sussex: The Harvester Press, 1978), 13.

[6] *Ibid.*, 9.

[7] "The concept of functional architecture", writes Pylyshyn [in Zenon W. Pylyshyn, *Computation and Cognition* (Cambridge, Mass.: MIT Press, 1984), 93] ". . . marks the distinction [in computer science] between what corresponds to the program of interest and the various incidental properties grouped under the general rubric 'implementation concerns'". I suggest that Carnap can be interpreted as concentrating on the "algorithmically significant features" of our epistemological "program", while presupposing that "the immutable, functional architecture of the machine" (namely, man's cognitive faculties) are capable of implementing that program. Presumably, a Carnap-style functional architecture would include (1) facilities for continually expanding the inventory of basic experiences related by the basic relation, and (2) all necessary hardware for executing procedures which can construct the logical classes derived by the system.

[8] As suggested by a referee for this paper, the following equivalent construction for Carnap's similcirc can be given in PM notation:

$$\text{Simil} =_{df} \hat{R}\{\hat{\beta}[\beta=\hat{\alpha}((x)(y)[x,y\in\alpha\supset R(x,y) \vee R(y,x)] \;\&\; \sim(\exists z)[z\notin\alpha \;\&\; (x)(x\in\alpha\supset(R(x,z) \vee R(z,x))])\}.$$

[9] See page 562 preceding.

[10] These objections have been discussed in note 3 for the similcirc construction in revised notation (p.562) and in the notes for my alternative construction (p.563).

[11] Nelson Goodman, *The Structure of Appearance*, 3rd Ed., Boston Studies in the Philosophy of Science, vol. 53 (Dordrecht and Boston: D. Reidel, 1977), 115 and fn. 4.

[12] *Ibid.*, 115f.

[13] *Ibid.*, 118.

[14] Parts of this project were funded by a grant from the Social Sciences and Humanities Research Council of Canada to construct a computer model of at least some aspect of Rudolf Carnap's constructional system in the *Aufbau*. The fruits of that research included the computer program discussed below and earlier drafts of the logical investigations leading to the "revised" and "alternative" versions of Carnap's constructions. These results have appeared previously in an unpublished thesis by the author.

[15] Interested readers may contact the author for additional details.

[16] This part of the procedure calculates and then prints each "non-akin pair" in both orders (e.g., "4 6" and "6 4"). Though theoretically redun-

dant, this approach saves program complexity and speeds the upcoming search for instances of such pairs within TCLASSes.

[17] Since the Apple II system available to me had limited memory capacity (64K), it was necessary to run the program in two stages--clearing and reloading program memory (but leaving data intact) between the two runs. Obviously these two "stages" have no theoretical significance with regard to the 5-step algorithm being implemented.

[18] Since the program was developed in Canada, Canadian spelling is retained in this simulated printout.

[19] That is to say, each of the following numbers represents an object whose TCLASS has been shown to be a genuine color class (or similcirc).