January 1990

From Simple Associations to Systemic Reasoning: A Connectionist Representation of Rules, Variables and Dynamic Bindings

Lokendra Shastri
University of Pennsylvania

Venkat Ajjanagadde
University of Pennsylvania

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-90-05.

From Simple Associations to Systemic Reasoning: A Connectionist Representation of Rules, Variables and Dynamic Bindings

Abstract

Human agents draw a variety of inferences effortlessly, spontaneously, and with remarkable efficiency - as though these inferences are a *reflex* response of their cognitive apparatus. The work presented in this paper is a step toward a computational account of this remarkable reasoning ability. We describe how a connectionist system made up of simple and slow neuron-like elements can encode millions of facts and rules involving *n*-ary predicates and variables, and yet perform a variety of inferences within *hundreds of milliseconds*. We observe that an efficient reasoning system must represent and propagate, dynamically, a large number of variable bindings. The proposed system does so by propagating *rhythmic* patterns of activity wherein dynamic bindings are represented as the *in-phase*, i.e., synchronous, firing of appropriate nodes. The mechanisms for representing and propagating dynamic bindings are biologically plausible. Neurophysiological evidence suggests that similar mechanisms may in fact be used by the brain to represent and process sensorimotor information.

Comments

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-90-05.

FROM SIMPLE ASSOCIATIONS TO SYSTEMATIC REASONING: A CONNECTIONIST REPRESENTATION OF RULES, VARIABLES AND DYNAMIC BINDINGS

Lokendra Shastri
Venkat Ajjanagadde
MS-CIS-90-05
LINC LAB 162

Department of Computer and Information Science
School of Engineering and Applied Science
University of Pennsylvania
Philadelphia, PA 19104

January 1990

ACKNOWLEDGEMENTS:
This work was supported by NSF grants IRI 88-054465, MCS-8219196-CER, MCS-83-05211, DARPA grants N00014-85-K-0018 and N00014-85-K-0807, and ARO grant ARO-DAA29-84-9-0027.

From simple associations to systematic reasoning: A connectionist representation of rules, variables and dynamic bindings*

Lokendra Shastri and Venkat Ajjanagadde
Computer and Information Science Department
University of Pennsylvania
Philadelphia, PA 19104

Technical report: MS-CIS-90-05
LINC Lab 162

December 1989

Abstract

Human agents draw a variety of inferences effortlessly, spontaneously, and with remarkable efficiency — as though these inferences are a *reflex* response of their cognitive apparatus. The work presented in this paper is a step toward a computational account of this remarkable reasoning ability. We describe how a connectionist system made up of simple and slow neuron-like elements can encode millions of facts and rules involving n-ary predicates and variables, and yet perform a variety of inferences within *hundreds of milliseconds*. We observe that an efficient reasoning system must represent and propagate, dynamically, a large number of variable bindings. The proposed system does so by propagating *rhythmic* patterns of activity wherein dynamic bindings are represented as the *in-phase*, i.e., synchronous, firing of appropriate nodes. The mechanisms for representing and propagating dynamic bindings are biologically plausible. Neurophysiological evidence suggests that similar mechanisms may in fact be used by the brain to represent and process sensorimotor information.

*This work was supported by NSF grants IRI 88-05465, MCS-8219196-CER, MCS-83-05211, DARPA grants N00014-85-K-0018 and N00014-85-K-0807, and ARO grant ARO-DAA29-84-9-0027.

1 Introduction

The ability to represent and compute with a large body of structured knowledge in an *effective* and *systematic* way is a central characteristic of cognition. The richness and speed of human reasoning ability is illustrated by the following example derived from (Schubert 1989). Consider a person reading a variation of the Little Red Riding Hood (LRRH) story in which the wolf intends to eat LRRH in the woods. The reader is at the point in the story where the wolf, who has followed LRRH into the woods, is about to attack her. The next sentence reads: "The wolf heard some wood cutters nearby and so he decided to wait." It seems reasonable to claim that the reader will understand this sentence spontaneously and without deliberate thought. However, a careful analysis of this sentence makes it apparent that even though the reader does not become aware of it, understanding this sentence requires a fairly elaborate chain of reasoning. This reasoning may informally be described as follows:

> To eat LRRH the wolf will have to approach her (because to eat something you have to be near it), if the wolf approaches LRRH she will scream (because a child is scared by an approaching wild animal), if LRRH screams, the wood cutters will hear her (because a loud noise can be heard at a distance and screaming generates a loud noise), if the wood cutters hear the scream they will know that a child is in danger (because a child's screaming suggests that the child is in danger) the wood cutters will come to the location of the scream (because people want to protect children in danger and in part, this involves determining the source of the danger), when the wood cutters see the wolf they will try to prevent it from attacking LRRH (because people want to protect children), and in doing so the wood cutters may hurt the wolf (preventing an animal from attacking a child may involve physical force ...), so the wolf will decide to wait (the wolf does not want to get hurt).

Clearly, in addition to accessing meanings of lexical items, parsing, and resolving pronominal reference, some computation equivalent to the above chain of reasoning must be occurring spontaneously when we process the sentence in question.[1] The above example illustrates that humans are extremely good at drawing a broad class of complex inferences with remarkable efficiency — within hundreds of milliseconds. Such inferences are drawn effortlessly, spontaneously, and without any conscious intervention — as though they are a *reflex* response of the agent's cognitive apparatus. Let us label such spontaneous reasoning as *reflexive* reasoning (Shastri 1990a). Any serious attempt at understanding intelligence must provide a detailed computational account of how such inferences may be drawn with the requisite efficiency.

In evaluating the remarkable efficiency of reflexive reasoning it must be borne in mind that such reasoning takes place with reference to an immense body of common sense knowledge. A detailed and exhaustive compilation of such knowledge in terms of 'rules' and 'facts' will easily result in a knowledge base containing millions (perhaps even more) of rules and facts. At the same time, the speed of human performance suggests that reflexive reasoning is computed in time that depends only on the *length* of the chain of inference and is independent of the *size* of the knowledge base. This implies that the process of reflexive reasoning is highly parallel at the knowledge level and involves pursuing a large number of *potentially relevant* inferential paths in parallel.

Complexity theory rules out the existence of a general purpose inference system that can derive all valid inferences in time proportional to the length of the chain derivation — such a system would be to the theory of computation what a perpetual motion machine is to thermodynamics. But then cognitive agents are not general purpose reasoners. They can only perform a limited — though fairly broad — class of inferences with extreme efficiency. This suggests that an appropriate research strategy would be to develop *limited inference systems* — systems that can perform a limited but interesting class of inference with requisite efficiency. The work reported in this paper is the result of pursuing such a research strategy.

1.1 A Connectionist Metaphor of Reasoning

With nearly 10^{12} computing elements and 10^{15} interconnections, the brain's capacity for encoding, communicating, and processing information is truly awesome. But if the brain is extremely powerful it is also extremely limited: the relative simplicity of individual processors, and the restriction on the complexity of individual messages also impose strong constraints on the nature of representations and processes that operate on them. Connectionist models (Feldman & Ballard 1982; Rumelhart & McClelland 1986) intend to emulate the information processing of the animal brain — albeit at an abstract computational level — and hence, reflect the same strengths and weaknesses.

The fine grained and massive parallelism supported by connectionism permits one to assign *a node to each unit of information.*[2] This has the following interesting consequence: Assume that besides enumerating facts about the world, we also identify the important *inferential dependencies* between these facts and encode these dependencies by explicit links between the appropriate nodes. Then we can view inference as *parallel spreading of activation* in a connectionist network. This connectionist reasoning metaphor has tremendous appeal because it suggests an extremely efficient way of performing inference. This metaphor can be attributed to McCullough and Pitts (1988) who recognized the ability of neural networks to encode propositional logic and to Quillian and Fahlman (Quillian 1968; Fahlman 1979) who applied the (related) spreading activation metaphor to the domain of conceptual knowledge. The connectionist semantic network (CSN) described in (Shastri 1988b) and the DICIFIL system for reasoning about adjective-noun combinations reported in (Weber 1989) also follow such an approach and perform quite sophisticated inferences with requisite efficiency. The expressive power of CSN and DICIFIL systems extends beyond propositional logic but is still limited to unary predicates and binary predicates with only one quantified variable.[3]

In order to perform general and flexible reasoning, however, a representation system must be expressive enough to encode *systematic* and *abstract* knowledge about structured and composite entities, i.e., it must be expressive enough to encode rules with *n-ary* predicates and variables. The ability of connectionist models to do so, however, is often questioned. It is argued that as connectionist nodes have extremely limited processing power and as they can only exchange *scalar* levels of activation, connectionist networks are only capable of encoding associations and simple inferential dependencies between minimally structured objects, but *not* complex inferential dependencies between structured and composite objects.

Let us try and examine the alleged limitations of the connectionist model. Consider a connectionist representation of the fact 'John gave Mary Book1'. This fact constitutes a piece of structured knowledge: it does not merely express an association between the constituents 'John', 'Mary', and 'Book1' but rather, it

expresses a specific relation between these constituents wherein each constituent plays a distinct role. Thus one cannot represent 'John gave Mary Book1' by simply activating the representations of the constituents 'John', 'Mary', and 'Book1'. In general, the connectionist encoding and the rules of spreading activation should be chosen carefully if the representation of 'John gave Mary Book1' is not to be confused with the representation of 'Mary gave John Book1' or 'Book1 gave John Mary'. For related reasons, unless one is careful, the simultaneous representation of the facts 'John gave Mary Book1' and 'Susan bought Car7' may result in the representation of the ghost fact 'Susan gave John Car7'! These problems are essentially problems of *cross-talk*.

A connectionist mechanism for representing structured objects without cross-talk is illustrated in Fig. 1 (cf. Shastri 1988b). Each triangular *binder* node binds the appropriate filler to the appropriate argument and the *focal* node *give*23 provides the requisite grouping between the set of bindings that constitute a fact. The binder nodes become active on receiving two inputs and thus serve to retrieve the correct filler given a fact and an argument (and vice-versa). Such an encoding using *dedicated* nodes and links is suitable for representing stable long-term knowledge; the required focal and binder nodes may be recruited over time in order to represent new — but stable — grouping of constituents. Such a scheme, however, is inadequate if such groupings have to be created *dynamically*.

As we discuss in Section 2, a connectionist reasoning system must not only be capable of representing structured knowledge but it must also be capable of operating on such representations in a systematic manner to dynamically *generate* representations of (new) structured objects. This amounts to solving a complex form of the cross-talk problem, namely, the *variable binding* problem (Feldman 1982; Smolensky 1987).

Perhaps prompted by the above concerns McCarthy has observed that connectionist systems suffer from "the unary or even propositional fixation" and the representational power of most connectionist systems is restricted to unary predicates applied to a fixed object (see McCarthy's commentary in (Smolensky 1988)). Fodor and Pylyshyn (1988) have also argued at length that the connectionist formulation is inadequate for representing structured knowledge and cannot embody systematicity and compositionality.

1.2 A preview

The research described in this paper addresses the question of effective reasoning and the representational adequacy of connectionism. We discuss the variable binding problem and describe a solution to this problem using very simple *phase-sensitive* binary threshold units. The solution involves maintaining and propagating variable bindings using temporally synchronous — i.e., in-phase — firing of appropriate nodes. The resulting system can maintain and propagate a large number of bindings *simultaneously* as long as the number of *distinct* individuals participating in the bindings during any given episode of reasoning, remains bounded. A psychologically as well as biologically plausible value of this bound is about seven or eight (see Sections 3.2 and 3.4).

Next we show that the solution to the variable binding problem leads to the formulation of a connectionist system that can represent structured knowledge expressed in terms of *facts* and *rules* involving *n-ary relations* and *variables* and perform a broad class of reasoning based on this knowledge with extreme efficiency. The time taken by the proposed system to draw an inference is only proportional to the *length* of the chain of

inference and is independent of the number of rules and facts encoded by the system. Thus we demonstrate that a system made of slow and simple computing elements with switching times and periods of oscillations in the msec. range can encode millions of facts and rules and yet perform inferences involving chains of reasoning in the hundred msec. range. The system's ability to do so is a direct consequence of the fact that the system can support the *simultaneous* application of a large number of rules. The proposed system also makes efficient use of nodes and links — the number of nodes and links required to encode knowledge is only *proportional* to the number of rules and facts encoded in the network's long-term memory.

The view of information processing implied by the proposed system is one where reasoning is the transient but systematic propagation of a *rhythmic* pattern of activation, where each *phase* in the rhythmic pattern corresponds to an object in the dynamic or short-term memory, where bindings are represented as the *in-phase* or *synchronous* firing of appropriate nodes, where long-term facts are subnetworks that act as temporal pattern matchers, and where rules are interconnection patterns that cause the propagation and transformation of rhythmic patterns of activation.

As we shall see, the rule-governed reasoning system does not embody any *explicit* inference mechanism. It does not manipulate or rewrite symbols in accordance with logical rules such as modus-ponens and universal instantiation. It is a simple physical device made up of simple nodes that just propagate binary activation levels without the intervention of a central controller, but inferences are the spontaneous and natural outcome of the system's behavior.

The binding mechanism and the resulting reasoning system is biologically plausible in that it strictly adheres to the core features of the connectionist model (these features are discussed in Section 3.1). At the same time, there exists neurophysiological evidence to suggest that the basic mechanisms used here for representing and propagating dynamic bindings, namely, the propagation of rhythmic patterns of activity and the synchronous activation of nodes, play a role in the representation and processing of information at the sensorimotor level (Freeman 1981; Gray & Singer 1989; Churchland 1986). Of course we cannot, and do not, claim that the brain encodes rules, facts, and dynamic bindings according to the scheme outlined in this paper but we think that it is significant that mechanisms implicated in low-level sensorimotor tasks seem sufficient to model high-level cognitive functions such as systematic reasoning.

2 Reasoning and the dynamic binding problem

In the previous section we observed that it may be possible to view inference as parallel spreading of activation in a connectionist network provided one could solve the problem of representing structured objects and performing systematic operations on such representations. We also mentioned that this gives rise to the variable binding problem. We examine this problem in greater detail below with the help of an example.[4]

Assume that our reasoning system encodes the following general knowledge about the world:

- If an agent gives a recipient an object then the recipient comes to own that object.
- Owners can sell what they own.

Given the above general knowledge, a reasoning system should be capable of inferring 'Mary owns Book1' and 'Mary can sell Book1' on being given the *additional* information that 'John gave Mary Book1'. Let us express the system's general knowledge as first-order *rules*:

- $\forall x, y, z\ [give(x, y, z) \Rightarrow own(y, z)]$
- $\forall x, y\ [own(x, y) \Rightarrow can\text{-}sell(x, y)]$

and let us express a *fact* such as 'John gave Mary Book1' as the atomic formula

- $give(John, Mary, Book1)$.

Before proceeding, two comments may be in order. First, we are assuming that knowledge about the world can be captured by 'hard' logical rules. The assumption that rules are 'hard' rules is not critical to our work at all. As we point out in Sections 6 and 8.4, the proposed mechanisms for encoding knowledge and reasoning with it can be generalized to 'soft' probabilistic/evidential rules. What *is* critical however, is the assumption that an agent's internal representation of the world embodies the regularities, dependencies, and causal relationships in the world, and such knowledge is not only *systematic* but also *abstract* and instantiation-independent. Hence, the use of rules, variables, and quantification.

Second, the rules and facts used in this example and elsewhere in this paper are only meant to illustrate the problem of dealing with systematic and abstract knowledge, and are not intended to be a detailed characterization of common sense knowledge. For example, the rule relating *giving* and *owning* is a gross oversimplification and does not capture the richness and complexity of the actual notions of *giving* and *owning*.

In the above rules and facts, predicates such as *give*, *own* and *can-sell* may be viewed as collections of ***arguments*** (or roles). For example, the predicate *give* may be viewed as the collection of arguments $(giver, recipient, give\text{-}object)$ while the predicate *own* may be viewed as the collection of arguments $(owner, own\text{-}object)$. A 'fact' is essentially an *instance* of a predicate and hence, may be viewed as a collection of ***argument bindings***. Thus, the fact $give(John, Mary, Book1)$ which is a particular instance of the predicate *give*, may be viewed as the collection of argument bindings: $(giver = John, recipient = Mary, give\text{-}object = Book1)$. Observe that instead of 'variables' and 'variable bindings' we will be referring to 'arguments' and 'argument bindings'.

If we view predicates as collections of arguments and facts as collections of argument bindings, it becomes apparent that a connectionist system that represents and reasons with structured knowledge must be capable of representing argument bindings in a rapid and dynamic fashion. First, a representation and reasoning system must interact with other (external) processes that communicate facts and pose queries to it. Therefore such a system must be capable of establishing argument bindings dynamically to represent such facts and queries. For example, our reasoning system should be capable of establishing the argument bindings: $(giver = John, recipient = Mary, give\text{-}object = Book1)$ when it is 'told' that 'John gave Mary Book1'. Second, the inference process dynamically creates *inferred* facts, and the system must be capable of representing argument bindings that constitute such inferred facts. For example, our reasoning system must represent the bindings $(owner = Mary, own\text{-}object = Book1)$ when the system infers $own(Mary, Book1)$.

In addition to representing argument bindings in a dynamic and rapid fashion, a reasoning system should also be capable of creating new argument bindings by ***systematically propagating existing bindings in accordance with the dependencies between predicate arguments mandated by the rules.***

A rule may be viewed as specifying an antecedent predicate, a consequent predicate, and a *correspondence* between the arguments of these predicates. Thus the rule $\forall x, y, z\ [(x, y, z)give(x, y, z) \Rightarrow own(y, z)]$ may be viewed as specifying the following correspondence between the arguments of the predicates *give* and *own*: The ***recipient*** of a ***give*** event maps to the ***owner*** of an ***own*** event, the ***give-object*** of a ***give*** event maps to the ***own-object*** of an ***own*** event, while the ***giver*** of a ***give*** event does not play any role in the resulting *own* event.

A step of inference or 'rule application' may therefore be viewed as taking an instance of the antecedent predicate and creating — dynamically — an instance of the consequent predicate, with the argument bindings of the latter being determined by applying the argument correspondence specified in the rule to the argument bindings of the former. Such a dynamically created instance of the consequent predicate corresponds to an inferred fact. For example, the application of the rule $\forall x, y, z\ [(x, y, z)give(x, y, z) \Rightarrow own(y, z)]$ in conjunction with an instance of the antecedent predicate *give*, namely, $give(John, Mary, Book1)$ creates a new instance of the consequent predicate *own* with the bindings $(owner = Mary, own\text{-}object = Book1)$ which constitute the inferred fact $own(Mary, Book1)$.

The system should also be capable of chaining inference steps, i.e., using an inferred fact in conjunction with other rules to create other inferred facts and so on. For example, the system should be capable of using the inferred fact $own(Mary, Book1)$ in conjunction with the rule $\forall x, y\ [own(x, y) \Rightarrow can\text{-}sell(x, y)]$ to create the inferred fact $can\text{-}sell(Mary, Book1)$.

The above discussion suggests that a connectionist system for representing and reasoning with structured knowledge must embody:

1. A mechanism for representing dynamic argument bindings.
2. A mechanism for creating new bindings by propagating argument bindings in accordance with the argument correspondences specified in the rules.

In the rest of the paper we will often refer to argument bindings simply as bindings and to argument bindings that must be represented in a rapid and dynamic fashion as dynamic bindings.

3 Solving the dynamic binding problem

The proposed solution to the problem of representing and propagating dynamic bindings involves several ideas that complement each other. These include i) representing an object during an episode of reasoning as a *phase* or time-slice within an oscillatory pattern of activity, ii) using temporally coherent firing of appropriate nodes to represent dynamic bindings, iii) using an object-centered representation of bindings rather than an argument-centered one, iv) distinct representation of each predicate and its arguments, and v) representation of rules via explicit representation of the inferential dependencies between predicates and the correspondence between predicate arguments. Before describing the solution and elaborating on these ideas we would like

to emphasize the basic computational features of connectionism and reiterate the constraints imposed on the solution by the task it is intended to serve, namely, reflexive reasoning. The constraints imposed by the connectionist architecture and reflexive inference preclude a number of solutions to the dynamic binding problem and provide criterion for evaluating alternate solutions.

3.1 Core computational features of connectionism

We enumerate below what we consider to be the core computational features of connectionism. This list is not intended to be a definition of connectionism but rather an enumeration of those features that capture the essence of the computational model underlying connectionism and lead to its biological plausibility (Feldman & Ballard 1982; Rumelhart & McClelland 1986). In addition to being neurally motivated these features also lead to an optimal use of massive parallelism (for a discussion see (Shastri 1990a; Shastri 1990b)).

- Processing elements (units) compute very simple functions of their inputs.
- Units are extremely limited in their ability to hold state information.
- Units output only levels of activation, i.e., a unit's output is only a *scalar* quantity with limited precision. Thus unit outputs cannot encode information such as symbol names, pointers, or addresses.
- Synaptic modifications cannot occur within tens of milliseconds.[5]
- There is no central controller that drives the network operation by instructing individual nodes to perform specific operations.

3.2 Some aspects of the dynamic binding problem

Some important aspects of the dynamic binding problem are summarized below:

1. Reflexive reasoning takes place with reference to an immense body of knowledge and the reasoning system must encode a large number, perhaps millions, of richly interconnected rules. At the same time, the speed of reflexive reasoning suggests that such inferences are drawn in time that depends only on the length of the chain of inference and is independent of the total number of rules in the system. This implies that the reasoning process must be highly parallel and must pursue a large number of inferential paths in parallel. The proposed mechanisms should therefore be capable of representing and propagating a large number of dynamic bindings simultaneously.

2. It is reasonable to assume that most cognitive tasks that are performed with extreme efficiency such as language understanding (e.g., dialog and story understanding) tend to involve only a small number of distinct entities[6] at a time. During normal discourse we have to deal with perhaps three or four distinct entities at a time, but a reasonable estimate of the maximum number of distinct entities we can deal with at a time is around seven (Miller 1956). Of course a large number – hundreds or perhaps thousands – of automatic inferences involving these entities may get drawn giving rise to a comparable number of bindings.

3. During an episode of reasoning, dynamic bindings must be created to represent inferred facts. Almost all these facts constitute intermediate results, and hence, dynamic bindings need only be represented temporarily for the duration of the reasoning process.[7]

4. The firing of a rule creates a new set of bindings, therefore the time taken to represent and/or propagate a set of bindings should be no more than the time it takes to apply a rule. We believe that human agents can perform reflexive inferences – even when they involve a chain of rule applications – within hundreds of milliseconds. This suggests that the time it takes for a rule to fire is of the order of tens of milliseconds. Consequently, the binding process should be capable of creating new bindings within tens of milliseconds.

3.3 Representing dynamic bindings

In this work we use the temporal structure of activation patterns to represent dynamic bindings. The use of the temporal dimension extends the basic connectionist framework in a very natural manner and provides it with the requisite power for representing and reasoning with complex structured information. The significance of temporally organized activity in neural nets has been recognized by several researchers such as Hebb (1949), von der Malsburg (1986), and Abeles (1982). In Section 5.3 we relate these explicitly neural models to the system proposed in this paper. The use of time to encode conceptual information also appears in the work of Clossman (1988) and Fanty (1988). Like the proposed system, Clossman's *broadcast net* also uses phase information to represent bindings. Clossman's system, however, cannot *apply* a rule and create new bindings and therefore, cannot perform inference. The solution to the dynamic binding problem presented here supports inference: in addition to representing bindings, it also propagates dynamic bindings along chains of rule applications to create new bindings (i.e., inferred facts).

Several other solutions to the binding problem have been suggested such as the use of dynamic connections (Feldman 1982), parallel constraint satisfaction (Touretzky & Hinton 1988), position specific encoding (Barnden 1989), tensor product representations (Dolan & Dyer 1988), and signatures (Lange & Dyer 1989). In Section 9 we discuss these alternatives and compare them with the temporal approach.

Let us refer to the schematic representation of some predicates and individual concepts shown in Fig. 2. A few comments about the figure before we proceed: i) Distinct predicates and their arguments are represented using distinct nodes. ii) For convenience we will assume that each argument and predicate node in Fig. 2 corresponds to an individual connectionist node. Later in Section 5.2 we show that each node corresponds to an *ensemble* of nodes. This change does not alter the binding mechanism but makes it robust with respect to noise and node malfunction. iii) Nodes such as $John$ and $Mary$ correspond to $focal$ nodes of the complete connectionist representations of the individuals 'John' and 'Mary'. Information about the attribute values (features) of 'John' and its membership in categories/concepts is encoded by linking the focal node $John$ to appropriate nodes, and this information can be accessed via the focal node. Details of such an encoding may be found in (Shastri 1988b; Fanty 1988; and Weber 1989). As Feldman (1989) points out, focal nodes may also be replaced by a small ensemble of nodes.

The proposed system represents dynamic bindings using *synchronous* — i.e., *in-phase* — firing

of appropriate nodes. In particular, the dynamic binding between a predicate argument and its filler is represented by the synchronous (in-phase) firing of nodes that represent the argument and the filler.

With reference to the nodes in Fig. 2, the dynamic representation of the bindings ($giver = John, recipient = Mary, give\text{-}object = Book1$) (i.e., the fact $give(John, Mary, Book1)$) will be represented by *rhythmic* pattern of activity shown in Fig. 3. (At this point, the activity of the predicate nodes is not of significance and is not specified. Also, argument names have been abbreviated in the figure.) Observe that all active nodes are firing with the *same frequency* but each node is firing in a specific *phase* (or time-slice) within the recurrent pattern. The binding between an argument and a filler is represented by the *in-phase* firing of associated nodes. The particular phase occupied by a filler and an argument is not critical. Thus the firing pattern shown in Fig. 4 also represents the dynamic bindings ($giver = John, recipient = Mary, give\text{-}object = Book1$) .

As another example, consider the firing pattern shown in Fig. 5. This pattern of activation represents the single dynamic argument binding ($giver = John$) and corresponds to the partially instantiated fact $give(John, x, y)$ ('John gave someone something').

Fig. 6 shows the firing pattern of nodes corresponding to the dynamic representation of the bindings ($giver = John, recipient = Mary, give\text{-}object = Book1, owner = Mary, own\text{-}object = Book1, potential\text{-}seller = Mary, can\text{-}sell\text{-}object = Book1$). These bindings encode the facts $give(John, Mary, Book1)$, $own(Mary, Book1)$, and $can\text{-}sell(Mary, Book1)$. Observe that the (multiple) bindings between $Mary$ and the arguments *recipient*, *owner*, and *potential-seller* are represented by the corresponding argument nodes firing in-phase with $Mary$. Similarly, the bindings between $Book1$ and the arguments *give-object*, *own-object*, and *can-sell-object*, are represented by the appropriate argument nodes firing in-phase with $Book1$. But observe that the individual concepts $Mary$, $Book1$, and $John$ are firing *out of* phase and occupy distinct phases in the rhythmic pattern of activity. This highlights an extremely significant aspect of the proposed solution:

- the transient or short-term representation of an entity is simply a *phase* within a rhythmic pattern of activity!
- the representation of dynamic bindings is not argument-centered but *object*-centered (or filler-centered).
- the number of *distinct* phases required to represent a set of dynamic bindings only equals the number of *distinct* entities participating in the bindings and does not in any way depend upon the *total number* of bindings.

What is perhaps equally significant about the temporal representation of dynamic bindings is that it requires rather simple computational mechanisms. No central controller is required to instruct or control individual nodes — in fact, the solution does not even require a global clock.

3.4 A realization using phase-sensitive nodes

The solution to the dynamic binding problem may be realized as follows:

- When active, nodes representing arguments and individual concepts fire with a fixed frequency and hence, a fixed *period* of oscillation π.[8]
- The pulse width ω, with which a node fires occupies a small fraction of the period of oscillation π.

A biologically plausible value of π may be about 10 msecs. and as a pulse is analogous to a *spike*, a plausible value of ω may be 1 msec. We discuss the question of biological plausibility in greater detail in Section 5 but the following observations appear relevant at this point.

An ideal system built out of nodes with frequency and pulse-width values of 100 *cps* and 1 msec. respectively, would be capable of representing arbitrarily many dynamic bindings as long as the number of distinct individuals involved in these bindings does not exceed *ten*. Any physical system, however, would be susceptible to noise and drift in the values of π and ω and it is unlikely that all ten phases within a period π can be occupied without introducing substantial overlap and cross-talk in the firing pattern. Restricting the total number of individuals participating in dynamic bindings at any given time to about seven would allow for moderate variations in π and ω, and still ensure that the phases in which distinct individuals fire do not overlap (See Section 5.2). If this number increases beyond seven the system may exhibit some cross-talk among bindings. As this number approaches or exceeds ten, the cross-talk will substantially degrade the system performance.

As observed earlier, cognitive tasks that are performed with extreme efficiency such as dialog or story understanding tend to involve only a small number of individuals at a given time, and although a large number of bindings may be generated at any time, the number of distinct individuals participating in these bindings remains small (about seven). It is also believed that if the number of distinct individuals that must be kept in focus exceeds seven, our ability to maintain coherence degrades. Thus, the limitation of the binding mechanism resulting from a biologically plausible choice of parameters appear to be psychologically plausible.

3.5 Propagating bindings: Encoding rules and effecting rule application

Our solution to the problem of encoding rules and propagating bindings during rule application follows naturally from the idea that distinct predicates and arguments have distinct representations (refer to Fig. 2). In Section 2 we described how a step of inference or rule application may be viewed as taking an instance of the antecedent predicate and dynamically creating an instance of the consequent predicate, with the argument bindings of the latter being determined by the argument bindings of the former and the argument correspondence specified by the rule. Hence what is required for encoding a rule is a mechanism for propagating argument bindings from the antecedent predicate to the consequent predicate in accordance with the argument mapping specified in the rule. With reference to the predicates in Fig. 2, encoding the rules

- $\forall x, y, z\ [give(x, y, z) \Rightarrow own(y, z)]$
- $\forall x, y\ [own(x, y) \Rightarrow can\text{-}sell(x, y)]$

should have the following effect:

The initial state of activation described by the rhythmic activation pattern shown in Fig. 3 should eventually lead to the rhythmic activation pattern shown in Fig. 6.

The desired behavior can be captured quite naturally by connecting the arguments of the antecedent and consequent predicates to reflect the correspondence between arguments embodied in the rules. Thus to encode the rules given above, the arguments *recipient* and *give-object* of *give* should be connected to the arguments *owner* and *own-object* of *own*, respectively. And the arguments *owner* and *own-object* of *own* should be connected to the arguments *potential-seller* and *can-sell-object* of *can-sell*, respectively. If we also wish to encode the rule: $\forall x, y\ [buy(x, y) \Rightarrow own(x, y)]$ (i.e., if an agent buys an object then he comes to own that object), we can do so by connecting the arguments *buyer* and *buy-object* of *buy* to the arguments *owner* and *own-object* of *own*, respectively. This encoding is illustrated in Fig. 7.

As before we assume that nodes encoding arguments and individual concepts fire with a fixed period of oscillation π and a fixed pulse width ω. For the present, let us also assume that the switching times of nodes equal π and signal propagation times along links are zero (the last assumption is not required and is relaxed in Section 5.1). In view of the above node behavior it is easy to see that the initial state of activation shown in Fig. 8 will lead to the state of activation shown in Fig. 9 after just one period of oscillation, and thereafter, to the state of activation shown in Fig. 10 after one more period of oscillation. No additional machinery is required for this to happen as long as argument nodes respond to inputs in a phase-sensitive manner. For a typical value of π, namely, 10 msecs, the pattern shown in Fig. 10 will result within 20 msecs. after the system state is initialized to the pattern shown in Fig. 8. Furthermore, the time required to propagate bindings is independent of the number of rules encoded in the system and the same response time will result even if millions of rules are encoded in the system.

Conceptually, the proposed encoding of rules amounts to creating a directed *inferential dependency* graph: Each predicate argument is represented by a node in this graph and each rule is represented by links from nodes denoting the arguments of the antecedent predicate to nodes denoting the arguments of the corresponding consequent predicate. In terms of this conceptualization it should be easy to see that the rule application process corresponds to a *parallel* breadth-first traversal of a directed graph and the encoding allows any number of rules to apply in parallel. It also follows that the time taken to generate a chain of inference will be independent of the total number of rules and facts and will just be equal to $l * \pi$ where l equals the *length* of the chain of inference and π equals the period of oscillation of the nodes used in the connectionist realization. Later is Sections 7.7 and 7.9 we show that this observation generalizes to rules with multiple predicates in the antecedent.

A distinct representation of each predicate and predicate argument together with an explicit encoding of the inferential dependency between predicate arguments seems essential if the system is expected to apply multiple rules in parallel and represent the large number of dynamic bindings that may result from these rule applications. In particular, the need for knowledge-level parallelism rules out purely distributed representations where predicates and arguments are represented as patterns over *common* ensemble of nodes (Hinton 1981; Touretzky & Hinton 1988).

3.6 Representing and propagating dynamic bindings — a summary

Given the representation of dynamic bindings and the encoding of rules described in the preceding sections, one may view i) reasoning as the transient but systematic propagation of a rhythmic pattern of activation, ii) an object in the dynamic memory as a phase in the above rhythmic activity, iii) bindings as the in-phase firing of argument and filler nodes, and iv) rules as interconnection patterns that cause the propagation and transformation of such rhythmic patterns of activation. During an episode of reasoning, all the arguments filled by constants currently in focus become active in the same phase as the concept thereby creating transient 'temporal frames' of knowledge grouped together by temporal synchrony. This can be contrasted with 'static' frames in a frame-based language that group together — spatially — knowledge about property values of concepts.

Inferences are the spontaneous and natural outcome of the system's behavior. The system does not embody any *explicit* inference mechanism. It does not manipulate or rewrite symbols in accordance with logical rules such as modus-ponens and universal instantiation. It is a simple physical device made up of simple nodes that just propagate binary activation levels, but in doing so cannot but infer what follows from a given state of affairs.

The representation of dynamic bindings as temporally synchronous patterns of activity allows a very large number of bindings to be represented simultaneously. This number is only bounded by the total number of arguments in the system. Similarly, the total number of dynamic facts that may be represented simultaneously is only bounded by the total number of predicates in the system. The relevant 'unit' of time here is π, i.e., about 10 msecs. Thus when we say 'simultaneous' we mean 'during a time interval π'.

The mechanism for propagating dynamic bindings also allows a very large number of rules to fire (apply) simultaneously, this number in only bounded by the total number of rules encoded in the system.

The representation of dynamic bindings does not require hard-wired connections between all possible argument filler pairs. This is extremely desirable, if not essential. In Section 1 we stated that the number of rules required to capture common sense knowledge may run into the millions. We also believe that the number of concepts, and hence, the number of potential argument fillers is also in the hundred thousand range.[9] Thus a binding scheme that depends on the existence of connections will require an unnecessarily large number of connections.

Although synchronous activity is central to the representation and propagation of binding, the system *does not require a global clock*. The propagation of in-phase activity occurs automatically, once an *initial* set of dynamic bindings is communicated to the system. For example, once the dynamic fact $give(John, Mary, Book1)$ is communicated to the system by forcing the nodes $John$, $Mary$, $Book1$, *giver*, *recipient*, and *give-object* to fire as shown in Fig. 8, the system automatically propagates — without the help of any other control/clock signals — the bindings corresponding to all the inferred facts and results in the pattern of activation shown in Fig. 10.

To simplify the presentation, all examples discussed thus far involved rules with a single predicate in the antecedent. This is not a limitation of the system and it can encode and reason with rules that have multiple predicates in the antecedent. Sections 7.7 and 7.9 specify how such rules are encoded.

A limitation of the proposed mechanism is that the number of distinct individuals involved in simultaneous

dynamic bindings is bounded by the ratio π/ω. As explained in Section 3.4, if π and ω equal 10 and 1 msecs., respectively, the number of distinct individuals that may participate in simultaneous dynamic bindings is bounded by 10. A more realistic estimate — taking noise and frequency drift into account — may be around 7. We do not view this limitation as a weakness of the system because we believe that not only reflexive reasoning but almost any form of common sense reasoning that does not make use of props is also subject to a similar limitation.

Another limitation of the binding mechanism as described above is that the system cannot represent two dynamic facts pertaining to the same predicate simultaneously. This limitation only applies to *dynamic* facts. In Section 4 we describe how *any* number of long-term facts about the *same* predicate may be encoded simultaneously. In Section 8.1 we outline a scheme that generalizes the use of temporal synchronization to represent several dynamic facts about the same predicate simultaneously.

4 Encoding long-term facts: Memory as a temporal pattern matcher

In addition to representing dynamic facts, a system for representing and reasoning with rules and facts must also represent *long-term* facts, i.e., facts that are stored relatively permanently in its memory. But what should be the nature of representation of long-term facts and what relation should it bear to the dynamic representation of facts?

The encoding of a long-term fact should allow the *detection* of situations that correspond to, or follow immediately from, the encoded fact. In other words, the encoding should allow the system to directly answer all questions whose answers depend on the fact alone and do not require the application of any rules (i.e., general domain knowledge). For example, the encoding of the long-term fact $give(John, Mary, Book1)$ should recognize situations such as: $give(John, Mary, Book1)$, $give(John, Mary, x)$ (i.e., John gave Mary something), $give(x, Mary, y)$, and $give(x, y, z)$ (i.e., Someone gave something to someone). However it should not recognize situations such as: $give(Mary, John, Book1)$, $give(John, Susan, x)$, or $own(Mary, Book1)$. Notice that the last fact cannot be recognized directly from the given fact even though it can be inferred from it via a rule application. Similarly, the encoding of the long-term fact: $give(John, Mary, x)$ should recognize situations such as $give(John, Mary, x)$, $give(x, Mary, y)$, and $give(x, y, z)$, but not $give(Mary, John, x)$, or $give(John, Mary, Book1)$ (this last situation specifies $Book1$ as the *give-object*, something not supported by the given fact).

In the context of the proposed system, a 'situation' essentially corresponds to the set of dynamic bindings represented in the system's state of activation. Therefore the encoding of a long-term fact should be capable of detecting whether the (static) bindings that constitute the fact, match the dynamic bindings represented in the system's state of activation. Given that the system represents dynamic bindings as temporal patterns, the representation of a long-term fact should behave like a *temporal pattern matcher*.

The design of such a temporal pattern matcher is illustrated in Figures 11 and 12. These figures encode the long-term facts $give(John, Mary, Book1)$ and $give(John, Mary, x)$, respectively. Although the complete rationale for choosing this particular encoding will become apparent in Section 7, the basic idea can be

conveyed at this point. The encoding of a long-term fact consists of a τ-and node (see below) that receives an input from the predicate associated with the fact. The input from the predicate is modified by inhibitory links (Kandel 1976) from the arguments of the predicate. If the fact specifies a filler for an argument then the inhibitory link from the argument is in turn modified by an inhibitory link from the filler. The τ-and node is a *temporal and* node, i.e., it becomes active if and only if it receives *uninterrupted* input for a duration equal to its period of oscillation. The output of the τ-and node constitutes the output of the long-term fact and this node is expected to become active if and only if the long-term fact matches the dynamic bindings represented in the network's state of activation.

It can be shown that the suggested encoding of a long-term fact behaves in the desired manner: First, assume that a predicate becomes active whenever any *query* or dynamic fact involving the predicate is represented in the system and generates a continuous output (how this happens will become apparent in Section 7). Now if a predicate argument is active, it will block the output of the predicate going into the τ-and node, *unless* the filler of this argument is also active *simultaneously*. But an argument and filler are active simultaneously if and only if they are bound. Therefore it follows that the τ-and node will receive uninterrupted input if and only if the dynamic bindings represented in the system's state of activation are such that

> Either an argument is unbound, or if it is bound, the argument filler in the dynamic binding matches the argument filler specified in the long-term fact.

The reader may wish to verify that the encodings given in Figures 11 and 12 will behave exactly as expected (see paragraph two of this section). The proposed encoding allows an arbitrary number of long-term facts involving the *same* predicate to be encoded in the system. It also allows any number of long-term facts, even if they involve the same predicate, to become active simultaneously.

5 Computational requirements and their biological plausibility

In this section we examine the computational requirements for representing and propagating dynamic bindings, encoding rules and long-term facts, and realizing rule-governed behavior. We also consider the problem of noise and show that the solutions suggested in the preceding sections generalize to a noise tolerant system in which arguments are represented as *ensembles* of nodes rather than single nodes. We also comment on the biological plausibility of the proposed solutions and relate them to some neural models of perception and cognition.

5.1 Computation requirements

The proposed representation and reasoning system can be realized using binary threshold units (BTUs) that are also sensitive to the temporal properties of their inputs. The critical property of these nodes is that

- in response to an oscillatory input these nodes respond with an oscillatory output and

- **the timing, i.e., the phase, of their output pulse is precisely governed by the timing or phase of their input**

In view of their temporal properties we refer to these nodes as *phase-sensitive* nodes. Phase-sensitive nodes may be differentiated based on the window of time over which they sample their inputs and the width of their output pulse. These differences give rise to several different types of phase-sensitive nodes — three of which are used in the proposed system and are discussed below:

- ρ-btu nodes: These nodes can fire with a stable period of oscillation, π, and a pulse width ω, where ω is only a fraction of π. Biologically motivated values of π and ω being 10 msecs. and 1 msec., respectively.

- τ-or nodes: These nodes are like ρ-btu nodes except that their firing pulse width is comparable to the period of oscillation π. For convenience we will assume that the pulse width of τ-or nodes is equal to π.

- τ-and nodes: These nodes are like τ-or nodes in that their firing pulse width is comparable to π, but unlike τ-or nodes, a τ-and node becomes active if and only if it receives *uninterrupted* input over a time interval equal to the period of oscillation π. Thus τ-and nodes may be viewed as temporal *and* nodes.

The temporal behavior of phase-sensitive nodes interacts with the thresholds of these nodes, and consequently, these nodes behave as temporal-spatial integrators. If we assume all link weights to be 1, the activation conditions of the above nodes may be described as follows: A ρ-btu node — and similarly, a τ-or node — with a threshold of k becomes active if it receives k simultaneous inputs (i.e., k inputs in-phase). A τ-and node with a threshold of k becomes active if it receives k simultaneous inputs throughout the time interval π. Unless stated otherwise, we will assume that ***all link weights and all thresholds equal 1.***[10]

In all illustrations, a ρ-btu node will be depicted as a circle, a τ-and node as a pointed pentagon, and a τ-or node as a triangle. Although plausible values of the period of oscillation (π) and the pulse width of ρ-btu nodes (ω) may be 10 msecs. and 1 msec., respectively, we have chosen π to be 6ω when illustrating patterns of activity. This is purely for making the depiction of such patterns easier.

In Section 3.5 we observed that we could obtain the desired in-phase firing of a chain of interconnected argument nodes by assuming that all signal propagation delays equal zero and all switching times equal π (by switching time we mean the time delay between the onset of input to a node and the onset of its output). Propagation delays include both, the conduction times for action potentials as well as synaptic delays, and although it may be reasonable to assume negligible conduction times, it would be unrealistic to assume zero synaptic delays.[11] Fortunately, it is not necessary to assume zero propagation delays — it suffices to assume that all links introduce the *same* propagation delay. It is straight forward to show that

> If the link delays equal λ, then the desired in-phase firing of nodes can be obtained by assuming that the switching time of nodes equals $\pi - \lambda$.

The use of the above modified switching time – adjusted for propagation delay — will ensure that interconnected argument nodes will fire in-phase during the propagation of dynamic bindings.

In a physical system the propagation delays may vary from link to link. Let λ_i — the delay along link l_i — be expressed as $\lambda_0 \pm \epsilon_i$ where λ_0 is a base delay and ϵ_i is a small link-specific deviation from λ_0. The proposed solution will be effective as long as the ϵ_i's, are small compared to ω. Consider the following example. Let ϵ_i's be limited to within $\pm 5\%$ of ω and let π equal 10ω. Assume that seven distinct individuals are involved in the bindings being propagated. From the above it follows that π will essentially consist of seven distinct phases, each of width ω, and hence, the phase separation will be about 0.4ω. Consequently, each phase can drift about 0.2ω before cross-talk becomes a significant problem. As each link introduces a drift equal to $\pm 0.05\omega$, upto four links can be traversed before the total drift approaches the critical value of 0.2ω. Thus even with fluctuations in propagation delays, the system can perform chains of reasoning of length four without cross-talk. A similar analysis assuming five distinct individuals in the bindings shows that the bindings can be propagated without cross-talk over chains of length ten. The above analyses assumed that the drifts interfered in the worst possible manner, and typically, the system will support longer chains of inference without cross-talk.

5.2 Use of node ensembles to deal with noise

A physical realization of the proposed system may have to deal with larger variations in propagation delays than those discussed above. Furthermore it must also be robust with respect to malfunction of individual nodes (perhaps, cell death) and other sources of noise such as spontaneous firings and drifts in firing frequencies. We believe that this problem can be dealt with in the classic manner, namely, by using ensembles of nodes instead of single nodes. The proposed solution to the binding problem directly generalizes to the case where individual argument and constant nodes are replaced by an *ensemble* of nodes. Fig. 13 illustrates such an ensemble based encoding of the predicates *give*, *own*, and *can-sell*, and the rules $\forall x, y, z[give(x, y, z) \Rightarrow own(y, z)]$ and $\forall x, y[own(x, y) \Rightarrow can\text{-}sell(x, y)]$. Notice that each argument is represented by an ensemble of nodes rather than a single node, and links between argument nodes have been replaced by a bundle of links between appropriate argument ensembles.

This encoding does not alter the basic mechanism for propagating and representing bindings. The dynamic binding between an argument and a filler is encoded by synchronous (in-phase) activation of the appropriate filler and argument ensembles. Thus the dynamic fact $give(John, Mary, Book1)$ is represented by having nodes in the argument ensembles *giver*, *recip*, and *g-obj* oscillating in phase with the nodes (or ensembles) representing $John$, $Mary$, and $Book1$, respectively. Propagation of bindings is realized via links between argument ensembles that cause nodes in interconnected ensembles to fire in synchrony. For example, the the dynamic binding $(recip = Mary)$ will propagate and lead to the argument ensembles *recip*, *owner* and *p-seller* firing in synchrony. Similarly, the binding $(g\text{-}obj = Book1)$ will propagate and lead to the argument ensembles *g-obj*, *o-obj* and *cs-obj* firing in synchrony. The encoding of long-term facts also remains unaffected except that individual links from argument nodes are now replaced by bundles of links from argument ensembles.

Representing bindings using the phase of activation of an ensemble of nodes rather than individual nodes

should greatly increase the robustness of the system and reduce its sensitivity to spontaneous firing and drift in firing frequency of individual cells, cell death, as well as variations in the propagation delays of individual links (Abeles 1982; Crick 1984; Strong & Whitehead 1989).

5.3 Biological plausibility

The computational properties of nodes stated above fall strictly within the constraints imposed by the core features of connectionism listed in Section 3.1. If one accepts those features as a reasonable abstraction of the computational characteristics of neurons, then clearly, the proposed representational mechanisms are biologically plausible. But we believe that the mechanisms proposed here are biologically plausible in a marginally stronger sense than what is implied by the above.

Before proceeding we would like to make it clear that we cannot — and do not wish to — claim that the brain encodes rules, facts, and dynamic bindings according to the scheme outlined in this paper. However, there is considerable evidence that suggests that i) the basic mechanisms required for sustaining synchronous (in-phase) rhythmic patterns of activity exist in the brain, and ii) such patterns may in fact be used by the animal brain to encode and process information.

The existence of rhythmic activity in the animal brain is well established. The fact that EEG recordings display rhythmic activity — even though they measure gross electrical activity summed over *millions* of neurons — strongly suggests that there exists significant temporal synchronization in neuronal activity (Sejnowsky 1981). More specific synchronous rhythmic activity has also been documented in the hippocampus, cerebellum, olfactory bulb, and the visual cortex (Freeman 1981; MacVicar & Dudek 1980; Gray & Singer 1989).

Freeman (1981) has put forth detailed arguments based on analysis, modeling, as well as experimental data to the effect that the phase of neuronal response plays a key role in the representation of information in the olfactory bulb. Churchland (1986) observes that the synchronization of rhythmic patterns in bands of Purkinge cells in the cerebellum may embody "a fundamental principle of neuronal organization underlying sensorimotor coordination".

Correlations between firing patterns of neurons in the visual cortex have been reported by several researchers. Gerstein (1970) and Toyama et al. (1981) have analyzed the cross-correlation in the firing of two simultaneously recorded cells in the visual cortex of the cat. These studies provide examples of closely correlated and even synchronous activity in pairs of cells in response to a moving light stimulus. For example, Fig. 6 (p. 656, (Gerstein 1970)) provides a striking example of phase relationship between the firing of a pair of neurons. The figure shows a cross-interval histogram of firings from two neurons.[12] Such histograms are particularly useful for detecting temporal relationships at time scales comparable to interspike intervals. In this particular case, the histogram reveals an extremely clear phase relationship between the spike trains produced by the two neurons. Gray and Singer have also studied the response of cells in orientation columns of the cat visual cortex (Gray & Singer 1989). They report that a moving light bar of optimal orientation, velocity, and direction of movement produces an oscillatory response that is tightly correlated with the phase (and amplitude) of an oscillatory local field potential. They conclude that synchronous oscillation of an ensemble of coactive neurons is an integral part of the neuronal response in the visual cortex and suggest that

the phase of the oscillatory response may be used as an additional dimension for encoding information in the cortex.

The experimental finding reported by Gray and Singer is particularly significant. It is a well known principle of perceptual organization and grouping that adjacent points moving with the same orientation are perceived as belonging to the same object. One may therefore conjecture that the synchronous oscillation of orientation specific cells responding to a moving light bar is the brain's way of encoding that all the cells responding to this stimulus represent the *same object*. This is analogous to the situation in Fig. 10 wherein the *in-phase* oscillation of the argument nodes *recipient*, *owner* and *potential-seller* and the individual node *Mary* is the system's way of encoding that all these roles are being filled by the *same object* Mary.

The use of temporally organized activity for encoding information has been a recurring theme in neural models of perception and cognition. It appears in the form of 'reverberating cell-assemblies' of Hebb, 'topological networks' of von der Malsburg, and 'synfire chains' of Abeles (Hebb 1949; von der Malsburg 1986; Abeles 1982). Recently, Strong and Whitehead (1989) have also proposed a mechanism for correcting illusionary conjuncts during visual perception using the idea of temporal synchrony.

More than forty years ago, Hebb introduced the notion of reverberating cell-assemblies — groups of simultaneously active cells — to explain perceptual integration at the feature level. In order to explain the integration of several features into a distinctive whole, Hebb augmented the notion of reverberating cell-assemblies and suggested that a complex object is represented by a 'phase sequence', i.e., by a sequence of organized and repeated activation of the cell-assemblies that correspond to the features of the object. In the words of Hebb, "Activity in a superordinate structure . . . is then best defined as being whatever determinate, organized activity results from repeated activity in the earlier-developed or subordinate structures giving rise to it."

Abeles argues that synchrony is a necessary feature of neural activity and suggests that computation in neural nets occurs via "synfire chains" — chains of synchronously firing groups of neurons. A synfire chain is distinct from a cell-assembly in that the transmission of activity along cells in a synfire chain is assumed to occur by the *synchronous* firing of cells. Abeles also argues that the existence of neural structures required to support synfire chains is consistent with neuroanatomy and neurophysiology. One can clearly identify synfire chains in our system when it propagates dynamic bindings during reasoning via the spread of rhythmic patterns of activity. With reference to the ensemble-based encoding described in Section 5.2 (see Fig. 13), the firing of rules and the propagation of bindings results in chains of argument ensembles firing in synchrony.

Von der Malsburg shows that synapses that can alter their weights extremely fast — within hundreds of milliseconds — can be used to create dynamic representations. A dynamic representation consists of several groups of cells, with cells within a group firing in synchrony and cells across different groups firing out of synchrony. As we have seen, our system also uses temporal synchrony and desynchrony among groups of cells to represent dynamic bindings but it does not require any synaptic modifications in order to do so (again refer to Section 5.2). This is fortunate because, as pointed out in Section 3.2, reflexive reasoning requires that dynamic bindings be established and propagated within tens of milliseconds, and hence, there is not enough time to realize such bindings via synaptic modifications unless one assumes the existence of even

faster synapses that can be modified within tens of milliseconds. The above is perhaps a moot issue because our system does not even require connections between arguments and their fillers to create dynamic bindings. The encoding of rules as long-term structures together with the use of temporal synchrony for representing dynamic bindings obviates the need for any hard-wired connections between arguments and their fillers (refer to Fig. 7). (We only need hard-wired connections to encode long-term facts). In contrast, any scheme for representing dynamic bindings that uses synaptic modification requires the existence of connections between *all possible* argument filler pairs.

The system described in Section 3 can rapidly and automatically infer what follows from a dynamically introduced fact. We have not said anything regarding the process or mechanism that might create dynamic bindings to represent a newly introduced fact via say, a linguistic input. The answer to this question is well beyond the scope of this paper but a mechanism similar to the 'searchlight' proposed by Crick (1984) may play a role in realizing this function. The searchlight mechanism proposed by Crick provides rapid bursts of activation *selectively*, to different groups of neurons. Thus one may envisage a multiple-beam searchlight mechanism that provides a burst of synchronous activation to *Mary* and *buyer* and simultaneously a phase-shifted burst of synchronous activation to *Porshe17* and *buy-obj* in response to the input 'Mary bought Porshe17'. Like Crick's searchlight, this mechanism should provide such bursts only for a few hundred milliseconds — long enough for the system to draw reflexive inferences but short enough to allow the system to represent the next input.

In the system described in this paper, new information is represented as a a transient trace of activation while long-term memory is encoded using 'hard-wired' interconnection between nodes. What remains to be seen is how *some* of the transient traces are converted into, and recorded as, synaptically encoded long-term structures. We do not offer any solution to this problem, but we would like to emphasize that the problem of learning the representation of rules and long-term facts proposed in this paper *is no more* difficult than the problem of learning any structured representation in connectionist networks. It might turn out that the fast synapses proposed by von der Malsburg play a role in sustaining short to 'medium-term' memories that must last beyond hundreds of milliseconds. Long-term potentiation in the hippocampus may also provide a 'medium-term' memory store that mediates and controls the selective transfer of transient information to long-term memory.

6 Putting things together

The proposed mechanisms for representing and propagating dynamic binding and encoding rules and long-terms facts provide the basic building blocks for a connectionist system that can represent and reason with structured knowledge involving multi-place predicates and variables. These mechanisms are fairly general and may be put together in different ways to realize different sorts of reasoning subsystems.

Our presentation in Section 3 centered around a forward reasoning system, i.e., a system that performed *predictive* inferences. Once the state of the system was initialized to represent an input situation by setting up appropriate dynamic bindings, the system automatically generated dynamic bindings that corresponded to situations that followed from the input situation.

The proposed mechanisms may also be used to create a backward reasoning system, i.e., a system that performs *explanatory* inferences. The behavior of such a system would be as follows: If the state of the system is initialized to represent a (input) situation by setting up appropriate dynamic bindings, the system will automatically determine whether this situation follows from the facts and rules encoded in the system. In other words, the input situation can be interpreted as a *query* and the system's behavior can be viewed as an attempt to answer the query using the knowledge encoded in the system.

So far we have been assuming that the reasoning carried out by the proposed systems is *deductive* in nature. This need not be the case. There is nothing inherent in the proposal that precludes the representation of probabilistic or evidential rules. This work proposes a connectionist model for representing and propagating dynamic bindings using the *temporal* aspect of node activations. Thus the strength of activation (the amplitude of the output pulse) and link weights may be used for encoding rule strengths and degree of beliefs in particular bindings. Based on our earlier work (Shastri 1988a; Shastri 1988b), where we used weighted links to encode evidential or probabilistic information, we believe that the proposed mechanism can be extended to encode evidential rules such as 'If an agent gives a book to a recipient then the recipient will probably read it'. When told that 'Tom gave Susan Book37', a forward reasoning system that encodes such evidential rules will not only infer 'Susan owns Book37' but also that 'probably, Susan will read Book37'. When told that 'Susan owns Book12' a backward reasoning system will generate explanations such as 'Someone probably gave Susan Book12', or 'Susan probably bought Book12'.

With the aid of some additional — but by no means trivial — control mechanisms it may be possible to design a system that combines forward as well as backward reasoning and admits evidential rules. Such a system will be capable of i) representing incoming information and making predictions based on this information by using its long-term knowledge, and ii) generating explanations for, and testing the consistency of, incoming information by referring to its long-term knowledge. We envision the design of a system that will perform a range of deductive and abductive inferences, make predictions and generate explanations, and do all this with remarkable efficiency. Such a system will interact with and complement several other representational and reasoning subsystems such as a frame-based hierarchical representation of concepts (i.e., a semantic network), *PART-OF* hierarchies, and procedural representations such as *routines* (Shastri & Feldman 1986) among others. The relevance of such a representation and reasoning system to the design of real-time intelligent systems should be apparent.

Having outlined the range of possibilities that may be realized with the proposed mechanisms, we turn to a specific reasoning system that has been designed using these mechanisms and analyzed in some depth.

7 A backward reasoning system

We now describe a *backward* reasoning system that can encode facts and rules involving *multi-place* predicates and *variables* in its *long-term* memory and perform an interesting class of inferences with extreme efficiency. The system may be described as follows: The system's long-term memory encodes any number of rules and facts. A query is posed to the system by initializing its state of activation to represent the (dynamic) argument bindings specified in the query. The system then automatically determines whether the *query*

follows from the facts and rules encoded in the long-term memory. The answers to queries are produced in optimal time, i.e., in time that is only proportional to the *length* of the shortest derivation of the query and is independent of the number of rules and facts encoded in the system. (In the context of the backward reasoning system, unless stated otherwise, we will refer to a 'long-term fact' simply as a 'fact'.)

7.1 A functional specification

In the notation of first order logic, rules have the form

$$\forall x_1, ..., x_m \; [P_1(...) \wedge P_2(...) ... \wedge P_n(...) \Rightarrow \exists z_1, ...z_l \; Q(...)]$$

where arguments for P_i's are subsets of $\{x_1, x_2, ...x_m\}$, while the arguments of Q consist of any number of arguments from among the x_i's, any number of constants, and single occurrences of existentially quantified arguments introduced in the consequent.

Facts are assumed to be partial or complete instantiations of predicates. In other words, facts may be viewed as atomic formulae of the form $P(t_1, t_2 ... t_k)$ where t_i's are either constants or — distinct — existentially quantified variables. Later in Section 8.3 we describe how the system may be extended so that the t_i's may refer to concepts in a *IS-A* hierarchy.

A query has the same form as a fact: it is a partially or fully instantiated predicate where the uninstantiated arguments are assumed to be existentially quantified. Observe that a query, all of whose arguments are bound to constants, corresponds to the *yes-no* question: 'Does the query follow from the rules and facts encoded in the long-term memory of the system?' A query with existentially quantified variables, however, has several interpretations. For example, the query $P(a, x)$, where a is a constant and x is an existentially quantified argument, may be viewed as the *yes-no* query: 'Does $P(a, x)$ follow from the rules and facts for some value of x?' Alternately this query may be viewed as the *wh*-query: 'For what values of x does $P(a, x)$ follow from the rules and facts in the system's long-term memory?'

Some examples of rules and facts are given below. Table 1 lists some queries, their interpretation(s), and their answer(s).

Rules:

$\forall x, y, z \; [give(x, y, z) \Rightarrow own(y, z)]$
$\forall x, y \; [buy(x, y) \Rightarrow own(x, y)]$
$\forall x, y \; [own(x, y) \Rightarrow can\text{-}sell(x, y)]$
$\forall x \; [omnipresent(x) \Rightarrow \forall y, t \; present(x, y, t)]$
; Anyone who is omnipresent is present everywhere at all times
$\forall x, y \; [born(x, y) \Rightarrow \exists t \; present(x, y, t)]$
; Everyone must have been present at his or her birthplace sometime.
$\forall x \; [triangle(x) \Rightarrow number\text{-}of\text{-}sides(x, 3)]$
; A triangle has three sides.
$\forall x, y \; [sibling(x, y) \wedge born\text{-}together(x, y) \Rightarrow twins(x, y)]$
; Two siblings born at the same time are twins.

Facts:

$give(John, Mary, Book1)$;	John gave Mary Book1.
$give(x, Susan, Ball2)$;	Someone gave Susan Ball2.
$buy(John, Car7)$;	John bought Car7.
$own(Mary, Ball1)$;	Mary owns Ball1.
$omnipresent(x)$;	There exists someone who is omnipresent.
$triangle(A3)$;	A3 is a triangle.
$sibling(Susan, Mary)$;	Susan and Mary are siblings.
$born\text{-}together(Susan, Mary)$;	Susan and Mary were born at the same time.

In describing the backward reasoner, we will begin by making several simplifying assumptions. We will assume that rules have a single predicate in the antecedent and that constants and existentially quantified variables do not appear in the consequents of rules. The rules and queries must also satisfy constraints that follow directly from the limitations of the mechanism for representing and propagating dynamic bindings. Subsequent sections will provide these details.

7.2 Encoding rules and facts in the long-term memory

Fig. 14 depicts a network that encodes the following rules and facts:

- $\forall x, y, z\ [give(x, y, z) \Rightarrow own(y, z)]$
- $\forall x, y\ [buy(x, y) \Rightarrow own(x, y)]$
- $\forall x, y\ [own(x, y) \Rightarrow can\text{-}sell(x, y)]$
- $give(John, Mary, Book1)$
- $buy(John, Car7)$
- $own(Mary, Ball1)$

Each constant in the domain is represented by a ρ-btu node. Constant nodes have been omitted from Fig. 14 but the links from these nodes to other nodes have been identified.

An *n-ary* predicate is represented by a pair of τ-and nodes and a cluster of n ρ-btu nodes. One of the τ-and nodes will be referred to as the *enabler* and the other as the *collector*. As we shall see, the *enabler* of a predicate P will become active if the network is being queried about P (this query might have been posed by an external process or may have been generated — internally — by the system in the process of answering a query). Similarly, the *collector* of a predicate P will be active whenever the network asserts something about P. As a matter of convention, *enabler* nodes will always point upwards while *collector* nodes will always point downwards. Also the *enabler* and *collector* of a predicate P will be labeled $e : P$ and $c : P$, respectively. With reference to Fig. 14, the ternary predicate *give* is represented by the *enabler* $e : give$, the *collector* $c : give$, and the three argument nodes: *giver*, *recip*, and *g-obj*. (We have abbreviated

the arguments *recipient* and *give-object* to *recip* and *g-obj*, respectively. We will do so often when labeling nodes in illustrations.)

A fact is encoded using a τ-and node which receives an input from the *enabler* node of the associated predicate. This input is modified by inhibitory links from argument nodes of the associated predicate. If an argument is bound to a constant in the fact then the modifier input from such an argument node is in turn modified by an inhibitory link from the appropriate constant node. The output of the τ-and node is connected to the *collector* of the associated predicate (refer to Fig. 14.)

Unless there is a potential for ambiguity, we will refer to a 'constant node representing the constant c', simply as the 'node c' or just 'c'. Similarly, for arguments.

A rule is encoded by connecting the *collector* node of the antecedent predicate to the *collector* node of the consequent predicate, the *enabler* node of the consequent predicate to the *enabler* node of the antecedent predicate, and by connecting the argument nodes of the consequent predicate to the argument nodes of the antecedent predicate in accordance with the correspondence between these arguments specified in the rule (refer to Fig. 7). Notice that the links are directed from the arguments of the consequent to the arguments of the antecedent. The 'reverse' direction of links is due to the fact that the system performs backward reasoning.

7.3 Inference process

The inference process is initiated by posing a query to the system by activating appropriate nodes. The inference process may be thought of as consisting of two stages (these stages overlap during the actual ro-cessing.) The first stage corresponds to a parallel breadth-first exploration of the potentially huge inferential dependency graph (see Section 3.5). During this stage, all the (long-term) facts that are relevant to the proof of the query become active. In the second stage the actual proof is constructed: activation from the relevant facts flows downwards along *collector* nodes to produce an answer to the query. A *yes* answer corresponds to the activation of the *collector* node of the query predicate. The time taken by the system to produce a *yes* answer equals $l * \pi$, where π equals the period of oscillation and l equals the *length* of the *shortest* derivation of the query. A *no* answer results if the *collector* node of the query predicate does not receive any activation for more than $2 * d$ periods of oscillations, where d equals the diameter of the inferential dependency graph associated with the rule-base.

7.4 Posing a query: specifying dynamic bindings

As described in Section 7.1 a query is a partially instantiated predicate of the form $P(t_1, ..., t_n)$, where t_is are either constants or existentially quantified variables. Therefore, posing a query to the system involves specifying the query predicate and the argument bindings specified in the query.

Let us choose an arbitrary point in time — say, t_0 — as our point of reference for initiating the query. We assume that the system is in a quiescent state just prior to t_0. The query predicate is specified by activating $e : P$, the *enabler* of the query predicate P, with a pulse train of width and periodicity p starting at time t_0.

The argument bindings specified in the query are communicated to the network as follows:

- Let the argument bindings in the query involve k distinct constants: $c_1, ..., c_k$. With each of these k constants, associate a delay δ_i such that no two delays are within ω of one another and the longest delay is less than $\pi - \omega$. Each of these delays may be viewed as a distinct *phase* within the period t_0 and $t_0 + \pi$, and therefore, by assigning the delays δ_i to appropriate constants, we have effectively associated a distinct phase with each distinct constant introduced in the query. (For simplicity, we will assume that the δ_i's are evenly spaced over the time interval π.)

- The argument bindings of a constant c_i are indicated to the system by providing an oscillatory pulse train of pulse width ω and periodicity π starting at $t_0 + \delta_i$, to c_i and all arguments to which c_i is bound. This is done for each constant c_i $(1 \leq i \leq k)$ and amounts to representing argument bindings by the in-phase or synchronous activation of the appropriate constant and argument nodes.

7.5 Propagation of dynamic bindings and activation of relevant facts

Once the query is posed, a parallel search for facts that are relevant to the proof of the query ensues. We illustrate this process with the help of an example (refer to Fig. 14). Consider the query *can-sell*($Mary, Book1$). This query is posed by providing inputs to the constants $Mary$ and $Book1$, the arguments *p-seller*, *cs-obj* and the *enabler* e : *can-sell* as shown in Fig. 15. $Mary$ and *p-seller* receive in-phase activation and so do $Book1$ and *cs-obj*. As would be expected, the two constants $Mary$ and $Book1$ receive activation in distinct phases. Let us refer to the phase of activation of $Mary$ and $Book1$ as phase-1 and phase-2 respectively. As a result of these inputs, $Mary$ and *p-seller* will fire synchronously in phase-1 of every period of oscillation, while $Book1$ and *cs-obj* will fire synchronously in phase-2 of every period of oscillation. The node *e:can-sell* will also oscillate and generate a pulse train of periodicity and pulse width π.

The activations from the arguments *p-seller* and *cs-obj* reach the arguments *owner* and *o-obj* of the predicate *own*, and consequently, starting with the second period of oscillation, *owner* and *o-obj* become active in phase-1 and phase-2, respectively. ¿From now on, the nodes $Mary$, *owner*, *p-seller* are active in phase-1, while the nodes $Book1$, *cs-obj*, and *o-obj* are active in phase-2. At the same time, the activation from e : *can-sell* activates e : *own* (Refer to Fig. 15). The system has essentially, created two new bindings for the predicate *own* – $Mary$ has been bound to the argument *owner*, and $Book1$ has been bound to the argument *own-object*. These newly created bindings of the arguments of *own* can be thought of as encoding the query *own*($Mary, Book1$) (i.e., 'Does Mary own Book1?')!

The τ-and node associated with the fact *own*($Mary, Ball1$) does not match the query and remains inactive. This may be verified by observing that during phase-2, the activation from e : *own* going into the τ-and node is blocked by the inhibitory activation from the argument node *owner*. (Notice that $Ball1$ is not firing).

The activations from *owner* and *o-obj* reach the arguments *recip* and *g-obj* of *give*, and *buyer* and *b-obj* of *buy* respectively. Thus beginning with the third period of oscillation, arguments *recip* and *buyer* become active in phase-1, while arguments *g-obj* and *b-obj* become active in phase-2. At this time, the active constant and argument nodes in phase-1 are: $Mary$, *p-seller*, *owner*, *recip* and *buyer*; and those active in phase-2 are: $Book1$, *cs-obj*, *o-obj*, *g-obj*, and *b-obj*. In essence, the system has created new bindings for the

predicates *can-sell* and *buy* that can be thought of as encoding two new queries: $give(x, Mary, Book1)$ (i.e., 'Did someone give Mary Book1?'), and $buy(Mary, Book1)$ (i.e., 'Did Mary buy Book1?').

The τ-and node associated with the fact $buy(John, Car7)$ does not become active because the activation from e : *buy* is blocked by the inhibitory activations from the arguments *buyer* and *b-obj*. (Notice that neither $John$ nor $Car7$ are active). The τ-and node associated with the fact $give(John, Mary, Book1)$, however, does become active as a result of the uninterrupted activation from e : *give*. It is easy to verify that the inhibitory inputs from *recip* and *g-obj* are blocked by the in-phase inputs from $Mary$ and $Book1$, respectively. The activation from this τ-and node causes c : *give* to become active and the output from c : *give* in turn causes c : *own* to become active and transmit an output to c : *can-sell*. Consequently, c : *can-sell* — the *collector* of *can-sell* — becomes active, resulting in an affirmative answer to the query $can\text{-}sell(Mary, Book1)$. The propagation of rhythmic oscillations resulting from the query are depicted in Fig. 15.

7.6 Encoding rules with constants and repeated variables in the consequent

In describing the encoding of rules and facts in the previous section we had assumed that constants or existentially quantified variables do not appear in the consequent. We show illustrates how such rules are encoded by illustrating the encoding of the following rule (see Fig. 16)

$\forall x1, x2\ [P(x1, x2) \Rightarrow \forall y \exists z\ Q(x1, x2, y, z, a)]$ ..... $R2$

The node labeled $g1$ is a τ-or node and it projects inhibitory modifiers that can block the firing of the $R2$. The node $g1$ ensures that the rule participates in an inference only if the conditions implicit in the consequent of the rule are met. The first condition concerns existentially quantified variables in the consequent of a rule. Observe that if a variable that is existentially quantified in the consequent of a rule gets bound in the reasoning process, the rule cannot be used to infer the consequent. The desired behavior is achieved by the link from the fourth argument of Q to $g1$ and the inhibitory modifiers emanating from $g1$. The node $g1$ will become active and block the firing of the rule whenever the fourth argument of Q gets bound.

The second condition concerns the occurrence of constants in the consequent of a rule. If a constant appears in the consequent of a rule then this rule cannot be used if the corresponding argument gets bound to any other constant during the reasoning process. This constraint is encoded by a link from the fifth argument of Q to $g1$ that is in turn modified by an inhibitory modifier from a. If the fifth argument of Q gets bound to any constant other than a, $g1$ will become active and block the firing of the rule.

In most cases, the argument corresponding to the universal quantifier in the consequent has no direct bearing on the activation of the rule, and hence, need not be connected to anything. There is however, one exception: if the same variable occurs in multiple argument positions in the consequent of a rule then we need to ensure that this variable is either unbound or bound to the same constant. This constraint is encoded by introducing a node that receives inputs from all the argument nodes that correspond to the same variable. Consider the network fragment shown in Fig. 17 that depicts the encoding of the rule $\forall x\ P(x) \Rightarrow \forall y Q(x, x, y, a)$. The node $g2$ is like a τ-or node except that it becomes active if it receives inputs in more than one phase within a period of oscillation. This behavior ensures that the firing of the associated rule is inhibited unless all the appropriate arguments are bound to the same constant.

7.7 Encoding multiple antecedent rules

A rule with conjunctive predicates in the antecedent, i.e., a rule of the form $P_1(...) \wedge P_2(...) \wedge ... P_m(...) \Rightarrow Q(...)$, is encoded using an additional τ-and node that has a threshold of m. The outputs of the *collector* nodes of $P_1, ..., P_m$ are connected to this node which in turn is connected to the *collector* of Q. This additional node becomes active if and only if it receives inputs from the *collector* nodes of all the m antecedent predicates. The interconnections between the argument nodes of the antecedent and consequent predicates remain unchanged. Fig. 18 illustrates the encoding of the multiple antecedent rule $\forall x, y P(x,y) \wedge Q(y,x) \Rightarrow R(x,y)$. The τ-and node labeled $g3$ has a threshold of 2.

7.8 Total node requirement

In addition to performing inference extremely efficiently, the rule-based reasoning system also makes efficient use of nodes. To encode r rules and f long-term facts, the system only requires $O(r+f+pa+c)$ phase sensitive binary threshold units. In the above expression, pa and c are the total number of predicate arguments and constants in the domain, respectively. Thus the total node requirement is only *linear* in the size of the knowledge-base.

7.9 Constraints

In this section we specify some constraints that govern the behavior of the backward reasoning system. Our aim was not to build a *general purpose* production system or *PROLOG* engine that would draw *all* inferences in time proportional to the length of the derivation. Such a system *cannot* exist — an efficient general purpose production system is to the theory of computation what a perpetual motion machine is to thermodynamics. Our *ultimate* goal is to develop a computational account of how an agent may perform certain inferences with extreme efficiency and the backward reasoning system described above is a step in this direction.

The first constraint on the backward reasoning system is that only a limited number of argument bindings may be specified in a query. A biologically, as well as psychologically, plausible estimate of this number appears to be about seven. This constraint is a direct consequence of the limitation that the number of *distinct* individuals involved in simultaneous dynamic bindings is bounded by the relative widths of π and ω.

As stated in Section 3.6, a limitation of the dynamic binding mechanism is that it cannot represent more than one dynamic fact about a predicate simultaneously. For example, the system cannot — simultaneously — maintain the *dynamic* facts: $give(John, Mary, Book1)$ and $give(Mary, Tom, Car7)$. The system behaves conservatively when such a situation arises; if *every* derivation of a query requires that a predicate be dynamically instantiated more than once, the system does not answer 'yes' (even though this query may *logically* follow from the rules and facts encoded in the system).[13] In Section 8.1 we show that this constraint may be relaxed by using the rhythmic and synchronous firing of nodes to simultaneously represent several dynamic facts pertaining to the same predicate.

The encoding of rules described in Section 3.5 enforces the correspondence between the arguments in the

antecedent and the arguments in the consequent of the rule. We have also described how the encoding of rules can be extended to enforce equality among arguments in the consequent of a rule (Section 7.6, last paragraph). However, it is not always possible for a parallel backward reasoning system to enforce equality among arguments in the antecedent of a rule. Consider the rule $\forall x, y P(x, x, y) \Rightarrow Q(y)$ and the query $Q(a)$. The processing of this query will result in the dynamic query $P(?, ?, a)$ — where the first and second arguments are left completely unspecified. Consequently, the system cannot enforce the condition that a long-term fact involving P should match the query $Q(a)$ only if its first and second arguments are bound to the same constant. Contrast this situation with one wherein the rule is: $\forall x, y P(x, x, y) \Rightarrow Q(x)$ and the query is $Q(a)$. The dynamic query resulting from the processing of the query $Q(a)$ will be $P(a, a, y)$. Notice that now the condition that the first and second arguments of P should be the same is incorporated in the dynamically generated query, and therefore, gets enforced automatically. The crucial feature of the second situation is that x, the *repeated* variable in the antecedent of the rule, also appears in the consequent *and* gets bound in the query. This explains the third and last constraint. We require that:

> a variable occurring in multiple argument positions in the *antecedent* of a rule that participates in the answering of a query, should also appear in the consequent of the rule and get bound during the query answering process.[14]

It is important to realize that the above constraint is required only in a backward reasoning system and *not* in a forward reasoning system. In a forward reasoning system, the rule $\forall x, y\ P(x, x, y) \Rightarrow R(x, y)$ would be encoded as shown in Fig. 19. The node g is a τ-or node with a threshold of 2 and receives inputs from the two argument nodes that should be bound to the same filler. It becomes active if it receives two inputs in the same phase and enables the firing of the rule via intermediary ρ-btu and τ-and nodes. This behavior ensures that the rule fires only if the first and second arguments of P are bound to the same filler. For an extended example of such a behavior see Fig. 20 that depicts the encoding of the multiple antecedent rule $\forall x, y, z\ P(x, y) \wedge Q(y, z) \Rightarrow R(x, z)$.

The restrictions imposed on the backward reasoning system do seem to exclude certain kinds of inferential behavior. For example, it seems to exclude rules that are required to express the transitivity of a relation. To assert that a relation P is transitive we need to say: $\forall x, y, z P(x, y) \wedge P(y, z) \Rightarrow P(x, z)$ — wherein the variable y occurs twice in the antecedent but does not appear in the consequent. This appears to be a major limitation because transitivity plays an important role in common sense reasoning — to wit, reasoning about sub and super-categories, ancestors and descendents, part-of relationships, property inheritance, and some cases of temporal reasoning. We fully agree that we are very good at dealing with transitivity of a *small number of very specific relations*. But we also believe that in all such situations, the transitivity of the relation is encoded *explicitly*. The organization of concepts in a *IS-A* hierarchy using *IS-A* links to capture the transitivity of sub-class/super-class relationship is an excellent case in point. In an *IS-A* hierarchy, the sub-class super-class relationship is explicitly represented using dedicated *IS-A* links that allow the system to efficiently compute the transitive closure of these relations. The use of dedicated links reduces the problem of applying the transitivity rule to a problem of link traversal — an operation that can be realized efficiently using spreading activation. Notice that a system that must compute the transitive closure of a set of tuples in the *IS-A* relation by applying the *rule*: $\forall x, y, z\ IS\text{-}A(x, y) \wedge IS\text{-}A(y, z) \Rightarrow IS\text{-}A(x, z)$ will not be efficient

(Allen & Frisch 1982). The above observation also applies to *part-of*, and *occurs-during* hierarchies. There is no evidence that as common sense reasoners — especially when operating in the automatic and reflexive mode — we are proficient in dealing with the abstract notion of transitivity in arbitrary situations. In Section 8.3 we show how the backward rule-based reasoner can be combined with a distinct representation of concepts organized in an *IS-A* hierarchy to result in an efficient reasoning system that makes use of the transitivity of the subclass-superclass relationships.

8 Some extensions of the reasoning system

In this section we outline some extensions of the backward rule-based reasoner that are under investigation.

8.1 Representing multiple dynamic facts about the same predicate

The representation for dynamic bindings described in Section 3 cannot simultaneously represent multiple dynamic facts about the same predicate. It turns out that the notion of rhythmic and synchronous firing of nodes can be extended to represent several dynamic facts pertaining to the same predicate.

The extension relies on the assumption that *during an episode of reflexive reasoning* any given predicate need only be instantiated a *bounded* number of times — a plausible value of this bound being around three to five. Thus we have two constraints on the nature of dynamic bindings: i) no more than seven or so distinct individuals may be involved in simultaneous dynamic bindings and ii) no more than three to five dynamic instantiations of the same predicate may exist simultaneously. The second assumption also seems plausible for the same reasons as the first one and anyone doubting this is requested to read the following text *without repetition*:

Susan loves Tom, John loves Lisa, Tom loves Mary, and Clara loves Tom.

and answer questions about who loves whom![15]

Fig. 21 depicts the encoding of the binary predicate *love* with arguments *lover* and *lovee*. The assumption here is that a predicate may be instantiated at most three times. Observe that we have three banks of predicate and argument nodes. In general, k banks are required if we want the system to be capable of dynamically instantiating a predicate k times. (As before, each node in the encoding may be replaced by an ensemble of nodes.)

Fig. 22 depicts the firing patterns of individual nodes when they represent the two dynamic facts $love(John, Mary)$ and $love(Mary, Tom)$, simultaneously. As before a distinct phase is associated with each constant and therefore, the firing pulses of $John$, $Mary$, and Tom occupy distinct phases. Also as before, dynamic bindings are represented by in-phase firing of appropriate argument and constant nodes. The first bank of nodes represents $love(John, Mary)$, and therefore, the argument nodes $lover_1$ and $lovee_1$ are firing in phase with $John$ and $Mary$, respectively. The second bank of nodes represents the fact $love(Mary, Tom)$, and therefore, the argument nodes $lover_2$ and $lovee_2$ are firing in synchrony with $Mary$ and Tom, respectively.

The predicate nodes $love_1$ and $love_2$ are involved in the representation of dynamic facts, and hence, are active. Observe that the period of oscillation of predicate nodes is 3π, i.e., *three times* that of argument nodes, while the pulse width of predicate nodes *equals* π. Thus a typical period of oscillation and pulse width of predicate nodes may be 30 msecs. and 10 msecs. respectively. Also notice that $love_1$ and $love_2$ occupy a distinct phase in this lower frequency oscillation. This separation in the phase of $love_1$ and $love_2$ is crucial in achieving the separation among the bindings belonging to two facts. This becomes quite clear if we examine the firing patterns obtained by *and*-ing (i.e., gating) the output of the argument nodes in a bank with the output of the predicate node in the associated bank. Fig. 23 depicts this firing pattern. Observe that now there is no temporal overlap – and hence, no cross-talk – in the representation of the dynamic bindings associated with the two facts. As before, the bindings between an argument and its filler are encoded using temporal synchrony at the 1 msec. level. At the same time argument-filler bindings pertaining to the same fact are grouped or bound together using temporal 'synchrony' at the 10 msec. level.

Notice that we need not postulate higher levels of temporal groupings: two levels suffice — one to bind arguments with fillers and another to bind argument-fillers that belong to the same fact. What we have not yet specified is how the correspondence between arguments of the antecedent and consequent predicates of a rule is encoded. This requires a mechanism (interconnection pattern) that would channel the temporally interleaved activation arriving along argument links into available banks of argument nodes — without interfering with the activity of argument nodes that are already in use. We hope to report a solution to this problem in the near future.

8.2 Answering *wh*-queries

As explained in Section 7.1, a query with unbound arguments can be interpreted either as a *yes-no* query or a *wh*-query. Treating such a query as a *yes-no* question assumes that the unbound arguments in the query as existentially quantified. To answer *yes* it must be determined that there exist *some* instantiations of the unbound arguments. The system described in Section 7 does this. To answer a *wh*-query, however, the system must determine the instantiations of unbound arguments for which the query is true. It turns out that there is a rather simple way of extending the system to do this.

Consider the proof of the query $can\text{-}sell(Mary, x)$ with respect to the network shown in Fig. 14. The *yes-no* version of this query will be answered in the affirmative and in the process the two relevant facts $own(Mary, Ball1)$ and $give(John, Mary, Book1)$ will become active. The answer to the *wh*-query 'What can Mary sell?', can be obtained by simply identifying the constants that are bound to the arguments *g-obj* and *b-obj*, respectively, of the two active facts. This is not a coincidence — notice that the arguments *g-obj* and *b-obj* are precisely the arguments that map to the unbound argument *cs-obj* of *can-sell* via the rules encoded in the system. The system can easily extract this information by making use of the same binding propagation mechanism it uses to map arguments bound in the query. A straightforward way of doing so is to posit a separate *answer extraction* stage that is carried out after the *yes-no* query associated with the *wh*-query has produced a *yes* answer. For example, given the query 'What can Mary sell?' the system first computes the answer to the *yes-no* query 'Can Mary sell something?' and identifies the facts $own(Mary, Ball1)$ and $give(John, Mary, Book1)$ that lead to a *yes* answer. The answer extraction stage

follows and picks out the constants *Ball*1 and *Book*1 as the answers.

The representation of a fact is augmented as shown in Fig. 24 in order to support answer extraction. The representation of a fact involving an n-ary predicate is modified to include $n+1$ additional nodes: for each of the n arguments of the associated predicate there exists a two input ρ-btu node with a threshold of two. For convenience we will refer to such a node as a *binder* node. The other node (shown as a filled-in pointed pentagon) is like a τ-and node except that once activated, it stays on for some time — say 10π, or a few hundred milliseconds — even after the inputs are withdrawn. This node, which we will refer to as the *latch* node, receives an *Answer* input in addition to an input from the τ-and node of the associated fact.

At the end of the first stage, the outputs of the τ-and nodes of all the relevant facts would be active. The output of these τ-and nodes in conjunction with the *Answer* signal will turn on the associated *latch* nodes and provide one of the two inputs to the *binder* nodes. If the associated *yes-no*-query results in a *yes* answer, the answer extraction stage is initiated. Inputs relating to the first stage are withdrawn and each *unbound* argument of the query predicate is activated in a distinct phase using a pulse train of width ω and periodicity π. In addition a network wide *Answer* signal is also propagated. The activation of unbound query arguments results in a phase-sensitive propagation of activation and eventually leads to the activation of arguments associated with facts relevant to the query. This provides an input to the appropriate *binder* nodes of these facts. As the *binder* nodes were already receiving an input from a *latch* node, they become active and produce a phase-sensitive output that in turn activates the associated constants in a phase-sensitive manner. The answer to the *wh*-query - i.e., the constant(s) that fill an unbound argument a_i of a query - will be precisely those constants that are active in phase with a_i. The time taken by the answer extraction step is bounded by the depth of the inferential dependency graph.

8.3 Combining the rule-based reasoner with an *IS-A* hierarchy

There is a direct way of interfacing the rule-based reasoner described thus far with a connectionist representation of a conceptual hierarchy. (We will refer to the latter as an *IS-A* hierarchy.) This results in an extended system that allows 'concepts' (or types) in an *IS-A* hierarchy to appear wherever constants could appear in the basic system. Such an extended system may be realized by assuming that the focal node of each concept (cf. Section 3.3) in the *IS-A* hierarchy is a ρ-btu node.

Each conceptual *IS-A* link is encoded using two connectionist links. Thus A *IS-A* B is encoded by a bottom-up link from A to B and a top-down link from B to A. The bottom-up links are enabled during the processing of *yes-no* queries while the top-down links are enabled during the answer extraction stage of a *wh*-query. The mechanism for selectively enabling bottom-up and top-down links is discussed in (Shastri 1988a). Fig. 25 illustrates the encoding of the rule and fact:

- $\forall x, y\ [preys\text{-}on(x, y) \Rightarrow scared\text{-}of(y, x)]$
- $preys\text{-}on(Cat, Bird)$

and the *IS-A* relationships:

1. $is\text{-}a(Bird, Animal)$ 2. $is\text{-}a(Cat, Animal)$

3. *is-a*(*Robin*, *Bird*) 4. *is-a*(*Canary*, *Bird*)
5. *is-a*(*Tweety*, *Canary*) 6. *is-a*(*Chirpy*, *Robin*)
7. *is-a*(*Sylvester*, *Cat*)

Table 2 lists some example queries and their answers based on the encoding given in Fig. 25. The time course of activation for the query: *scared-of*(Tweety,Sylvester) (Is Tweety scared of Sylvester?) is given in Fig. 26. The proposed encoding and the inferential behavior of the system is such that it interprets 'Cats eat birds' to mean 'All cats eat all birds'. This is at best controversial and raises complex issues relating to the treatment of quantification. These issues, however, are beyond the scope of this paper.[16]

Interfacing the rule-based reasoner with an *IS-A* hierarchy further illustrates the interesting role played by the temporal encoding of information during reasoning. It was observed in Section 3.6 that during an episode of reasoning, all the arguments filled by a constant become active in the same phase thereby creating a dynamic 'temporal frame' of knowledge grouped together by temporal synchrony. The notion of a 'temporal frame' gets even richer when the rule-based reasoner is interfaced with an *IS-A* hierarchy because in the augmented system, the local as well as inherited property values of an entity, its type and supertype, as well as the arguments it fills can all be brought together in a dynamic 'temporal frame' with extreme efficiency.

We are also investigating the possibility of interconnecting the rule-based reasoner with an *IS-A* hierarchy to encode rules with *sorted* (or typed) variables. The use of typed variables simplifies the form of rules. For example, the following multiple antecedent rule, with repeated variables in the antecedent: $\forall x, y\ collide(x, y) \wedge animate(x) \wedge solidobj(y) \Rightarrow hurt(x)$ may be transformed into the single antecedent rule: $\forall x : animate, y : solidobj\ collide(x, y) \Rightarrow hurt(x)$, where x and y are variables of type *animate* and *solidobj*, respectively. We hope to show that some of the type restrictions on the variables can be imposed by appropriate inhibitory or excitatory links between concepts (i.e., types) in the *IS-A* hierarchy and the rule-based component. These links will allow the rule to fire only if the relevant argument fillers belong to the appropriate types (concepts).

8.4 Encoding defeasible rules

As mentioned earlier, there is nothing inherent in the proposed solution to the binding problem that precludes the representation of probabilistic or defeasible rules. On the contrary, it is rather easy to extend the system to incorporate such rules. The proposed system makes use of the phase of activation to encode binding information. This leaves open the possibility of using the *amplitude* of activation and link weights to encode the strength of probabilistic rules and the 'degree of belief' in the dynamic bindings. Based on our experience with the encoding of evidential or probabilistic information we intend to extend the proposed mechanism to encode probabilistic rules.[17]

When told that 'Tom gave Susan Book37', a forward reasoning system that incorporates probabilistic rules will not only infer that 'Susan owns Book37' but also *predict* that 'Susan will probably read Book37' by making use of soft rules such as 'If an agent gives a book to a recipient then the recipient will probably read it'. Similarly, when told that 'Susan owns Book12' a backward reasoning system will generate explanations such as 'Probably, someone gave Susan Book12', or 'Probably, Susan bought Book12'.

8.5 Admitting function terms

We are also looking at the possibility of extending the expressive and reasoning power of the rule-based reasoner by allowing restricted occurrences of function-terms in rules and facts. The key idea is as follows: Function-terms introduce new uninstantiated objects during the reasoning process. In our representation, objects are represented as a phase in a rhythmic pattern of activation during an episode of reasoning. Therefore, objects introduced by function-terms can be represented by allocating them an unused phase. *Thus during an episode of reasoning, the reference to mother-of(Tom) should result in the association of an* 'unused' phase — say, one adjacent to the one associated with Tom — for this uninstantiated entity 'mother of Tom'. All arguments that are bound to 'mother of Tom' will oscillate in this phase. An easy of realizing this would be to use a subnetwork that acts as a *phase shifter*. Clearly, the number of function-terms that can be introduced will be limited to a small number. In this context it must be remembered that general reasoning with function-terms is computationally intractable.

9 Some connectionist systems for encoding rules and dynamic bindings

Reasoning systems that only have single variable rules can get by without actually solving the dynamic binding problem. For example, if we only consider the *structure* of information encoded in the connectionist semantic network (CSN) (Shastri 1988b), and ignore the evidential nature of information encoded in it, we find that the 'rules' in the system are of the form: $\forall x P(x) \Rightarrow Q(x)$ and $\forall x P(x) \Rightarrow R(x, a)$. Observe that both these rules have just one variable and hence, during an episode of reasoning, only one constant participates in all the variable bindings. Consequently, CSN does not require any *explicit* representation of dynamic bindings.

Touretzky and Hinton's DCPS (Distributed Connectionist Production System) (1988) does represent dynamic bindings explicitly, but its ability to maintain and propagate dynamic bindings is limited in critical ways. First, DCPS can only deal with *single variable* rules. Second, it is serial at the knowledge level and allows only *one* rule to fire during each processing cycle. Consequently, it does not satisfy the efficiency requirements outlined in Sections 1 and 3.2. Finally, in spite of allowing only a single variable in a rule and only dealing with a single rule firing at a time, the dynamic binding mechanism used by DCPS is susceptible to cross-talk.

Let us consider systems that can represent rules with multiple variable. Lange and Dyer (1989) describe the ROBIN system that uses *signatures* to solve the binding problem. They permanently allocate a distinct signature to each constant and represent a dynamic binding by propagating the signature of the appropriate constant to the argument to which it is bound. Unfortunately, passing signatures amounts to passing names/addresses and thus this solution violates the connectionist constraint that messages be simple scalars (refer to Section 3.1). It also requires that nodes be capable of storing and matching long bit strings that make up the signatures (alternately, the nodes must be capable of storing and comparing *high-precision* analog values).

A significant improvement over the signature based solution can be arrived at by noting that although the total number of constants in any domain may be very large, the number of constants involved in a particular reasoning episode is small (cf. Section 3.2). Hence, instead of permanently assigning a distinct signature to every domain constant, it suffices to assign distinct signatures to those constants that participate in an episode of reasoning. Furthermore, this assignment can be temporary and need only exist during a reasoning episode. What we have just said is that we do not need signatures, we can get by with *markers* (*a la* Fahlman (1979)!

In fact, the solution to the binding problem based on temporal synchrony is *functionally* equivalent to a solution based on passing markers. This can be recognized by observing that the phase associated with a concept during the propagation of bindings may be viewed as a 'marker'. The in-phase or synchronous activation of all arguments bound to a concept is functionally equivalent to marking these arguments with the same marker. The temporal synchronization approach adopted by us, however, offers significant advantages over the marker-passing approach:

- While nodes in a system based on temporal synchrony need only communicate a simple (ON/OFF) level of activation, nodes in a marker passing system must *selectively* propagate and *store* distinct markers. Thus nodes in the marker-passing system will be more complex – both in their memory capacity as well as their processing power.

- Compared to the marker passing approach, the temporal synchronization approach requires considerably simple machinery to support reasoning. In the latter, dynamic bindings are represented by the synchronous activation of appropriate argument and constant nodes, and the detection of dynamic bindings requires determining whether two signals are synchronized or not. In a marker-passing system dynamic bindings will have to be realized by propagating the same marker to the appropriate nodes, and therefore, the system will have to detect bindings by determining whether two k-bit vectors are equal or not (assuming that markers are encoded using k-bits). But detecting temporal synchrony is easier than comparing bit strings, and hence, the 'mechanisms' required to detect and check bindings in a marker-passing system will have to be considerably more complex. To make this concrete, compare the encoding of long-term facts shown in Figs. 11 and 12 with a 'device' that will be required to perform the same function in a marker passing system. Observe that each inhibitory modifier in these figures will have to be replaced by a k-bit comparator.

- The use of temporal synchronization to create dynamic bindings is biologically plausible.

- Another important difference will arise if we wish to encode defeasible rules. Our system uses the phase of activation to encode binding information. Thus the *amplitude* of output pulses and link weights can be used to encode the strengths of probabilistic rules and the 'degree of belief' in the dynamic bindings.

Another solution to the binding problem is based on frequency modulation (Tomabechi & Kitano 1989). In this approach, dynamic bindings may be encoded by having the appropriate nodes fire with the same frequency. There are advantages to encoding bindings using phases rather than frequencies. In the phase-based approach binding is represented by the simultaneity of activation and checking for simultaneous activations

is simpler and faster than checking that two signals have the same frequency. While the latter involves sampling the signal over a time interval spanning several periods of oscillation, the former only requires checking for simultaneity over one period of oscillation (cf. the design of a long-term fact in Section 4).

Barnden (1989) discusses 'Conposit', a connectionist production system. Central to Conposit is a scratch pad of registers called Configuration Matrices. A fairly complex interpreter reads the contents of these registers and updates them. In Conposit, patterns are associated by virtue of the *relative position* of registers that contain these patterns, as well as by virtue of the *similarity* between these patterns. The task of propagating argument bindings is performed by the interpreter. The use of an 'interpreter' leads to something akin the von Neumann bottleneck. Consequently, Conposit, like DCPS, only allows one rule to fire during each processing cycle. Conposit appears inappropriate for modeling reflexive reasoning but we believe that it is a plausible connectionist architecture for modeling reasoning that requires complex representations and elaborate rule applications.

Feldman (1982) addresses the problem of dynamically associating any element of a group of N entities with any element of another group of N entities, i.e., a total of N^2 bindings, using an interconnection network. He shows how it is possible to achieve the association task with an interconnection network having only $4N^{3/2}$ nodes as opposed to the obvious one of N^2 nodes.

Smolensky (1987) describes a connectionist network that uses a tensor product representation of dynamic bindings. The system encodes arguments (roles) and fillers as distributed patterns (vectors) over pools of argument and filler nodes, respectively. The proposed network requires N^2 nodes to encode N^2 bindings without cross-talk. The network can however, store a greater number of bindings if some cross-talk among bindings is acceptable.

Table 3 compares four connectionist rule-based reasoning systems along the following dimensions:

1. Nature of parallelism: How many rules can fire in a processing cycle, one or several.
2. Number of variables in a rule.
3. Reasoning speed: How long does it take to execute a processing cycle. In DCPS each rule application involves several cycles of relaxation, and hence, is a slow process. There is also significant serialization among the subprocesses underlying each reasoning step in Conposit. As ROBIN performs operations over signatures, we expect it to be marginally slower than our system.
4. Complexity of messages: Are messages scalars or addresses/pointers.
5. Complexity of nodes:
6. Network size: With reference to the number of rules and facts encoded in the system

9.1 Using Pattern containment for encoding dynamic bindings

An important aspect of the reasoning system proposed in this paper is the organization of rules into a directed graph wherein the inferential dependencies between antecedent and consequent predicates together with the correspondence between the arguments of these predicates are represented explicitly. This encoding in

conjunction with the temporal representation of dynamic bindings leads to an extremely efficient reasoning system. The proposed encoding of rules is significant in its own right. One may take this framework for organizing rules and obtain other organizationally and functionally isomorphic connectionist systems corresponding to alternate techniques for representing dynamic bindings. All such systems will share the same effective mechanism for propagating bindings inherent in the explicit organization of rules and arguments. These systems, however, will differ in the network size, reasoning speed, and biological plausibility. We illustrate this by using the proposed encoding of rules in conjunction with an alternate representation of dynamic bindings, namely, the *pattern-containment* approach (this approach was suggested by Geoff Hinton in a personal communication).

In the pattern-containment approach. each argument is represented by an ensemble of nodes and a pattern of activity is associated with each concept. A dynamic binding between a concept and an argument is represented by inducing the pattern of activation associated with the concept in the argument ensemble. With reference to the network in Fig. 27, the dynamic bindings corresponding to $give(John, Mary, Book1)$ will be represented by the presence of the patterns corresponding to $John$, $Mary$ and $Book1$ in the argument ensembles *giver*, *recipient*, and *give-object*, respectively. Contrast this with the temporal encoding approach where these bindings would be represented by having the nodes in the argument ensembles *giver*, *recipient*, and *give-object* oscillate in phase with the nodes (or ensembles) representing $John$, $Mary$, and $Book1$, respectively (refer to Section 5.2).

The propagation of bindings in the pattern-containment approach will occur exactly as it would in the temporal encoding approach. Links between nodes in the argument ensembles of the antecedent and consequent predicates will propagate the pattern of activity present along interconnected argument ensembles resulting in the propagation of bindings.

It is instructive to compare the pattern containment approach with the temporal synchronization approach. The key question that must be asked is: 'What is the significance of the pattern of activity that is associated with a concept and propagated across argument ensembles?' One possibility is that this pattern encodes all the micro-features of the concept (Hinton, 1981; Rumelhart and McClelland, 1986). Such a pattern, however, will be extremely large, and result in an inefficient use of computational resources: each argument ensemble will have to be large enough to encode the entire pattern of micro-features associated with concepts, and propagating bindings will involve propagating large patterns consisting mostly of irrelevant information. A more feasible alternative would be to assume that the patterns associated with concepts are *reduced descriptions*. But then the temporal synchronization approach already makes use of reduced descriptions — albeit of a very unusual sort. During the propagation of bindings, the phase associated with a concept acts as a reduced description — a very reduced one at that — of the concept!

10 Conclusion

We have described a solution to the dynamic (variable) binding problem. The solution involves maintaining and propagating variable bindings by propagating rhythmic patterns of activity wherein bindings are represented as the in-phase oscillations of appropriate nodes. We have also described how this solution may

be incorporated into a connectionist system that can represent long-term knowledge expressed in terms of facts and rules involving n-ary relations and variables and perform a limited class of reasoning based on this knowledge with extreme efficiency. The time taken by the proposed system to draw an inference is only proportional to the *length* of the chain of inference and is independent of the number of rules and facts encoded by the system. Thus the system offers a computational account of a limited class of reflexive reasoning.

The proposed system obeys all the constraints applicable to connectionist models. In particular, the system uses simple phase-sensitive binary threshold units that exchange simple 1-bit messages and the system operates without a global clock. The number of nodes required to encode knowledge is only *linear* in the size of the knowledge-base and the representation of dynamic bindings does not require the existence of hard-wired connections between all possible argument-filler pairs.

We have outlined several extensions of the proposed system. These include dealing with several dynamic instantiations of the same predicate, extracting argument bindings to answer *wh*-queries, interfacing the rule-based system with an *IS-A* hierarchy so that individuals as well as abstract concepts may occur within facts and rules, admitting function-terms in rules and facts, and encoding soft (probabilistic) rules instead of purely logical rules. In the near future we envision the design of a system that will perform a range of deductive and abductive inferences — make predictions and generate explanations, and do so with remarkable efficiency.

We believe that the work described in this paper is a reasonable demonstration of how a system made of slow and simple computing elements with switching times and periods of oscillations in the msec. range, can encode millions of facts and rules and yet perform systematic inferences involving chains of reasoning in the hundred msec. range.

The work also demonstrates that mechanisms implicated in the solution of low-level sensorimotor tasks are perhaps *sufficient* to model high-level cognitive functions such as systematic thought. We consider this to be of significance to the study of cognition.

It has often been argued that a deep understanding of intelligence will accrue only if we adopt an integrated approach that synthesizes computational, behavioral, as well as neurobiological issues. We hope that the work described in this paper is a small step in this direction.

11 Acknowledgements

We wish to thank the AI Group at ICSI, Berkeley — in particular, Jerry Feldman, Mark Fanty and Bob Wilensky; R. Chandrasekar, Pat Hayes, Geoff Hinton, George Gerstein and Simon Thorpe whose comments, suggestions, and criticisms are greatly appreciated. Also thanks to Denis Dancanet for drawing the numerous figures that appear in this paper. This research was supported by NSF grants IRI 88-05465, MCS-8219196-CER, MCS-83-05211, DARPA grants N00014-85-K-0018 and N00014-85-K-0807, and ARO grant ARO-DAA29-84-9-0027.

End Notes

1. One could argue that some of the steps in the above reasoning process are pre-compiled or 'chunked'. However, it would be preposterous to claim that the entire chain of reasoning given above can be construed as retrieval or even a single step inference!

2. A reference to a single node may be replaced by a reference to a small ensemble of nodes (See Section 5.2).

3. CSN performs evidential or probabilistic reasoning and can deal with exceptions and multiple inheritance situations stemming from default attribute values. However, these aspects of the system are not relevant here.

4. In this example we only consider rules that have a single antecedent and a single consequent. We do this to keep our discussion simple and the observations apply to complex rules as well.

5. Fast synapses that can modify their weights in the hundred millisecond range have been proposed by von der Malsburg (1986).

6. By 'entity' we mean both, individual concepts such as 'John' as well as abstract concepts such as 'Dog'. With reference to formal logic, entities correspond to *terms*.

7. In fact these bindings *can* only be represented temporarily! In view of points 1 and 2 the number of such inferred facts would be enormous and the system will run out of memory rather soon if it had to remember all the intermediate results.

8. If one is willing to accept the existence of some independent mechanism for sustaining phase-sensitive oscillations then it is easy to envision a system in which the firing frequency of a node varies during each episode of reasoning and is governed by the number of distinct individuals participating in the bindings. In earlier reports we had described such a realization that used a global clock to provide the necessary control. (Shastri & Ajjanagadde 1989; Ajjanagadde & Shastri 1989).

9. Thorpe and Imbert (1988) argue that even the number of visually identifiable objects is upwards of a hundred thousand.

10. Link weights will play a crucial role in a system that encodes evidential rules (see Section 8.4).

11. The assumption that signal propagation delays are zero is realistic in case of electrically coupled neurons (Llinas et al. 1974). Encoding a rich body of conceptual knowledge, however, would require connecting spatially distant neurons and it is unlikely that electrical coupling can be effective over the required distances.

12. A cross-interval histogram of the spike trains produced by two cells, A and B, is obtained as follows (Gerstein 1970): Time intervals are measured from a given spike of train A to the nearest preceding and succeeding spikes of train B. This is repeated using each spike of train A as origin, and a histogram is compiled.

13. The encoding described in Section 7 naturally detects multiple dynamic instantiations of a predicate – no addition machinery is required.

14. Given the rules $\forall x, y\ P(x, x, y) \Rightarrow R(x, y)$ and $\forall x, y\ R(x, y) \Rightarrow Q(y, x)$, the system will respond correctly to queries such as $R(a, b)$, $R(a, y)$, $Q(a, b)$ and $Q(y, a)$ because the bindings specified in these query result in the repeated variable (x) of P getting bound. The system *may* respond incorrectly to queries such as $R(x, b)$, $R(x, y)$, and $Q(a, y)$ as these leave x unbound.

15. Any reasoning system not subject to this limitation would be capable of handling recursion to arbitrary depths — something that is known to be computationally intractable.

16. There exists an alternate solution that uses only a single stage. This involves assigning distinct phases to unbound as well as bound arguments and using two different levels of activation, one for bound arguments/constants and another for unbound arguments.

17. The use of phase to encode entities when generalized to concepts in an *IS-A* hierarchy may shed some light on the ***attribute binding problem*** discussed in (Stenning et al. 1988). Although the argument binding and the attribute binding problems are related, they differ in significant ways. The attribute binding problem concerns phenomena that occur over relatively long time intervals and do not involve any *inference*: a subject reads a sequence of sentences over several seconds in a self paced mode before attempting to recall the attribute bindings. The argument binding problem involves propagating argument bindings ***very rapidly*** in order to support inferences that must occur within hundreds of milliseconds. Attribute bindings, on the other hand, are influenced by — and must interact with — existing semantic information in the form of attribute values associated with existing concepts in the *IS-A* hierarchy.

18. In (Shastri, 1988a; Shastri,1988b) we describe a connectionist semantic memory that solves an interesting class of inheritance and recognition problems based on an *evidential* treatment of knowledge. The system deals with ***exceptional*** and ***conflicting*** multiple inheritance situations as well as ***best match*** or ***partial match*** situations during recognition.

References

Abeles, M., (1982). *Local Cortical Circuits: Studies of Brain Function* vol. 6. Springer, New York.

Ajjanagadde, V.G. and Shastri, L. (1989). Efficient inference with multi-place predicates and variables in a connectionist system, *Proceedings of the Cognitive Science Conference*, August, pp. 396-403.

Allen, J.A. and Frisch, A.M. (1982). What is in a Semantic Net? In *Proceedings ACL-82*, Toronto.

Barnden, J. (1989). Neural-net implementation of complex symbol-processing in a mental model approach to syllogistic reasoning, *Proceedings of IJCAI-89*, pp. 568-573.

Churchland, P.S. (1986). Neurophilosophy: Toward a Unified Science of the Mind/Brain. The MIT Press, Cambridge, MA.

Clossman G. (1988). A model of categorization and learning in a connectionist broadcast system. Ph.D. Dissertation, Department of Computer Science, Indiana University.

Crick, F. (1984). Function of the thalamic reticular complex: The searchlight hypothesis. *PNAS*, Vol. 81, pp. 4586-4590.

Dolan, C., and Dyer, M. (1988). Parallel retrieval and application of conceptual knowledge, Technical Report TR UCLA-AI-88-3, University of California, Los Angeles, January.

Fahlman, S. (1979). *NETL: A system for representing real-world knowledge*, MIT Press, Cambridge, MA.

Fanty, M.A. (1988). Learning in Structured Connectionist Networks. Ph.D. Dissertation, Computer Science Department, University of Rochester, Rochester, NY.

Feldman, J.A. (1989). Neural Representation of Conceptual Knowledge. In *Neural Connections, Mental Computation.* L. Nadel, L.A. Cooper, P. Culicover, and R.M. Harnish (eds). The MIT Press, Cambridge, MA.

Feldman, J.A. (1982). Dynamic connections in neural networks, *Bio-Cybernetics*, 46:27-39.

Feldman, J.A. and Ballard D.H. (1982). Connectionist models and their properties. *Cognitive Science*, 6 (3): 205-254.

Fodor, J.A. and Pylyshyn Z.W. (1988). Connectionism and cognitive architecture: A critical analysis. In *Connections and Symbols* S. Pinker and J. Mehler (eds.) The MIT Press, Cambridge, MA.

Freeman, W.J. (1981). A physiological basis of perception. In *Perspectives in Biology and Medicine*, Univ. of Chicago Press.

Frisch, A.M. and Allen, J.F. (1982). Knowledge retrieval as limited inference. In D.W. Loveland(Ed.), Lecture Notes in Computer Science: 6th Conference on Automated Deduction, Springer-Verlag, New York.

Gray, C.M. and Singer, W. (1989). Stimulus-specific neuronal oscillations in orientation specific columns of cat visual cortex. *PNAS*, Vol. 86, pp. 1698-1702.

Gerstein, G.L. (1970). Functional Association of Neurons: Detection and Interpretation. In *The Neurosciences: Second Study Program.* (ed) F.O. Schmitt. The Rockfeller University Press. New York.

Hebb, D.O. (1949). *The Organization of Behavior.* Wiley, New York.

Hinton, G.E. (1981). Implementing semantic networks in parallel hardware, In G.E. Hinton and J.A. Anderson (Eds.), *Parallel Models of Associative Memory*, Erlbaum.

Kandel, E.R. (1976). *The Cellular Basis of Behavior.* Freeman, San Francisco, CA.

Lange, T., and Dyer, M. (1989). High-Level inferencing in a Connectionist Neural Network. Technical Report, UCLA-AI-89-12, October. Computer Science Department, UCLA.

Llinas, R.R., Baker, R. and Sotelo, C. (1974). Electronic coupling between neurons in the cat inferior olive. *Journal of Neurophysiology.* 37: 560-571.

von der Malsburg, C. (1986). Am I thinking assemblies? In *Brain Theory.* In G. Palm and A. Aertsen (Eds). Springer-Verlag, Berlin Heidelberg. (Also Internal Report 81-2. Department of Neurobiology, Max-Planck-Institute for Biophysical Chemistry, Gottingen, FRG. 1981.)

MacVicar B. and Dudek, F.E. (1980). Dye-coupling between CA3 pyramidal cells in slices of rat hippocampus. *Brain Research*, 196: 494-497.

McCulloch W.S. and Pitts, W. (1988). A logical calculus of the ideas immanent in nervous activity. In J.A. Anderson and E. Rosenfeld (eds.), *Neurocomputing: Foundations of Research.* MIT Press, Cambridge, MA.

Miller, G.A. (1956). The magical number seven, plus or minus two: Some limits on our capacity for processing information, *The Psychological Review*, 63(2), March, pp. 81-97.

Quillian, M.R. (1968). Semantic Memory. In em Semantic Information Processing, (Ed.) M. Minsky, MIT Press, Cambridge, MA.

Rumelhart, D.E. and McClelland, J.L. (1986). (Eds.) *Parallel Distributed Processing: Explorations in the Microstructure of Cognition.* Vol I. Cambridge, MA: Bradford Books/MIT Press.

Schubert, L.K. (1989). An Episodic Knowledge Representation for Narrative Texts. In *Proceedings of KR-89.* Toronto, May.

Sejnowski, T.J. (1981). Skeleton filters in the brain. In G.E. Hinton and J.A. Anderson (eds.), *Parallel models of associative memory*, Erlbaum.

Shastri, L. and Feldman. J.A. (1986). Semantic Nets, Neural Nets, and Routines. In *Advances in Cognitive Science*, (Ed.) N. Sharkey, Ellis Harwood Limited/John Wiley & Sons, Chichester, UK.

Shastri, L. (1990a). Connectionism and the Computational Effectiveness of Reasoning. To appear in *Theoretical Linguistics.*

Shastri, L. (1990b). The Relevance of Connectionism to AI: A representation and reasoning perspective. In *Advances in Connectionist and Neural Computation Theory*, vol. 1., J. Barnden (ed.), Ablex Publishing Company, Norwood, N.J. (To appear).

Shastri, L. (1988a). *Semantic networks : An evidential formulation and its connectionist realization*, Pitman London/ Morgan Kaufman Los Altos.

Shastri, L. (1988b). A connectionist approach to knowledge representation and limited inference, *Cognitive Science*, 12(3), pp. 331-392.

Shastri, L. and Ajjanagadde, V. (1989). A connectionist system for rule based reasoning with multi-place predicates and variables, Tech. Report MS-CIS-8905, Dept. of Computer Science, Univ. of Pennsylvania, January.

Smolensky, P. (1988). Proper treatment of Connectionism, *Behavioral and Brain Sciences*, 11:1.

Smolensky, P. (1987). On variable binding and the representation of symbolic structures in connectionist systems, Technical Report CU-CS-355-87, Department of Computer Science, University of Colorado at Boulder.

Strong, G.W. and Whitehead, B.A. (1989). A solution to the tag-assignment problem for neural nets. *Behavioral and Brain Sciences*, 12, 381-433.

Stenning, K., Shepard, M. and Levy J. (1988). On the Construction of Representations for Individuals from Descriptions in Text. *Language and Cognitive Processes*, 3(2), pp. 129-164.

Thorpe, S.J. and Imbert, M. (1988). Biological constraints on connectionist modelling. In *Connectionism in Perspective*, R. Pfeiffer (ed). Springer.

Tomabechi, H. and Kitano, H. (1989). Beyond PDP: The Frequency Modulation Neural Network Approach. In *Proceedings IJCAI-89*, Detroit, August.

Touretzky, D. and Hinton, G.E. (1988). A Distributed Connectionist Production System. *Cognitive Science*, 12(3), pp. 423-466.

Toyama, K., Kimura, M., and Tanaka, T. (1981). Cross correlation analysis of interneuronal connectivity in cat visual cortex, *Journal. of Neurophysiology*, 46(2), December, pp. 191-201.

Weber S.H. (1989). A structured connectionist approach to direct inferences and figurative adjective-noun combinations. Technical Report 289, Computer Science Department, University of Rochester, May.

Table 1: Interpretation of some queries and their answers.

QUERY	yes-no-form (answer)	wh-form (answer)
$own(Mary, Ball1)$	Does Mary own Ball1? (yes)	-
can-sell(Mary,Book1)	Can Mary sell Book1? (yes)	-
can-sell(Mary,x)	Can Mary sell something? (yes)	What can Mary sell? (Book1, Ball1)
$own(x, y)$	Does someone own something? (yes)	Who owns something? (Susan,Mary,John) What is owned by someone? (Book1,Ball1,Ball2,Car7)
can-sell(John,x)	Can John sell something? (yes)	What can John sell? (Car7)
$present(x, Northpole, 1/1/89)$	Was someone present at northpole on 1/1/89? (yes)	Who was present at northpole on 1/1/89? (There was someone; but, don't know who)
number-of-sides(A3,4)	Does A3 have 4 sides? (no)	-
can-sell(Mary,Ball2)	Can Mary sell Ball2? (no)	-
$twins(Susan, Mary)$	Are Mary and Susan twins? (yes)	-

Table 2: Some queries and answers with reference to the rule-base and *IS-A* hierarchy shown in Fig. 26.

QUERY	**yes-no-form (answer)**	**wh-form (answer)**
$preys\text{-}on(Sylvester, Tweety)$	Does Sylvester prey on Tweety? (yes)	-
scared-of(Tweety,Sylvester)	Is Tweety scared of Sylvester? (yes)	-
scared-of(x,Sylvester)	Is someone scared of Sylvester? (yes)	Who is scared of Sylvester? (Bird)
$preys\text{-}on(Cat, Bird)$	Do cats prey on birds? (yes)	-
scared-of(Bird,Cat)	Are birds scared of cats? (yes)	-

Table 3: A comparison of four connectionist rule-based reasoning systems

	Is it parallel at the rule-level	Number of variables in a rule	Reasoning speed	Complexity of messages	Complexity of nodes	Size of the network
Our System	Yes	Virtually unlimited	Very fast	Simple messages	Simple nodes	Small
ROBIN (Lange & Dyer	Yes	Virtually unlimited	Fast	Complex messages	Complex nodes that can process signatures	Small
COMPOSIT (Barnden	No	Virtually unlimited	Medium	Mostly simple messages	Simple nodes	Large
DCPS (Touretzky & Hinton)	No	One	Slow	Simple messages	Simple nodes	Large

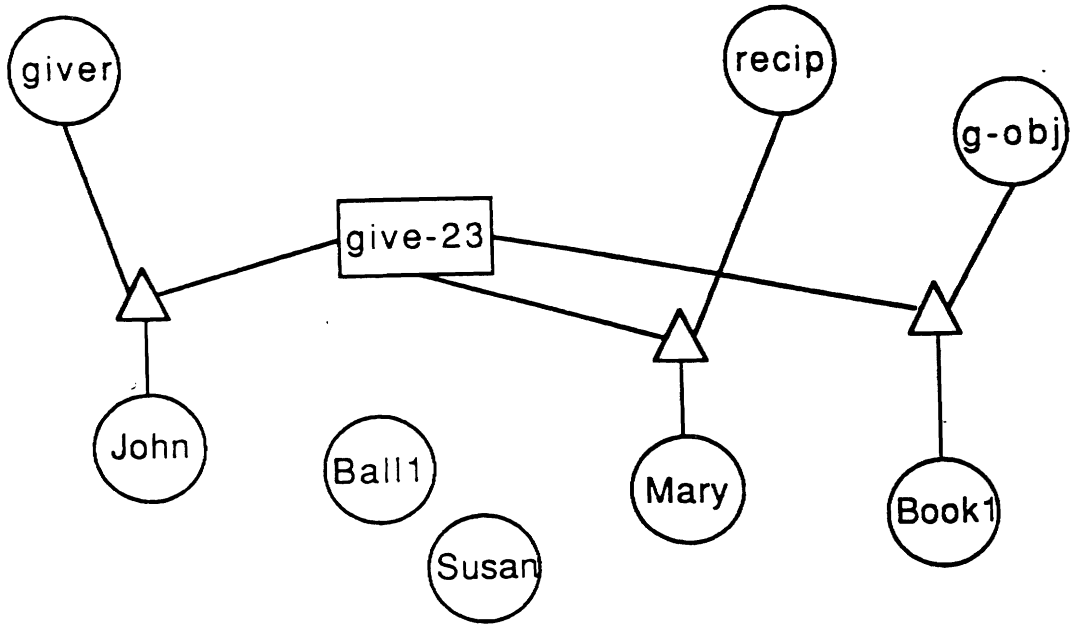

Fig. 1: Encoding static bindings using dedicated nodes and links. *give*23 is a *focal* node and the triangular nodes are *binder* nodes.

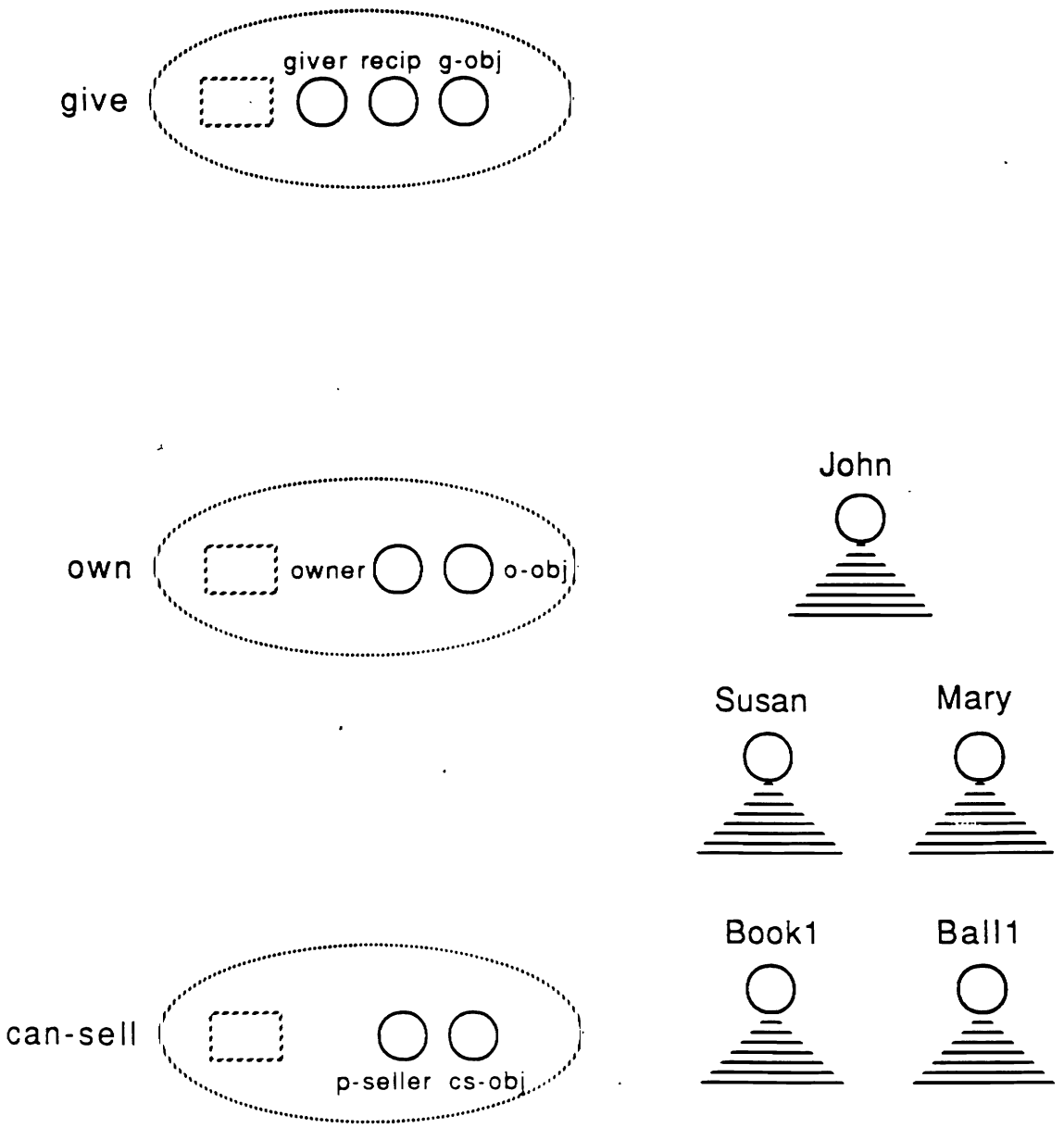

Fig. 2: Encoding predicates and individual concepts. Distinct predicates and arguments are encoded using distinct nodes. Each circular node in the figure can either be a single node or an ensemble of nodes.

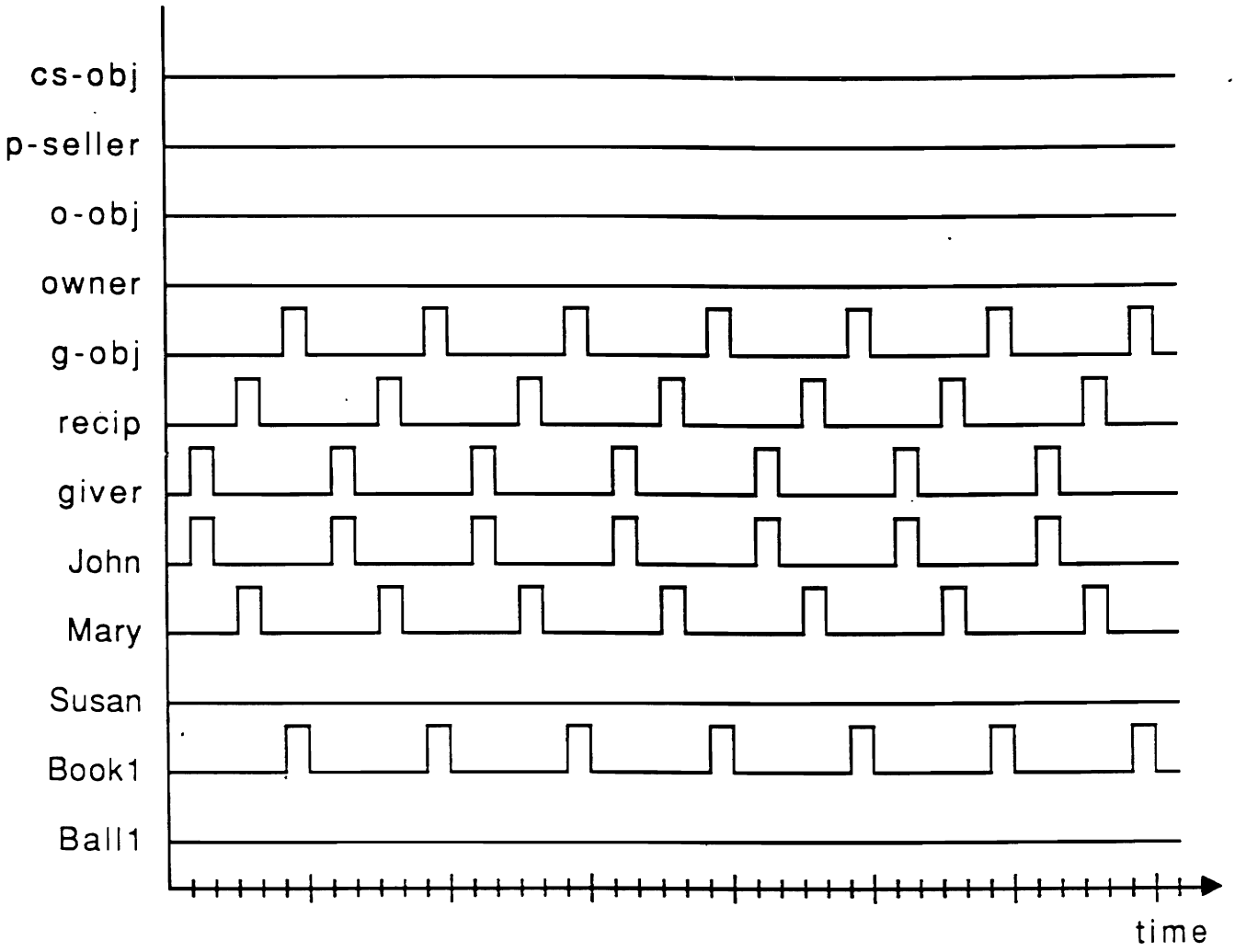

Fig. 3: Rythmic pattern of activation representing the dynamic bindings ($giver = John, recipient = Mary, give\text{-}object = Book1$). These bindings constitute the fact $give(John, Mary, Book1)$. All active nodes fire with the same frequency but occupy distinct phases in the firing pattern. The binding between an argument and a filler is represented by the in-phase firing of associated nodes.

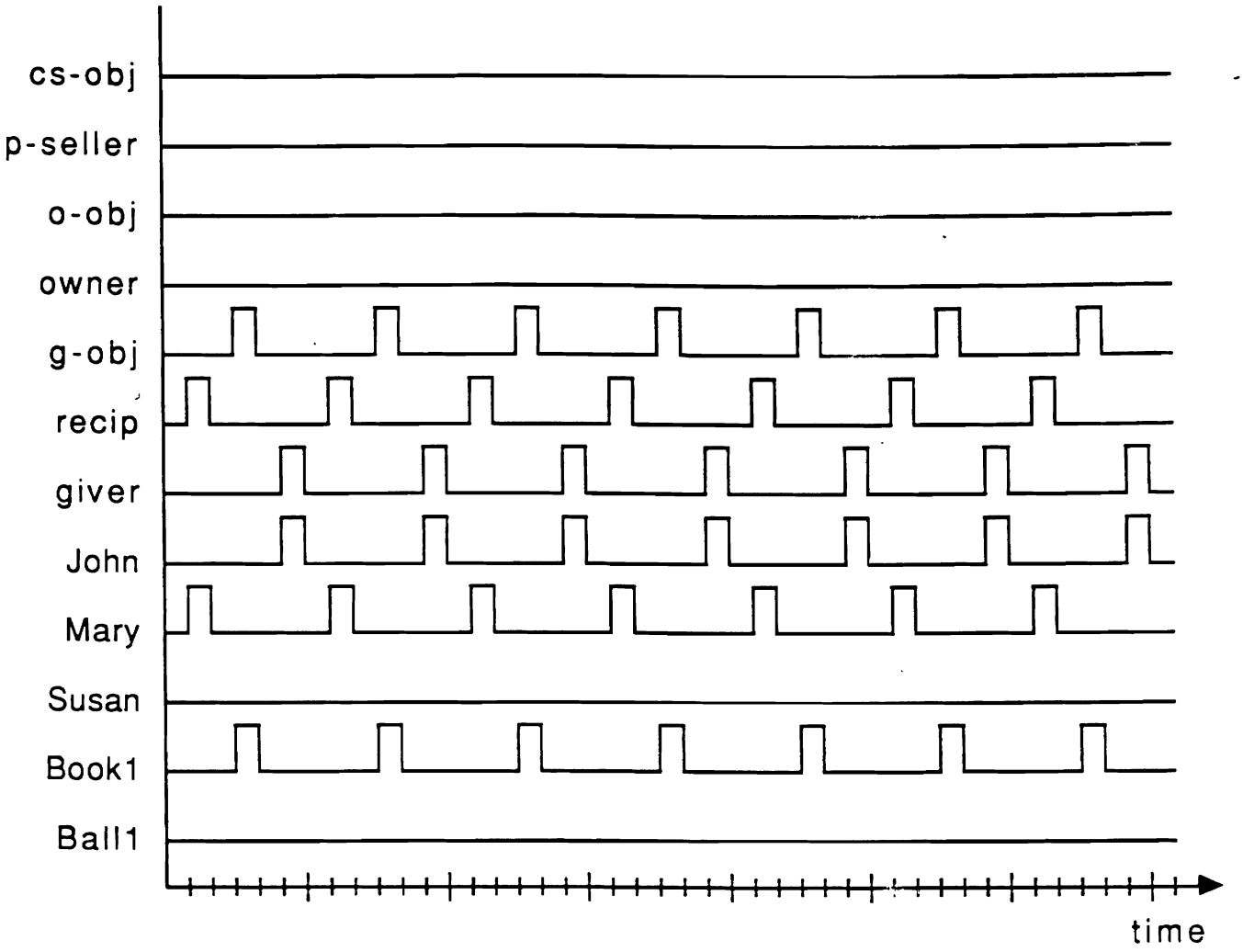

Fig. 4: An alternate representation of the bindings ($giver = John, recipient = Mary, give\text{-}object = Book1$). The particular phase occupied by a filler and argument nodes is not critical.

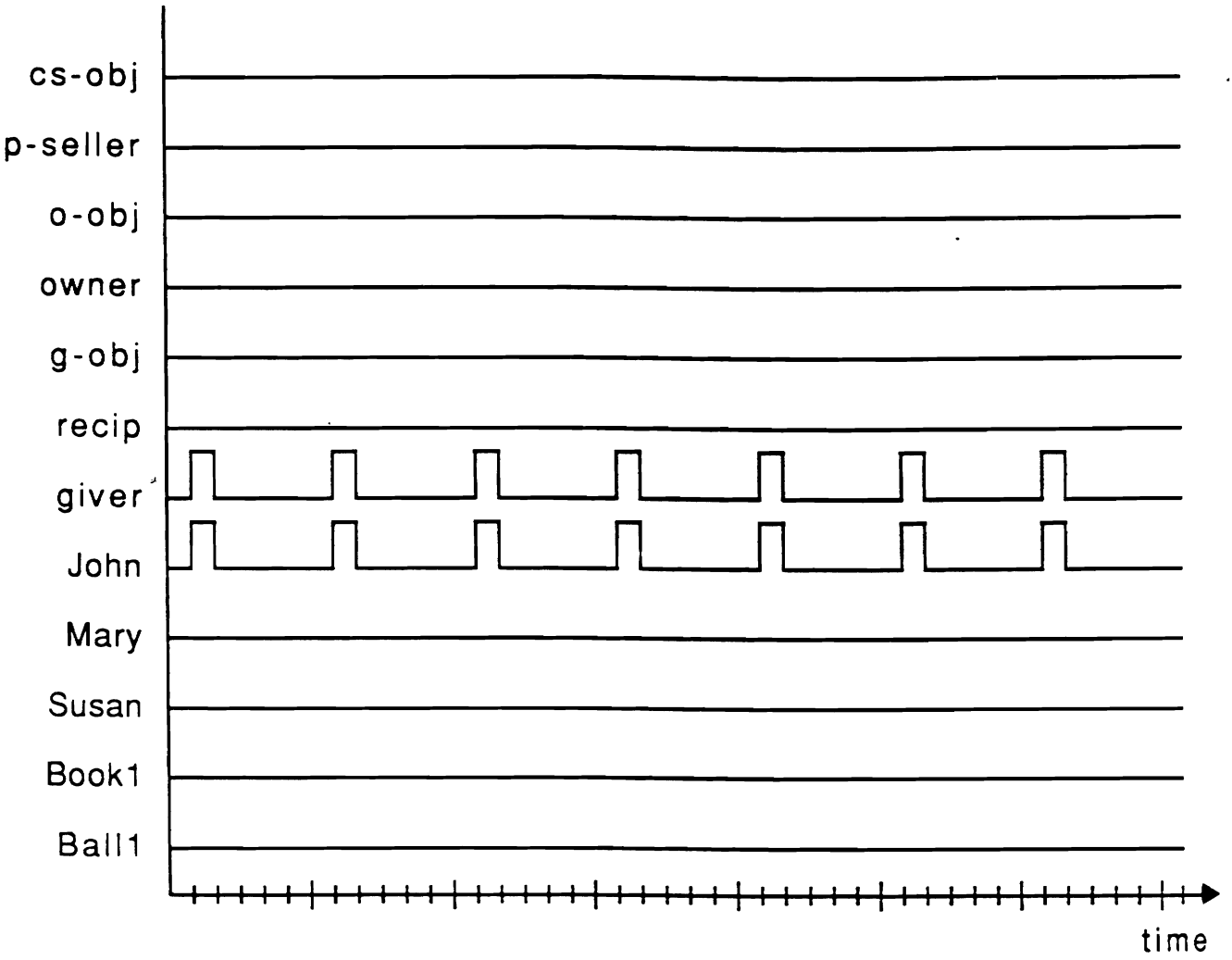

Fig. 5: Representation of the dynamic binding (*giver* = *John*) that constitutes the partially instantiated fact 'John gave someone something'.

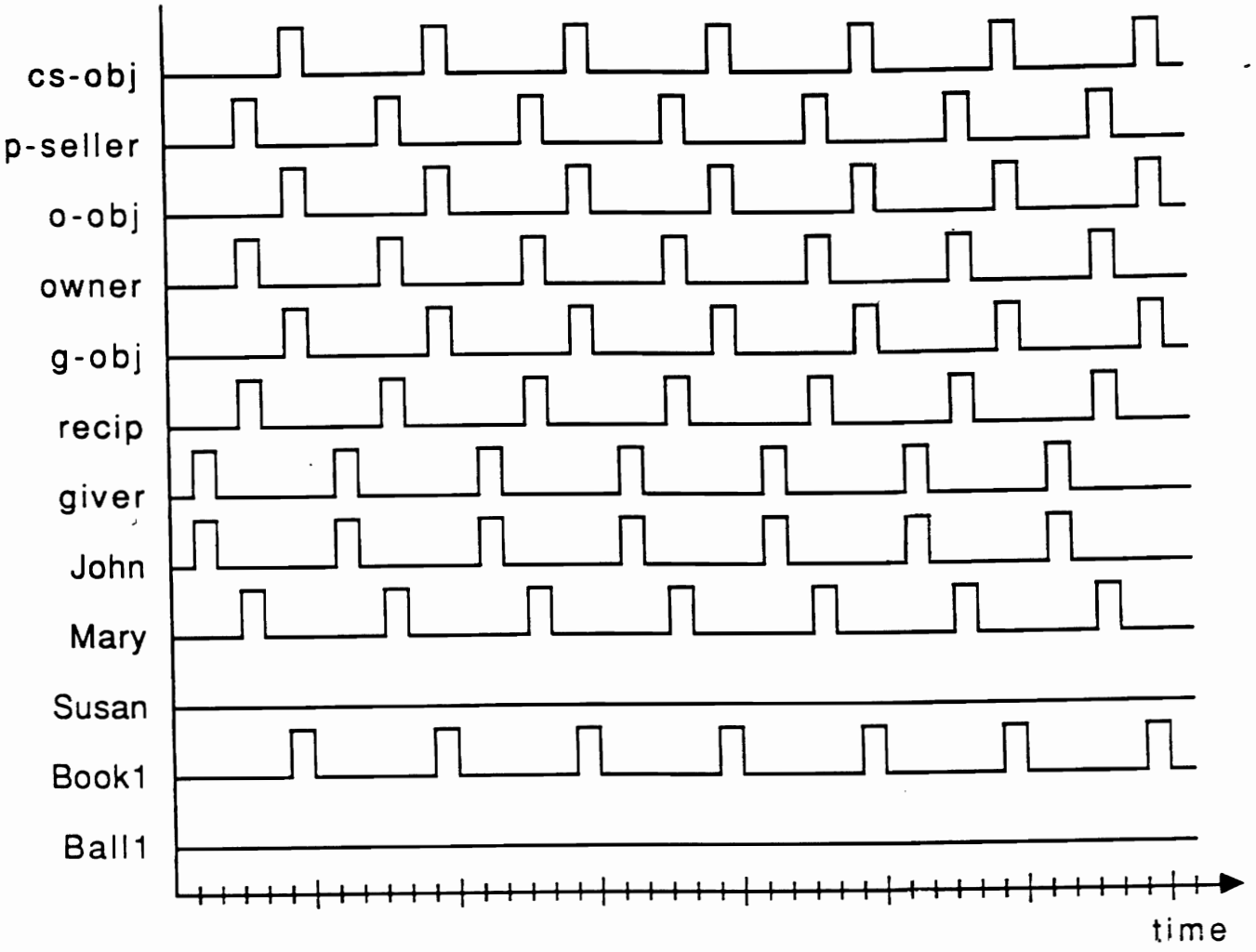

Fig. 6: Pattern of activation representing the dynamic bindings (*giver* = *John*, *recipient* = *Mary*, *give-object* = *Book*1, *owner* = *Mary*, *own-object* = *Book*1, *potential-seller* = *Mary*, *can-sell-object* = *Book*1). These bindings constitute the facts *give*(*John*, *Mary*, *Book*1), *own*(*Mary*, *Book*1), and *can-sell*(*Mary*, *Book*1). The bindings between *Mary* and the arguments *recipient*, *owner*, and *potential-seller* are represented by the in-phase firing of the appropriate nodes. The transient representation of an entity is simply a *phase* or time-slice within an oscillatory pattern of activity. The number of *distinct* phases required to represent a set of dynamic bindings only equals the number of *distinct individuals* participating in the bindings and is independent of the *total number of bindings*. In this example, only three distinct phases are required to represent seven bindings.

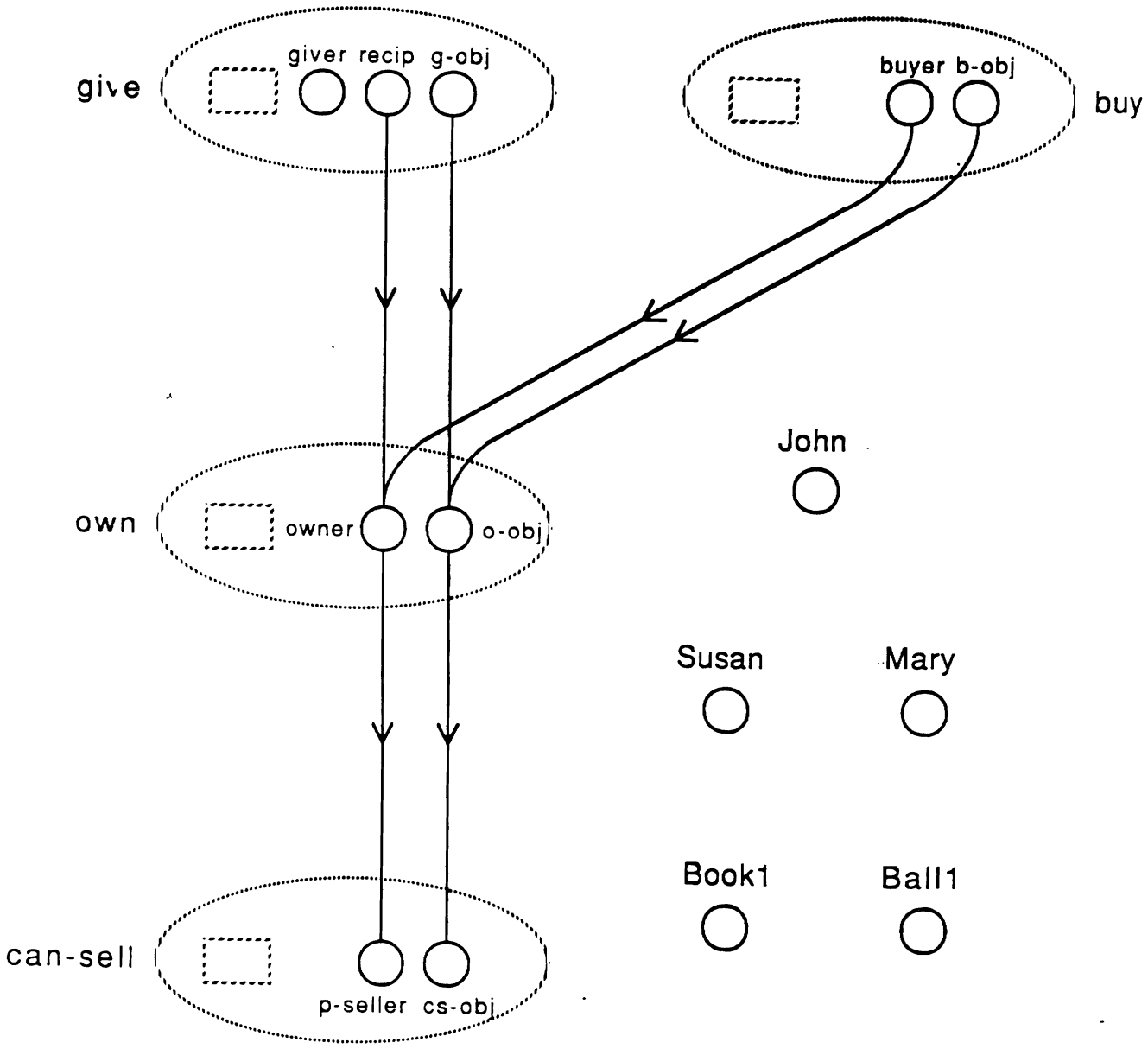

Fig. 7: Encoding of predicates, individual concepts, and the rules: $\forall x, y, z\ [give(x, y, z) \Rightarrow own(y, z)]$, $\forall x, y\ [own(x, y) \Rightarrow can\text{-}sell(x, y)]$, and $\forall x, y\ [buy(x, y) \Rightarrow own(x, y)]$. Links between arguments reflect the correspondence between arguments in the antecedents and consequents of rules.

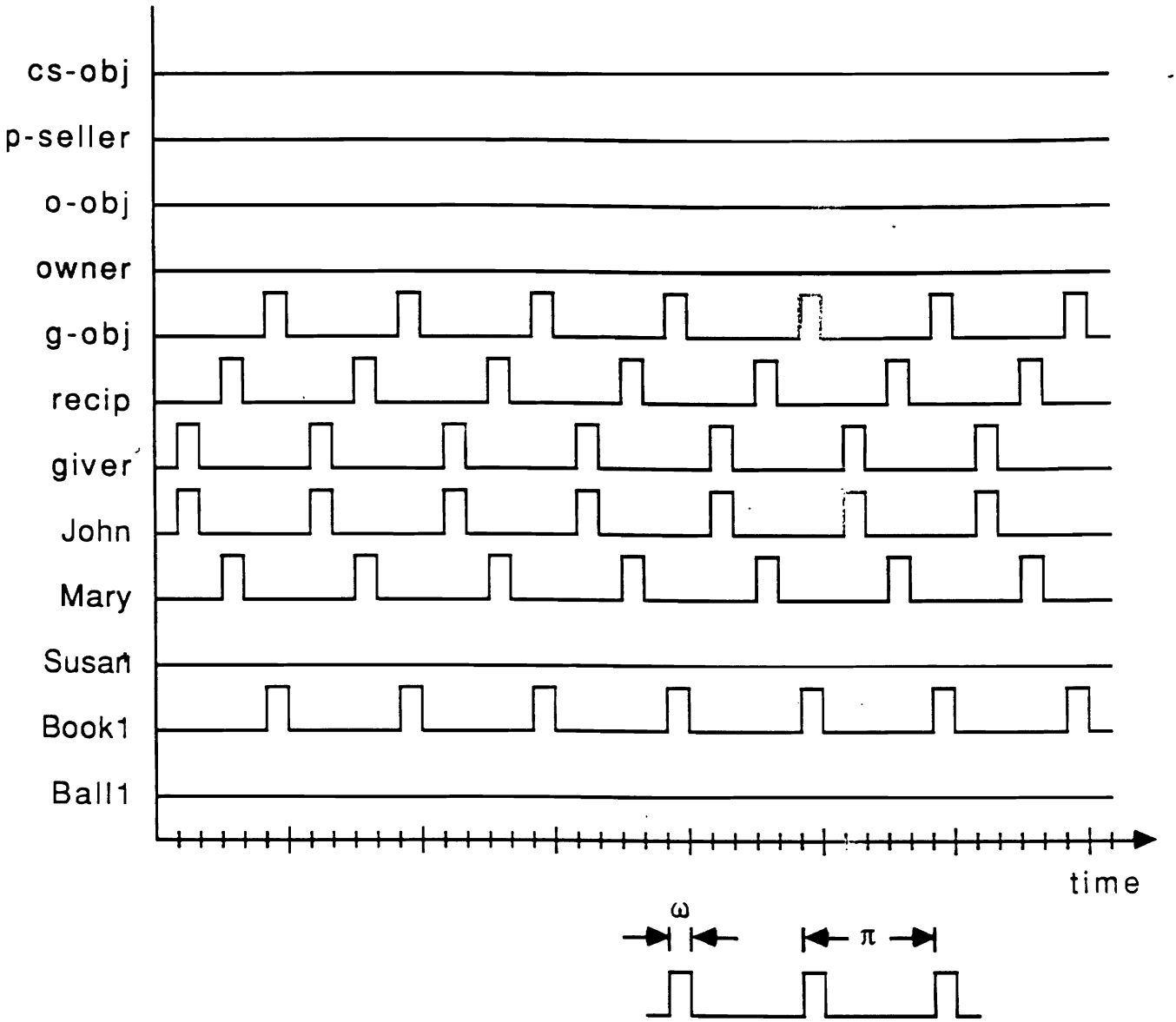

Fig 8: Initial pattern of activation representing the bindings (*giver* = *John*, *recipient* = *Mary*, *give-object* = *Book*1)

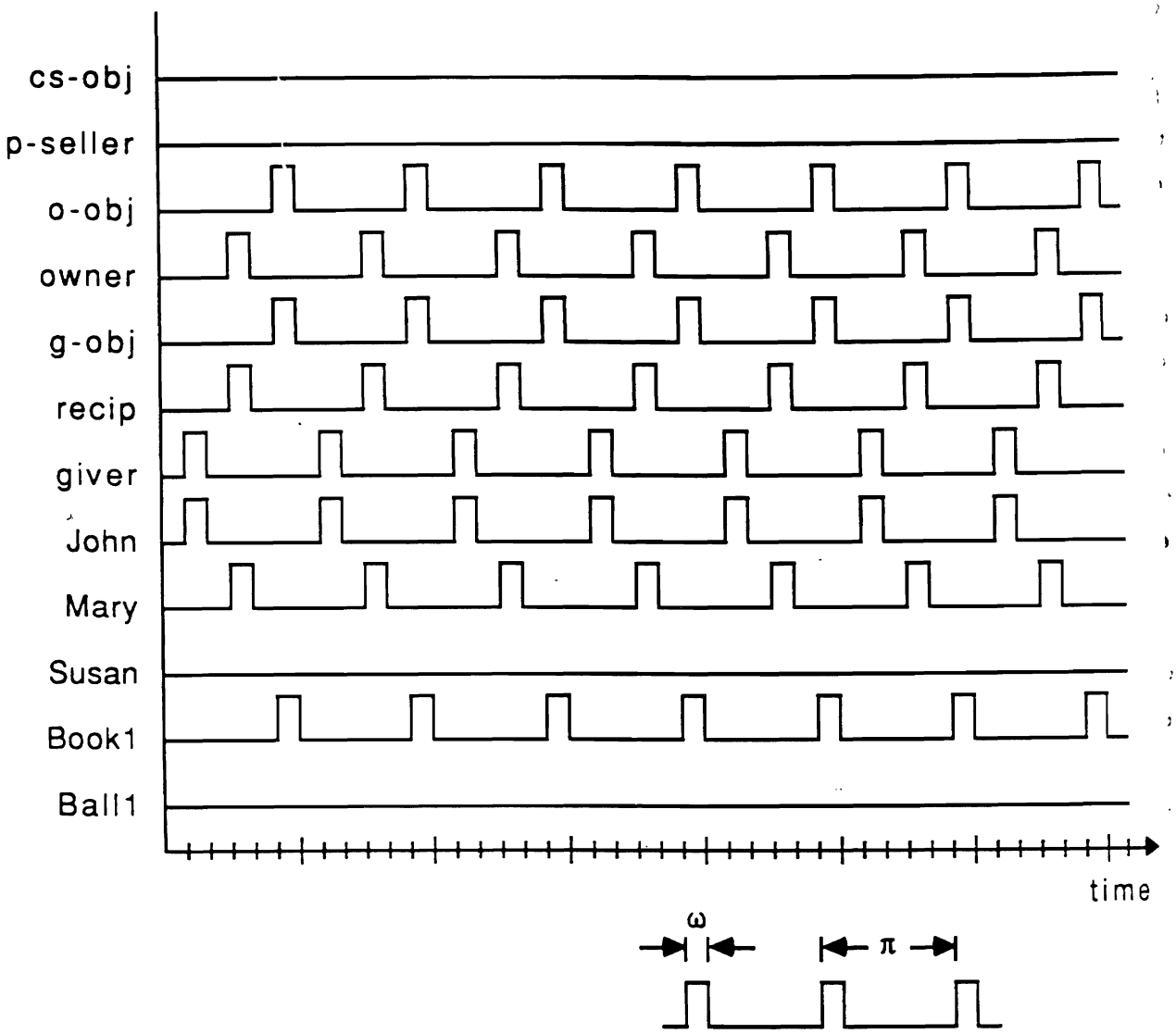

Fig. 9: Pattern of activation after one period of oscillation (with reference to the state of activation in Fig. 8). This state represents the dynamic bindings: $(giver = John, recipient = Mary, give\text{-}object = Book1, owner = Mary, own\text{-}object = Book1)$. The system has essentially inferred the fact $own(Mary, Book1)$.

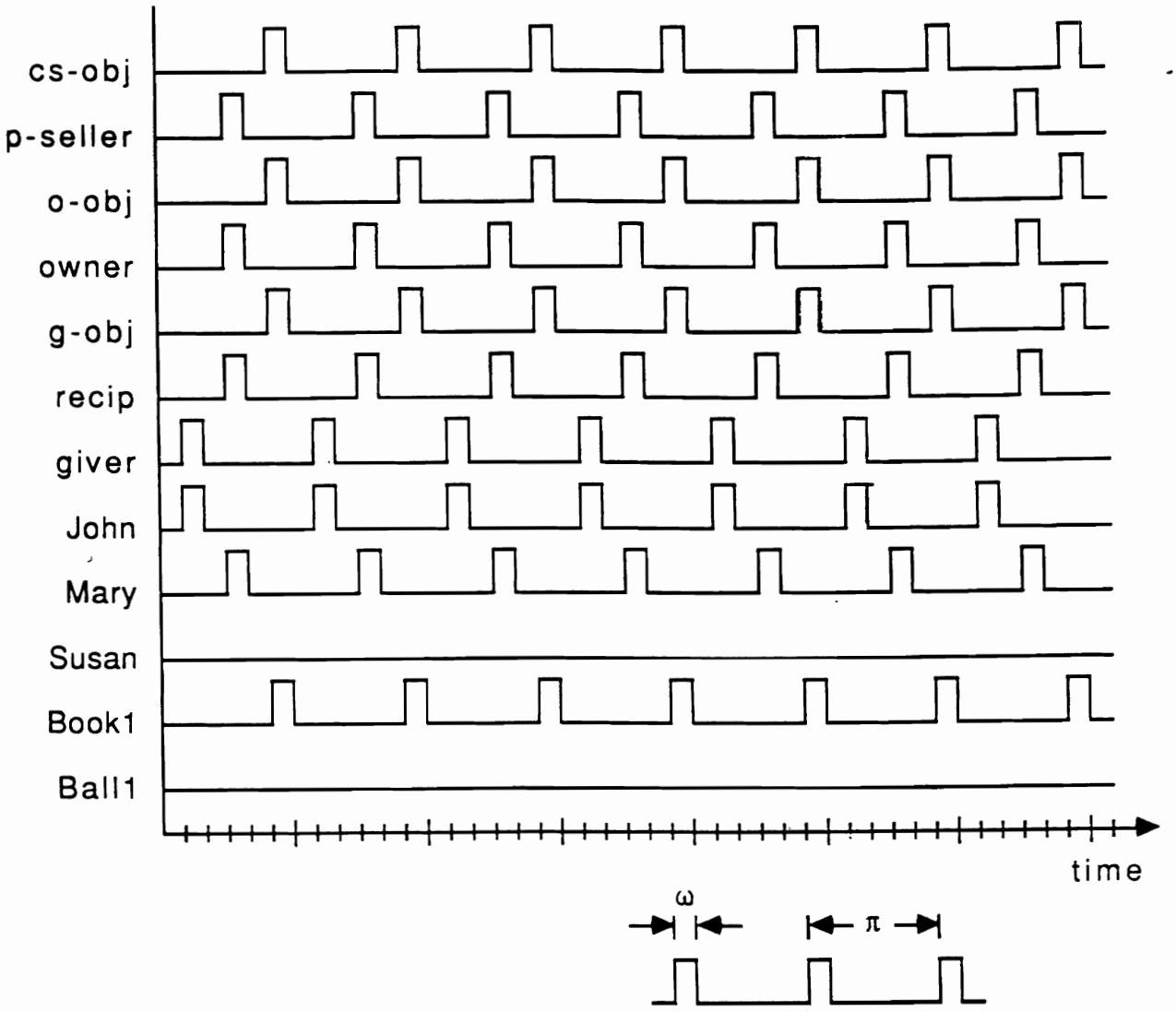

Fig. 10: Pattern of activation after two periods of oscillation (with reference to the state of activation in Fig. 8). This state represents the dynamic bindings: $(giver = John, recipient = Mary, give\text{-}object = Book1, owner = Mary, own\text{-}object = Book1, potential\text{-}seller = Mary, can\text{-}sell\text{-}object = Book1)$. The system has essentially inferred the facts $own(Mary, Book1)$ and $can\text{-}sell(Mary, Book1)$.

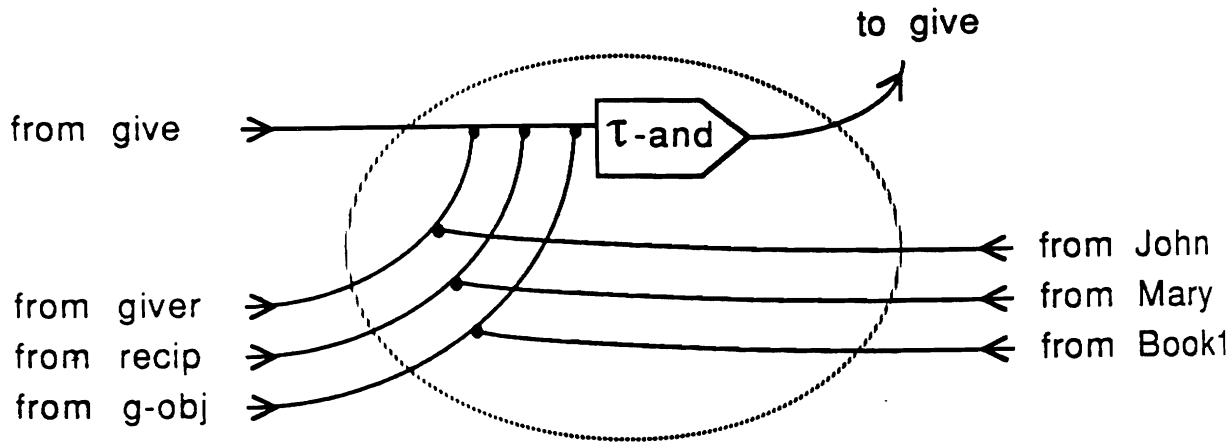

Fig. 11: Encoding of long-term facts. The interconnections shown here encode the *static* bindings ($giver = John, recipient = Mary, give\text{-}object = Book1$) that constitute the long-term fact $give(John, Mary, Book1)$. The pentagon shaped node in a τ-and node. Such a node becomes active if it receives uninterrupted input for the duration of its period of oscillation.

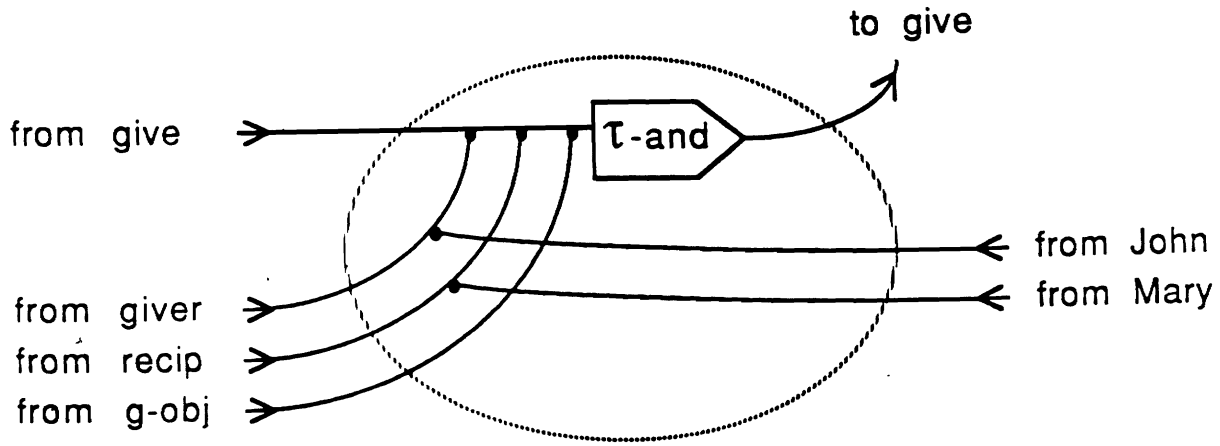

Fig. 12: Encoding of the partially instantiated long-term fact $give(John, Mary, x)$, i.e., 'John gave Mary something'. The input from *g-obj* does not receive a inhibitory input from any filler.

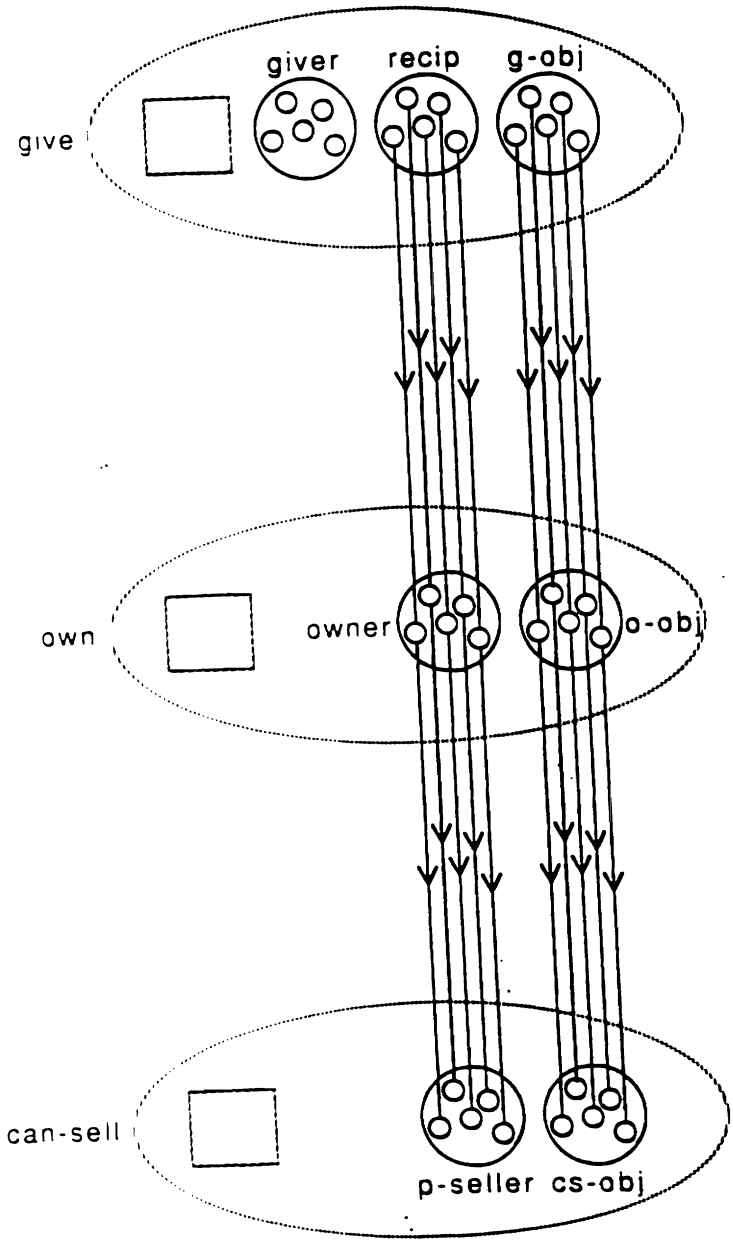

Fig. 13: Ensemble based encoding of the rules $\forall x, y, z[give(x, y, z) \Rightarrow own(y, z)]$ and $\forall x, y[own(x, y) \Rightarrow can\text{-}sell(x, y)]$. Arguments are encoded as ensembles of nodes rather than single nodes. A link between argument nodes is replaced by a bundle of links between appropriate argument ensembles. The dynamic binding between an argument and its filler is encoded by the in-phase activation of the appropriate filler and argument ensembles. Propagation of bindings is is realized via links between appropriate argument ensembles that cause nodes in these ensembles to fire in synchrony.

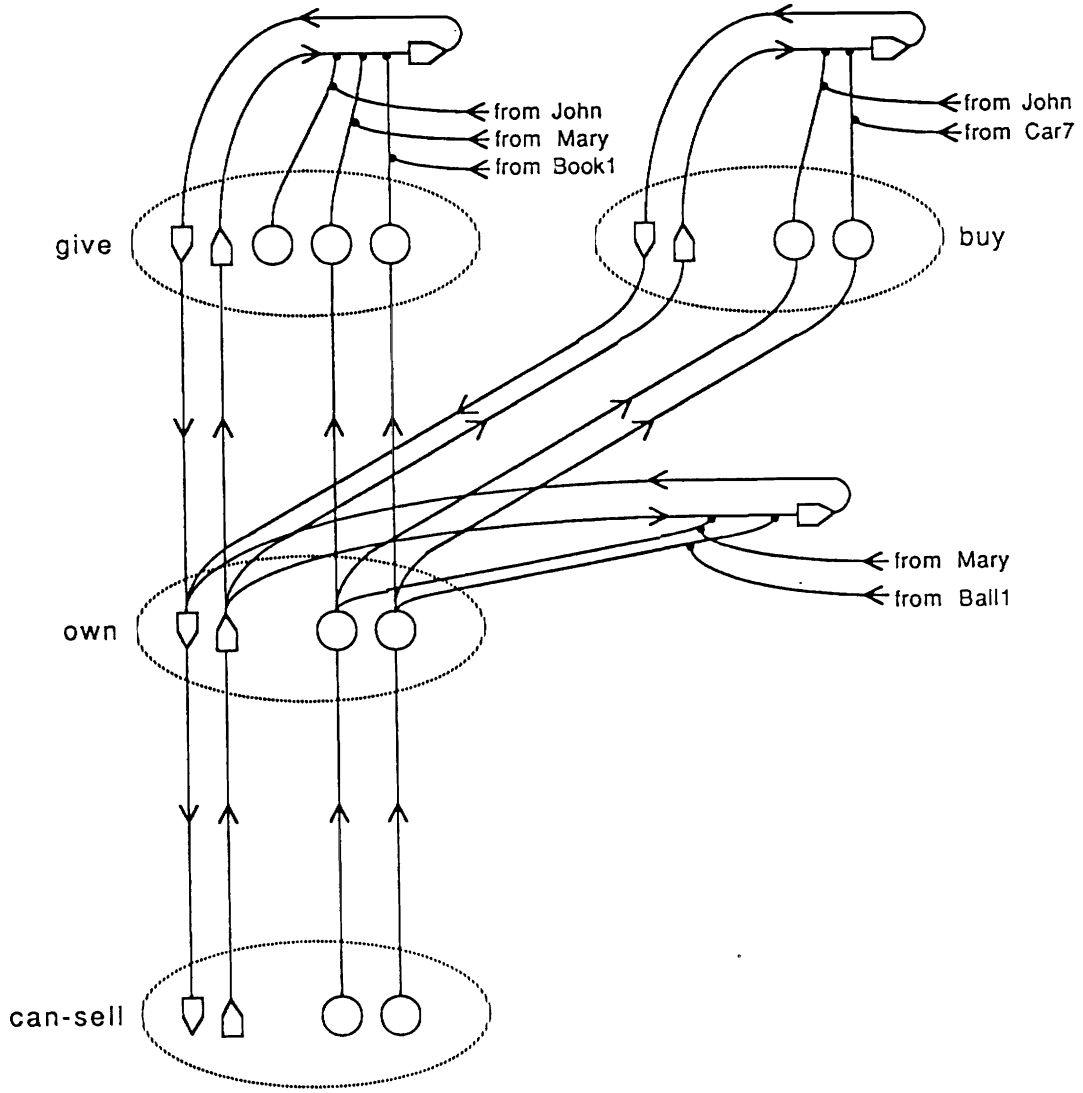

Fig. 14: A network encoding the rules: $\forall x, y, z\ [give(x, y, z) \Rightarrow own(y, z)]$, $\forall x, y\ [buy(x, y) \Rightarrow own(x, y)]$, and $\forall x, y\ [own(x, y) \Rightarrow can\text{-}sell(x, y)]$; and the long-term facts: $give(John, Mary, Book1)$, $buy(John, Car7)$, and $own(Mary, Book2)$. The links between arguments are in the reverse direction because the rules are wired for 'backward reasoning'.

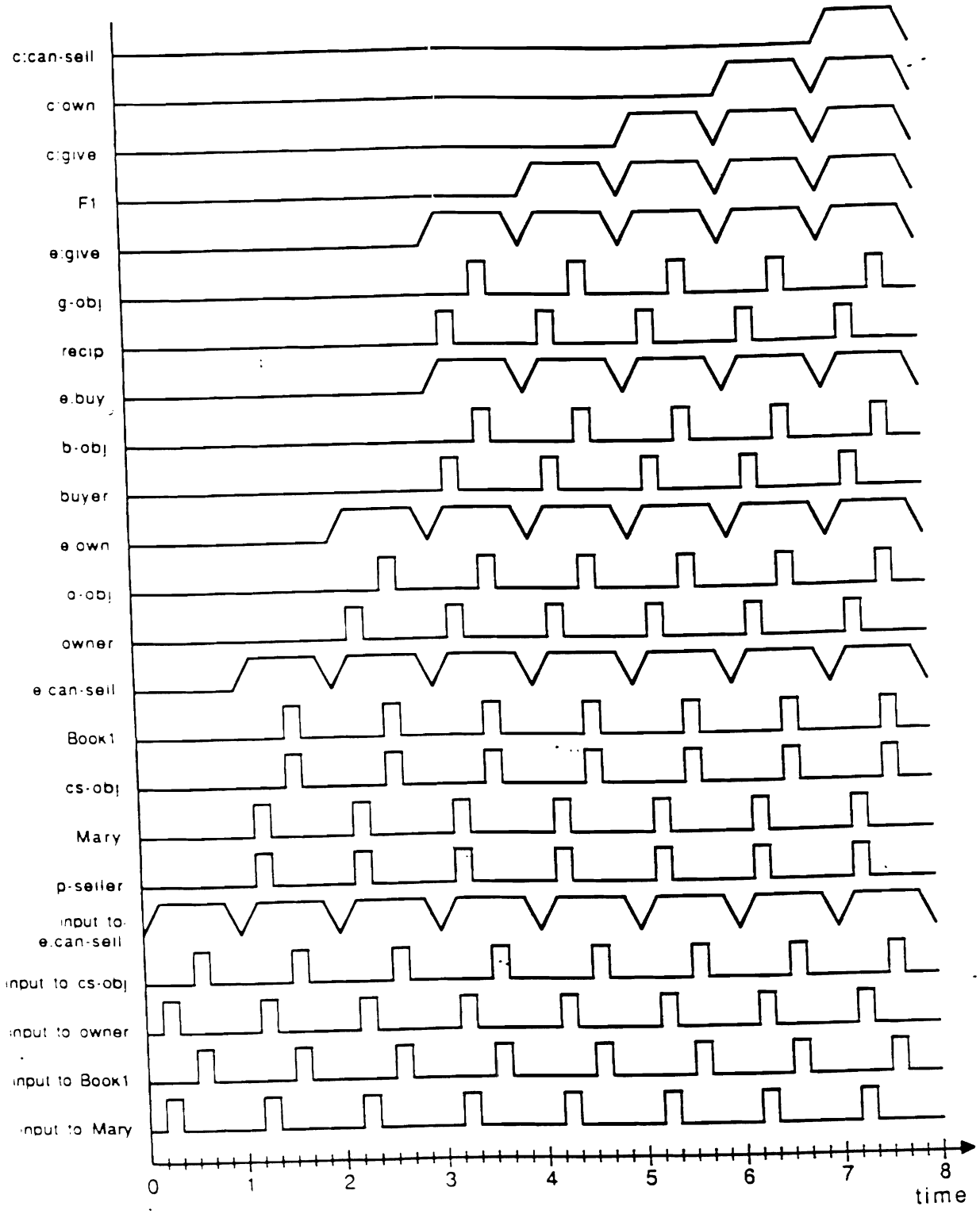

Fig. 15: Activation trace for the query *can-sell*(*Mary*, *Book*1) (Can Mary sell Book1?). The query is posed by providing appropriate inputs to *e:can-sell*, *Mary*, *Book*1, *p-seller* and *cs-obj* as shown. The activation of *c* : ***can-sell***, the *collector* of the query predicate *can-sell*, indicates a *yes* answer.

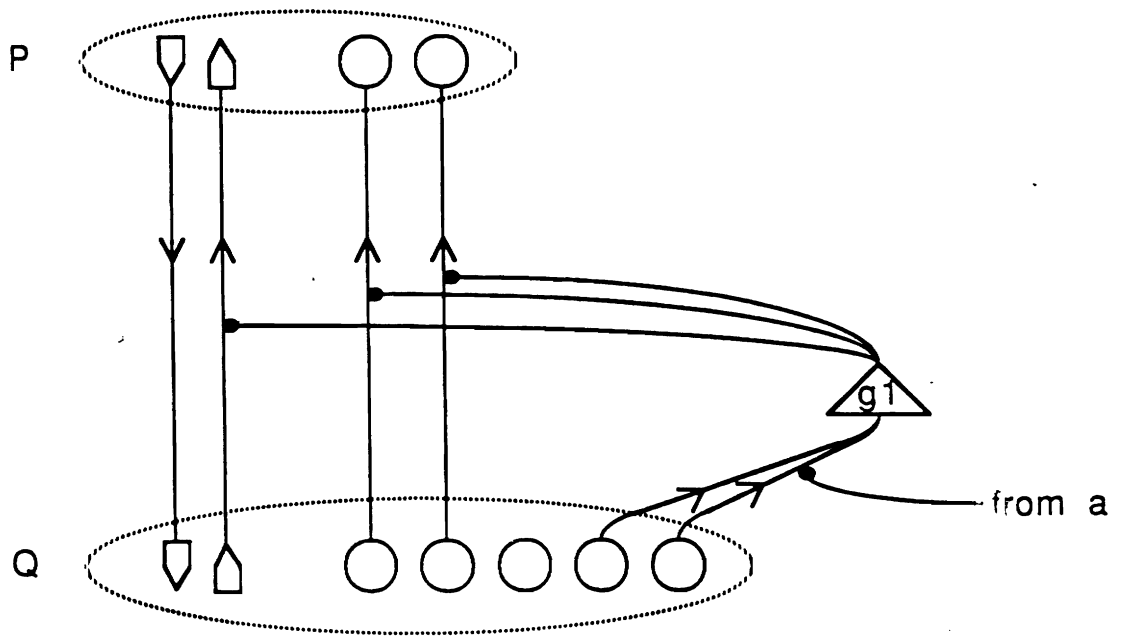

Fig. 16:: Encoding rules with existentially quantified variables and constants in the consequent. The above network encodes the rule $\forall x1, x2\ [P(x1, x2) \Rightarrow \forall y \exists z\ Q(x1, x2, y, z, a)]$. This rule must not fire if during the processing of a query, the existentially bound argument gets bound or if the argument bound to a constant gets bound to a different constant. The node labeled $g1$ is a τ-or node. It projects inhibitory modifiers that block the firing of the rule if during the processing of a query, the existentially quantified argument of Q gets bound or if the argument of Q bound to a gets bound to a constant other than a.

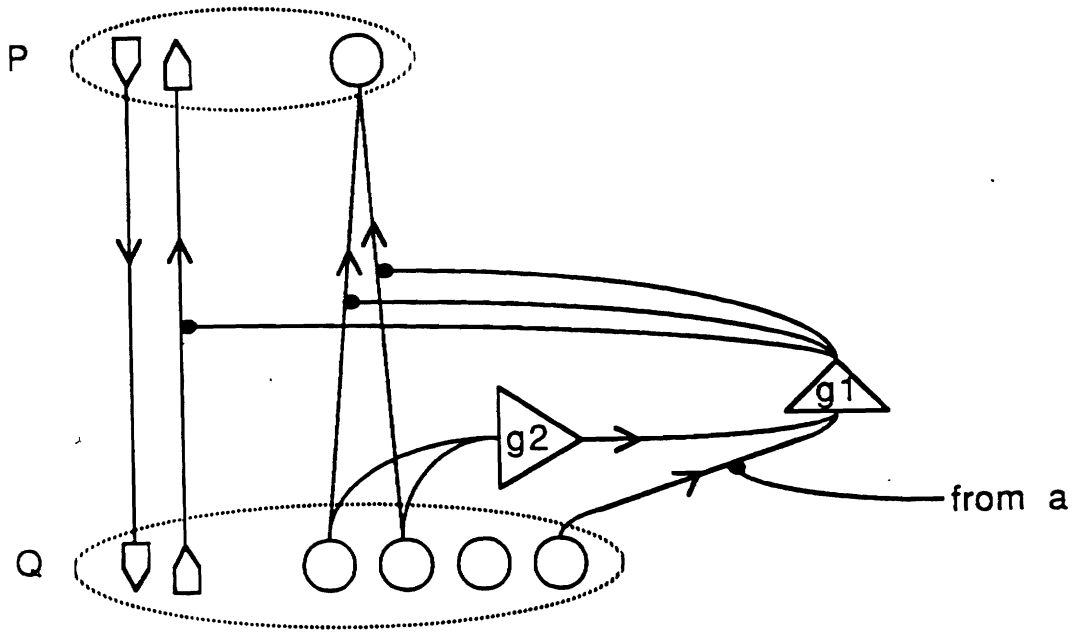

Fig. 17: Encoding rules where the same variable occurs in multiple argument positions in the consequent. The network encodes the rule $\forall x\ P(x) \Rightarrow \forall y Q(x, x, y, a)$. The rule must fire only if a multiply occuring variable is unbound, or if all occurrences of the variable are bound to the same constant. The node $g2$ is like a τ-or node except that it becomes active if it receives inputs in more than one phase within a period of oscillation. On becoming active it activates the node $g1$ which is a τ-or node. The firing of $g1$ blocks the firing of the rule whenever the first and second arguments of Q get bound to different constants. (The above encoding also enforces the constraint that last argument of Q should not be bound to any constant other than a.

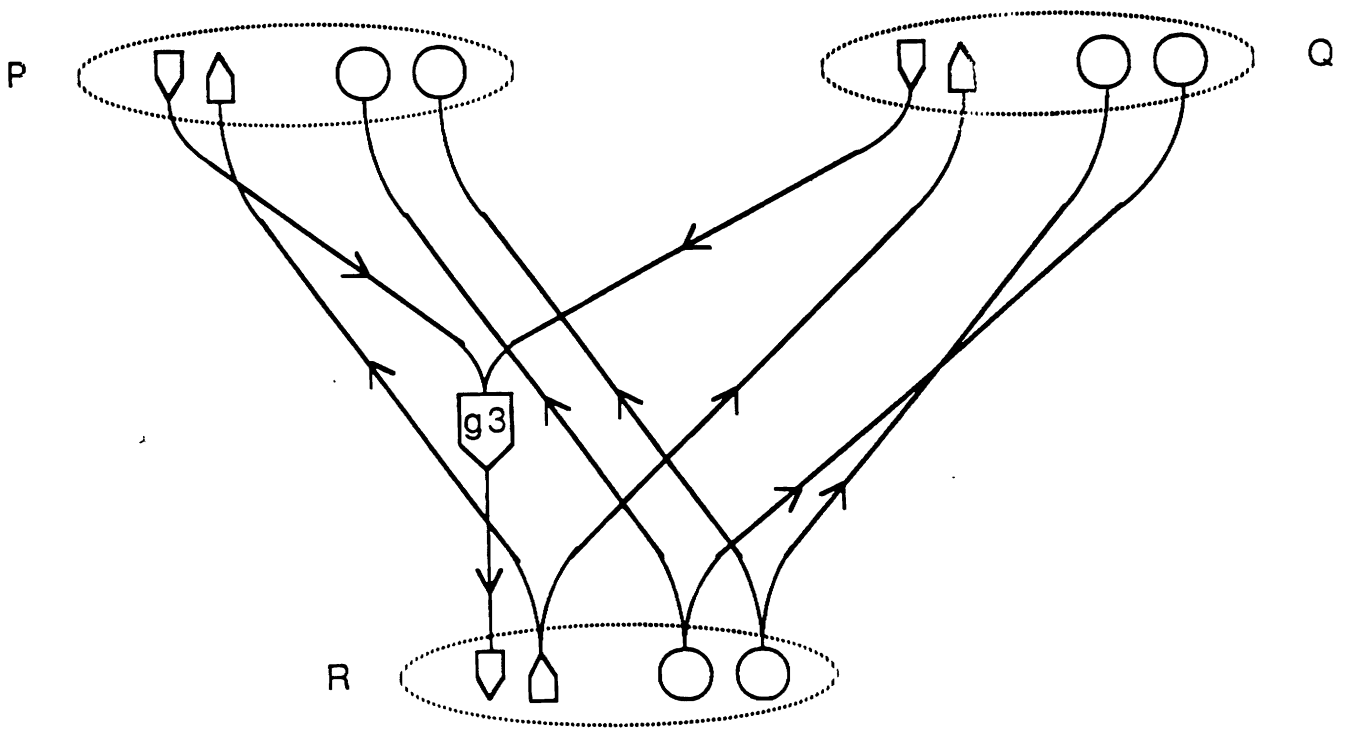

Fig. 18: Multiple antecedent rules are encoded using an additional τ-and node whose threshold equals the number of predicates in the antecedent. This additional node becomes active if and only if it receives inputs from the *collector* nodes of all the antecedent predicates. The interconnections between the argument nodes of the antecedent and consequent predicates remain unchanged. The figure illustrates the encoding of the rule $\forall x, y P(x,y) \wedge Q(y,x) \Rightarrow R(x,y)$. The τ-and node labeled $g3$ has a threshold of 2.

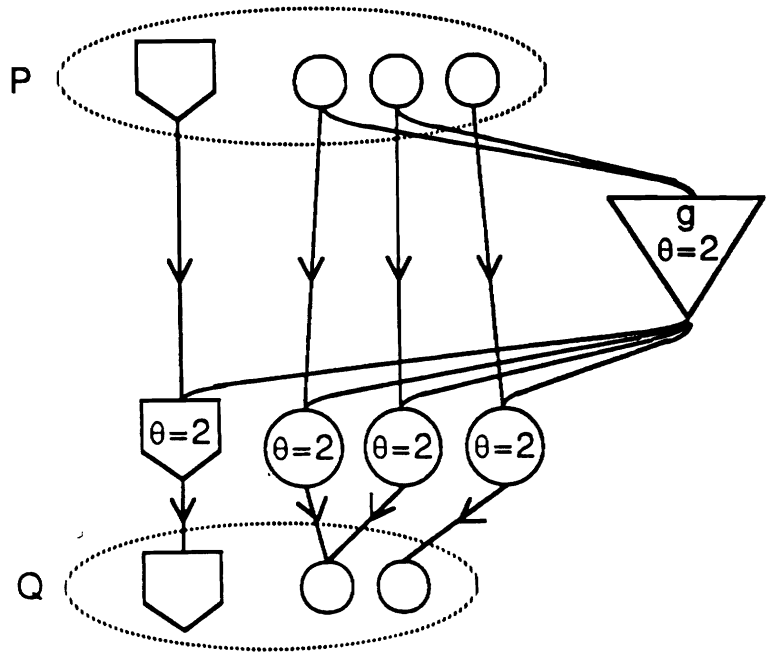

Fig. 19: Encoding a rule with repeated variables in the antecedent within a forward reasoning system. The figure shows the encoding of the rule $\forall x, y\ P(x, x, y) \Rightarrow R(x, y)$. This rule should fire only if all the arguments corresponding to the repeated variable (x) get bound to the same constant. The node g is a τ-or node with a threshold of 2 and receives inputs from the two argument nodes that should be bound to the same filler. It becomes active if it receives two inputs in the same phase and enables the firing of the rule via intermediary ρ-btu and τ-and nodes that also have a threshold of 2.

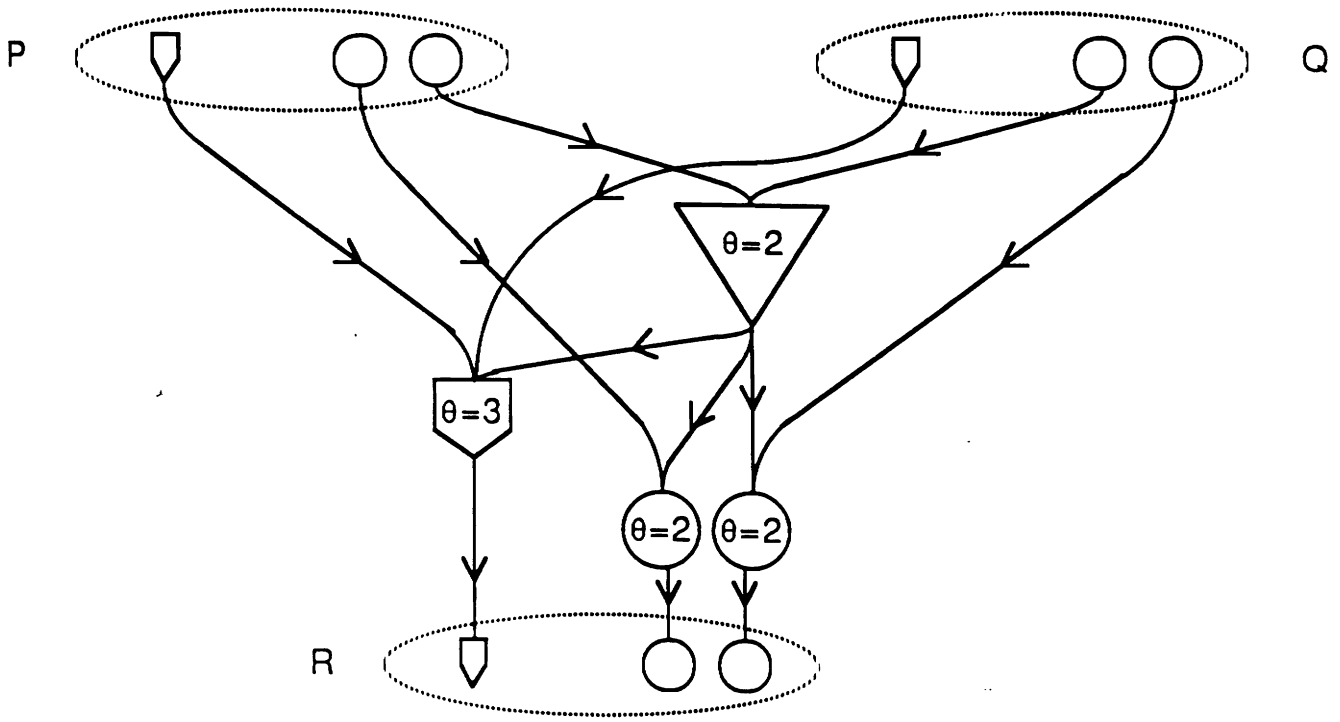

Fig. 20: The encoding of the rule $\forall x, y, z\ P(x, y) \wedge Q(y, z) \Rightarrow R(x, z)$ if a forward reasoning system.

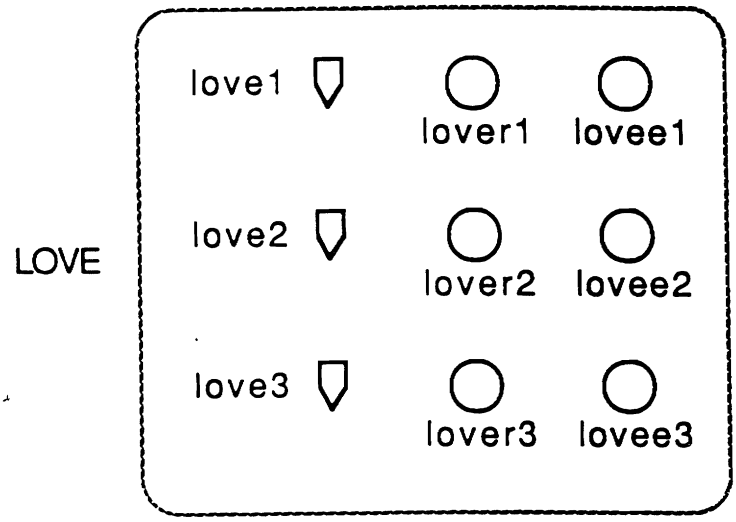

Fig. 21: The subnetwork encoding the binary predicate *love*. A single bank of predicate and argument nodes has been replaced by three banks of nodes. In general, k banks are required if a predicate needs to be dynamically instantiated k times.

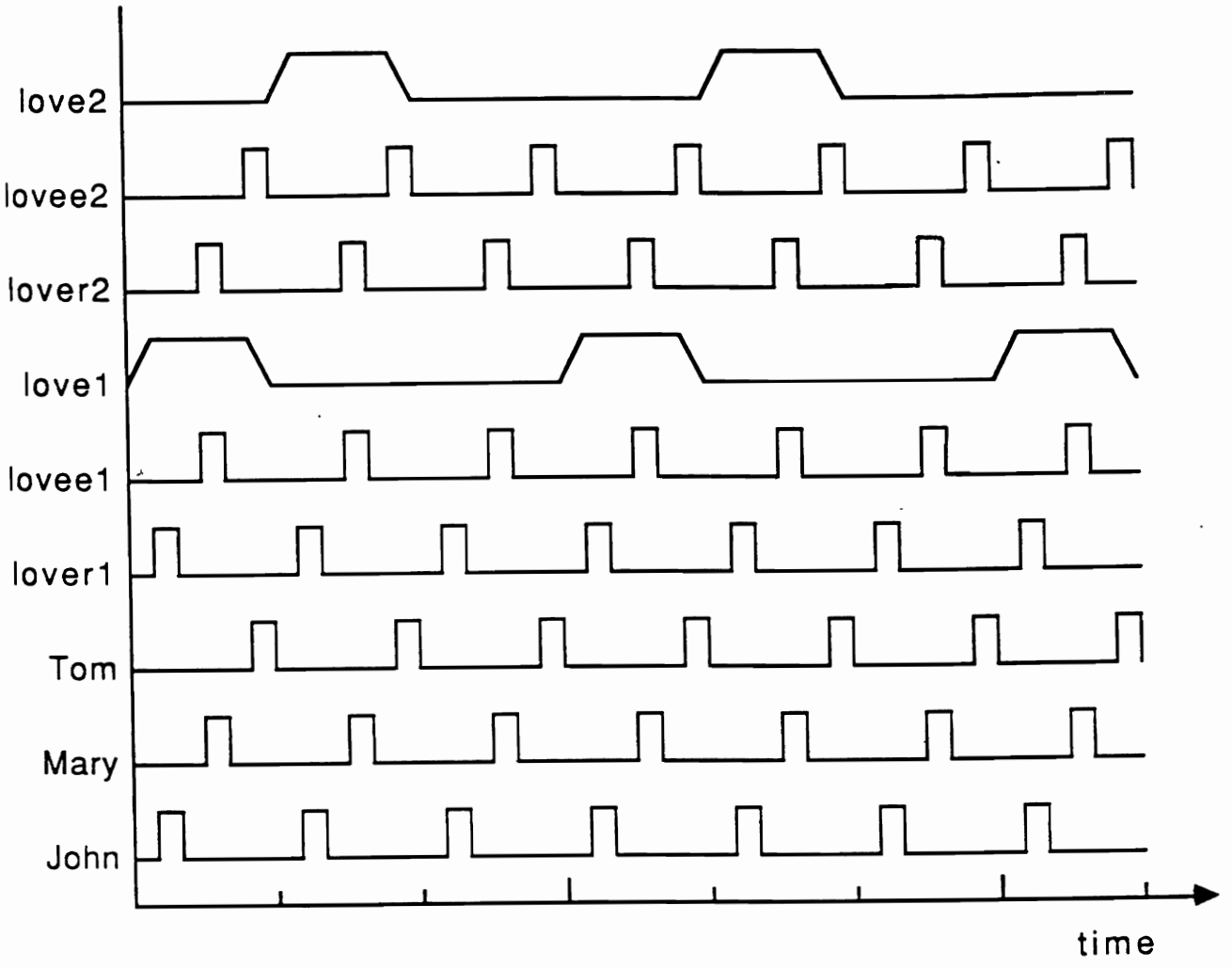

Fig. 22: The rythmic firing pattern of nodes represents the two dynamic facts $love(John, Mary)$ and $love(Mary, Tom)$, simultaneously. The constants $John$, $Mary$, and Tom occupy distinct phases but fire with the same frequency. The first bank of nodes represents $love(John, Mary)$, therefore the argument nodes $lover_1$ and $lovee_1$ are firing in-phase with $John$ and $Mary$, respectively. The second bank of nodes represents the fact $love(Mary, Tom)$, therefore the nodes $lover_2$ and $lovee_2$ are firing in-phase with $Mary$ and Tom, respectively. Predicate nodes $love_1$ and $love_2$ occupy distinct phases in a lower frequency oscillation whose period is three times π, the period of oscillation of argument nodes. The pulse width of predicate nodes *equals* π.

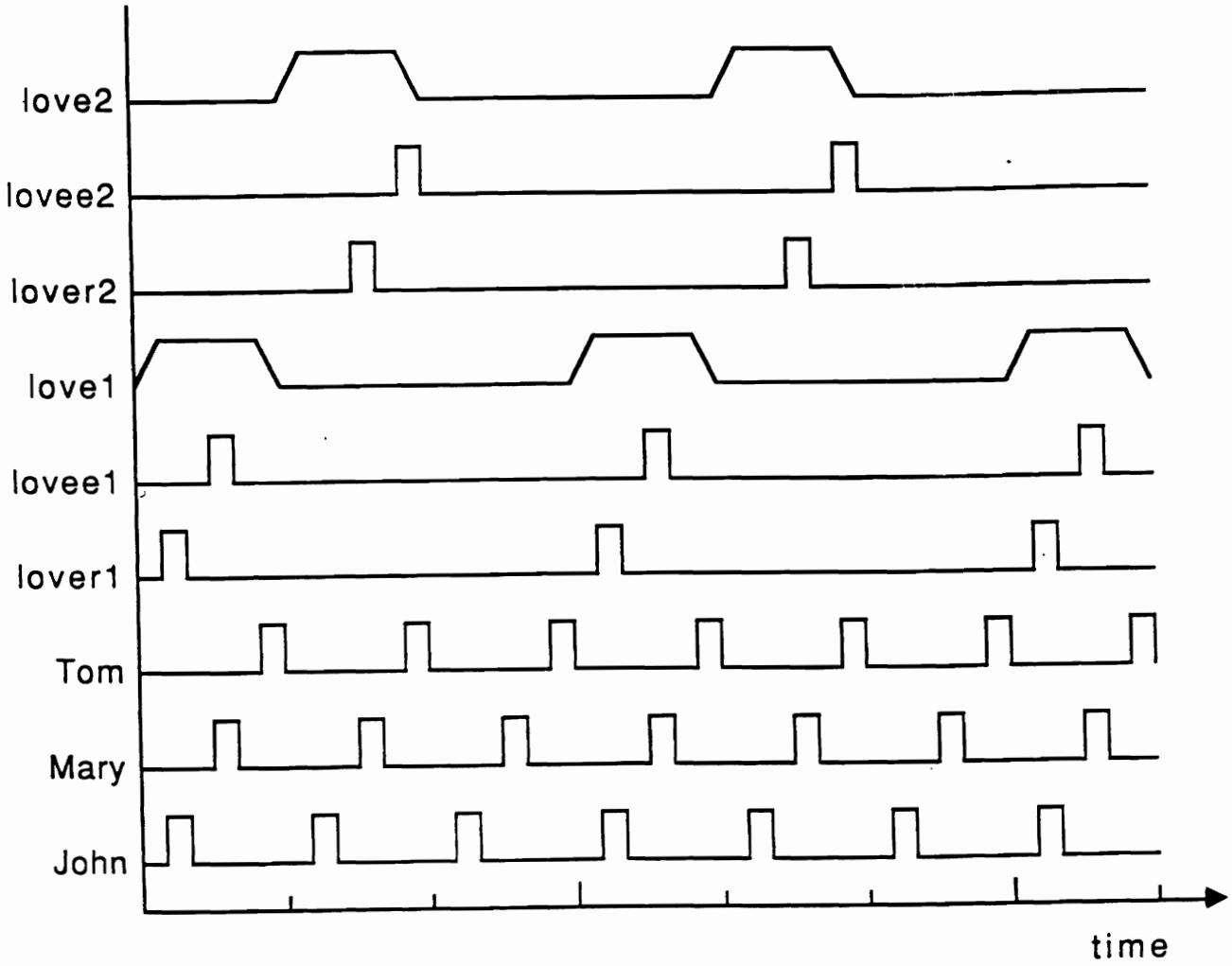

Fig. 23: The output of the *love* subnetwork when it encodes the facts $love(John, Mary)$ and $love(Mary, Tom)$. This pattern is obtained by gating the output of the argument nodes with the output of the predicate node in the associated bank. Dynamic bindings associated with the two facts no longer overlap temporally. The bindings between an argument and its filler are encoded using temporal synchrony at the time-scale of ω (1 msec.). The argument-filler bindings pertaining to the same fact are grouped together using temporal 'synchrony' at the time scale of π (10 msec.).

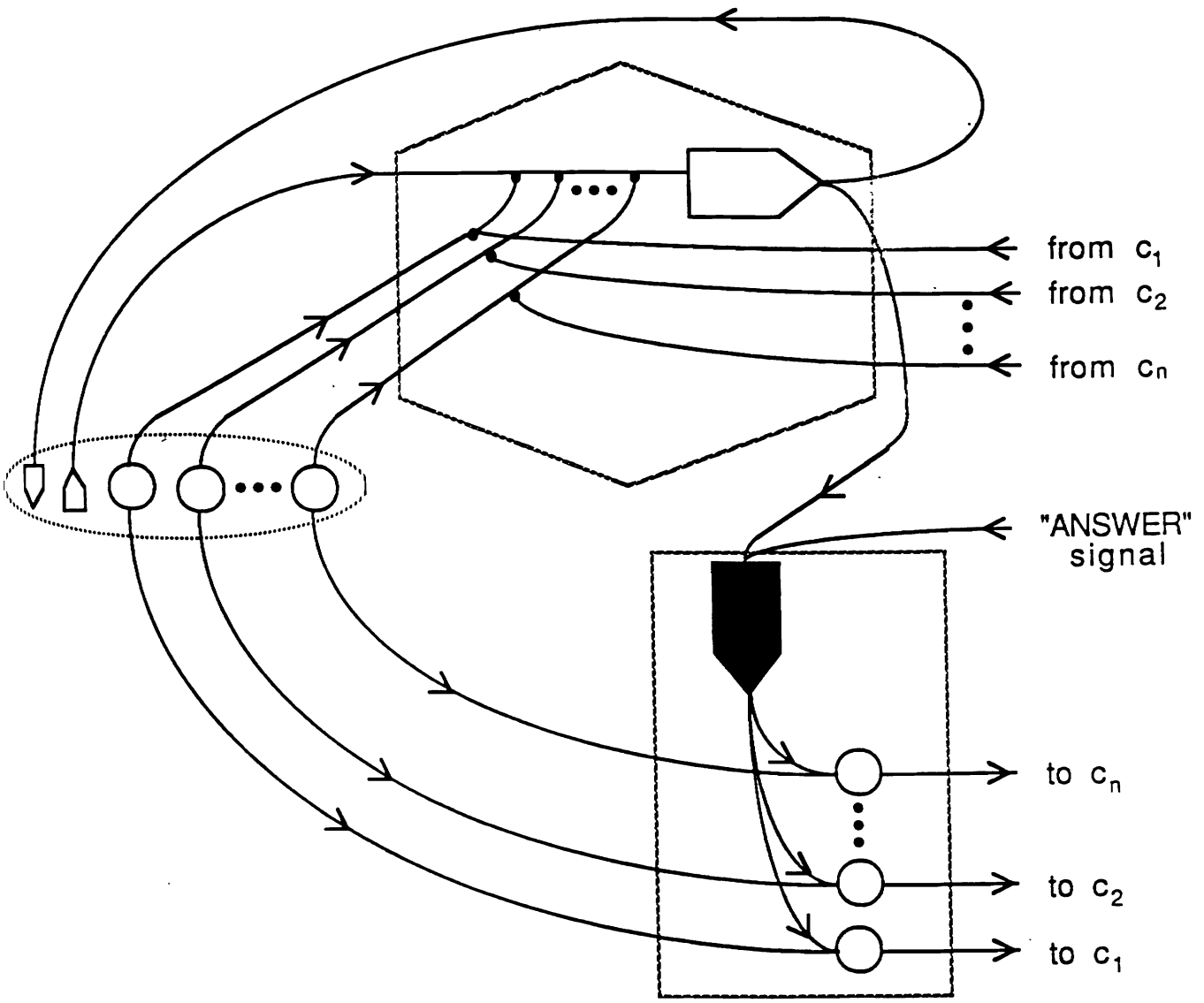

Fig. 24: Augmented representation of a long-term fact in order to support answer extraction. For each of the n arguments of the associated predicate there exists a two input ρ-btu node with a threshold of two. The node shown as a filled-in pointed pentagon behaves like a τ-and node except that once activated, it stays active for some time - say 15π, or about 150 msecs. – even after the inputs are withdrawn.

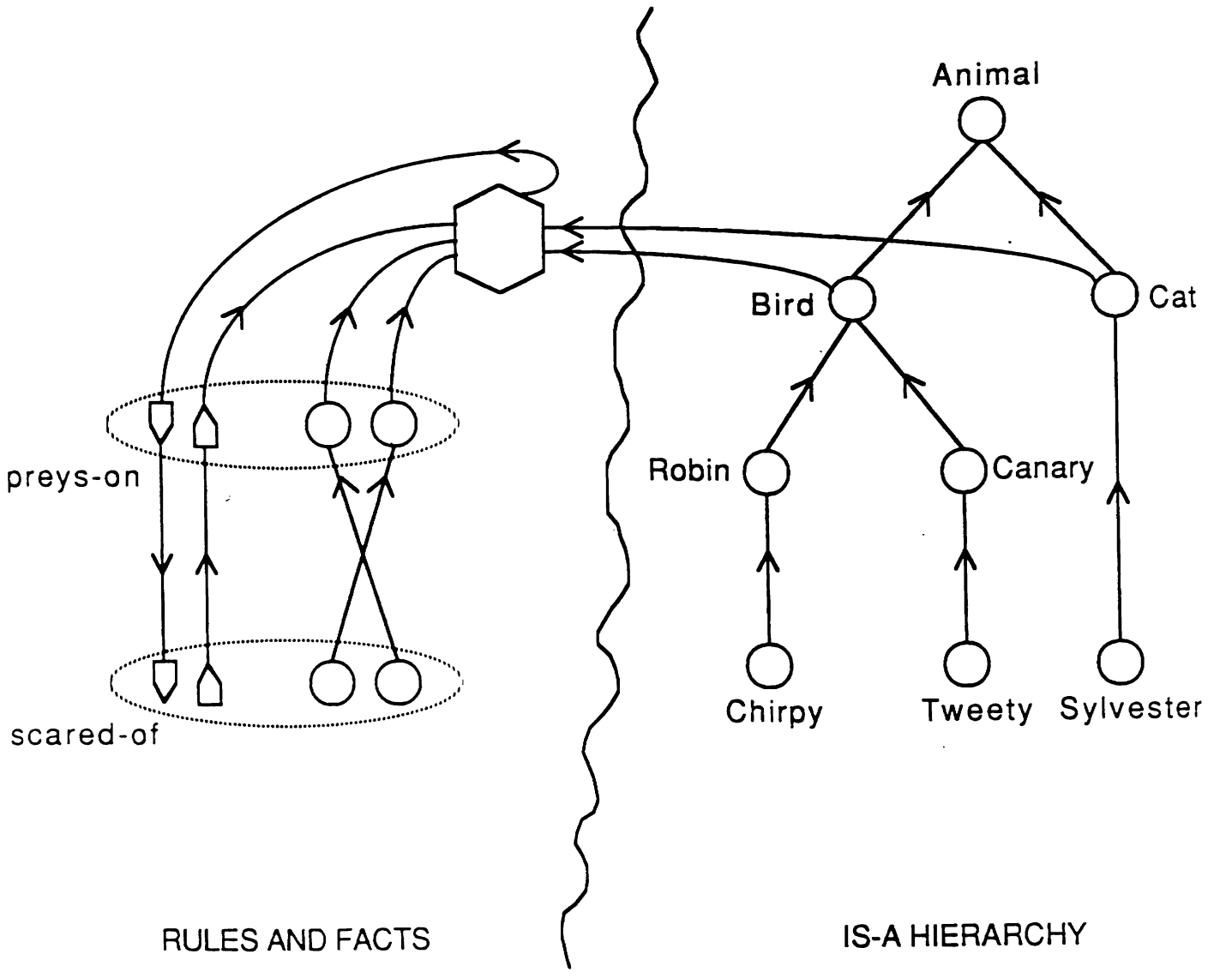

Fig. 25: Interaction between a rule-based reasoner and an *IS-A* hierarchy (conceptual hierarchy). The rule component encodes the rule $\forall x, y$ *preys-on*$(x, y) \Rightarrow$ *scared-of*(y, x) and the fact *preys-on*$(Cat, Bird)$.

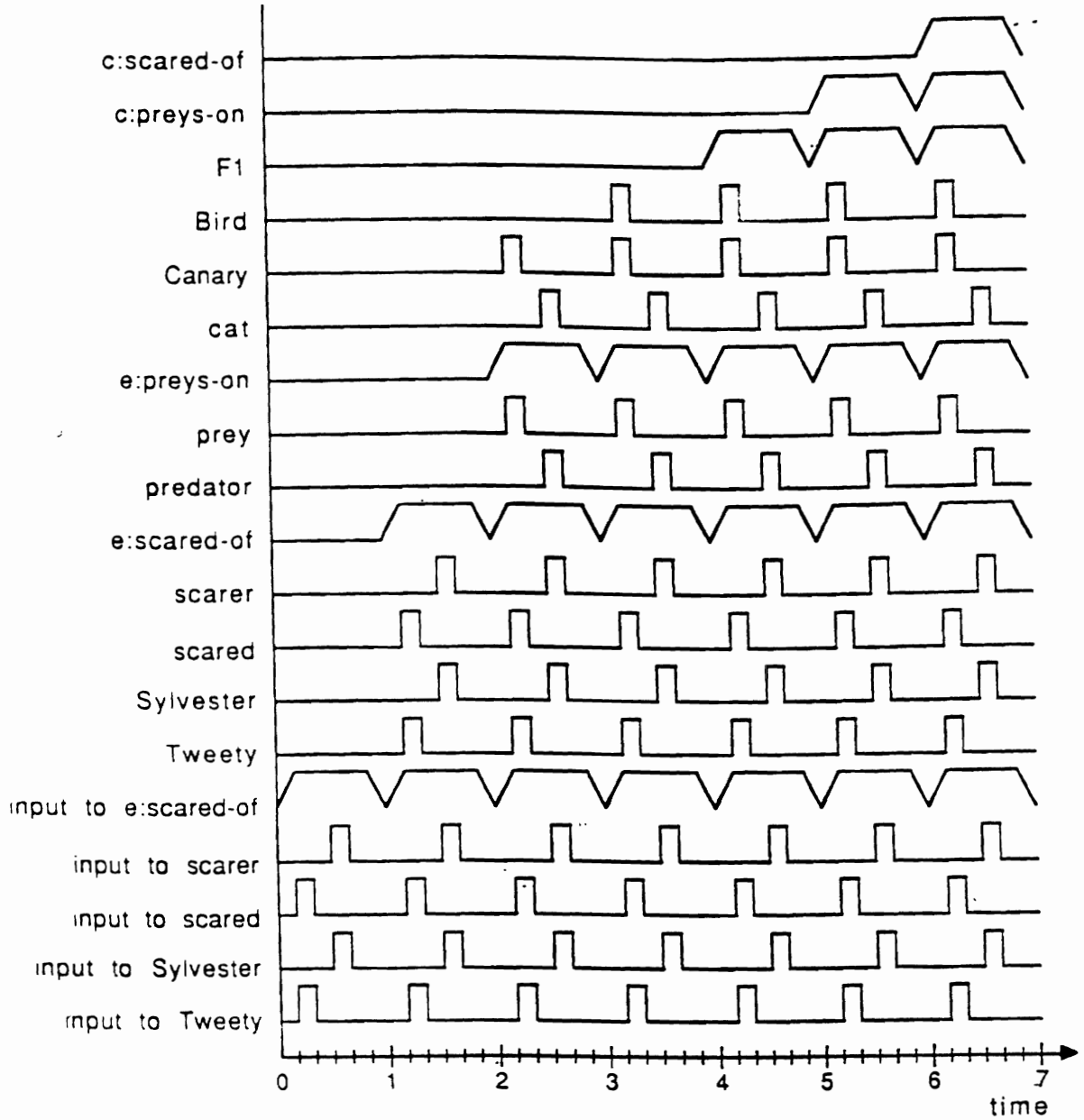

Fig. 26: Activation trace for the query *scared*-of($Tweety, Sylvester$).

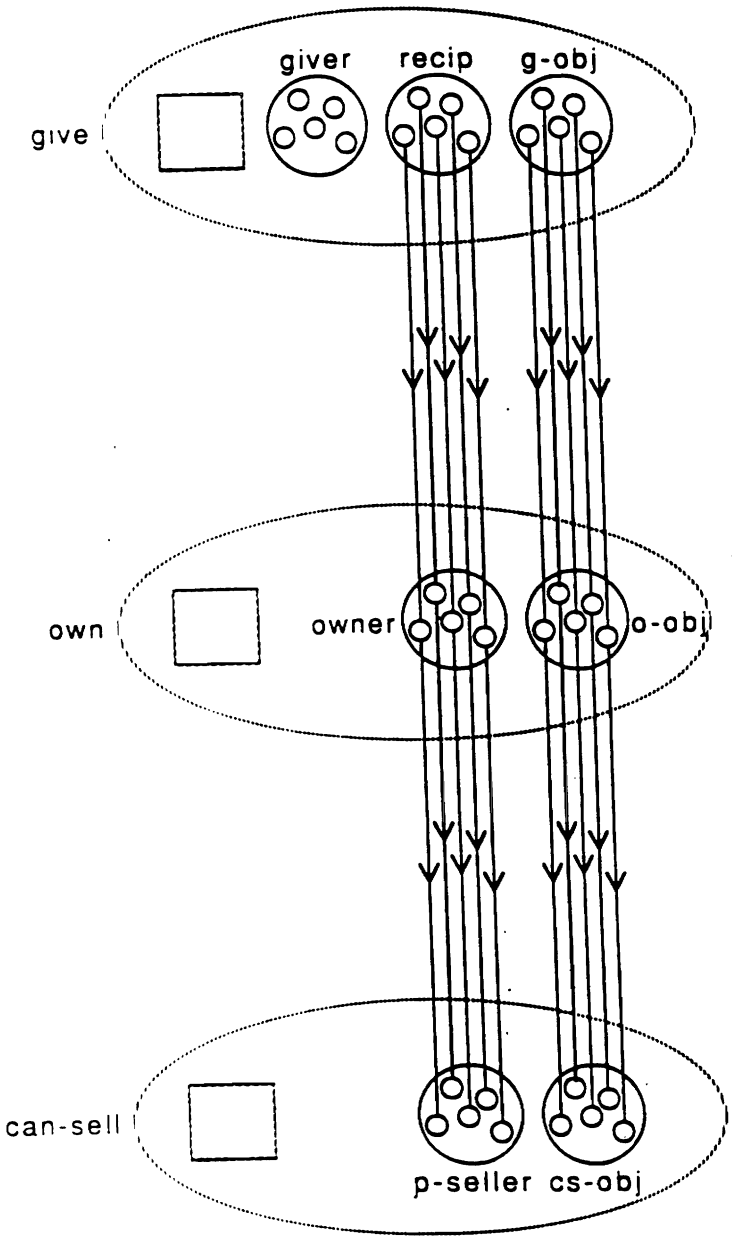

Fig. 27: Propagating bindings by propagating patterns over argument ensembles.