On Short-Term Information-Processing in Connectionist Theories

John A. Barnden

Computer Science Department

Indiana University

Bloomington, Indiana 47405

Running Head: Short-Term Connectionist Processing.

TECHNICAL REPORT NO. 152

ON SHORT-TERM INFORMATION-PROCESSING IN CONNECTIONIST THEORIES

BY

JOHN A. BARNDEN

JANUARY, 1934

ABSTRACT

Connectionist theories of the neural basis of cognition have concentrated on long-term memory and certain specialized short-term processing operations. They have not been much concerned with general issues of short-term information-processing. Several varieties of connectionist theory are investigated here. It is pointed out that short-term information-processing must be pervaded by the manipulation of ancillary information structures. For instance, neural assemblies must frequently be marked to indicate that they are temporarily playing some special rôle. Also, lists of assemblies must be rapidly created and updated. It is shown that most varieties of connectionist theory face problems of efficiency in trying to account for the manipulation of ancillary information structures. The considerations lead to a preference for a particular variety of theory. That variety acts as a base from which the author's own theory was derived. (This theory is not described here.)

1 INTRODUCTION

In recent years there has been a growing amount of work on "connectionist" theories of cognition. [Anderson & Hinton (1981), Anderson & Mozer (1981), Anderson et al (1977), Cottrell & Small (1983), Edelman (1978), Feldman (1981a, 1981b, 1982), Feldman & Ballard (1982), Fahlman (1979, 1981), Goddard (1980) {refining the theory of Hebb (1949)}, Hinton (1981a, 1981b), Kohonen (1981), McClelland & Rumelhart (1981), Pollack & Waltz (1982), Rumelhart & McClelland (1982), Small (1982), Wickelgren (1979).] Connectionists have concentrated on showing how their systems can perform certain specialized computational operations (often concerned with retrieval from long-term memory) [e.g. Fahlman & al (1983)], on devising systems for perceptual processing [e.g. Hinton (1981a); Feldman & Ballard (1982)] or on devising systems that are meant to produce human-like behaviour on certain aspects of natural-language processing. [E.g. Cottrell & Small (1983), McClelland & Rumelhart (1981), Pollack & Waltz (1982), Rumelhart & McClelland (1982), Small (1982). See also Gigley (1982) for related work at a less "physiological" level.]. Connectionists often acknowledge that the specialized operations or systems they study will eventually have to be woven into the larger tapestry of general, complex, short-term information processing, involving activities such as inference, planning, and rule execution [Cottrell & Small (1983), Fahlman et al (1983), Hinton (1981b), Kohonen (1981), Rumelhart & Norman (1981)]. Fahlman (1979, 1981) has perhaps been the one to take most detailed note of this fact, by including a centralized, conventional sequential processor. Also, Hinton has noted [1981b, p.183f] the need to account eventually for such things as rule execution, variable instantiation, and quantification in representation structures. As Rumelhart & Norman (1981, pp.2,5) observe, it is a matter of faith that the models in Hinton & Anderson (1981) (i.e. those of Hinton, Kohonen, Anderson, Sejnowski, Feldman, Fahlman and others) can capture complex global goal-directed processing. Feldman (1983) states the current importance of seeking to close the gap between the complex symbolic information-processing models used in artificial intelligence and cognitive-psychology research on the one hand and present neurally-based work, such as connectionism, on the other.

It is therefore of interest to see whether current connectionist theories can be extended to encompass *more general, short-term* information-processing, without being deprived of their essential connectionist nature. This paper shows that the inclusion of such processing leads to some tricky problems for some varieties of connectionism. We are therefore led to a preference for particular varieties, though no variety appears entirely free from problems. The problems are more in the nature of challenges to connectionist researchers than necessarily insurmountable obstacles. It would be dangerous to claim they are insurmountable, especially since what is under attack is not a specific theory but rather a broad paradigm.

Because of space limitations, the reader is assumed to have a basic familiarity with connectionist theories. The sets of neurons that are the basic "atoms of representation" in a connectionist theory will be called *neural assemblies*. Data structures take the form of sets of assemblies joined by *connections*, which are facilitated transmission paths. (The creation of a connection is assumed to be the conferring of a non-zero weight on a transmission path. Deletion consists of placing a zero weight on the path. For simplicity, all connections are taken to be excitatory. This restriction in the discussion does not lead to artificial problems. Facilitation of a transmission path may consist of increased synaptic efficacy, lowered thresholds of neurons acting as relay stations on the path, other special states of such neurons (as in Feldman (1982)), and so on.)

In concentrating on short-term matters I shall ignore long-term issues almost entirely. I do not, for instance, discuss how short-term pieces of information are converted into long-term pieces of information, or how a short-term structure could arise from retrieval processes acting on long-term memory.

The considerations expressed in this paper have led me to propose a class of theories specifically geared towards general manipulations of short-term symbolic information-structures. This class of theories will be described elsewhere. <<Note to referees: it is sketched in the optional appendix included with this paper.>> Early versions of the ideas are presented in Barnden (1982a, 1982b, 1983). The theory class can be viewed as a development from a certain type of connectionist theory.

2 PLAN AND PREAMBLE

We shall assume throughout that short-term information structures are implemented in the brain as active connectionist structures. These are networks of neural assemblies joined by connections, where each neural assembly has a level of activity different from its "resting" activity. As a simple example, the proposition that John loves Mary might be implemented, in some connectionist theories, as the active connectionist structure shown in Fig.1(a).[1] In this illustration the network is very similar to a semantic network [Findler (1979)] that might be used to represent the proposition (see Fig.1(b)), but in other connectionist theories the correspondence is not so straightforward.

Fig. 1

We assume that (much) short-term information-processing consists of a sequence of manipulations of active connectionist structures. Each manipulation is the execution of some *processing rule* whose triggering condition is satisfied by the current set of active connectionist structures. (The paradigm being appealed to here is of course that of "production systems" [Davis & King (1977), Waterman & Hayes-Roth (1978)].) This sequence-of-manipulations view of cognition is not antithetical to connectionism as normally portrayed – it is simply that most reports of connectionist work only discuss operations that could form parts of individual rule-executions. Thus, for instance, the relaxation process by which the system of Cottrell & Small (1983) recognizes word-meanings in an English sentence could be viewed as a part or the whole of a rule execution. Similarly, an associative retrieval operation could be an operation in a rule execution. Many comments in the connectionist literature recognize that eventually the operations discussed will have to be seen as part of a larger computation, and the usual implication is that the computation would consist of a manipulation sequence of some sort.[2]

Consider a rule-execution sub-sequence that effects a complex computation such as the understanding of a spoken English sentence of ordinary complexity, under normal conversational circumstances. Presumably the sub-sequence contains many rule-executions. Therefore, the processes of triggering and executing rules must be fast. The main points to be made in this paper are that (a) rule execution and triggering is pervaded by the *necessarily-rapid manipulation of very short-lived ancillary data structures,* and that (b) it is *not easy to see how a connectionist theory could provide sufficiently rapid manipulation* of these structures (without a certain sacrifice of elegance). We shall see that some types of connectionist theory can plausibly provide enough speed but are implausible in the amount of mechanism they demand (and in other ways), while other types require an amount of *connection-probing* that is probably excessive. (Connection-probing will be defined below: roughly, it is an attempt to determine what connections impinge upon a given neural assembly.)

I shall categorize connectionist theories using four orthogonal dichotomies: the *separable-assembly* versus *inseparable-assembly* dichotomy; the *permanent-reservation* versus *recruitment* dichotomy; the *visible-association* versus *invisible-association* dichotomy; and the *local-state* versus *non-local-state* dichotomy. Fortunately, we shall not need to pay explicit attention to the differences between all sixteen induced theory types.

Before proceeding we must introduce some terminology. Consider the problem of representing, in a connectionist scheme, a group of facts or hypotheses such as that John loves Mary, Mary loves Bill, and so on. Assume that for each person p_i there is a neural assembly P_i representing that person, and that there is a neural assembly L representing the relationship "love". It is clearly not enough to connect each P_i to L, because the relationships would then become confused. What we need[3] are "instance satellites" – neural assemblies that stand for particular instances of relationships. Thus, the statement that John loves Mary would be represented by an instance satellite S_1 for "love", suitably connected to the neural assemblies for John and Mary; and the statement that Mary loves Bill would be implemented by a different instance satellite S_2 for "love". Presumably all instance satellites for a relationship are connected in some way to the neural assembly for the relationship itself (e.g. L above).

3 VARIETIES OF (EXTENDED) CONNECTIONIST THEORY

Here we define the four dichotomies mentioned above and make some preliminary observations.

A separable-assembly connectionist theory is one in which in any neuron assembly N there is a subset N′ of neurons that are not members of any other neuron assembly. Therefore, each neuron in N′ determines N and can be viewed as a "representative" of N. An inseparable-assembly connectionist theory is one in which that condition is not fulfilled – and we assume in fact that for *most* neuron assemblies N there is no subset N′ of neurons that do not appear in any other neuron assembly. Theories such as that of Hinton (1981b) are inseparable-assembly theories (and are usually said to be "distributed" theories). Theories such as those of Fahlman (1979, 1981), Feldman (1981b, 1982), Feldman & Ballard

(1982) and Wickelgren (1979) appear to be separable-assembly theories. In fact it is often the case in such theories that N′ is the whole of N. We say a theory is *disjoint-assembly* if N′ is always the whole of N. Most of the time I shall be concerned with the distinction between disjoint-assembly and non-disjoint-assembly theories, rather than that between separable-assembly and inseparable-set theories.

We turn now to the distinction between theories that have *"permanently reserved"* instance satellites and theories that have *"recruited"* instance satellites. We shall see that the former are impractical, and concentrate on the latter in the rest of the paper.

3.1 Permanent-Reservation Methods

In a permanent-reservation theory, neural assemblies are permanently reserved for particular propositions. There would be an instance satellite LSAT (for "love") permanently reserved for the task of representing the proposition P that John loves Mary. Similarly, there would be an instance satellite BSAT permanently reserved for the task of representing the proposition B that Bill believes that John loves Mary. We do not specify here the methods that might be used to actually indicate that, say, LSAT is currently part of either short-term or long-term memory. All we say is that LSAT is available for representing the proposition P, and cannot be used for any other proposition.

The huge number of permanently-reserved instance satellites needed makes permanent-reservation theories implausible. Let us count roughly how many propositions there are of certain types at certain levels of complexity. Let us confine attention to descriptions and propositions of the following forms. The descriptions are single names or are of the form "the person who loves D" or of the form "the person who believes that P", where D is a description and P is a proposition. The propositions are of the form "D loves D′ " and "D believes that P". (Thus, a possible proposition is "the person who loves Mike believes that John loves the person who believes that Jim loves Mary". Note that the replacement of a description D occurring within a proposition or description X by another description D′ with the same referent as D is taken to yield a proposition or description X′ different from X. We are concerned here with information *structures,* not with their meanings.) We can then ascribe complexity levels to the descriptions and propositions. A name is a level-0 description, and there are no level-0 propositions. A description of form "the person who loves D" or "the person who believes that P" is at a level one greater than the level of D or P respectively. A proposition of the form "D loves D′ " or "D believes that P" is at a level one greater than the maximum of the levels of D, D′ (for the first form) or of the levels of D, P (for the second form). Of course, there is presumably some limit on the complexity level of propositions and descriptions implementable in the brain.

Let the number of distinct names be N. Then it can easily be shown that the number of propositions at level L (>0) is greater than $N^{F(L+2)}$ where F(k) is the kth. number in the Fibonacci series 1, 1, 2, 3, 5, 8, 13, 21, 34, ... (each number being the sum of the two previous ones). Thus, the number of propositions at level 3 (e.g. "John loves the person who believes that Jim loves Mary") is greater than N^5, and the number of propositions at level 4 is greater than N^8. Even if we take N to be as low as 200, this means that the number of propositions at level 3 is greater than 32.10^{10}, which is greater than the number of neurons in cortex (that number being about 5.10^{10}, according to Mountcastle (1978)). Therefore, there is a dramatic combinatorial explosion of propositions as levels are ascended, and a permanent-reservation theory is totally implausible unless only propositions (and descriptions) at the very lowest few levels of complexity are allowed. Moreover, we have the rather curious result that far more instance satellites are reserved at the higher levels than are at the lower levels, although presumably the lower level propositions are more common.

Nothing in the foregoing argument depends on any peculiar properties of belief: belief was merely taken as a convenient way of illustrating levels of proposition that result from nesting. We get exactly similar phenomena if we nest by means of logical connectives. Indeed, propositions at the higher levels of this hierarchy will surely be more common than high-level propositions resulting from iterated belief. (We assume that a proposition to the effect that X implies Y, say, is implemented by an instance satellite of an "imply" neural assembly. The treatment of connectives as analogous to relationships is based on a respectable tradition, in artificial intelligence research at least [see papers in Findler (1979)].

An inseparable-assembly theory has a greater chance of coping with the mentioned combinatorial explosion than a separable-assembly theory has, since the former does not imply that there are at least as many neurons as assemblies (whereas the latter does). Nevertheless, it would be incumbent on a inseparable-assembly, permanent-reservation connectionist theory to show that it could provide enough neural assemblies to cope with a respectable number of complexity levels.

3.2 Recruitment Theories

From now on we ignore permanent-reservation theories and consider only recruitment theories. A recruitment theory assumes a pool of "free" neural assemblies that can be recruited for the purpose of representing propositions. So, if a short-term structure for the proposition that John loves Mary has to be set up, an instance satellite LSAT is grabbed from the pool. (We are assuming here that the proposition does not already exist in long-term memory. If it does, it may well be that some neural assembly is permanently reserved as an instance satellite for the proposition, and that that neural assembly is also used when the proposition is accessed in short-term information processing.) The way LSAT might be deployed in the representation of the proposition differs according to whether a *"visible-association"* recruitment method or an *"invisible-association"* recruitment method is used. As the paper unfolds we shall argue that both visible-association recruitment methods and invisible-association recruitment methods are problematical.

As in the case of permanent-reservation theories, I ignore the problem of distinguishing between free neural assemblies, neural assemblies in use as instance satellites in short-term structures, neural assemblies in use as instance satellites in long-term structures, and neural assemblies in use as instance satellites in both.

3.2.1 Visible-Association Recruitment Theories

In a visible-association recruitment theory, the presence of a short-term association between two neuron assemblies is signalled by a special neuron assembly, called an "association-indicator", being active. There is a separate association-indicator for each possible short-term association. An association-indicator is connected permanently to two other neural assemblies, and when the association-indicator is active we are to regard those two neural assemblies as being temporarily associated. Consider the proposition P that John loves Mary. P would be implemented by the active connectionist structure shown in Fig. 2. JohnLSAT, MaryLSAT and LoveLSAT are association-indicators which are activated as well as J (the neuron assembly for John), M (the neuron assembly for Mary) and LSAT (a recruited neuron assembly).

Fig. 2

An association-indicator together with its two special connections is called a "connection-duple". It may or may not be the case that an active connection-duple can be regarded as an (indirect) temporary connection. That is, it may or may not be the case that the fact that the association-indicator is active allows activity to flow between the two associated neuron assemblies. That is why the term "association" has been used rather than "(indirect) connection".

It is important to understand the benefits and limitations of association-indicators. If a processing mechanism needs to "know" whether *two given* neural assemblies are to be taken to be associated, then the presence or absence of activity at the association-indicator on a connection-duple joining the neural assemblies is enough in theory to provide the answer. However, if a processing mechanism wishes to find one or several neuron assemblies associated to a *single* given[4] neural assembly, where the mechanism has no advance knowledge of which neural assemblies they might be, then association-indicators do not provide much help. The mechanism presumably has to use a probing process similar to that which will be postulated below in the case of invisible-association theories: we postpone the issue of probing until then.

Feldman (1981a, 1982) mentions association-indicators in passing, though he does not call them that. (The active connection-duples in his case do act as indirect, temporary connections.) The neural assemblies associated by his duples are "long-term" ones, in that they permanently represent something. He points out that the number of association-indicators required is implausibly large if it is to be possible to associate any pair of normal neural assemblies. However, a visible-association connectionist theory does not necessarily propose that any pair of long-term neural assemblies must be able to be temporarily associated. It may only be necessary to temporarily associate pairs where at least one of the assemblies is recruited (such as LSAT in Fig. 2). If then the pool of recruitable neural assemblies is not very large, the number of association-indicators needed is not so troublesome.

In order to get round the problem of the number of association-indicators, Feldman proposes more complex temporary connections. The creation of a temporary connection now consists of the stimulation of several intermediate neural assemblies on a permanent connection path between the assemblies to be associated. A given intermediate assembly is shared amongst many such connection paths, so that the presence of activity in any one intermediate assembly does not determine a unique temporary connection.

Nevertheless, the presence of activity in all the intermediate assemblies between two normal assemblies does determine a unique temporary connection. Hence, that whole set of intermediate assemblies can be regarded as an association-indicator (and we have connection-tuples, not connection-duples). For simplicity, I shall ignore this more complex variant of the idea; but the discussion can easily be modified to encompass it.

3.2.2 Invisible-Association Recruitment Theories

In an invisible-association theory, the creation of temporary structure consists of the creation or deletion of some connections, and there are no special neural assemblies signalling the presence of connections. So, for instance, to represent the heard sentence "John loves Mary" the system might recruit a "love" instance satellite LSAT and create connections between it and the neural assemblies for John and Mary; the creation might take the form of the facilitation of synapses or the reduction of neuron thresholds on transmissin paths. See Fig.1(a). (It is possible to envisage connection-*deletion* methods, which inhibit (temporarily delete) unwanted connections. So, LSAT would be already connected to many neural assemblies that (could) represent people, and all but two of these would be deleted. From now on, however, we consider only connection-creation methods. The discussion can be modified to encompass connection-deletion methods.)

Since connection creation in an invisible-association theory may be a matter of synapse facilitation, a potential problem is that facilitation may just not be fast enough for the purposes of short-term structure creation. Most claims about connection creation in the connectionist literature concern the acquisition of long-term connections over some learning period much longer than the time scales we are interested in. Also, Feldman (1981a, p.18) agrees with the common assumption that synaptic weights cannot change quickly enough for the purposes of rapid, short-term connection creation and deletion. However, it does appear conceivable that synapse modification could happen at the necessary speed (within a very small fraction of a second [Routtenberg (1982)]). Therefore the creation-speed problem would be a difficult one to argue at present. The next observation about connection-creation theories is that a short-term term information structure will often have to decay or be deleted, so that there must be two types of connection: short-term term and long-term. Although extra complication is hereby introduced, transmission paths may well be able to entertain several different types of facilitation [Matthies (1982)]. Goddard (1980) presents a version of Hebb's (1949) theory that uses different types of facilitation.

A processing mechanism M that works on the short-term information structures must have some way of knowing or finding out which neural assemblies are connected to which in that information structure. In an invisible-association theory it is obviously not enough for M to detect the activity in neural assemblies. Presumably, M must detect connections by *probing* them. The activity at one end of the connection must be changed in some way, and then the change in activity of other neural assemblies noted. Problems arise when we look at how probing might actually be done. Suppose it must be determined whether there is a connection from neural assembly N to neural assembly N′ . An initial suggestion would be to make N active and then observe whether N′ becomes active. The trouble with this is that it seems to be the case in connectionist theories that at any given moment all the neural assemblies in the data structure currently being attended to are active, so that N and N′ would already be active. The idea of probing by momentarily switching *off* the activity of N is also problematical, because the presence of other active neural assemblies connected to N′ is likely to preserve the activity of N′ . A more workable idea is to temporarily give N a higher activity level than any neural assembly normally has, and then to wait for an increase in the activity level of N′ (or, equivalently, decreasing the activity level of all the other active neural assemblies: however, the need to restore their activity a moment later introduces further complication into the story); but in all of these probing methods it is not clear how to avoid interference from paths of connections from N to N′ via other neural assemblies N′ ′ .

It was noted that a visible-association theory has a need for probing (of connection-duples). This probing would proceed much on the lines indicated for invisible-association theories, and faces the same sorts of difficulty.

3.3 Local-State and Mark-Passing Connectionist Theories

We come now to the fourth and final dichotomy. Some connectionist theories [see e.g. Feldman (1981b), Feldman & Ballard (1982), Fahlman (1979, 1981)] allow neural assemblies to be in one of a set of discrete, local states. We say these theories are "local-state" theories. The way the states are used makes them symbolically significant. I consider such theories to depart from ordinary connectionist principles, whereby the only "state" that a neural assembly has is a level of activity, which is interpreted as indicating the degree of attention being devoted to that assembly or the degree of confidence that the assembly is appropriate in some sense. Once discrete, local states are allowed in, the door is open to suggesting something more general and sweeping: namely that neural assemblies are somewhat like computer-memory locations, having at any moment a state as rich in significance as a bit-string. This would be entirely against connectionist principles. Yet it seems ad hoc to restrict the rôle of states in the way they are in fact restricted in connectionist theories that use them.

Some theories [e.g. Fahlman (1979, 1981)] allow discrete marks, rather than just levels of activity, to be transmitted along connections. I regard such theories also as departing from pure connectionism, for somewhat similar reasons. For example, it appears to be a central tenet of connectionism that neural loci in the brain do not transmit symbols to each other. But transmitted marks are a type of symbol.

I shall not presume to banish discrete-state and mark-passing theories from consideration. Indeed, I shall finally conclude that certain varieties of local-state connectionist theories are to be preferred over connectionist theories without local states.

3.4 An Orthogonality

There is a danger of confusing the separable-assembly/inseparable-assembly distinction and the permanent-reservation/recruitment distinction, which are orthogonal. *A separable-assembly theory dedicates some neurons to particular neural assemblies* (so every neural assembly contains some neurons that do not appear in any other neural assembly, and are thus "representatives" of that set), whereas *a permanent-reservation theory dedicates neural assemblies to particular tasks* (such as being an instance satellite for a particular relationship). Thus, a separable-assembly, recruitment theory postulates a pool of free neural assemblies, where in each neural assembly there is a subset of representative neurons. On the other hand, in an inseparable-assembly, permanent-reservation theory each neural assembly has a fixed representational function, but generally do not contain representative neurons.

4 CONNECTION DELETION IN INVISIBLE-ASSOCIATION THEORIES

A particular problem arises in *non-disjoint-assembly* invisible-association connectionist theories. In the absence of strong reasons to the contrary, it is fair to assume that short-term information-processing in an invisible-association theory often requires *the deletion of individual connections in short-term structures* . (As pointed out below, it is conceivable that all or most connection-deletion could be avoided, but the cost involved is high.)

Suppose a connection from neural assembly N to neural assembly N′ must be deleted. Because we are considering a non-disjoint-assembly theory, N in general shares neurons with some other neural assemblies N′ and M shares neurons with other neural assemblies M′ . Therefore, the degree of facilitation on fibre paths from neurons in N to neurons in M reflects not only the connection from N to M that is to be deleted, but also connections from N′ assemblies to M′ assemblies. If the N-M connection were to be deleted then facilitation on N-M paths would be reduced, but there is no way of knowing the size of the required reduction without somehow analysing the other connections that use those paths. Any process, if there is one, to manage this would be cumbersome and would run the risk of being too time-consuming, especially if the connection-deletion is meant to be a small part of some larger modification.

It is possible in principle to avoid all or most connection-deletion by refusing ever to change a network of connections, but rather to "copy it with modifications". That is, a copy of the network is made, except that the portions to be deleted are skipped over (and new portions may be added). By this means we could perhaps ensure that connections are only ever created and never deleted (though we must presume that old networks that are no longer of interest eventually disappear by virtue of their connections disappearing by some decay-of-facilitation process). Clearly, however, the reconstruction involved in the copy-with-modifications technique makes it undesirable as a way of avoiding the problems of connection-deletion.

5 PROCESSING-LOCUS IDENTIFICATION

Suppose the following inference rule must be implemented in a connectionist scheme: "if a man m loves a woman w who loves a man m′ (other than m), then m hates m′ ". What we want to happen here is that if connectionist structures representing a particular example of the antecedent of the rule are recognized, then a connectionist structure implementing the consequent is set up. The consequent structure must involve, in the right way, the neural assemblies M, M′ representing the particular men m and m′ . Thus, the act of recognition of an example of the antecedent must communicate the right "processing loci" – the identities of those neural assemblies – to the mechanism that sets up the consequent structure.[5] We shall point out various broad possibilities for how this could be done, splitting the discussion between visible-association and invisible-association theories. We shall confine ourselves to a very simple example of a processing rule. This rule should respond to any proposition that some particular man is hungry by outputting the word "hungry" and then outputting the man's name. We assume there is a set of neural assemblies permanently representing some particular people. By outputting the name of person x we mean stimulating a neural assembly $NAME_x$ that represents the name. We assume the existence of a connection from the neural assembly X representing the person to $NAME_x$. We assume that there is a neural assembly HUNGRY representing the property of being hungry. By outputting the word "hungry" we mean stimulating a neural assembly $WORD_{HUNGRY}$.

The main conclusion of the discussion will be that, in order for a connectionist theory to account for locus identification in a way that is practical and in tune with the spirit of connectionism, there is a need for the *marking* of neural assemblies and, paradoxically, for sequential traversal of connectionist structures. Marking is viewed as a simple sort of ancillary data structuring.

5.1 Locus-Identification in the Visible-Association Case.

Fig. 3 shows some connection structure needed for the representation of some propositions of hungriness. We assume for simplicity that at any time only one proposition about a man being hungry is active, though there may be some active propositions about women being hungry. We shall look at some ways of implementing the desired processing rule.

Fig. 3

(I) The first method of implementing the rule is illustrated in Fig. 4. We have separate pieces $RULE_M$ of neural mechanism for each man-neuron-set M. For each neural assembly H that is recruitable as an instance satellite for "hungry", $RULE_M$ receives both an input connection $c_{M,H,HUNGRY}$ from the association-indicator between H and HUNGRY and an input connection $c_{M,H,M}$ from the association-indicator between H and M. We call H the source of the input-connection pair. $RULE_M$ has an output connection to $NAME_m$ (for the man m represented by M) and to $WORD_{hungry}$. When for some H both $c_{M,H,HUNGRY}$ and $c_{M,H,M}$ transmit activity to $RULE_M$ (i.e. the relevant association-indicators are active) then $RULE_M$ transmits activity along its output connections.

Fig. 4

The method solves the locus-identification problem by encoding processing loci as the very identities of the $RULE_M$ mechanisms. The action of the rule is simple and fast. Clearly, however, the price paid is great: the number of distinct $RULE_M$ mechanisms is the number of man-neuron-sets. (Note also that other rules might have to deal with neural assemblies representing complex descriptions of entities, not just with neural assemblies that represent entities "directly" and permanently.) In the simple example chosen, the mechanism in $RULE_M$ is fairly trivial since the rule has a trivial task to do, but of course for more complex tasks the replication of that mechanism for each man would be intolerable. The amount of replication itself becomes much greater when we turn to more complex trigger conditions, containing more variables like M. Also, it is difficult to see how the mechanism could plausibly have been created by a learning process. Further, in the simple example chosen the triggering proposition is not affected by the rule. Suppose instead the rule had to, say, delete the proposition by eliminating the activity in the active H (and attendant association-indicators). We then might have each input-connection pair to $RULE_M$ gating an output connection to the H that is the "source" of that input pair. Such measures merely make the method more cumbersome.

(II) Let us now reduce the amount-of-mechanism problem encountered in (I) by having just one piece of mechanism, RULE, which has an input-connection pair as in (I) for every H and every M. The result is illustrated in Fig. 5. RULE is not, of course, associated with any particular M, so must presumably have an output connection to each $NAME_M$ and have the input-connection pairs gating these connections. The

Fig. 5

action of the rule is still simple and fast, but now the computation in the rule's action part is embodied only once. However, we still have the problem of how the input-connections could have arisen in a learning process. Also, even if the necessary number of connections in our simple example is tolerable, it is unclear that it remains so when we try to extend the method to more complex trigger conditions. For instance, consider a rule that responds to the presence of any proposition of the form "woman w believes that man m is hungry" by outputing the name of the woman. Such a proposition would be represented by an active connectionist structure like that illustrated in Fig. 6. Instead of input-connection pairs we have input-connection pentuples, as there are five association-indicators needed to identify each one of the relevant propositions. The number of pentuples needed is the product of: the number #W of woman neural assemblies W, the number #M of man neural assemblies M, and the square of the number #I of neural assemblies recruitable as instance satellites.[6] If we take #W and #M to be 500 and #I to be only 100 or so, then we are talking about 250 million pentuples, for just one rule. Fig. 6

(III) The third method seeks to reduce the amount of circuitry drastically by having a mechanism RULE that has an input connection $c_{H,HUNGRY}$ from the association-indicator mediating between HUNGRY and each neural assembly H recruitable as an instance satellite. RULE also has an input connection c_M from each man neural assembly M. See Fig. 7. Clearly, both the triggering and the action of RULE must be more elaborate. In fact, we see that both must depend on *probing* operations. In the triggering case, it is not just that the particular hungry man is not yet known, but also that it is not even yet known that a *man* is hungry. When RULE receives activity on $c_{H,HUNGRY}$, a probe operation applied to H (or to the association-indicator between H and HUNGRY) must be initiated. $c_{H,HUNGRY}$ must presumably perform a gating function to get the probe to go to the right place (say H). Also, measures must be taken to ensure that the probe does not simply detect the connection-duple to HUNGRY. Ignoring this problem, and assuming that the only active hungriness proposition is that man m is hungry, let us suppose that the probe is successful in somehow singling out the correct man neural assembly M: let us say by putting it into an abnormally high state of activity. Let us now say that this in turn has the effect of sending stronger activity down the c_M connection to RULE. RULE then "knows" both that it is indeed a man who is hungry and the identity of the man; the c_M connection can gate the output connection to $NAME_M$.

This account has been vague about the nature and control of probing. We have pointed out that probing in connectionist theories is intrinsically problematical. Quite apart from these intrinsic problems, however, the present method has the following drawbacks. First, it departs from the appealing "associative retrieval" flavour of methods (I) and (II), and is thus less in the spirit of connectionism as usually conceived; in particular, we are making heavy use of a sequential style of computation (which amounts to a *traversal* of a structure). Second, we have possible interference from active propositions about *women* being hungry. The probing operations must be able to try again if the wrong H is tried first and found to be associated with a woman, and must *also* be able to avoid looking at the very same piece of structure on the second try. Third, in a more complex rule there could well be *several* neural assemblies whose identities must become known to RULE, not just the one M in our simple example. In the above account, we assumed that M was "marked" by virtue of having an abnormally high activity level, for the sake of argument. But, if several identities have to be communicated to RULE, then in general several different neural assemblies will have to be "marked" *in different ways.*

It might be objected that the detection of the right M could be done without any need for traversal, by a process that engages in massively-parallel transmission of distinguishable marks between neural assemblies, on the lines of the proposal of Fahlman (1979, 1981). This observation is correct, but ignores the point that the passing of distinguishable markers departs somewhat from pure connectionism, as noted in section 3.3. In any case, one of the main conclusions we need to draw from the present discussion is precisely that marking is a necessity in connectionist information-processing.

A final observation about the present method is that we have almost entirely ignored the association-indicators (except for those mediating between the Hs and HUNGRY). We were reduced to probing to find associations, whereas the putative task of association-indicators is precisely that of signalling the presence of associations. Recall the limitations of association-indicators that we observed in section 3.2.1.

(IV) A variant of (III) is to have RULE having just one input connection: a connection from HUNGRY. This method has consequences very similar to those of (III).

These four methods are not meant to exhaust all possibilities, but it seems that other methods will be essentially similar to them. The main conclusions are the following.

(a) The only method that has a reasonable chance of being practical is one (namely (III)) which appears to make little use of association-indicators. This would suggest that we plump for invisible-association schemes anyway; however, we shall have to look again at this suggestion in section 6.

(b) That method, (III), involves the marking of neural assemblies, where there must be several distinguishable ways of marking, not just one.

(c) Some of the need for marking comes from a need for traversal involved in a search for processing loci (unless, possibly, a Fahlman-like mark-transmission scheme is used).

(d) Some of the need for marking comes directly from the need to specify and distinguish processing loci.

(e) Methods like (I) and (II) allow the processes necessary for testing the condition parts of different rules to proceed in parallel, but rules (III) and (IV) probably force rules to be tested one at a time (except when some rules are so different in their domain of discourse that there is no possibility of interference).

As regards (d), it might be objected that instead of marking a neural assembly we could instead transmit to RULE a pulse train that uniquely identifies that neural assembly (much as a bit-string in a computer can be used to identify a memory location uniquely). Such a possibility is interesting, implausible, and completely against the spirit of connectionism.

5.2 Locus-Identification in the Invisible-Association Case.

There are only two basic methods, which we shall call (III′) and (IV′), for implementing the simple rule we have been looking at. The methods correspond to methods (III) and (IV) in the visible-association case. In (III′), RULE just has an input connection from every neural assembly recruitable as an instance satellite, an input connection from HUNGRY, and an input connection from the neural assembly M for each man m. In (IV′), RULE simply has an input connection from HUNGRY. In both (III′) and (IV′), the rule cannot fire unless the HUNGRY input is active. Also, triggering of the rule requires probing much as in (III) and (IV). However, in (III′) extra probing is needed in order to determine that the instance satellite sending input to rule is actually connected to HUNGRY. The issue of marking is exactly as portrayed for the visible-association case.

Variants of the methods can be suggested. For instance, RULE might have an input connection from the neural assembly representing the class of men, where this input must be transmitting activity for RULE to respond. A variant of (III′) arises when some neural assemblies are set aside to be recruitable only as "hungry" instance satellites; then, there is no need for a check to see whether the instance satellite sending input to RULE is connected to HUNGRY.

5.3 Locus Identification: Summary

Locus identification requires a traversal in the form of a sequence of probes (unless a Fahlman-like scheme can be used), and requires the marking of parts of a structure (whether or not a Fahlman-like scheme is used). Marks must be introduced during rule satisfaction, and must be detected and (usually) deleted during rule execution. Further, the possibility of interference between the probing and marking of different structures often forces rules to be examined one by one.

The association-indicators of a visible-association theory provide little help, it seems. However, we shall see in section 6 that they could be of great use in facilitating traversal.

6 TRAVERSAL OF CONNECTIONIST STRUCTURES

We saw in section 5 that connectionist structures may need to be traversed. This is despite the existence of the associative matching operations discussed in the connectionist literature [e.g. Fahlman (1979, 1981), Fahlman et al (1983), Hinton (1981), Kohonen (1981)]. Now, the associative-matching operations obviate the need for traversal during searches for particular, given connectionist structures: so we might have expected traversal to be unnecessary in rule triggering. But the triggering methods that are most like the associative match processes (methods I and II in the visible-association case) are impractical. The reason is that we are looking not for a particular connectionist structure but rather for a particular *class* of connectionist structures.

Associative match operations also do not afford a mechanism for "blind" traversal. Blind traversal does not seek particular predefined substructures, but instead examines a connectionist structure systematically in order to apply some particular operation to the substructures encountered, whatever they are. For instance, if information held in network form has to be converted into some linear form (such as a spoken sentence or a series of actions) then some appropriately organized blind traversal must presumably take place. Similarly, transformation of one network N into another network M (e.g., during sentence production, a transformation from an active-voice network representation, N, of the sentence to a passive-voice network representation, M) presumably involves an organized blind traversal of N. It is hard to see how a "one-shot", non-sequential operation, or a non-systematic sequential examination, could do the required transformation to all possible relevant networks N.

Blind traversal is also necessitated by the *copying* of connectionist structures from one region of the neural mass to another, at least in the invisible-association case. (This is shown in Barnden (1983).) It could be that the different regions are able to perform different sorts of manipulation on the structures. Copying might be used in the interests of redundancy and reliability.

In conclusion, rapid traversal of connectionist structures must be accounted for by a connectionist theory extended to deal with manipulation of short-term data structures. The next step is to note that, in general, traversal necessitates the manipulation of temporary ancillary data structures that serve to record information needed by backtracking mechanisms and to "mark" nodes as having been visited. For the purposes of backtracking there must be a temporary, varying list of neural assemblies; this list corresponds to the lists used in traversal of graph structures implemented in computers [see e.g. Knuth (1973), Standish (1980)].[7] Figs. 8(a,b) illustrate the abstract nature of such lists and suggests one form they might take in an invisible-association theory. Such lists are an important consideration for us, but for brevity we shall look in detail only at the implications of neural-assembly marking, while bearing in mind that this is but a *simple special case* of the maintenance of ancillary data structures.

~ Fig. 8 (a,b)

It would be possible to dispense with the special neural assemblies in Fig. 8(b) by creating special connections between the "normal" neural assemblies, but this technique would require far more transmission paths (assuming that the pool from which the extra neural assemblies are taken is considerably smaller than the number of normal neural assemblies.) In an invisible-association theory, another way of avoiding both the need for extra neural assemblies and the need for extra connections would be to *mark connections* during traversal. See Fig. 8(c). A connection is marked in some way when it is followed in a traversal step; when a backtracking step is performed, the marked connection from the first neural assembly F on the backtrack list is found, the mark is erased, and the assembly at the other end of the connection from F becomes the new first assembly on the list. The identification of the first assembly on the list could be done by marking as in Fig. 8(c) or by pointing as in Fig. 8(b). However, as will be noted in section 7, it is not at all clear how connections are to be marked, especially as we do not want to introduce extra assemblies or connections.

~ Fig. 8 (c)

Traversal presents essentially the same difficulties in a visible-association theory as it does in an invisible-association theory, with one very important reservation. The analogy of connection-marking is the marking of connection-duples. But this can be done by marking the association-indicators (though it does not have to be). Hence, the implementation of backtrack lists in a visible-association theory can easily be reduced to neural-assembly marking. It is at this point that we go back to the suggestion in section 5.3 that from the point of view of locus-identification we might as well adopt an invisible-association theory. We now see that a visible-association theory may allow the traversal required by rule-triggering to be more efficient than it could be in an invisible-association theory, because less machinery is needed to implement backtrack lists.

It is not often pointed out that we can abstractly view the backtrack list at any moment as being a set of marks on nodes. For instance, when the list is (d,c,b,a) in the example, we can view d as being marked with the number 4, c with number 3, b with 2 and a with 1. A node is added to the list by being marked with the lowest-valued unused integer. The node at the front of the list is removed from it by being unmarked, and having its mark deemed to be unused now. (It is also necessary to keep a record of what the lowest unused mark is, or of what the highest used one is.) Thus backtrack-list manipulation can in principle be reduced to manipulation of neural-assembly marks, even in an invisible-association theory. This view turns out to be unhelpful, as we shall see when we consider marking methods.

7 MARKING METHODS

We have seen that locus identification and traversal impose a need for marking. In this section we look at various possible ways to implement the marking of neural assemblies. (We look briefly at connection-marking at the end of the section.) We assume that there are several distinct types of marks that can be placed "on" neural assemblies, and that neural assemblies must also be able to be *un*marked (unless the structure containing the neural assemblies marked is no longer of interest). Note also that we must assume that at any given time there may be *several* different marks on a given neural assembly. I shall point out that the implementation of marking causes awkward problems. The argument will not be to the effect that connectionist theories *cannot* encompass marking (though real difficulties will be pointed out for some varieties of theory) but rather that to encompass it they must include ad hoc or cumbersome provisions.

We shall look first at three ways in which marking might be done in a connectionist theory. The first method is called the *local-state* method, and assumes a local-state theory. Neural assemblies are marked by putting them into a special mode of activity, or by giving the neurons special membrane potentials, or by changing the concentration of some chemical in or around the neurons, or by changing the local state of the assembly in some other way. (We must remember here that different sorts of marking must be accounted for, and that an assembly may have several marks simultaneously.) The first point is that there is the difficulty in a non-disjoint-assembly theory that neural assemblies often share neurons, so that there could be conflict between the ways in which those neural assemblies are marked. This problem of conflict may not be insoluble, but at present it is a great impediment to accepting the local-state method in non-disjoint-assembly theories. The problem disappears when we turn to disjoint-assembly theories, and indeed the local-state method is then attractive. However, there is the methodological objection (mentioned in section 3.3) that it requires neural assemblies to have several symbolically distinct states. A technical objection to the method is that there is often a need to discover neural assemblies that are marked in a given way, rather than to discover the marks on a given neural assembly; but it is by no means clear how the former type of discovery can occur if marks take the form of local states. Finally, we must remember that marking is just a special case of the use of ancillary data structures. Even if the marking method being considered is deemed satisfactory, we are left with the problem of other ancillary data (e.g. lists used during backtracking in traversal).

The *replacement* method is a second conceivable marking approach. A neural assembly n is marked by replacing it by a neural assembly n′ that acts as the "marked version of n". Processing mechanisms are assumed to be able to recognize that n′ is a marked version of something and that n is not. Replacement of n by n′ (or vice versa for unmarking) is effected by somehow putting n out of play and linking n′ into the rest of the connectionist structure in just the way n is connected. Presumably, in an invisible-association theory, "putting n out of play" means actually deleting its connections with the rest of the connectionist structure, and, in a visible-association theory, it means deactivating the analogous connection-duples: we certainly do not want to get into the vicious circle of putting n out of play by marking it! (We could instead put n out of play by inhibiting it in some way which ensures that it can never be stimulated until a later special disinhibition signal is sent to it. But then we might just as well adopt the local-state approach.) The deletion or deactivation would be a time-consuming operation if there were many such connections or duples: and, of course, in a non-disjoint-assembly invisible-association theory the connection-deletion needed by the neural assembly replacement is problematical (see section 4). (Clearly, we would not want to avoid the connection-deletion by means of the "copying with modifications" technique explained in section 4: this would be a highly time-consuming thing to do just for the purposes of marking, and in any case we would be risking a vicious circle because the copying involves traversal, which involves marking in general.) A further difficulty is that it is not at all clear how processing mechanisms are to recognize that a neural assembly is a marked version of something, and that processing mechanisms that do not care whether a particular neuron set is a marked version or not will have to be able to deal with both cases. All this is quite apart from the fact that for each neural assembly we need a special companion marked-version neural assembly – indeed, we need a different one for each type of marking.

Our last suggested marking method is the *pointer* method. It appears to be the most natural approach to take in a connectionist theory. There are some special neural assemblies M_i (at least one for each of the different ways in which neural assemblies can be marked). In an invisible-association theory, we mark an ordinary neural assembly n by creating a connection between n and one of the special neural assemblies M_i. Unmarking consists in deleting such connections. The special neural assembly acts as a pointer to n. An immediate objection in the case of theories of the non-disjoint-assembly invisible-association variety is that we have seen that connection deletion (which is required by unmarking) is highly problematic. But there is another objection, in the case of both disjoint-assembly and non-disjoint-assembly theories. In order for a processing mechanism to take account of the fact that a neural assembly N is marked, it has to detect the connection between N and the mark neural assembly – and this detection requires probing. Bearing in mind that mark-detection is an extremely frequent operation during traversal and rule-execution, and that probing is a troublesome and (potentially time-consuming) operation, we see that an invisible-association theory faces a significant efficiency problem. We call this the *pervasive probing problem.*

We might suggest modifying the pointer method in the invisible-association case by allowing the use of association-indicators on the connections to the special mark neural assemblies. This is tantamount to using a visible-association scheme just for the limited purposes of marking. Such a step is possible, of course, but is distinctly ad hoc.

If we now generalize away from marking to more general ancillary data structures, which are presumably implemented as connectionist structures of some sort, we come to essentially the same conclusion. Use of ancillary data structures requires connection probing in general – but such use is all-pervasive and such probing is troublesome.

In the visible-association case, the direct analogue of the pointer method is to have special mark neural assemblies M_t just as before, and to have a connection-duple between each non-mark neural assembly and one mark neural assembly of each type t. The neural assembly n is t-marked if and only if the association-indicator between n and the t-mark neural assembly is active. Hence no probing is necessary to detect *a particular, given mark on a particular, given neural assembly.* Probing is still necessary to discover neural assemblies marked by a given mark, and probably also to discover the mark(s) on a given neural assembly. Therefore, in a visible-association theory the manipulation of marks leads to pervasive probing much as it does in the invisible-association case. The reason is that it is not true in general that connection-detection (of which mark detection is a special case) is necessarily detection of a connection between *two given* neural assemblies.

Backtrack lists cause problems for a visible-association theory. For instance, consider the backtrack-list implementation shown in Fig. 8(b) with connections replaced by connection-duples. Suppose the front element is to be deleted. Then a probe is needed to find out which special neural assembly A is pointed at by the "list start" neural assembly, and then a further probe is needed to find out which special neural assembly B is pointed at by A (because now the list-start assembly must be made to point to B). Let us now see whether we can avoid the pervasive probing problem if we adopt connection-duple marking as the implementation of backtrack lists. We assume duples are marked by virtue of the association-indicators being marked with a mark we call BT, and that this marking is done by the local-state or pointer method. The first neural assembly on the backtrack list is indicated by being marked with a mark we call FIRST. Consider a backtrack step: that is, the first neural assembly A is to be deleted from the list by having its FIRST mark removed, and that neural assembly B associated with A by a BT-marked connection-duple D is to be FIRST-marked. (Also, the BT mark on D must be removed.) The backtrack step could be effected as follows. Peculiar to each ordered pair (A, B) of neural assemblies recruitable for use in short-term connectionist structures there is a rule-like mechanism $\mathrm{STEP}_{A,B}$ that effects a backtrack step for the case when A is FIRST-marked and D is BT-marked.

Suppose first that marking is done by the local-state method. See Fig. 9(a). $\mathrm{STEP}_{A,B}$ has input connections from A and the association-indicator D on the connection between A and B (for simplicity we

~ Fig. 9

assume there is only one connection between A and B). $STEP_{A,B}$ also has a control input that tells it to try to do a backtrack step. $STEP_{A,B}$ has output connections to A, B and D. $STEP_{A,B}$ is triggered if and only if the following holds: the control input is active; A and D are active; A is FIRST-marked (we assume that the input from A is only active when A is active and FIRST-marked); and D is BT-marked (similar assumption). $STEP_{A,B}$ then immediately sends output to A, B and D, with the effect of making the right mark-changes. Hence, a backtrack step is simple and fast, and does not require any sort of probing. Of course, the method being proposed is analogous to method (I) in section 5.1, and suffers from a similar problem of replication of machinery – k^2 mechanisms $STEP_{A,B}$ are needed, where k is the number of recruitable neural assemblies. To try to avoid the replication, we could turn to a method involving probing. (The test for a BT mark on a *given* association-indicator may be fast, but probing is needed to find the association-indicators of duples impinging upon F.) It is not clear, however, that much mechanism, if any, will be saved by doing this.

If the pointer method is used for marking, then we run into a very similar situation. See Fig. 9(b). We assume the existence of special mark assemblies M_{FIRST} and M_{STEP}. We have mechanisms $STEP_{A,B}$ much as before. $STEP_{A,B}$ has a control input, an input from the association-indicator $A{:}M_{FIRST}$ on the connection between A and M_{FIRST}, and an input from the association-indicator $D{:}M_{STEP}$ on the connection between the association-indicator D (D as before) and M_{STEP}. (Notice the new factor here: association-indicators subordinate to other association-indicators.) $STEP_{A,B}$ has outputs to $A{:}M_{FIRST}$ and $D{:}M_{STEP}$, and also to $B{:}M_{FIRST}$, the association-indicator between B and M_{FIRST}. $STEP_{A,B}$ is triggered when and only when all three of its inputs are active, and then immediately sends output that switches $A{:}M_{FIRST}$ and $D{:}M_{STEP}$ off and $B{:}M_{FIRST}$ on. We again have k^2 mechanisms $STEP_{A,B}$. In an attempt to avoid this replication, we could either (a) probe from F to find the association-indicators on duples impinging upon F, and test those association-indicators for being BT marked, or (b) probe from the special neural assembly M_{STEP} to find BT-marked association-indicators and test them for impingement on F, or (c) perform a dual probing process starting from F and M_{STEP} (such a process being reminiscent of the techniques used by Fahlman (1979, 1981)). Again, it is not clear that there is actually any saving of mechanism.

In section 6 we mentioned that backtrack-lists could be implemented by neural-assembly marking even in an invisible-association theory. Could we implement the necessary marking using the local-state, replacement or pointer methods? The local-state approach would cause difficulties, because of the large size of the mark set. (Nor is it obvious how the mark ordering would be implemented and used.) The replacement method would likewise cause difficulties because of the large number of possible marks, and yet more circuitry would be needed to effect the ordering. The pointer method could be used, and the mark neural assemblies could be connected together to form an ordered chain. But the result is much the same as the implementation shown in Fig. 8(b). Therefore, the assembly-marking implementation of backtrack lists in an invisible-association theory is not beneficial.

Finally, we look briefly at *connection*-marking in invisible-association theories. (We noted in section 6 that the analogy in visible-association theories, connection-duple marking, can reduce to neural-assembly marking.) One option is to modify the theory by splitting each connection into a pair of connections joined by a neural assembly much like an association-indicator in a visible-association theory, and then marking the special neural assembly. But then it seems that we might as well use a visible-association theory anyway. Other options are to give connections local states similar to those of assemblies, or to allow connections to impinge upon connections as well as upon assemblies. Such options depart considerably from standard connectionist ideas.

8 SUMMARY

We have defined four orthogonal ways of categorizing connectionist theories: as separable-assembly versus inseparable-assembly (with the related dichotomy: disjoint-assembly versus non-disjoint-assembly); as permanent-reservation versus recruitment; as visible-association versus invisible-association; and as local-state versus not local-state. We rejected permanent-reservation theories of all types, as they demand implausibly large numbers of neural assemblies. We then saw that locus-identification (concerned with how "variables" in rules receive values as a result of rule satisfaction) requires mark creation (during rule satisfaction) and detection and deletion of those marks (during rule execution). Traversal of structures is also needed in rule triggering, unless a Fahlman-style parallel mark-passing scheme is used (but of course such a scheme reinforces the need for, and complexity of, marking). We saw that traversal is needed also for other purposes. Traversal itself imposes a need for marking and for the use of other ancillary data structures. We observed that marking, and the manipulation of ancillary data structures in general, must be done frequently and rapidly. However, we encounter difficulties in accounting for such manipulation in (recruitment) connectionist theories.

The points made lead to the following preferences:

(a) disjoint non-local-state invisible-association theories over non-disjoint non-local-state invisible-association theories (because the former simplify connection-deletion, which is needed in particular for the removal of marks);

(b) visible-association theories over invisible-association theories (because the detection of an association between two given neural assemblies is in principle easier in the former variety, and because back-track lists can be implemented more simply and efficiently, though still possibly with a great need for probing);

(c) non-local-state theories over local-state theories, from the point of view of connectionist purity; yet:

(d) disjoint local-state theories over other types of theory, from the point of view that the former allow the use of the local-state marking method, which is the simplest method (without disjointness there would be interference between the local states of different neural assemblies).

Combining (b) and (d), we arrive at a preference for *local-state, disjoint-assembly, visible-association theories*. (provided that, with local-state marking, there is some way in which neural assemblies marked with a given mark can be efficiently discovered).

The remarks in this paper may serve to focus research in connectionism in a new way. My own reaction to the points raised, however, has been to propose a rather different class of theories of the nature of short-term information-processing in the brain A detailed, up-to-date description of this class is in preparation, and there is no space even for an outline here. The class can be regarded as a (radical) development of local-state, disjoint-assembly, visible-association connectionist theories; however, it is less prone to the difficulties we have raised in this paper. The development is radical because connections are no longer the means by which associations between information items are encoded! An impression of the theory class can be gained from Barnden (1982a,b, 1983).

FOOTNOTES

(1) A complication we are ignoring here is that the rôles of John and Mary in the loving must of course be distinguished. The problem of rôles is important and interesting, but for simplicity we do not discuss it. In any case, it is not specifically to do with *short*-term structuring. Particular approaches to the problem can be found in Cottrell & Small (1983), Hinton (1981b), and Kohonen (1981). Similarly, we ignore the strongly related problem of deciding on the directions of connections. Whenever we mention connections in this paper we assume that the direction of a connection is appropriate to the representation task at hand.

(2) The sequence-of-manipulations view is employed in this paper for simplicity and definiteness, and is probably an over-simplification. It may in reality be more appropriate to replace it by a more complex view involving parallelism – for instance, by a view in which there is a merely-partially ordered set of rule executions.

(3) This need should be argued for in detail, but we do not have the space to do that here. The need is frequently expressed or implied in the literature.

(4) The term "given" here begs for an analysis which is difficult to undertake unless a particular, precise, given theory is at hand.

(5) The communication of processing loci corresponds to "variable binding" in computer-implemented computation. We avoid this term, lest it give the impression that there necessarily are things in the brain identifiable as variables.

(6) The last factor could be reduced by assuming different pools of neural assemblies for different relationships/properties; but this measure seems ad hoc, and is positively harmful if a situation arises in which an instance satellite for relationship R is needed, there is none such, but there is a free instance satellite for some other relationship.

(7) (a) It is important in what follows that a "list" is an abstract mathematical object: it is convenient for us to take a list to be a function from some (possibly empty) set $\{1,2,...,L\}$ of integers to the set of nodes in some graph structure. What is at issue is the neural implementation of such lists.

(b) Backtracking can be avoided in some circumstances by the use of special links, called threads, in the implementation of the structure [Knuth (1973), Standish (1980)]. However, threading amounts to a trick which cannot plausibly be ascribed to the workings of the brain.

FIGURE LEGENDS

Fig. 1.

(a) A possible active connectionist structure implementing the proposition that John loves Mary (ignoring any consideration of the direction of connections and of how rôles in relationships are distinguished).

(b) A semantic-network fragment which could be used to encode the proposition that John loves Mary.

Fig. 2. An active connectionist structure implementing the proposition "John loves Mary" in a visible-association, recruitment theory. (Once again we ignore the connection-direction and rôle issues.) Triangles stand for association-indicators. All neural assemblies shown (including the association-indicators) are active.

Fig. 3. Several hungriness propositions. Triangles stand for association-indicators.

Fig. 4. Method I in a visible-association, recruitment theory for implementing the rule specified in the text. Lines with arrowheads indicate input and output connections for the rule mechanisms. Double lines indicate other connections. H and H' are just two of the set of neural assemblies recruitable as instance satellites. Neural assembly M is a neural assembly representing some man.

Fig. 5. Method II in a visible-association, recruitment theory for implementing the rule specified in the text. Lines with arrowheads indicate input and output connections for the rule mechanisms. Only one instance satellite is shown for each man, but, for a given M, RULE receives an input pair ($c_{M,H,HUNGRY}$, $c_{M,H,M}$) for each recruitable neural assembly H. (So, in particular, RULE has input pairs ($c_{M,H',HUNGRY}$, $c_{M,H',M}$) and ($c_{M',H,HUNGRY}$, $c_{M',H,M'}$) as well as the ones shown.)

Fig. 6. If all the neural assemblies are active, this structure implements the proposition that woman w believes that man m is hungry. Note that detection of activity in fewer than all five association-indicators is insufficient to determine the presence of the proposition (because each instance satellite has connection-duples linking it to many neural assemblies not shown).

Fig. 7. Method III in a visible-association, recruitment theory for implementing the rule specified in the text. There are extra connections (not shown) to effect probing (see text). The probing operations are started when RULE receives stimulation along $c_{H,HUNGRY}$ for at least one H and along c_M for at least one M. Such input is not in itself enough to make the rule fire, because, for instance, there is a connection-duple between H and M' and a connection-duple between H' and M. (These connection-duples are not shown.) All that the mentioned input conveys to RULE is that there is *some* active hungriness proposition and that there is *some* proposition involving a man.

Fig. 8.

(a) Showing how a node list can be used to organize the traversal of a directed graph structure. The list always holds a path from the starting node to the current node in the graph. The list currently contains the nodes d, c, b, a in that order. When there is a move from the current node to a new node, the new node is added to the front of the list. When a node which has no unvisited successors is reached (and has just been put on the list), backtracking occurs. This consists of retracing steps by removing nodes one by one from the front of the list until a node with an unvisited successor is found. Visited nodes are indicated by ticks. Thus in the graph shown the list is about to contract and expand as follows: (c b a), (e c b a), (g e c b a), (e c b a), (c b a), (b a), (f b a), (b a), a, null. (For details of this and other traversal methods, see e.g. Knuth (1973), Standish (1980).)

(b) Possible implementation of a backtracking list in an invisible-association, recruitment theory. The uncrossed boxes show neural assemblies corresponding to some nodes in (a). The crossed boxes are neural assemblies recruited as backtrack-list elements. The list-head neural assembly is present throughout the traversal, and always "points to" the neural assembly acting as the first list element. The connections shown are created and deleted as necessary to reflect the expansions and contractions of the list. The horizontal connections implement the ordering within the list. The vertical connections establish the significance of the list-element neural assemblies.

Fig. 8(c). The connection-marking implementation of backtrack lists in an invisible-association theory. A connection is BT-marked when it is followed (in either direction) during traversal. A backtrack step causes one BT mark to be removed. The front assembly of the list is indicated by being marked with a special label, FIRST.

Fig. 9 $STEP_{A,B}$ performs a backtrack step when it receives a pulse on its control input, A is active and FIRST-marked, and D is active and BT-marked. $STEP_{A,B}$ unmarks A and D, and FIRST-marks B. In (a), local-state marking is used. In (b), the pointer marking method is used.

Fig. 10

A network diagram with five instances of the node-pattern, two instances each of the AGENT-pattern and the OBJECT-pattern, and numerous instances of the dot-pattern. The nodes are circles, all of same size. All instances of a given label-pattern are of the same size and orientation. The label-patterns are words for the purposes of illustration only. Link lines are not necessarily straight, and may cross each other.

Fig. 11. The function of the pattern-recognition mechanism. LA_{node} is the node-pattern location array for the network configuration in Fig. 10. LA_{AGENT} and LA_{dot} are the AGENT-pattern and dot-pattern location arrays for that network configuration. Dots in this array boxes in the figure indicate the 1s. All the other values in the arrays are zero. Not shown are the location arrays for the node labels and for the link labels other than AGENT.

For OPTIONAL APPENDIX

Fig. 12. Sketch of mechanism needed for a rule triggered by the presence in the PM of two nodes one of which is labelled with label A and the other of which is labelled with label B. (We take a single node labelled with both A and B to be a valid trigger.) LM_{node+A} and LM_{node+B} are like location matrices, but specify instances of composite patterns. LM_{node+A} has high activity at position (x,y) when LM_{node} and LM_A have high activity at or near position (x,y). (Similarly for B.) The rule fires (i.e. RULE starts some processing) when LM_{node+A} and LM_{node+B} both contain at least one element with high activity. Highlighting (see later in text) can be performed by RULE sending output to LM_{node+A} and LM_{node+B}. This output causes any active elements in those matrices to be highlighted. There are outputs from LM_{node+A} and LM_{node+B} to LM_{node} (and other LMs). The purpose of these outputs is for the highlighting at a position (x,y) in LM_{node+A} or LM_{node+B} to cause highlighting of elements at or near (x,y) in LM_{node}. The pattern-recognition mechanism is such that highlighting of an element in LM_{node} can be used to highlight the node centred at (x,y) in the PM.

Fig. 13. Implementation of a backtrack list in a network-configuration traversal. The network is the same as the one in Fig. 8(a). No extra nodes or links are used. The list is implemented by instances of a special mark, BT, on *links*. A link is BT-marked when it is followed (in either direction) during traversal. A backtrack step causes one BT mark to be removed. The front node of the list is indicated by being marked with a special label, FIRST. Recall that we toyed with an analogous implementation in the case of connectionist theories.

REFERENCES

Anderson, J.A. & G.E. Hinton. Models of information processing in the brain. In Hinton & Anderson (1981).

Anderson, J.A. and Mozer, M.C. Categorization and selective neurons. In Hinton and Anderson (1981).

Anderson, J.A., Silverstein, J.W., Ritz, S.A. and Jones R.S. Distinctive features, categorical perception, and probability learning: Some applications of a neural model. *Psychological Review,* 1977, *84, 413-451.*

Barnden, J.A. The integrated implementation of imaginal and propositional data structures in the brain. *Procs. 4th. Cognitive Science Conf.,* Ann Arbor, Michigan, 1982a.

Barnden, J.A. A continuum of diagrammatic data structures in human cognition. Tech. Rep. 131, Computer Science Dept., Indiana University, October 1982b.

Barnden, J.A. On association techniques in neural representation schemes. *Procs. 5th. Conf. of the Cognitive Science Society,* Rochester, NY, May 1983.

Campbell, F.W. The transmission of spatial information through the visual system. In F.O. Schmitt & F.G. Worden (eds), *The Neurosciences Third Study Program,* MIT Press, 1974.

Cottrell, G.W. and Small, S.L. A connectionist scheme for modelling word sense disambiguation. *Cognition and Brain Theory,* 1983, *6,* 89-120.

Davis, R. and King, J. An overview of production systems. In E. Elcock and D. Michie (Eds.), *Machine Intelligence 8,* Chichester: Ellis Horwood, 1977.

Duda, R.O. & Hart, P.E. *Pattern Classification and Scene Analysis.* Wiley: NY, 1973.

Edelman, G.M. Group selection and phasic reentrant signalling: a theory of higher brain function. In G.M. Edelman and V.B. Mountcastle, *The mindful brain.* Cambridge, Mass.: MIT Press, 1978.

Fahlman, S.E. *NETL: A system for representing and using real-world knowledge.* Cambridge, Mass.: MIT Press, 1979.

Fahlman, S.E. Representing implicit knowledge. In Hinton and Anderson (1981).

Fahlman, S.E., Hinton, G.E. & Sejnowski, T.J. Massively parallel architectures for artificial intelligence. In *Procs. Natl. Conf. on Artificial Intelligence,* Washington, D.C., Aug. 1983. Kaufmann: Los Altos, 1983.

Feldman, J.A. Memory and change in connection networks. Tech. Rep. 96, Dept. of Comp. Sci., Univ. of Rochester, Dec. 1981a.

Feldman, J.A. A connectionist model of visual memory. In Hinton and Anderson (1981b).

Feldman, J.A. Dynamic Connections in Neural Networks. *Biol. Cybern.,* 1982, *46,* 27-39.

Feldman, J.A. Introduction: Advanced computational models in cognition and brain theory. *Cognition and Brain Theory,* 1983, *6,* 1-3.

Feldman, J.A. & Ballard, D.H. Connectionist models and their properties. *Cognitive Science,* 1982, *6,* 205-254.

Findler, N.V. (Ed). *Associative networks.* New York: Academic Press, 1979.

Gigley, H.M. HOPE – AI and the dynamic process of language behavior. *Cognition and Brain Theory,* 1983, *6,* 39-88.

Goddard, G.V. Component properties of the memory machine: Hebb revisited. In P.W. Jusczyk and R.M. Klein (Eds.), *The nature of thought: Essays in honor of D.O. Hebb.* Hillsdale, N.J.: Lawrence Erlbaum, 1980.

Hebb, D.O. *Organization of behaviour.* New York: Wiley, 1949.

Hinton, G.E. Shape representation in parallel systems. In *Procs. 7th. Int. Joint Conf. on Artificial Intelligence,* Vancouver, B.C., Aug. 1981a.

Hinton, G.E. Implementing semantic networks in parallel hardware. In Hinton and Anderson (1981). 1981b.

Hinton, G.E. and Anderson, J.A. (eds) *Parallel models of associative memory.* Hillsdale, NJ: Lawrence Erlbaum, 1981.

Knuth, D.E. *The art of computer programming. Vol. 1: Fundamental algorithms.* (2nd. Ed.) Addison-Wesley: Reading, Mass., 1973.

Kohonen, T., Oja, E. and Lehtiö. Storage and processing of information in distributed associative memory systems. In Hinton and Anderson (1981).

Matthies, H. Plasticity in the nervous system – an approach to memory research. In C. Ajmone-Marsan & H. Matthies (Eds), *Neuronal Plasticity and Memory Formation,* IBRO Monog. Series Vol. 9, Raven: New York, 1982.

McClelland, J.L. & Rumelhart, D.E. An interactive activation model of context effects in letter perception: Part 1. *Psychological Review,* 1981, *88,* 375-407.

Mountcastle, V.B. An organizing principle for cerebral function: the unit module and the distributed system. In G.M. Edelman and V.B. Mountcastle, *The mindful brain.* Cambridge, Mass.: MIT Press, 1978.

Pollack, J. & Waltz D. Natural-language processing using spreading activation and lateral inhibition. *Procs. 4th. Cognitive Science Conf.,* Ann Arbor, Michigan, 1982.

Pollen, D.A. & Taylor, J.H. The striate cortex and the spatial analysis of visual space. In F.O. Schmitt & F.G. Worden (eds), *The Neurosciences Third Study Program,* MIT Press, 1974.

Routtenberg, A. Memory formation as a post-translational modification of brain proteins. In C. Ajmone-Marsan & H. Matthies (Eds), *Neuronal Plasticity and Memory Formation,* IBRO Monog. Series Vol. 9, Raven: New York, 1982.

Rumelhart, D.E. & McClelland, J.L. An interactive activation model of context effects in letter perception: Part 2. *Psychological Review,* 1982, *89,* 60-94.

Rumelhart, D.E. and Norman, D.A. A comparison of models. Introduction to Hinton and Anderson (1981).

Small, S.L. Exploded connections: unchunking schematic knowledge. *Procs. 4th. Cognitive Science Conf.,* Ann Arbor, Michigan, 1982.

Standish, T.A. *Data Structure Techniques.* Addison-Wesley: Reading, Mass., 1980.

Waterman, D.A. and Hayes-Roth, F. *Pattern-directed inference systems.* New York: Academic Press, 1978.

Wickelgren, W.A. Chunking and consolidation. *Psychological Review,* 1979, *86,* 44-60.

FOR POSSIBLE INCLUSION if referees think it appropriate and it does not make the paper too long. This appendix is a summary of a paper in preparation.

APPENDIX: SKETCH OF AN ALTERNATIVE PARADIGM

I propose a paradigm called "patternism" as an alternative to connectionism. According to patternism, a short-term data structure is a configuration of neural activity made up of inter-associated sub-configurations, where an association between two sub-configurations rests on their *relative positions* (rather than on connections). The basic sub-configurations are instances of certain "unanchored" primitive activity-patterns. These patterns are unanchored in not being tied to a particular set of neurons, as they can be instantiated in different neural positions. In fact, they can be instantiated anywhere within certain neural areas called *pattern matrices (PMs)*. PMs are, abstractly speaking, two-dimensional rectangular arrays of small neural networks called *basic elements*. There is a set of *activity modes,* and each basic element has some intensity of activity in each mode. Normally, a basic element has a "resting level" of activity in each mode, but when the element is part of a set of basic elements supporting a pattern instance it has non-resting activity in at least one mode.

Fig. 10 shows an example of an important sort of PM activity configuration, playing the rôle that the active connectionist structures have played in this paper. Such a configuration is called a *network configuration.* The primitive patterns are the node pattern, a large set of label patterns, and the dot pattern. Any instance of the node pattern occupies, roughly, a "circle" of basic elements. Any instance of the dot pattern occupies a small localized group of basic elements; instances are placed in a PM so as to form lines (called "links") that have much the significance that links have in diagrams of semantic networks. Label-pattern instances are placed next to nodes and links, the intention being to label them in much the way, again, that labels are used in semantic network diagrams. Another view is that nodes in network configurations play the rôle that neuron sets play in the connectionist paradigm, and the links play the rôle of connections. There are types of PM configuration other than network configurations, but for brevity we ignore them here – see Barnden (1982a, 1982b). There are are also some issues and details concerning network configurations which are not discussed here; for example, the lines in network configurations can be given directions, in a number of different ways. Network configurations themselves in the form depicted in Fig. 10 are not to be taken too seriously – they are intended only to be first approximations to activity configurations appearing in the brain. Fig. 10

The most fundamental form of association between sub-configurations is relative position – and in fact for network configurations we need only consider *adjacency* relationships between sub-configurations. For instance, labels are associated with particular nodes and links by virtue of being adjacent to those nodes and links in a PM. Links are associated with particular nodes by virtue of the adjacency of their ends to the nodes. Links are made up of dot-pattern instances associated by their adjacency. On the other hand, adjacency does not necessarily imply association: thus, two links or labels might be near to each other without their being taken to be associated. Whether an adjacency constitutes an association depends on the use made of PM configurations by processing rules (see below). Association by adjacency is closely analogous to the association between bit-strings that arises in computer-implemented data structures when the bit-strings are in neighbouring locations. However, adjacency in patternism is a much fuzzier matter than that occurring in computer memory.

Links serve to associate nodes, of course. This form of association is parasitic on adjacency-association, but can conveniently be considered to be primitive most of the time. A third form of association is *similarity association,* and is closely analogous to "content-addressing" in computers. There are some special labels that can be placed next to nodes, the intention being that two nodes with the same special label are considered to represent the same thing (to be, in a sense, the same node). This form of association allows individual PM configurations to be simplified and allows PM configurations in different PMs to be tied together.

We assumed that short-term information processing in the connectionist paradigm consists of a sequence of rule-executions manipulating connectionist structures. In patternism, the processing consists of a sequence of rule-executions manipulating network configurations. The action parts of the rules are sequences of primitive operations that move pattern instances around in PMs, introduce new instances of patterns, delete pattern instances, and interact with other mechanisms (for instance, move PM configurations into and out of long-term storage – an issue not discussed here). A rule is fired by the presence or absence of instances of particular patterns, where the instances must have particular adjacency relationships. For example, a rule might be fired by the presence of a node that has an instance of a particular label-pattern next to it provided that it does not also have an instance of some other particular label-pattern next to it. Rules are not generally interested in the absolute positions of pattern instances.

Clearly, the rules require the existence of a pattern-recognition mechanism capable of finding the locations of the instances of the different patterns. A particular pattern-recognition mechanism has been devised. For simplicity, we assume in what follows that there is just one PM. The output of the pattern-recognition mechanism consists of neural activity configurations in a set of *location matrices.* There is one location matrix LM_p for each primitive pattern p. A location matrix is an array of elements that is isomorphic to the PM. The configuration in a location matrix consists of some standard non-zero level of activity at each position (x,y) such that in the PM there is an instance of the pattern p centred at (x,y). The mode that that activity is in is the same as the mode used by the pattern instance (we assume for simplicity in this account that a pattern instance involves non-resting activity in only one mode.) See Fig. 11. The mechanisms which implement rules actually take their input from the location matrices, not from the PM itself. Fig. 12 shows an outline of the mechanism needed for a certain rule to be fired. Fig. 11. Fig. 12.

The important point is that adjacency associations can be detected easily *without a process analogous to connection-probing,* yet the action mechanism for a rule does not have to be replicated for all the possible PM positions in which the pattern instances in its condition part might be. The avoidance of connection-probing in adjacency-association detection means that marking can be conveniently and efficiently done by placing special labels next to nodes (or, indeed, next to labels or links). A slightly different type of marking is used for locus identification. When a rule condition part is satisfied, pattern instances that led to the satisfaction can be *highlighted* in different modes. (A pattern instance that does not use mode h becomes highlighted in mode h when all basic elements occupied by the instance are stimulated into non-resting activity in mode h. Also, highlighting of a pattern instance in the PM has the effect of highlighting the corresponding point in the LM for the pattern.) Activity modes used for highlighting can therefore be employed by rule action parts to identify pieces of the PM configuration. The ability of the rule-satisfaction mechanisms to cause highlighting rests on the ability of the pattern-recognition mechanism to be "run backwards" in order to *insert* as well as to detect patterns (see below). Highlighting can be viewed as a special case of the previous type of marking – placing labels next to pattern instances. For, we can take the "label" to be another instance of the same pattern, only using the mode h this time; and we have identity of position rather than adjacency.

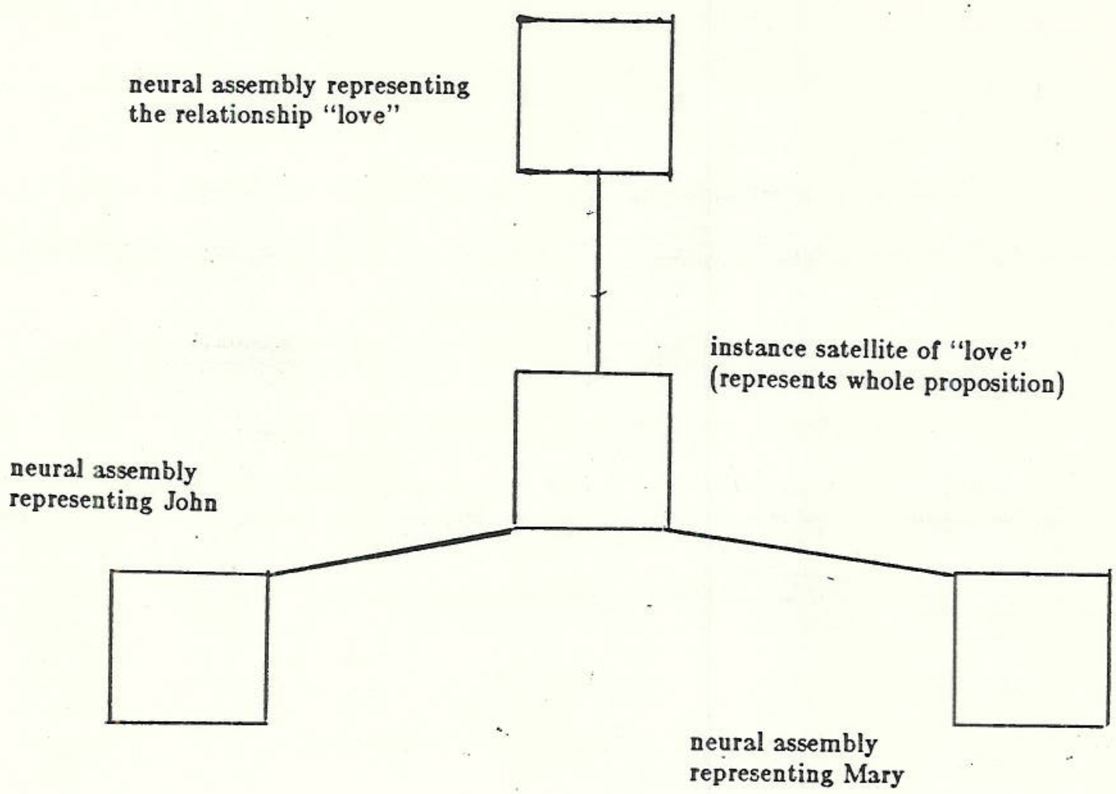
The detection of structures involving links may require a probing process (although the process is less problematic than connection-probing in connectionist theories). It therefore might appear that patternism does not counter the efficiency problems that face connectionism in regard to the manipulation of ancillary data structures more general than marks. However, patternism has less need of links in ancillary data structures than connectionism has need of connections in them. Consider, for instance, a list used in backtracking during traversal of a network. Fig. 13 shows the possible implementation of the list in a PM: no extra nodes or links are needed, because link-marking is used. The figure should be contrasted with Fig. 8(b). Section 6 toyed with the idea of connection-marking, but section 7 found it to be problematic. In a patternist theory it is just as natural and easy to mark a link (or even a label – or a mark) as to mark a node. Fig. 13

The proposed pattern-recognition mechanism is based on Fourier transforms. It has often been noted that Fourier transforms can be used to perform template matching [Duda & Hart (1973), pp.305ff]. If Fourier transforms are produced by parallel computation in neural nets, templates can be matched against an image with no scanning or sweeping. The proposed mechanism uses a variant of the usual Fourier technique. To "sharpen" the configurations in LMs, a relaxation (settling down) method is used in combination with mutual inhibition among the elements of each individual LM. An important feature of the mechanism is that it is easy to run it backwards to achieve insertion of pattern instances (whose positions are specified by active points in the LMs for the patterns). Instance deletion is also easy to achieve. It has been suggested that the brain uses Fourier techniques to perform visual pattern-recognition (which is strongly related to the pattern-recognition needed). [See e.g. Campbell (1974), Pollen & Taylor (1974).]

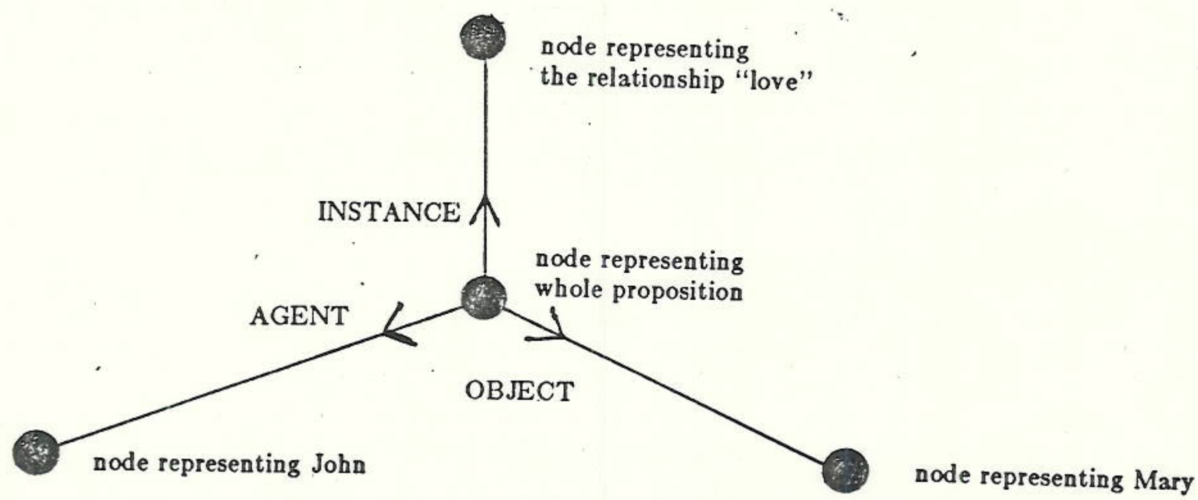
It might appear that, since the rule mechanisms take input from the LMs, there is no need to have the PM itself in the first place, and therefore no need for patterns or pattern-recognition. However, there are important uses for them, such as providing channels between long-term memory and short-term memory.

Patternism can be viewed as springing from local-state, disjoint-assembly, visible-association connectionist theories (the preferred variety – see section 8). We can regard the set of basic elements used by a pattern-instance as forming a recruited neural assembly similar to those in a recruitment connectionist theory. Pattern-instances use more-or-less disjoint sets of basic elements. The idea of line-links joining pattern-instances can be viewed as being derived from the idea of association-indicators in visible-association connectionist theories. Highlighting is derived from local-state ideas in connectionist theories. However, highlighting is more in tune with the theory as a whole than local states are with pure connectionism.

My present research concerns the simulation of the action of the pattern-recognition mechanism (at an abstract level – not for the moment at the level of neural circuits). When the logical adequacy of the mechanism has been confirmed by simulation, it will be appropriate to investigate its neural plausibility in more detail.



(a) A possible active connectionist structure implementing the proposition that John loves Mary (ignoring any consideration of the direction of connections and of how roles in relationships are distinguished).



(b) A semantic-network fragment which could be used to encode the proposition that John loves Mary.

Fig. 1.

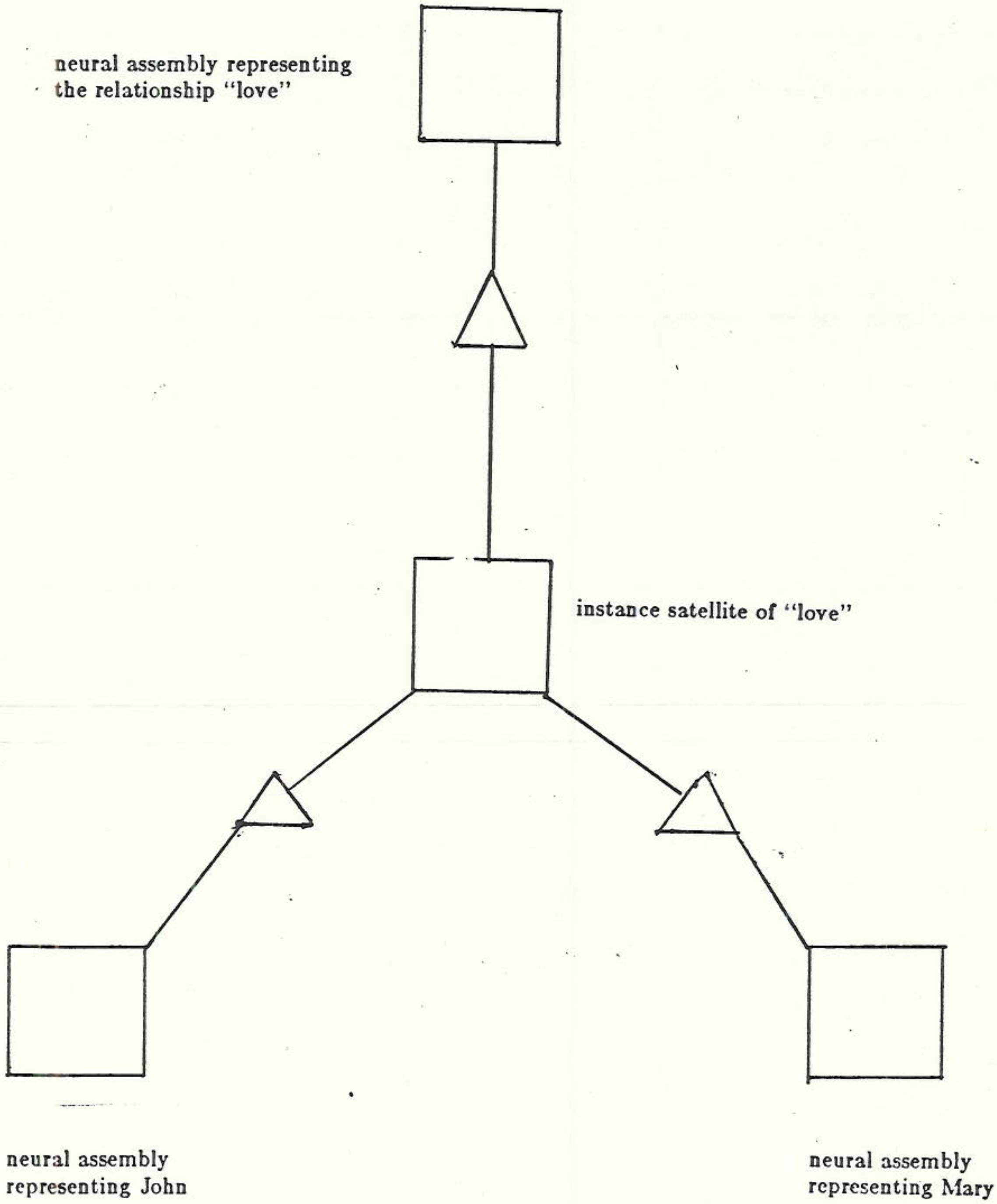


Fig. 2. An active connectionist structure implementing the proposition "John loves Mary" in a visible-association, recruitment theory. (Once again we ignore the connection-direction and role issues.) Triangles stand for association-indicators. All neural assemblies shown (including the association-indicators) are active.

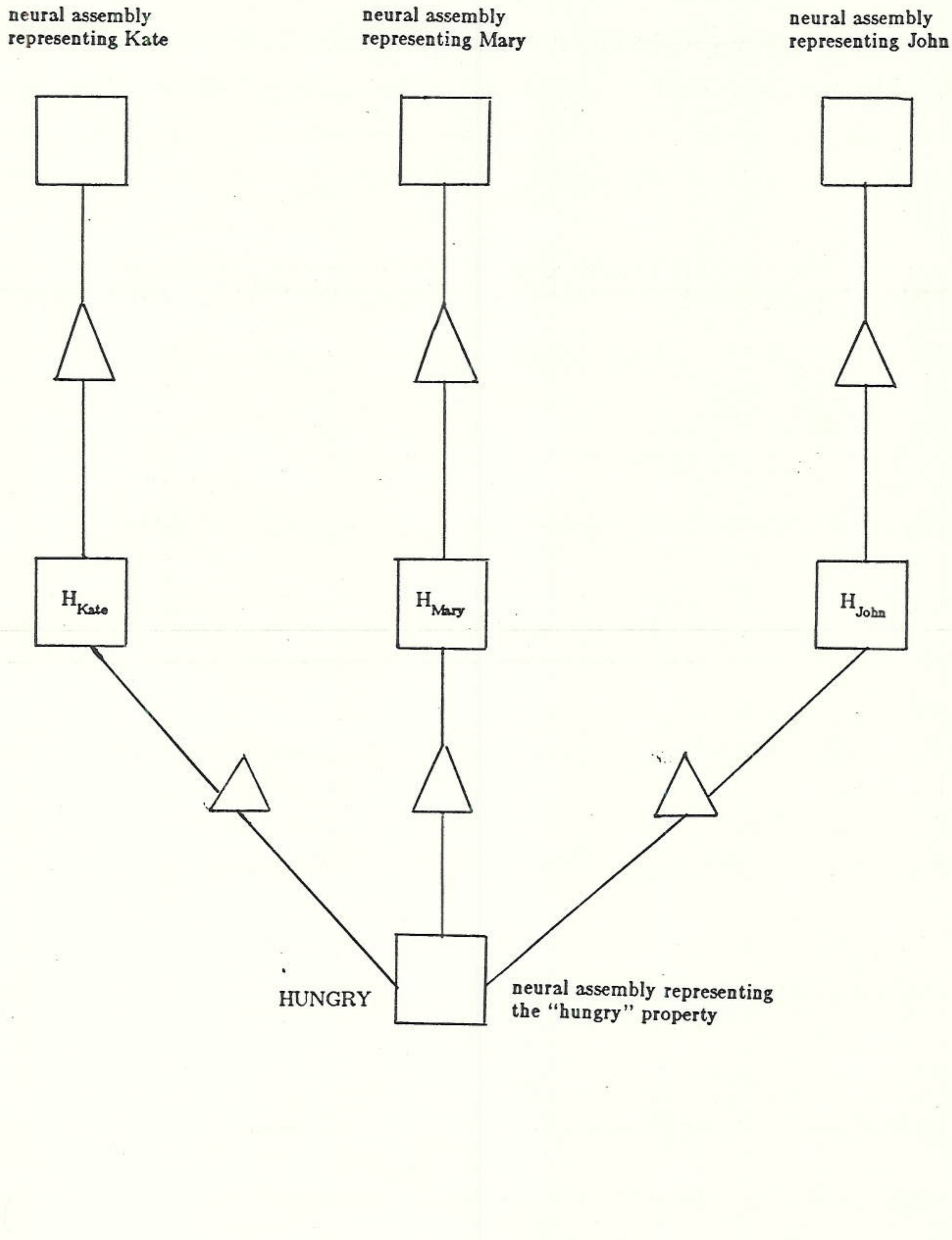


Fig. 3. Several hungriness propositions. Triangles stand for association-indicators.

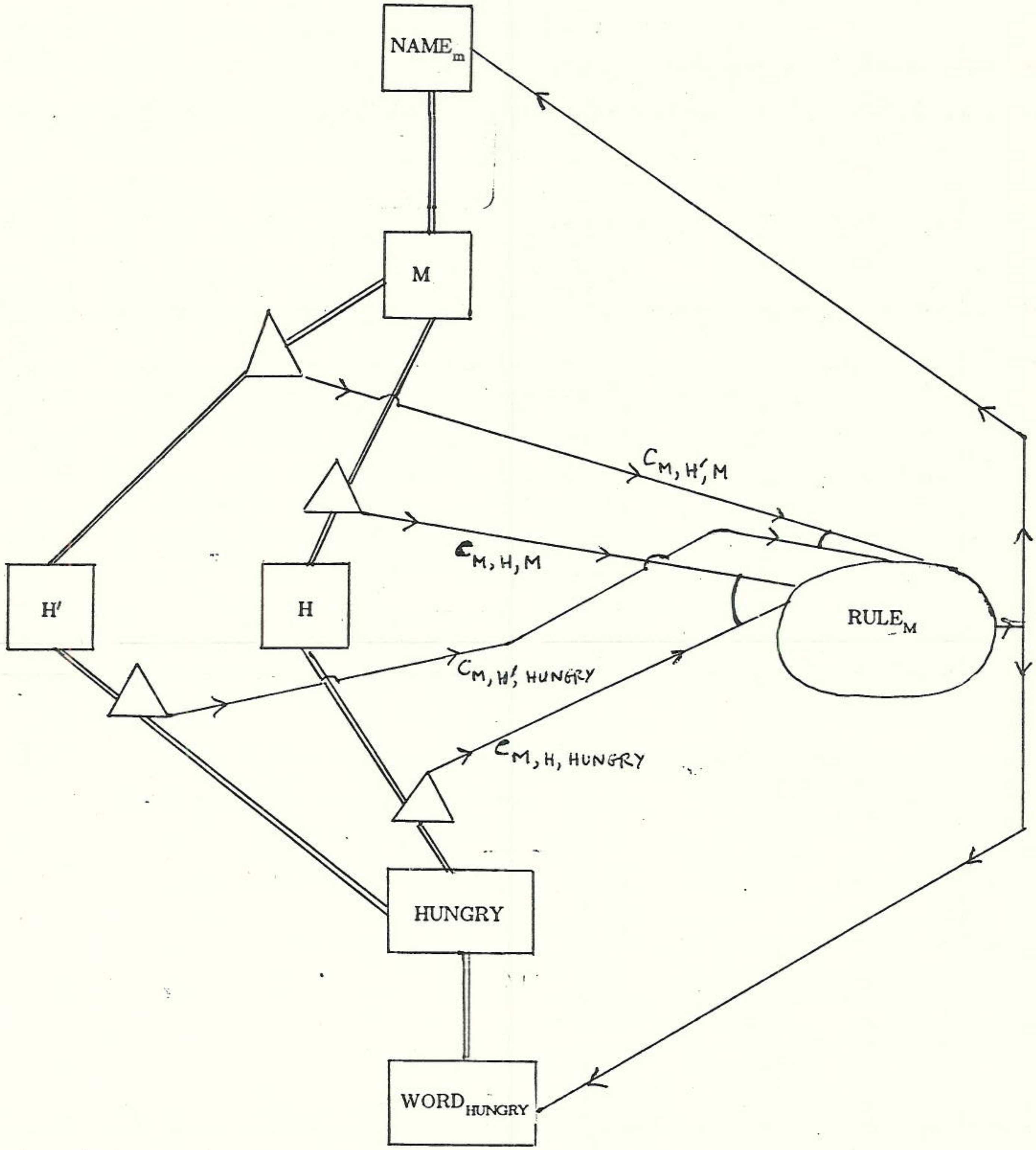


Fig. 4. Method I in a visible-association, recruitment theory for implementing the rule specified in the text. Lines with arrowheads indicate input and output connections for the rule mechanisms. Double lines indicate other connections. H and H′ are just two of the set of neural assemblies recruitable as instance satellites. Neural assembly M is a neural assembly representing some man.

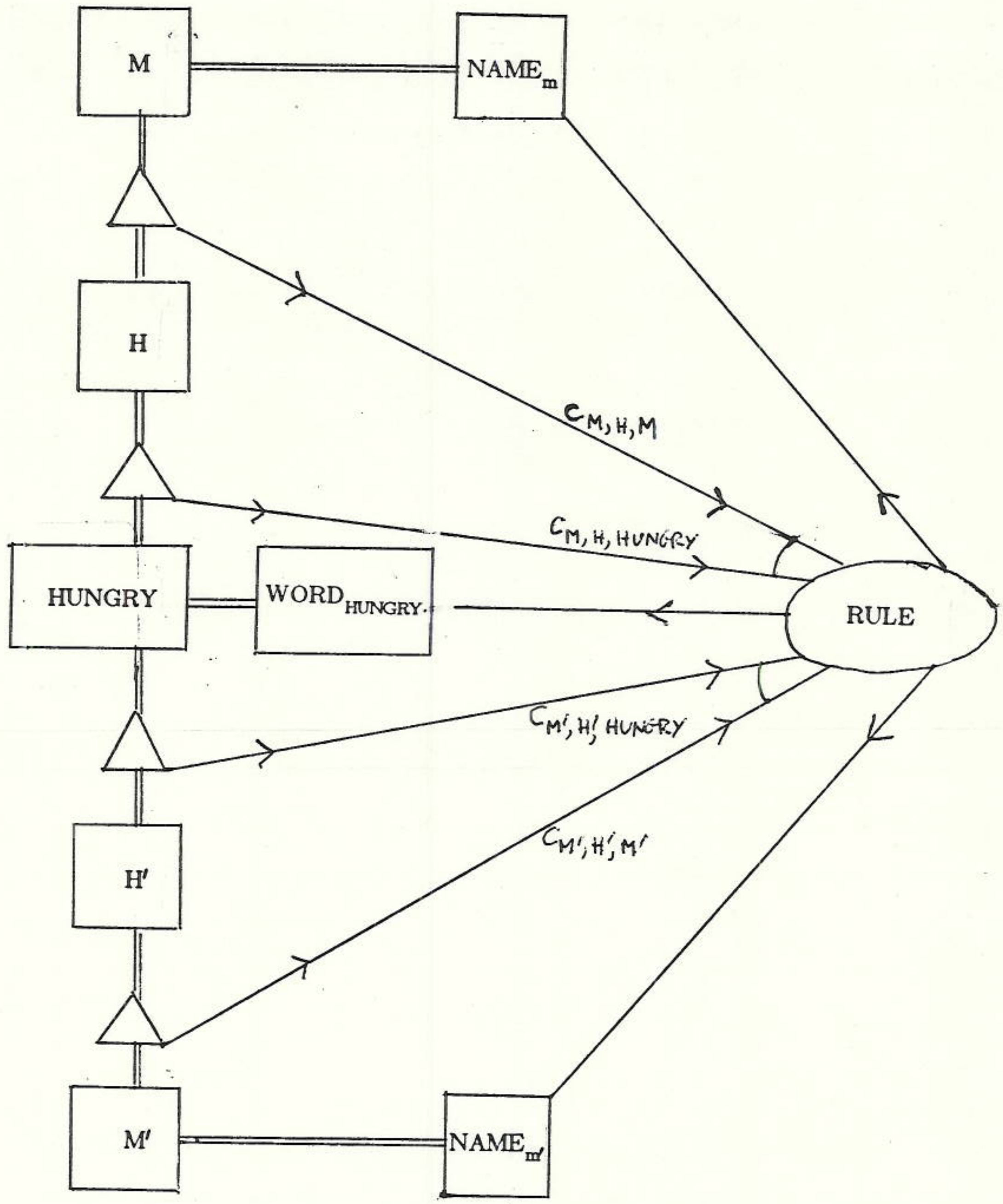


Fig. 5. Method II in a visible-association, recruitment theory for implementing the rule specified in the text. Lines with arrowheads indicate input and output connections for the rule mechanisms. Only one instance satellite is shown for each man, but, for a given M, RULE receives an input pair ($c_{M,H,HUNGRY}$, $c_{M,H,M}$) for each recruitable neural assembly H. (So, in particular, RULE has input pairs ($c_{M,H',HUNGRY}$, $c_{M,H',M}$) and ($c_{M',H,HUNGRY}$, $c_{M',H,M'}$) as well as the ones shown.)

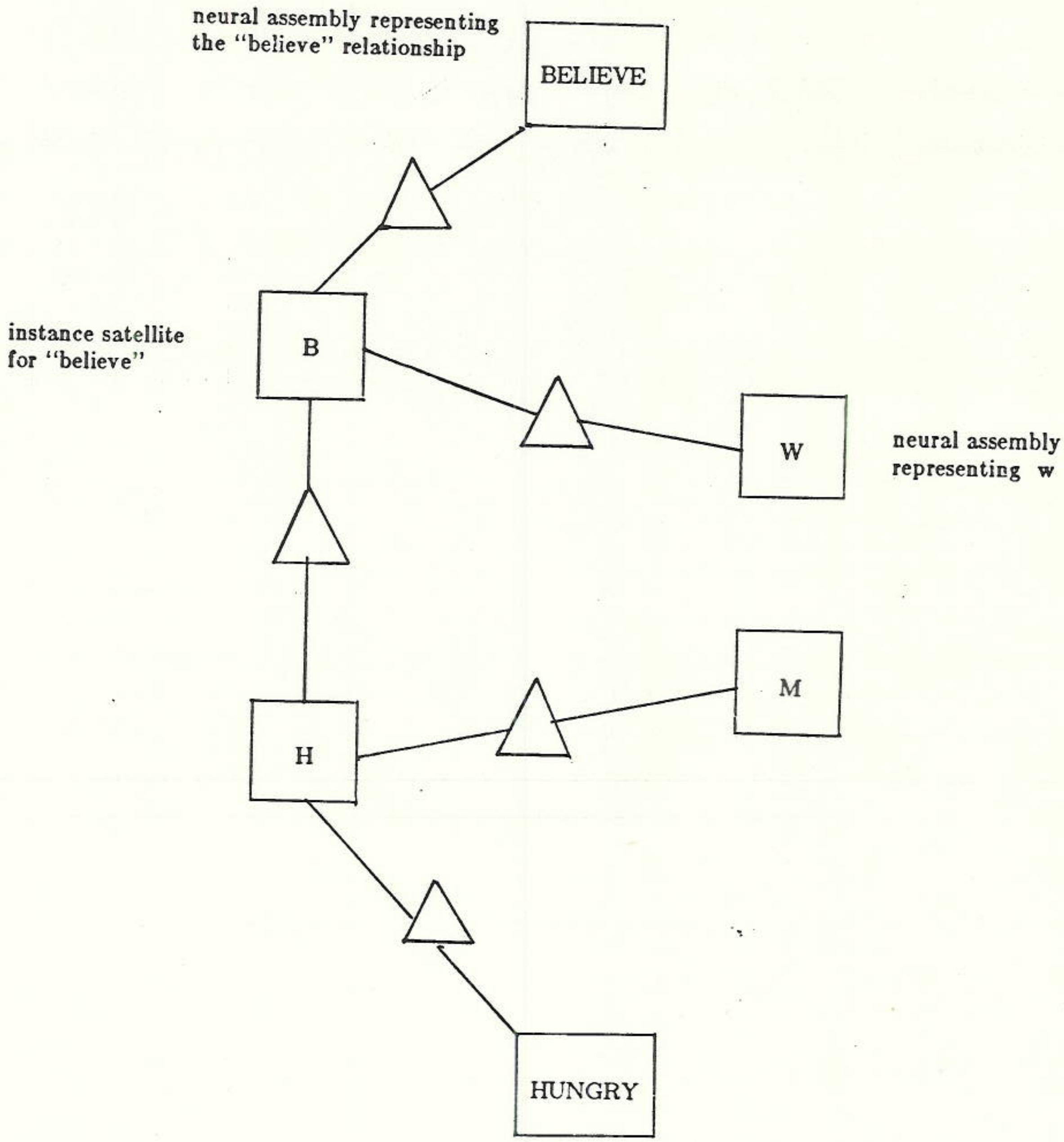


Fig. 6. If all the neural assemblies are active, this structure implements the proposition that woman w believes that man m is hungry. Note that detection of activity in fewer than all five association-indicators is insufficient to determine the presence of the proposition (because each instance satellite has connection-duples linking it to many neural assemblies not shown).

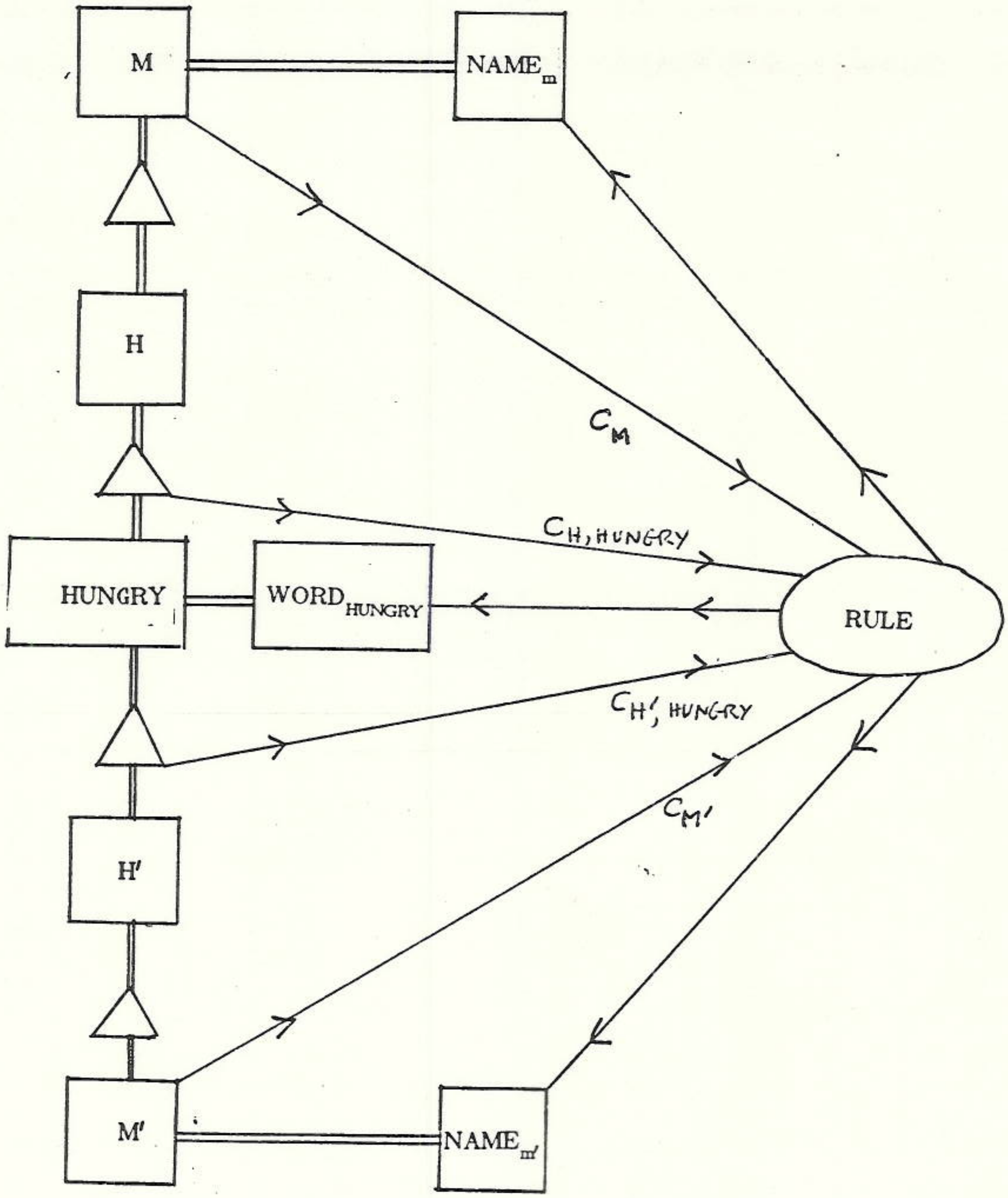
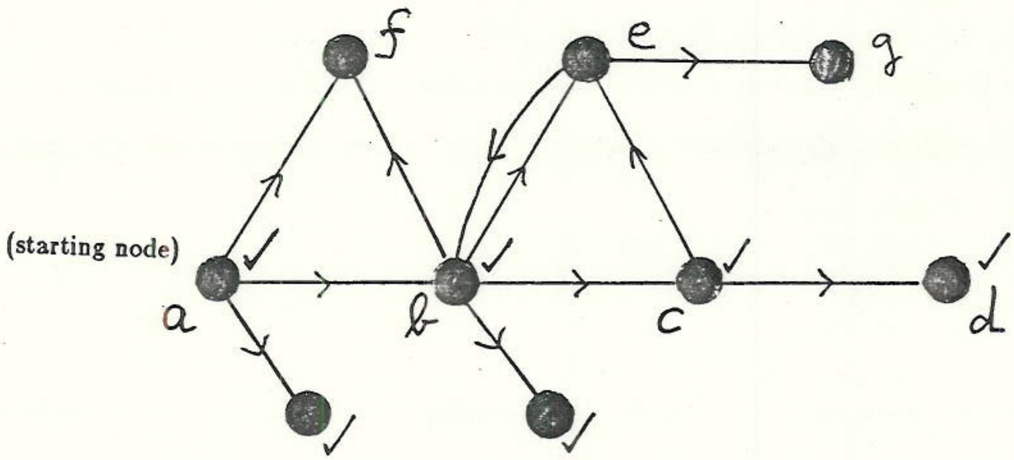
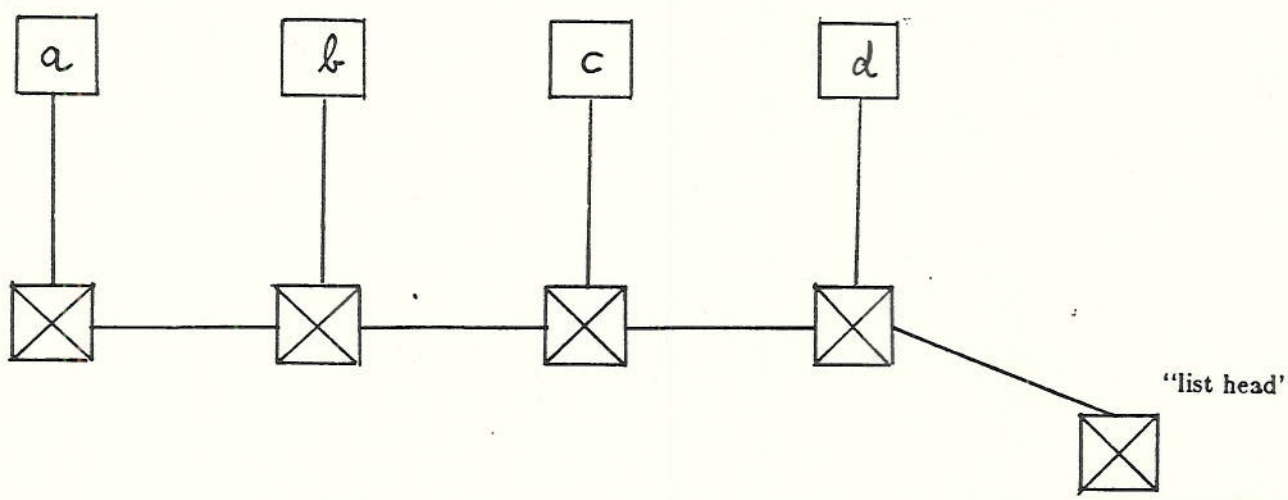


Fig. 7. Method III in a visible-association, recruitment theory for implementing the rule specified in the text. There are extra connections (not shown) to effect probing (see text). The probing operations are started when RULE receives stimulation along $c_{H,HUNGRY}$ for at least one H and along c_M for at least one M. Such input is not in itself enough to make the rule fire, because, for instance, there is a connection-duple between H and M′ and a connection-duple between H′ and M. (These connection-duples are not shown.) All that the mentioned input conveys to RULE is that there is *some* active hungriness proposition and that there is *some* proposition involving a man.



(a) Showing how a node list can be used to organise the traversal of a directed graph structure. The list always holds a path from the starting node to the current node in the graph. The list currently contains the nodes d, c, b, a in that order. When there is a move from the current node to a new node, the new node is added to the front of the list. When a node which has no unvisited successors is reached (and has just been put on the list), backtracking occurs. This consists of retracing steps by removing nodes one by one from the front of the list until a node with an unvisited successor is found. Visited nodes are indicated by ticks. Thus in the graph shown the list is about to contract and expand as follows: (c b a), (e c b a), (g e c b a), (e c b a), (c b a), (b a), (f b a), (b a), a, null. (For details of this and other traversal methods, see e.g. Knuth (1973), Standish (1980).)



(b) Possible implementation of a backtracking list in an invisible-association, recruitment theory. The uncrossed boxes show neural assemblies corresponding to some nodes in (a). The crossed boxes are neural assemblies recruited as backtrack-list elements. The list-head neural assembly is present throughout the traversal, and always "points to" the neural assembly acting as the first list element. The connections shown are created and deleted as necessary to reflect the expansions and contractions of the list. The horizontal connections implement the ordering within the list. The vertical connections establish the significance of the list-element neural assemblies.

Fig. 8.

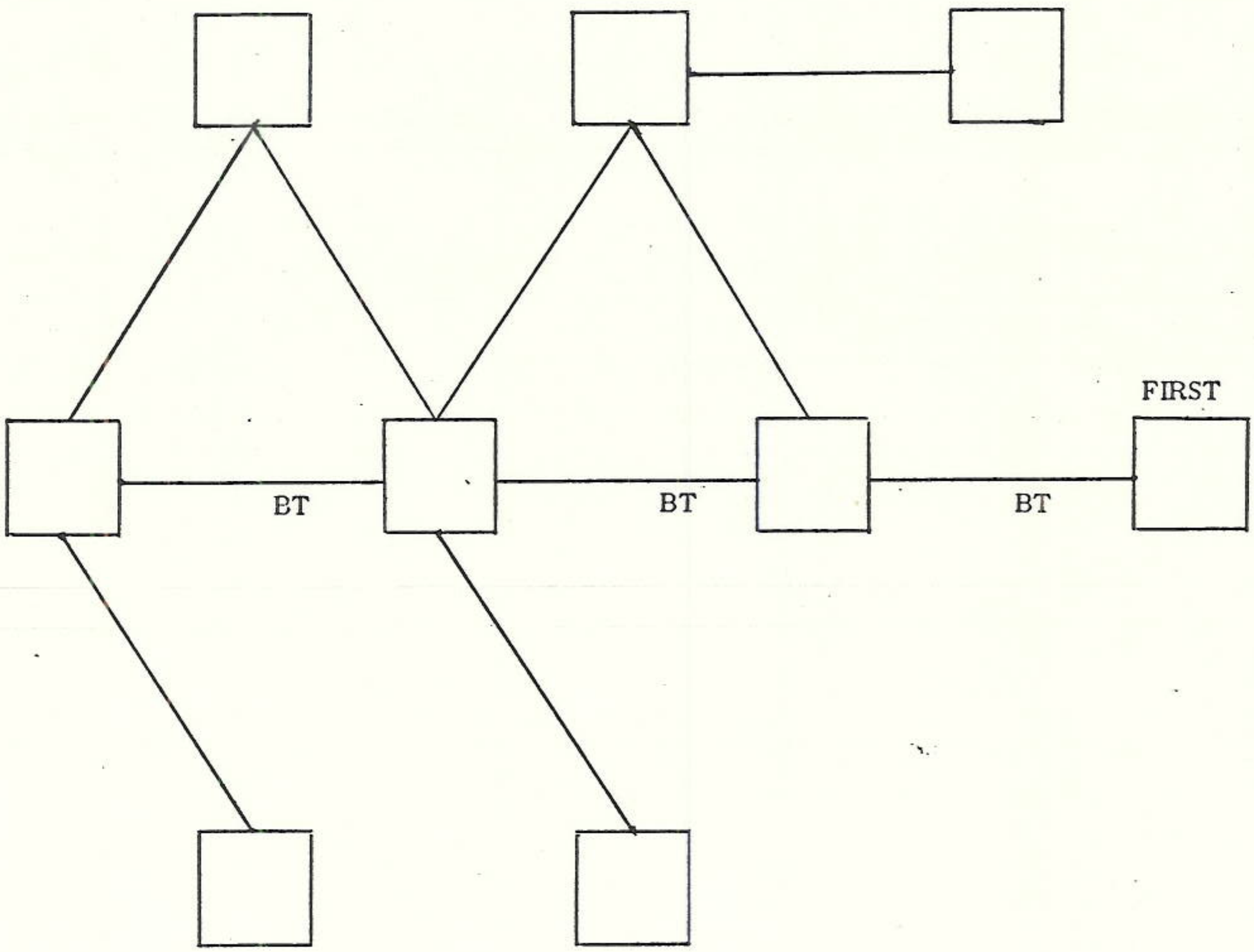


Fig. 8(c). The connection-marking implementation of backtrack lists in an invisible-association theory. A connection is BT-marked when it is followed (in either direction) during traversal. A backtrack step causes one BT mark to be removed. The front assembly of the list is indicated by being marked with a special label, FIRST.

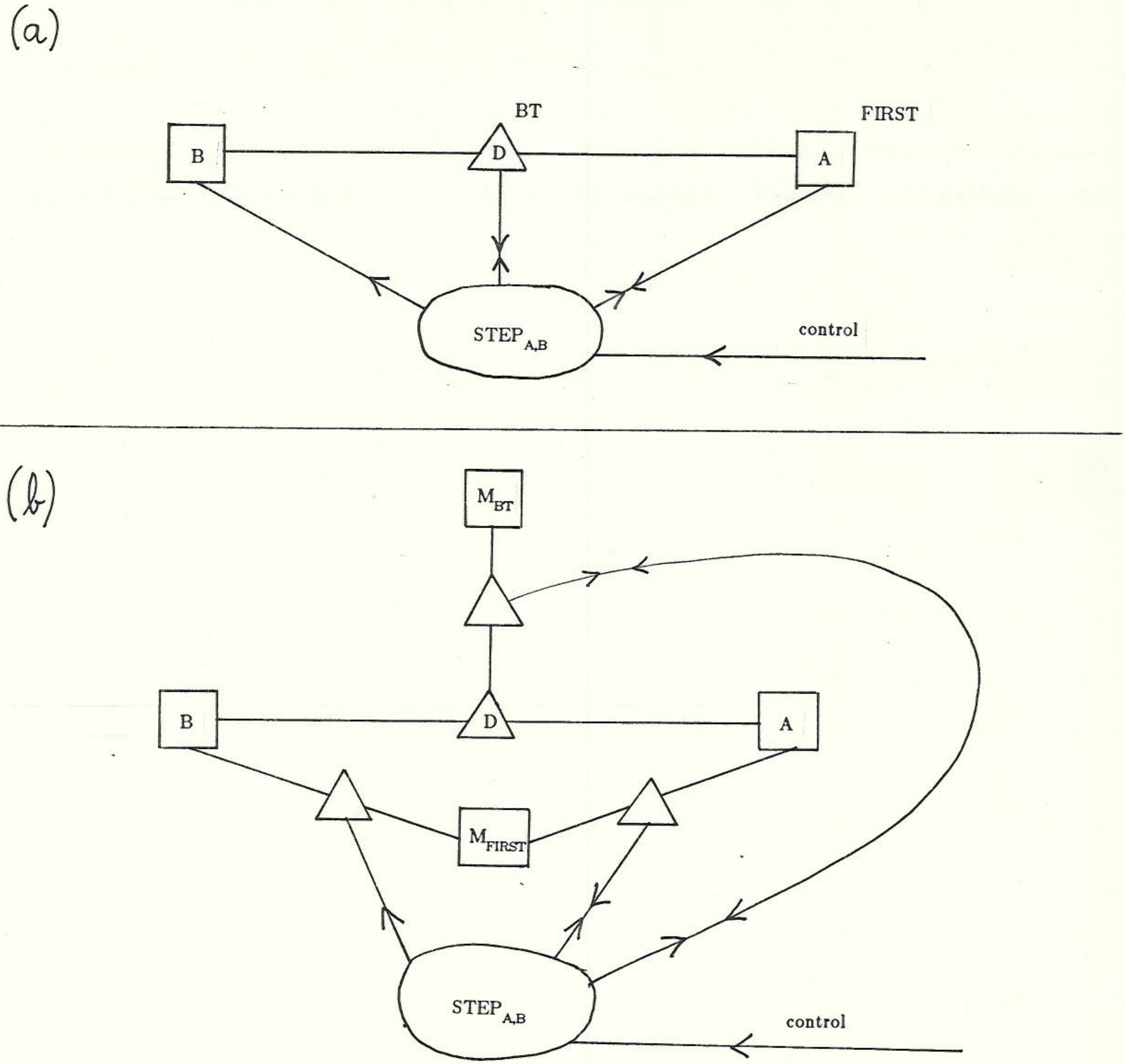


Fig. 9 STEP$_{A,B}$ performs a backtrack step when it receives a pulse on its control input, A is active and FIRST-marked, and D is active and BT-marked. STEP$_{A,B}$ unmarks A and D, and FIRST-marks B. In (a), local-state marking is used. In (b), the pointer marking method is used.

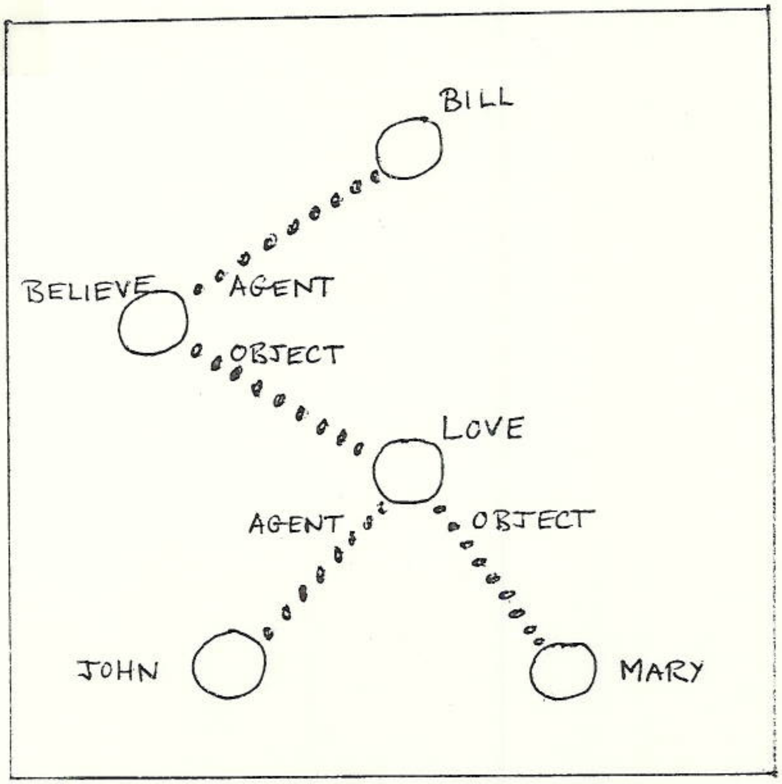


Fig. 10

A network diagram with five instances of the node-pattern, two instances each of the AGENT-pattern and the OBJECT-pattern, and numerous instances of the dot-pattern. The nodes are circles, all of same size. All instances of a given label-pattern are of the same size and orientation. The label-patterns are words for the purposes of illustration only. Link lines are not necessarily straight, and may cross each other.

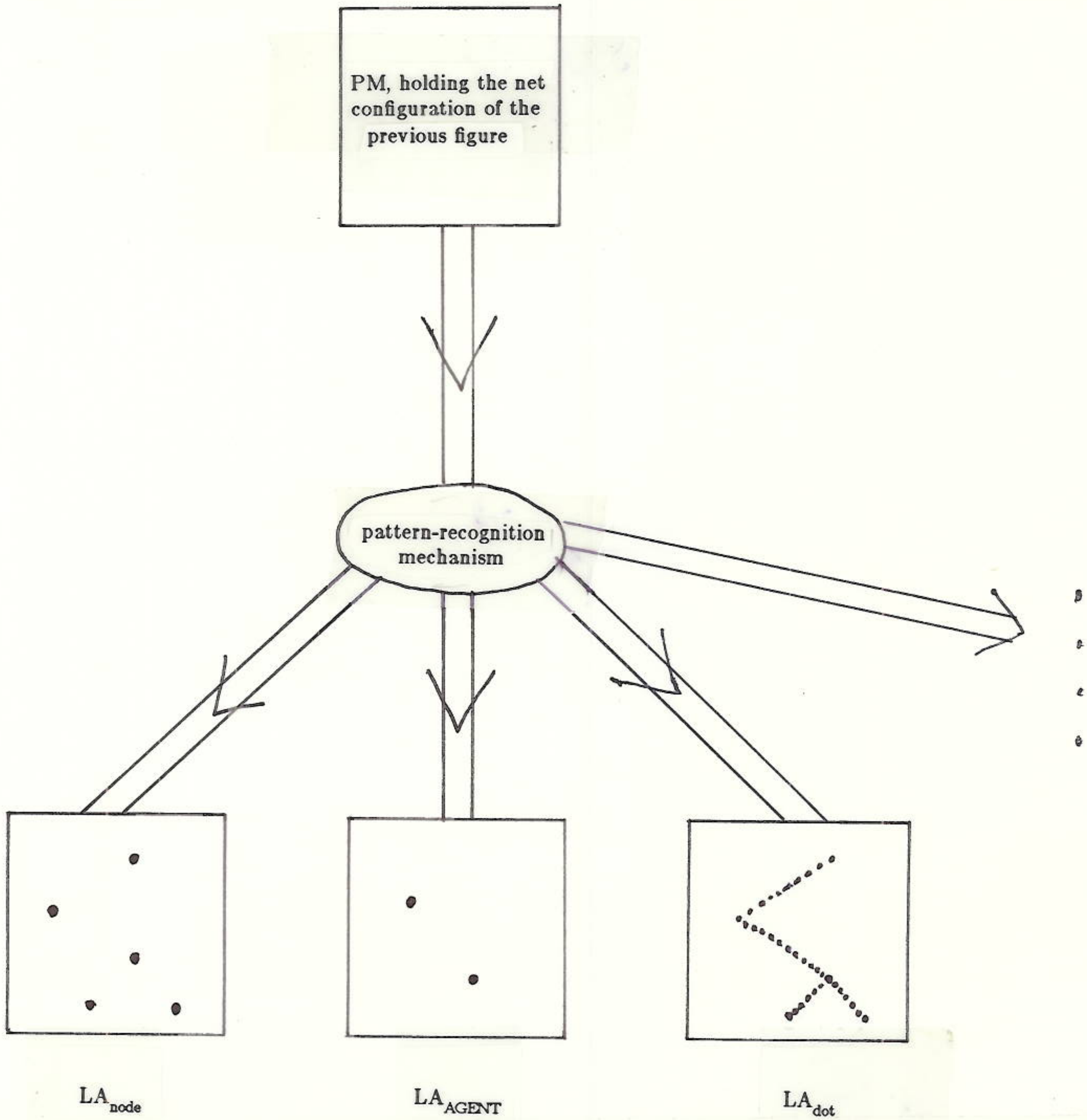


Fig. 11 The function of the pattern-recognition mechanism. LA_{node} is the node-pattern location array for the network configuration in Fig. 10. LA_{AGENT} and LA_{dot} are the AGENT-pattern and dot-pattern location arrays for that network configuration. Dots in this array boxes in the figure indicate the 1s. All the other values in the arrays are zero. Not shown are the location arrays for the node labels and for the link labels other than AGENT.

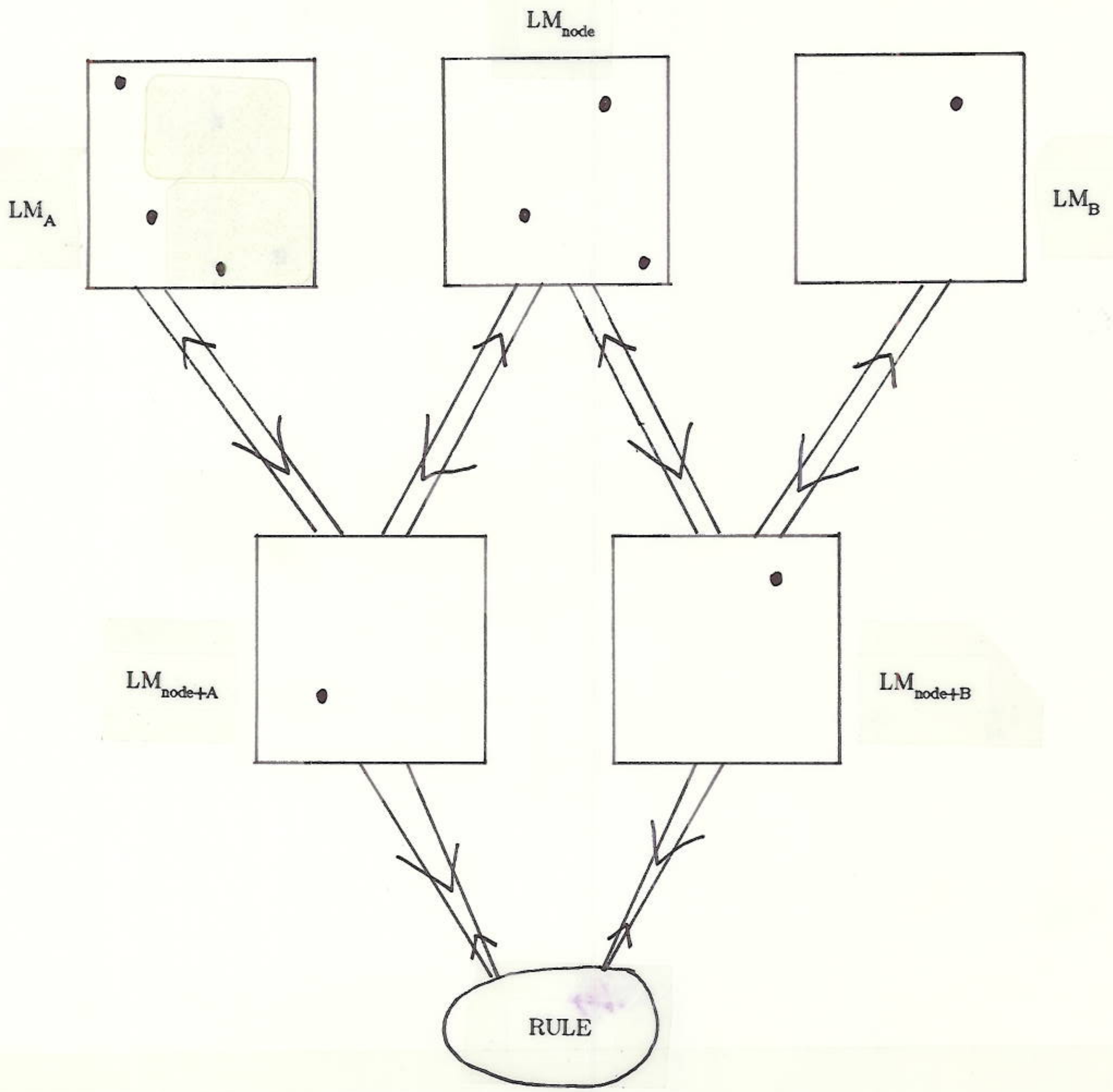


Fig. 12. Sketch of mechanism needed for a rule triggered by the presence in the PM of two nodes one of which is labelled with label A and the other of which is labelled with label B. (We take a single node labelled with both A and B to be a valid trigger.) LM_{node+A} and LM_{node+B} are like location matrices, but specify instances of composite patterns. LM_{node+A} has high activity at position (x,y) when LM_{node} and LM_A have high activity at or near position (x,y). (Similarly for B.) The rule fires (i.e. RULE starts some processing) when LM_{node+A} and LM_{node+B} both contain at least one element with high activity. Highlighting (see later in text) can be performed by RULE sending output to LM_{node+A} and LM_{node+B}. This output causes any active elements in those matrices to be highlighted. There are outputs from LM_{node+A} and LM_{node+B} to LM_{node} (and other LMs). The purpose of these outputs is that the highlighting at a position (x,y) in LM_{node+A} or LM_{node+B} may cause highlighting of elements at or near (x,y) in LM_{node}. The pattern-recognition mechanism is such that highlighting of an element in LM_{node} can be used to highlight the node centred at (x,y) in the PM.

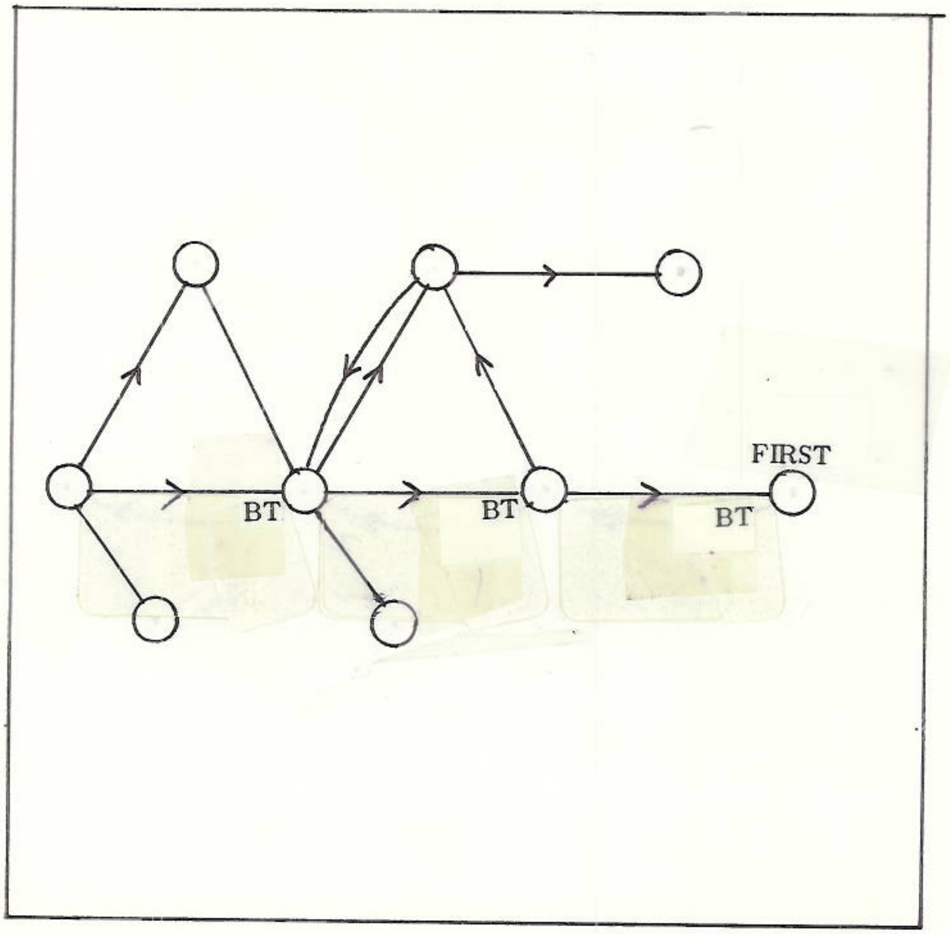


Fig. 13. Implementation of a backtrack list in a network-configuration traversal. The network is the same as the one in Fig. 8(a). No extra nodes or links are used. The list is implemented by instances of a special mark, BT, on *links*. A link is BT-marked when it is followed (in either direction) during traversal. A backtrack step causes one BT mark to be removed. The front node of the list is indicated by being marked with a special label, FIRST. Recall that we toyed with an analogous implementation in the case of connectionist theories.