FROM SIMPLE ASSOCIATIONS TO SYSTEMATIC REASONING: A Connectionist representation of rules, variables, and dynamic bindings using temporal synchrony

Lokendra Shastri
Computer and Information Science Department
University of Pennsylvania
Philadelphia, PA 19104
shastri@central.cis.upenn.edu

Venkat Ajjanagadde
Wilhelm-Schickard-Institut
University of Tuebingen
Sand 13 W-7400 Tuebingen, Germany
nnsaj01@mailserv.zdv.uni-tuebingen.de

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. Furthermore, these inferences are drawn with reference to a large body of background knowledge. This remarkable human ability seems paradoxical given the results about the complexity of reasoning reported by researchers in artificial intelligence. It also poses a challenge for cognitive science and computational neuroscience: How can a system of simple and slow neuron-like elements represent a large body of systematic knowledge and perform a range of inferences with such speed? We describe a computational model that is a step toward addressing the cognitive science challenge and resolving the artificial intelligence paradox. We show how a connectionist network can encode millions of facts and rules involving n-ary predicates and variables, and perform a class of inferences in a few hundred msec. Efficient reasoning requires the rapid representation and propagation of dynamic bindings. Our model achieves this by i) representing dynamic bindings as the synchronous firing of appropriate nodes, ii) rules as interconnection patterns that direct the propagation of rhythmic activity, iii) and long-term facts as temporal pattern-matching sub-networks. The model is consistent with recent neurophysiological findings which suggest that synchronous activity occurs in the brain and may play a representational role in neural information processing. The model also makes specific predictions about the nature of reflexive reasoning that are psychologically significant. It identifies constraints on the form of rules that may participate in such reasoning and relates the capacity of the working memory underlying reflexive reasoning to biological parameters such as the lowest frequency at which nodes can sustain synchronous oscillations and the coarseness of synchronization.

Keywords: knowledge representation; reasoning; connectionism; binding problem; temporal synchrony, neural oscillations, short-term memory; long-term memory; working memory; systematicity.

1 Introduction

The ability to represent and reason with a large body of knowledge in an *effective* and *systematic* manner is a central characteristic of cognition. This is borne out by research in artificial intelligence and cognitive science which suggests that reasoning underlies even the most commonplace intelligent behavior. For example, language understanding, a task that we usually perform rapidly and effortlessly, depends upon our ability to make predictions, generate explanations, and recognize speaker's plans.[1] To appreciate the richness and speed of human reasoning, consider the following example derived from (Schubert 1989). Imagine 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 conscious effort. However, a careful analysis suggests that even though the reader remains unaware of it, understanding this sentence requires a chain of reasoning that may informally be described as follows (parenthetical text identifies the background knowledge that might mediate the reasoning process):

> The wolf will approach LRRH (to eat something you have to be near it); LRRH will scream (because a child is scared by an approaching wild animal); upon hearing the scream the wood-cutters will know that a child is in danger (because a child's screaming suggests that it is in danger); the wood-cutters will go to the child (people want to protect children in danger and in part, this involves determining the source of the danger); the wood-cutters will try to prevent the wolf from attacking LRRH (people want to protect children); in doing so the wood-cutters may hurt the wolf (preventing an animal from attacking may involve physical force ...); so the wolf decides to wait (because an animal does not want to get hurt).

One could argue that some of the steps in the above reasoning process are pre-compiled or 'chunked', but it would be unreasonable to claim that this entire chain of reasoning can be construed as direct retrieval or even a single step inference! Hence, in addition to accessing lexical items, parsing, and resolving anaphoric reference, some computation similar to the above chain of reasoning must occur when the sentence in question is processed. As another example consider the sentence 'John seems to have suicidal tendencies, he has joined the Columbian drug enforcement agency.' In spite of it being novel, we can understand the sentence spontaneously and without conscious effort. This sentence, however, could not have been understood without using background knowledge and dynamically inferring that joining the Columbian drug enforcement agency has dangerous consequences and since John probably knows this, his decision to join the agency suggests that he has suicidal tendencies.

As the above examples suggest, we can draw a variety of inferences rapidly, spontaneously and without conscious effort — as though they were a *reflex* response of our cognitive apparatus. In view of this let us describe such reasoning as *reflexive* (Shastri 1990).[2] Reflexive reasoning may be contrasted with *reflective* reasoning which requires reflection, conscious deliberation, and often an *overt* consideration of alternatives and weighing of possibilities. Reflective reasoning

takes longer and often requires the use of external props such as a paper and pencil. Some examples of such reasoning are solving logic puzzles, doing cryptarithmetic, or planning a vacation.[3]

Our remarkable ability to perform reflexive reasoning poses a challenge for cognitive science and neuroscience: How can a system of simple and slow neuron-like elements represent a large body of systematic knowledge and perform a range of inferences with such speed? With nearly 10^{12} computing elements and 10^{15} interconnections, the brain's capacity for encoding, communicating, and processing information seems awesome. But if the brain is extremely powerful it is also extremely limited: First, neurons are slow computing devices. Second, they communicate relatively simple messages that can encode only a few bits of information. Hence a neuron's output cannot encode names, pointers, or complex structures.[4] Finally, the computation performed by a neuron is best described as an *analog* spatio-temporal *integration* of its inputs. The relative simplicity of a neuron's processing ability with reference to the needs of symbolic computation, and the restriction on the complexity of messages exchanged by neurons, impose strong constraints on the nature of neural representations and processes (Feldman & Ballard 1982; Feldman 1989; Shastri 1991). As we discuss in Section 2, a reasoning system must be capable of encoding systematic and abstract knowledge and *instantiating* it in specific situations to draw appropriate inferences. This means that the system must solve a complex version of the *variable binding problem* (Feldman 1982; Malsburg 1986). In particular, the system must be capable of representing *composite* structures in a *dynamic* fashion and systematically propagating them to instantiate other composite structures. This turns out to be a difficult problem for neurally motivated models. As McCarthy (1988) observed most connectionist systems suffer from the "unary or even propositional fixation" with their representational power restricted to unary predicates applied to a fixed object. Fodor and Pylyshyn (1988) have even questioned the ability of connectionist networks to embody systematicity and compositionality.

1.1 Reflexive reasoning: some assumptions and hypotheses

Reflexive reasoning occurs with reference to a large body of *long-term* knowledge. This knowledge forms an integral part of an agent's conceptual representation and is retained for a considerable period of time once it is acquired. We wish to distinguish long-term knowledge from *short-term* knowledge as well as *medium-term* knowledge. By the latter we mean knowledge that persists longer than short-term knowledge and may be remembered for days or even weeks. Such medium-term knowledge however, may be forgotten without being integrated into the agent's long-term conceptual representation. The distinction between medium and long-term knowledge is not arbitrary and seems to have a neurological basis. It is likely that medium-term memories are encoded via long-term potentiation (LTP) (Lynch 1986) and some of them subsequently converted into long-term memories and encoded via essentially permanent structural changes (see e.g., Marr 1971; Squire 1987; Squire & Zola-Morgan 1991).

An agent's long-term knowledge base (LTKB) encodes several kinds of knowledge. These include specific knowledge about particular entities, relations, events and situations, as well as general systematic knowledge about the regularities and dependencies in the agent's environment. For example, an agent's LTKB may contain specific knowledge such as 'Paris is the capital of

France' and 'Susan bought a Rolls Royce', as well as systematic and *instantiation independent* knowledge such as 'if one buys something then one owns it'. We will refer to specific knowledge as *facts*, and general instantiation independent knowledge as *rules* (note that by a 'rule' we do not mean a 'rule of inference' such as modus-ponens). The LTKB may also include knowledge about the attributes or features of concepts and the super/sub-concept relationships among concepts, and also procedural knowledge such as 'how to mow a lawn'.

In discussing the LTKB we are focusing on representational adequacy — i.e., the need for representing entities, relations, inferential dependencies, and specific as well as general (instantiation independent) knowledge. The expressiveness implied by this generic specification, however, is sufficient to represent knowledge structures such as *frames* (Minsky 1975), *scripts* (Schank & Abelson 1977), and *productions* or *if-then* rules (Newell & Simon 1972).

A serious attempt at compiling common sense knowledge suggests that the LTKB may contain as many as 10^8 items (Guha & Lenat 1990). This should not be very surprising given that it must include, besides other things, our knowledge of naive physics and naive psychology; facts about ourselves, our family and friends; facts about history and geography; our knowledge of artifacts; sports, art, and music trivia; and our models of social and civic interactions.

Space and time constraints on a reflexive reasoner

Given that there are about 10^{12} cells in the brain, the expected size of the LTKB (10^8) rules out any encoding scheme whose node requirement is quadratic (or higher) in the size of the LTKB.[5] In view of this we adopt the working hypothesis that *the node requirement of a model of reflexive reasoning should be no more than linear in (i.e., proportional to) the size of the LTKB.* This is a reasonable hypothesis. Observe that i) a node in an idealized computational model may easily correspond to a hundred or so actual cells and ii) the number of cells available for encoding the LTKB can only be a fraction of the total number of cells.

We believe that although the size of an agent's LTKB increases considerably from, say, age ten to thirty, the time taken by an agent to understand natural language does not. This leads us to suspect that the time taken by an episode of reflexive reasoning does not depend on the overall size of the LTKB, but only on the complexity of the particular episode of reasoning. Hence we adopt the working hypothesis that *the time required to perform reflexive reasoning is independent of the size of the LTKB.*[6]

The independence of i) the time taken by reflexive reasoning and ii) the size of the LTKB, implies that reflexive reasoning is a parallel process and involves the simultaneous exploration of a number of inferential paths. Hence a model of reflexive reasoning must be parallel at the level of rule application and reasoning, i.e., it must support knowledge-level parallelism. This is a critical constraint and one that is not necessarily satisfied by a connectionist model simply because it is 'connectionist' (also see Sumida & Dyer 1989).

We understand written language at the rate of somewhere between 150 and 400 words per minute (Carpenter & Just 1977). In other words, we can understand a typical sentence in a matter of one to two seconds. Given that reflexive reasoning occurs during language understanding, it follows that *episodes of reflexive reasoning may take as little as a few hundred msec.*

Reflexive reasoning is limited reasoning

Complexity theory rules out the existence of a general purpose reasoning system that derives all inferences efficiently. This entails that there must exist constraints on the class of reasoning that may be performed in a reflexive manner. Not surprisingly, cognitive agents can perform only a limited class of inferences with extreme efficiency. Naturally, we *expect that the representational and reasoning ability of the proposed system will also be constrained and limited in a number of ways*. However, we would like the strengths and limitations of the system to be psychologically plausible and mirror some of the strengths and limitations of human reasoning.

1.2 Computational constraints

Connectionist models (Feldman & Ballard 1982; Rumelhart & McClelland 1986) are intended to emulate the information processing characteristics of the brain — albeit at an abstract computational level — and reflect its strengths and weaknesses. Typically, a node in a connectionist network corresponds to an idealized neuron, and a link corresponds to an idealized synaptic connection. Let us enumerate some core computational features of connectionist models: i) Nodes compute very simple functions of their inputs, ii) They can only hold limited state information — while a node may maintain a scalar 'potential', it cannot store and selectively manipulate bit strings. iii) Node outputs do not have sufficient resolution to encode symbolic names or pointers. iv) There is no central controller that instructs individual nodes to perform specific operations at each step of processing.

1.3 A preview

We discuss the variable binding problem as it arises in the context of reasoning and describe a neurally plausible solution to this problem. The solution involves maintaining and propagating dynamic bindings using synchronous firing of appropriate nodes. We show how our solution leads to a connectionist knowledge representation and reasoning system that can encode a large LTKB consisting of *facts* and *rules* involving *n-ary predicates* and *variables*, and perform a broad class of reasoning with extreme efficiency. Once a query is posed to the system by initializing the activity of appropriate nodes, the system computes an answer automatically and in time proportional to the *length* of the shortest chain of reasoning leading to the conclusion. The ability to reason rapidly is a consequence, in part, of the system's ability to maintain and propagate a large number of dynamic bindings simultaneously.

The view of information processing implied by the proposed system is one where i) reasoning is the transient but systematic propagation of a *rhythmic* pattern of activity, ii) each entity in the dynamic memory is a phase in the above rhythmic activity, iii) dynamic bindings are represented as the *synchronous* firing of appropriate nodes, iv) long-term facts are subnetworks that act as temporal pattern matchers, and v) rules are interconnection patterns that cause the propagation and transformation of rhythmic patterns of activity.

We cite neurophysiological data which suggests that the basic mechanisms proposed for representing and propagating dynamic variable bindings, namely, the propagation of rhythmic

patterns of activity and the synchronous activation of nodes, exist in the brain and appear to play a role in the representation and processing of information.

Our system predicts a number of constraints on reflexive reasoning that have psychological implications. These predictions concern the capacity of the working memory underlying reflexive reasoning (WMRR) and the form of rules that can participate in such reasoning. The predictions also relate the capacity of the WMRR and the time it would take to perform one step of reasoning to biological parameters such as the lowest frequency at which nodes can sustain synchronous oscillations, the coarseness of synchronization, and the time it takes connected nodes to synchronize. By choosing biologically plausible system parameters we show that it is possible for a system of neuron-like elements to encode millions of facts and rules and yet perform multi-step inferences in a few hundred msec.

Reasoning is the spontaneous and natural outcome of the system's behavior. The system does not apply syntactic rules of inference such as *modus-ponens*. There is no separate interpreter or inference mechanism that manipulates and rewrites symbols. The network encoding of the LTKB is best viewed as a vivid internal *model* of the agent's environment, where the interconnections between (internal) representations directly encode the dependencies between the associated (external) entities. When the nodes in this model are activated to reflect a given state of affairs in the environment, the model spontaneously simulates the behavior of the external world and in doing so makes predictions and draws inferences.

The representational and inferential machinery developed in this work has wider significance and can be applied to other problems whose formulation requires the expressive power of *n*-ary predicates, and whose solution requires the rapid and systematic interaction between long-term and dynamic structures. Some examples of such problems are i) parsing and the dynamic linking of syntactic and semantic structures during language processing and ii) model-based visual object recognition requiring the dynamic representation and analysis of spatial relations between objects and/or parts of objects. Recently, Henderson (1991) has proposed the design of a natural language parser based on our computational model.

1.4 Some caveats

Our primary concern has been to extend the representational and inferential power of neurally plausible (connectionist) models and to demonstrate their scalability. We are also concerned that the strengths and limitations of our system be psychologically plausible. However, our aim has not been to model data from specific psychological experiments. What we describe is a partial model of reflexive reasoning. It demonstrates how a range of reasoning can be performed in a reflexive manner and also identifies certain types of reasoning that cannot be performed in a reflexive manner. Our system, however, does not model all aspects of reflexive reasoning. For example, we focus primarily on declarative and semantic knowledge and do not model reflexive analogical reasoning, or reflexive reasoning involving episodic memory (Tulving 1983) and imagery. We do not say much about what the actual contents of an agent's LTKB ought to be. We do not provide a detailed answer to the question of learning. We do however, discuss in brief how specific facts may be learned and existing rules modified (Section 10.6). Neural plausibility is an important aspect of this work — we show that the proposed system can be

realized using neurally plausible nodes and mechanisms, and we investigate the consequences of choosing biologically motivated values of system parameters. Needless to say, what we describe is an idealized computational model and it is not intended to be a blue-print of how the brain encodes a LTKB and performs reflexive reasoning.

An outline of the paper

Section 2 discusses the dynamic binding problem in the context of reasoning. Section 3 presents our solution to this problem and the encoding of long-term rules and facts. Section 4 describes a reasoning system capable of encoding a LTKB and answering queries based on the encoded knowledge. The interface of the basic reasoning system with an *IS-A* hierarchy that represents entities, types (categories), and the super/sub-concept relations between them is described in Section 5. Section 6 discusses a solution to the multiple instantiation problem. Section 7 discusses the biological plausibility of our system and identifies neurally plausible values of certain system parameters. Section 8 points out the psychological implications of the constraints on reflexive reasoning suggested by the system. Section 9 discusses related connectionist models and the marker passing system NETL. Finally, Section 10 discusses some open problems related to integrating the proposed reflexive reasoning system with an extended cognitive system. Certain portions of the text are set in small type. These cover detailed technical material and may be skipped without loss of continuity.

2 Reasoning and the dynamic binding problem

Assume that an agent's LTKB embodies the following rules:[7]

1. If someone gives a recipient an object then the recipient comes to own that object.
2. Owners can sell what they own.

Given the above knowledge an agent would be capable of inferring 'Mary owns Book1' and 'Mary can sell Book1' on being told 'John gave Mary Book1'. A connectionist reasoning system that embodies the same knowledge should also be capable of making similar inferences, and hence, exhibit the following behavior: If the network's pattern of activity is initialized to represent the fact 'John gave Mary Book1' then very soon, its activity should evolve to include the representations of the facts 'Mary owns Book1' and 'Mary can sell Book1'.

Before proceeding further let us point out that the knowledge embodied in a rule may be viewed as having two distinct aspects. A rule specifies a systematic *correspondence* between the arguments of certain 'predicates' (where a predicate may be thought of as a relation, a frame, or a schema). For example, rule (1) specifies that a 'give' event leads to an 'own' event where the *recipient* of 'give' corresponds to the *owner* of 'own', and the *object* of 'give' corresponds to the *object* of 'own'. Let us refer to this aspect of a rule as *systematicity*.[8] The second aspect of the knowledge embodied in a rule concerns the *appropriateness* of the specified argument correspondence in a given situation depending upon the types (or features) of the argument-fillers involved in that situation. Thus appropriateness may capture type restrictions that argument fillers

must satisfy in order for a rule to fire. It may also indicate type preferences and provide a graded measure of a rule's applicability in a given situation based on the types of the argument-fillers in that situation.

We will first focus on the problems that must be solved in order to incorporate systematicity in a connectionist system. Later in Section 5 we will discuss how the solutions proposed to deal with systematicity may be augmented to incorporate appropriateness and represent context dependent rules that are sensitive to the types of the argument-fillers.

If we focus on systematicity, then rules can be succinctly described using the notation of first-order logic. For example, rules (1) and (2) can be expressed as the following first-order rules:

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

wherein $give$ is a three place predicate with arguments: $giver$, $recipient$, and $give\text{-}object$; own is a two place predicate with arguments: $owner$ and $own\text{-}object$; and $can\text{-}sell$ is also a two place predicate with arguments: $potential\text{-}seller$ and $can\text{-}sell\text{-}object$. The use of quantifiers and variables allows the expression of general, instantiation independent knowledge and helps in specifying the systematic correspondence between predicate arguments.[9]

A fact may be expressed as a predicate instance (atomic formula). For example, the fact 'John gave Mary Book1' may be expressed as *give(John, Mary, Book1)*.

A connectionist network must solve three technical problems in order to incorporate systematicity. We discuss these problems in the following three sections.

2.1 Dynamic representation of facts: instantiating predicates

A reflexive reasoning system should be capable of representing facts in a rapid and dynamic fashion. Observe that the reasoning process generates inferred facts dynamically and the reasoning system should be capable of representing these inferred facts. Furthermore, the reasoning system must interact with other processes that communicate facts and pose queries to it, and the system should be capable of dynamically representing such facts and queries.

The dynamic representation of facts poses a problem for standard connectionist models. Consider the fact *give(John, Mary, Book1)*. This fact cannot be represented dynamically by simply activating the representations of the arguments *giver*, *recipient*, and *give-object*, and the constituents 'John', 'Mary', and 'Book1'. Such a representation would suffer from cross-talk and would be indistinguishable from the representations of *give(Mary, John, Book1)* and *give(Book1, Mary, John)*. The problem is that this fact — like any other instantiation of an n-ary predicate — is a composite structure: it does not merely express an association between the constituents 'John', 'Mary', and 'Book1', rather it expresses a specific relation wherein each constituent plays a distinct role. Thus a fact is essentially a collection of *bindings* between predicate arguments and fillers. For example, the fact *give(John, Mary, Book1)* is the collection of argument-filler bindings *(giver=John, recipient=Mary, give-object=Book1)*. Hence representing a dynamic fact amounts to representing — in a dynamic fashion — the appropriate bindings between predicate arguments and fillers.

The dynamic representation of facts should also support the simultaneous representation of

multiple facts such as *give(John, Mary, Book1)* and *give(Mary, John, Car3)* without 'creating' ghost facts such as *give(Mary, John, Book1)*.

Static versus dynamic bindings

A connectionist encoding that represents the bindings associated with the fact *give(John, Mary, Book1)* without cross-talk is illustrated in Fig. 1 (cf. Shastri & Feldman 1986; Shastri 1988b). Each triangular *binder* node binds the appropriate filler to the appropriate argument and the *focal* node *give23* provides the requisite grouping between the set of bindings that make up the 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 a *static* encoding using physically interconnected nodes and links to represent argument-filler bindings is suitable for representing stable and long-term knowledge because the required focal and binder nodes may be learned (or recruited) over time in order to represent new but stable bindings of constituents.[10] This scheme however, is implausible for representing bindings that will be required to encode dynamic structures that will arise during language understanding and visual processing. Such dynamic bindings may have to be represented very rapidly — within a hundred msec — and it is unlikely that there exist mechanisms that can support wide spread *structural* changes and growth of new links within such time scales. An alternative would be to assume that interconnections between *all possible* pairs of arguments and fillers already exist. These links normally remain "inactive" but the appropriate subset of these links become "active" temporarily to represent dynamic bindings (Feldman 1982; von der Malsburg 1986). This approach, however, is also problematic because the number of all possible argument-filler bindings is extremely large and having pre-existing structures for representing all these bindings will require a prohibitively large number of nodes and links. Techniques for representing argument-filler bindings based on the von Neumann architecture also pose difficulties because they require communicating names or pointers of fillers to appropriate argument nodes and vice versa. As pointed out earlier, the storage and processing capacity of nodes as well as the resolution of their outputs is not sufficient to store, process, and communicate names or pointers.

********** Figure 1 goes about here **********

2.2 Inference, propagation of dynamic bindings, and the encoding of rules

The second technical problem that a connectionist reasoning system must solve concerns the dynamic generation of inferred facts. For example, starting with a dynamic representation of *give(John, Mary, Book1)* the state of a network encoding rules (1) and (2) should evolve rapidly to include the dynamic representations of the inferred facts: *own(Mary, Book1)* and *can-sell(Mary, Book1)*. This process should also be free of cross-talk and not lead to spurious bindings.

Generating inferred facts involves the systematic propagation of dynamic bindings in accordance with the rules embodied in the system. A rule specifies antecedent and consequent predicates and a correspondence between the arguments of these predicates. For example, the rule $\forall x, y, z\ [give(x, y, z) \Rightarrow own(y, z)]$ specifies that a *give* event results in an *own* event wherein the *recipient* of a *give* event corresponds to the *owner* of an *own* event and the *give-object* of a

give event corresponds to the *own-object* of an *own* event. An application of a rule (i.e., a step of inference), therefore amounts to taking an instance of the antecedent predicate(s) 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. Thus the application of the rule $\forall x, y, z\ [give(x, y, z) \Rightarrow own(y, z)]$ in conjunction with an instance of *give*, *give(John, Mary, Book1)*, creates an instance of *own* with the bindings *(owner=Mary,own-object=Book1)*. These bindings constitute the inferred fact *own(Mary, Book1)*. Once the representation of an inferred fact is established, it may be used in conjunction with other domain rules to create other inferred facts. Such a chain of inference may lead to a proliferation of inferred facts and the associated dynamic bindings.

2.3 Encoding long-term facts

In addition to encoding domain rules such as (1) and (2), a connectionist reasoning system must also be capable of encoding facts in its LTKB and using them during recall, recognition, query answering, and reasoning. For example, we expect our system to be capable of encoding a fact such as 'John bought a Rolls-Royce' in its LTKB and use it to rapidly answer the query 'Did John buy a Rolls-Royce?'. We also expect it to use this fact in conjunction with other knowledge to rapidly answer queries such as 'Does John own a car?' Observe that storing a long-term fact would require storing the associated bindings as a *static* long-term structure. This structure should interact with dynamic bindings and recognize those that *match* it.

2.4 Dynamic binding and categorization

As discussed at the beginning of Section 2, the appropriateness of a rule in a specific situation may depend on the types/features of the argument-fillers involved in that situation. Thus categorization plays a crucial role in the propagation of dynamic bindings during reasoning. Consider the rule: $\forall x, y\ walk\text{-}into(x, y) \Rightarrow hurt(x)$ (i.e., if one walks into something then one gets hurt'). As stated the rule only encodes systematicity and underspecifies the relation between 'walking into' and 'getting hurt'. It would fire even in the situation "John walked into the mist" and lead to the inference "John got hurt". A complete encoding of the knowledge embodied in the rule would also specify the types/features of the argument-fillers of 'walk-into' for which the application of this rule would be appropriate. Given such an encoding the propagation of binding from the first argument of *walk-into* to the argument of *hurt* will occur *only if* the fillers of the arguments of *walk-into* belong to the appropriate type (we discuss the encoding of such rules in Section 5).

The use of categorization can also prevent certain cases of cross-talk in the representation of dynamic facts. For example, categorization may prevent cross-talk in the representation of *buy(Mary, Book1)* because spurious versions of this fact such as *buy(Book1, Mary)* would violate category restrictions, and hence, would be unstable. However, categorization cannot in of itself, *solve* the dynamic binding problem. This because it alone cannot enforce systematicity. For example, categorization cannot determine that the dynamic fact *give(John, Mary, Book1)* should result in the inferred fact *own(Mary, Book1)* but not *own(John, Book1)*.

2.5 The dynamic binding problem in vision and language

The need for systematically dealing with composite objects in a dynamic manner immediately gives rise to the dynamic binding problem. Thus the dynamic binding problem arises in any cognitive activity that admits systematicity and compositionality. Consider vision. Visual object recognition involves the rapid grouping of information over the spatial extent of an object and across different feature maps so that features belonging to one object are not confused with those of another (Treisman and Gelade 1980). The binding of features during visual processing is similar to the binding of argument fillers during reasoning. In terms of representational power, however, the grouping of all features belonging to the same object can be expressed using *unary*-predicates.[11] But as we have seen, reasoning requires the representation of unary as well as n-ary predicates. A similar need would arise in a more sophisticated vision system that dynamically represents and analyzes spatial relations between objects and/or parts of object.

While there may be considerable disagreement over the choice of primitives and the functional relationship between the 'meaning' of a composite structure and that of its constituents, it seems apparent that a computational model of language should be capable of computing and representing composite structures in a systematic and dynamic manner. Thus language understanding requires a solution to the dynamic binding problem, not only to support reasoning, but also to support syntactic processing and the dynamic linking of syntactic and semantic structures.

3 Solving the dynamic binding problem

In this section we describe solutions to the three technical problems discussed in sections 2.1 through 2.3. The solution involves several ideas that complement each other and together lead to a connectionist model of knowledge representation and reflexive reasoning.

As pointed out in Section 2.1, it is implausible to represent dynamic bindings using structural changes, pre-wired interconnection networks, or by communicating names/pointers of arguments and fillers. Instead, what is required is a neurally plausible mechanism for rapidly and temporarily labeling the representations of fillers and predicate arguments to dynamically encode argument-filler bindings. Also required are mechanisms for systematically propagating such transient labels and allowing them to interact with long-term structures.

In the proposed system we use the temporal structure of node activity to provide the necessary labeling. Specifically, we represent dynamic bindings between arguments and fillers by the *synchronous* firing of appropriate nodes. We also propose appropriate representations for n-ary predicates, rules, long-term facts, and an *IS-A* hierarchy that facilitate the efficient propagation and recognition of dynamic bindings.[12]

The significance of temporally organized neural activity has long been recognized (Hebb 1949; Freeman 1981; Sejnowski 1981). In particular, von der Malsburg (1981,1986) has proposed that correlated activity within a group of cells can be used to represent dynamic grouping of cells. He also used temporal synchrony and synapses that can alter their weights within hundreds of msec to model sensory segmentation and the human ability to attend to a specific speaker in a noisy environment (von der Malsburg & Schneider 1986). Abeles (1982, 1991) has put forth the hypothesis that computations in the cortex occur via "synfire chains" — propagation

of synchronous activity along diverging and converging pathways between richly interconnected cell assemblies. Crick (1984) has also suggested the use of fine temporal coincidence to represent dynamic bindings and synchronized activity across distant regions forms the keystone of Damasio's (1991) general framework for memory and consciousness. Several researchers have reported the occurrence of synchronous activity in the cat and monkey visual cortex and presented evidence in support of the conjecture that the visual cortex may be using synchronous and/or oscillatory activity to solve the binding problem (see Section 7).

Recently, other researchers have used temporal synchrony to solve various aspects of the binding problem in visual perception (Strong & Whitehead 1989; Hummel & Biederman 1991; Horn, Sagi & Usher 1991). In this work we use temporal synchrony to solve a different problem, namely, the representation of, and systematic reasoning with, conceptual knowledge. In solving this problem we also demonstrate that temporal synchrony can support more complex representations. The expressiveness and inferential power of our model exceeds that of the models cited above because our system can represent dynamic instantiations of *n*-ary predicates — including *multiple* instantiations of the same predicate.[13]

Clossman (1988) had used synchronous activity to represent argument-filler bindings but he had not suggested an effective representation of 'rules'. Consequently, his system could not *propagate* dynamic bindings to perform inferences.

As an abstract computational mechanism, temporal synchrony can be related to the notion of *marker passing* (Quillian 1968; Fahlman 1979).[14] Fahlman had proposed the design of a *marker passing* machine consisting of a parallel network of simple processors and a serial computer that controlled the operation of the parallel network. Each node could store a small number of discrete 'markers' (or tags) and each link could propagate markers between nodes under the supervision of the network controller. Fahlman showed how his machine could compute transitive closure and set intersection in parallel, and in turn, solve a class of inheritance and recognition problems efficiently. Fahlman's system however, was not neurally plausible. First, nodes in the system were required to *store*, *match*, and *selectively* propagate marker bits. Although units with the appropriate memory and processing characteristics may be readily realized using electronic hardware, they do not have any direct neural analog. Second, the marker passing system operated under the strict control of a serial computer that specified, at *every step* of the propagation, exactly which types of links were to pass which markers in which directions.

The relation between marker passing and temporal synchrony can be recognized by noting that nodes firing in synchrony may be viewed as being marked with the *same* marker and the propagation of synchronous activity along a chain of connected nodes can be viewed as the propagation of markers. Thus in developing our reasoning system using temporal synchrony we have also established that marker passing systems can be realized in a neurally plausible manner. In the proposed system, nothing has to be stored at a node in order to mark it with a marker. Instead, the *time* of firing of a node *relative* to other nodes and the *coincidence* between the time of firing of a node and that of other nodes, has the *effect* of marking a node with a particular marker! A node in our system is not required to match any markers, it simply has to detect whether appropriate inputs are coincident. Thus our approach enables us to realize the abstract notion of markers by using time, a dimension that is forever present, and giving it added representational status.

As we shall see, the neural plausibility of our system also results from its ability to operate without a central controller. Once a query is posed to the system by activating appropriate nodes, it computes the solution without an external controller directing the activity of nodes at each step of processing. (Also see Section 9.1).

Several other connectionist solutions to the binding problem have been suggested (Feldman 1982; Touretzky & Hinton 1988; Barnden & Srinivas 1991; Dolan & Smolensky 1989; and Lange & Dyer 1989). These alternatives are discussed in Section 9.3.

3.1 Representing dynamic bindings

Refer to the representation of some predicates and entities shown in Fig. 2. Observe that predicates, their arguments, and entities are represented using distinct nodes. For example, the ternary predicate *give* is represented by the three argument nodes labeled *giver, recip,* and *g-obj* together with an associated 'node' depicted as a dotted rectangle. The role of this 'node' will be specified later in Section 3.3. For simplicity we will assume that each argument node corresponds to an individual connectionist node. This is an idealization. Later in Section 7.3 we discuss how each argument node corresponds to an *ensemble* of nodes. Nodes such as *John* and *Mary* correspond to *focal* nodes of more elaborate connectionist representations of the entities 'John' and 'Mary'. Information about the attribute values (features) of 'John' and its relationship to other concepts is encoded by linking the focal node *John* to appropriate nodes. Details of such an encoding may be found in (Shastri & Feldman 1986; Shastri 1991). As explained in (Feldman 1989), a focal node may also be realized by a small ensemble of nodes.

********** Figure 2 goes about here **********

Dynamic bindings are represented in the system by the *synchronous* firing of appropriate nodes. Specifically, a dynamic binding between a predicate argument and its filler is represented by the synchronous 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-object=Book1),* i.e., the dynamic fact *give(John, Mary, Book1),* is represented by *rhythmic* pattern of activity shown in Fig. 3. The absolute phase of firing of fillers and arguments nodes is not significant — what matters is the *coincidence* (or the lack thereof) in the firing of nodes. The activity of the dotted rectangular nodes is not significant at this point and is not specified. As another example, consider the firing pattern shown in Fig. 4. This pattern of activation represents the single binding *(giver=John)* and corresponds to the partially instantiated fact *give(John,x,y)* (i.e., 'John gave someone something').

********** Figures 3, 4 and 5 go about here **********

Fig. 5 shows the firing pattern of nodes corresponding to the dynamic representation of the bindings *(giver=John, recipient=Mary, give-object=Book1, owner=Mary, own-object=Book1, potential-seller=Mary, can-sell-object=Book1).* These bindings encode the facts *give(John, Mary, Book1), own(Mary,Book1),* and *can-sell(Mary,Book1).* Observe that the (multiple) bindings between *Mary* and the arguments *recipient, owner,* and *potential-seller* are represented by these argument nodes firing in-phase with *Mary.* 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 some significant aspects of the proposed solution:

- The transient or short-term representation of an entity is simply a *phase* within a rhythmic pattern of activity
- The number of *distinct* phases within the rhythmic activation pattern only equals the number of *distinct* entities participating in the dynamic bindings. In particular, this does not depend on the total number of dynamic bindings being represented by the activation pattern.
- The number of distinct entities that can participate in dynamic bindings at the same time is limited by the ratio of the period of the rhythmic activity and the width of individual spikes.

Thus far we have assumed that nodes firing in synchrony fire precisely in-phase. This is an idealization. In general we would assume a coarser form of synchrony where nodes firing with a lag or lead of less than $\omega/2$ of one another would be considered to be firing in synchrony. This corresponds to treating the width of the 'window of synchrony' to be ω.

3.2 Encoding rules and propagating dynamic bindings

In Section 2.2 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 (i) the argument bindings of the former and (ii) the argument correspondence specified by the rule. Consequently, the encoding of a rule should provide a means for propagating bindings from the arguments of the antecedent predicate to the arguments of the consequent predicate in accordance with the argument correspondence specified in the rule. With reference to Fig. 2, encoding the rules $\forall x, y, z\ [give(x, y, z) \Rightarrow own(y, z)]$ and $\forall u, v\ [own(u, v) \Rightarrow can\text{-}sell(u, v)]$ should have the following effect: The state of activation described by the rhythmic activation pattern shown in Fig. 3 should eventually lead to the rhythmic activation pattern shown in Fig. 5.

The desired behavior may be realized if a rule is encoded by linking the arguments of the antecedent and consequent predicates so as to reflect the correspondence between arguments specified by the rule. For example, the rule $\forall x, y, z\ [give(x, y, z) \Rightarrow own(y, z)]$ can be encoded by establishing links between the arguments *recipient* and *give-object* of *give* and the arguments *owner* and *own-object* of *own*, respectively. If we also wish to encode the rule: $\forall x, y\ [buy(x, y) \Rightarrow own(x, y)]$, 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. 6. In the idealized model we are assuming that each argument is represented as a single node and each argument correspondence is encoded by a one to one connection between the appropriate argument nodes. As discussed in Section 7.3, however, each argument will be encoded as an ensemble of nodes and each argument correspondence will be encoded by many-to-many connections between the appropriate ensembles (for a preview see Fig. 26).

********** Figure 6 goes about here **********

Arguments and concepts are encoded using what we call ρ-**btu** nodes. These nodes have the following idealized behavior:

- If a node A is connected to node B then the activity of node B will synchronize with the activity of node A. In particular, a periodic firing of A will lead to a periodic and *in-phase* firing of B. We assume that ρ-btu nodes can respond in this manner as long as the period of firing, π, lies in the interval $[\pi_{min}, \pi_{max}]$. This interval can be interpreted as defining the frequency range over which ρ-btu nodes can sustain a synchronized response.

- To simplify the description of our model we will assume that periodic activity in a node can lead to synchronous periodic activity in a connected node *within one period*.

- A threshold, n, associated with a node indicates that the node will fire only if it receives n or more *synchronous* inputs.[15] If unspecified, a node's threshold is assumed to be 1.[16]

As described above, interconnected ρ-btu nodes can propagate synchronous activity and form chains of nodes firing in synchrony. In Section 7 we point to evidence from neurophysiology and cite work on neural modeling which suggests that the propagation of synchronous activity is neurally plausible.

********** Figures 7, 8 and 9 go about here **********

Given the above interconnection pattern and node behavior, the initial state of activation shown in Fig. 7 will lead to the state of activation shown in Fig. 8 after one period, and to the state of activation shown in Fig. 9 after another period.

The encoding of rules by the explicit encoding of the inferential dependency between predicates and predicate arguments, in conjunction with the use of temporal synchrony provides an efficient mechanism for propagating dynamic bindings and performing systematic reasoning. Conceptually, the proposed encoding of rules creates a directed *inferential dependency graph*: Each predicate argument is represented by a node in this graph and each rule is represented by links between nodes denoting the arguments of the antecedent and consequent predicates. In terms of this conceptualization, it should be easy to see that the evolution of the system's state of activity corresponds to a *parallel* breadth-first traversal of the directed inferential dependency graph. This means that i) a large number of rules can fire in parallel and ii) the time taken to generate a chain of inference is independent of the total number of rules and just equals $l\pi$ where l is the *length* of the chain of inference and π is the period of oscillatory activity.

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

As stated in Section 2.3, our system must also be capable of representing long-term facts which are essentially a permanent record of a set of bindings describing a particular situation. The representation of a long-term fact should encode the bindings pertaining to the fact in a manner that allows the system to rapidly recognize dynamic bindings that match the encoded fact. Given that dynamic bindings are represented as temporal patterns, it follows that the encoding of a long-term fact should behave like a *temporal pattern matcher* that becomes active whenever the static bindings it encodes match the dynamic bindings represented in the system's state of activation.

********** Figures 10 and 11 go about here **********

The design of such a temporal pattern matcher is illustrated in Figs. 10 and 11 that depict the encoding of the long-term facts *give(John,Mary,Book1)* and *give(John,Susan,x)*, respectively

(the latter means 'John gave Susan something'). The encoding fully specifies how a predicate is encoded. Observe that in addition to ρ-btu nodes, the encoding also makes use of pentagon shaped τ-and nodes that have the following idealized behavior:

- A τ-and node becomes active on receiving an *uninterrupted* pulse train, i.e., a pulse train such that the gap between adjacent pulses is *less than* a spike width. Thus a τ-and node behaves like a *temporal and* node. On becoming active, such a node produces an output pulse train similar to the input pulse train.

- Note that a τ-and node driven by a periodic input consisting of a train of pulses of width comparable to the period π, will produce a periodic train of pulses of width and periodicity π. We assume that a τ-and node can behave in this manner as long as the period of the input pulse train lies in the interval $[\pi_{min}, \pi_{max}]$.

- A threshold, n, associated with a τ-and node indicates that the node will fire only if it receives n or more synchronous pulse trains. If unspecified, n is assumed to be 1.

An n-ary predicate P is encoded using two τ-and nodes and n ρ-btu nodes. One of these τ-and nodes is referred to as the *enabler* and the other as the *collector*. An *enabler* will be referred to as *e:P* and drawn pointing upwards while a *collector* will be referred to as *c:P* and drawn pointing downwards. With reference to Figs. 10 and 11, 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*. The representational significance of the *enabler* and *collector* nodes is as follows. The *enabler e:P* of a predicate P has to be activated whenever the system is queried about P. Such a query may be posed by an external process or generated internally by the system itself during an episode of reasoning (see Section 4.4). On the other hand, the system activates the *collector c:P* of a predicate P whenever the dynamic bindings of the arguments of P are consistent with the knowledge encoded in the system.

A long-term 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 an entity, the modifier input from the argument node is in turn modified by an inhibitory link from the appropriate entity node. The output of the τ-and node encoding a long-term fact is connected to the *collector* of the associated predicate. We will refer to the τ-and node associated with a long-term fact as a fact node. Note that there is only one *enabler* node, one *collector* node, and one set of argument nodes for each predicate. These nodes are shared by all the long-term facts pertaining to that predicate.

It can be shown that a fact node becomes active if and only if the static bindings it encodes match the dynamic bindings represented in the network's state of activation. As stated above, *e:P* becomes active whenever any *query* involving the predicate P is represented in the system. Once active, *e:P* outputs an uninterrupted pulse train which propagates to various fact nodes attached to *e:P*. Now the pulse train arriving at a fact node will be interrupted by an active argument of *P*, unless the filler of this argument specified by the long-term fact is firing in synchrony with the argument. But a filler and an argument will be firing in synchrony if and only if they are bound in the dynamic bindings. Thus a fact node will receive an uninterrupted pulse 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 Figs. 10 and 11 will behave as expected.

The encoding of the long-term fact *give(John, Mary, Book1)* will recognize dynamic bindings that represent dynamic facts such as: *give(John, Mary, Book1), give(John, Mary, x), give(x, Mary, y),* and *give(x, y, z).* However it will not recognize those that represent *give(Mary, John, Book1)* or *give(John, Susan, x).* Similarly, the encoding of the long-term fact *give(John, Susan, x)* will recognize dynamic bindings that encode: *give(John, Susan, x), give(x, Susan, y),* and *give(x, y, z),* but not *give(Susan, John, x)* or *give(John, Susan, Car7).*

3.4 Dynamic bindings and temporal synchrony— some observations

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, iv) rules as interconnection patterns that cause the propagation and transformation of such rhythmic patterns of activation, and v) facts as temporal pattern matchers. During an episode of reasoning, all the arguments bound to a filler become active in the same phase as the filler thereby creating transient 'temporal frames' of knowledge grouped together by temporal synchrony. This can be contrasted with 'static' frames of knowledge where knowledge is grouped together — spatially — using hard-wired links and nodes.

The system can represent a large number of dynamic bindings at the same time, provided the number of distinct entities involved in these bindings does not exceed $\lfloor \pi_{max}/\omega \rfloor$, where π_{max} is the maximum period (or the lowest frequency) at which ρ-btu nodes can sustain synchronous oscillations and ω is the width of the window of synchrony. Recall that a window of synchrony of ω implies that nodes firing with a lag or lead of less than $\omega/2$ of one other are considered to be in synchrony. We discuss biologically plausible values of π and ω in Section 7.2 and the psychological implications of these limits in Section 8.

As described thus far, the system allows the simultaneous representation of a large number of dynamic facts but only supports the representation of one *dynamic* fact *per* predicate. In Section 6 we discuss a generalization of the proposed representation that allows multiple dynamic facts pertaining to each predicate to be active simultaneously.

Although synchronous activity is central to the representation and propagation of binding, the system does not require a global clock or a central controller. The propagation of in-phase activity occurs automatically — once the system's state of activation is initialized to represent an input situation by setting up appropriate dynamic bindings, the system state evolves automatically to represent the dynamic bindings corresponding to situations that follow from the input situation.

Reasoning is the spontaneous outcome of the system's behavior. The system does not encode syntactic rules of inference such as *modus-ponens*. There is no separate interpreter or inference mechanism in the system that manipulates and rewrites symbols. The encoding of the LTKB is best viewed as a vivid *internal model* of the agent's environment. When the nodes in this model are activated to reflect a particular situation in the environment, the model simulates the behavior

of the external world and dynamically creates a vivid model of the state of affairs resulting from the given situation. The system is clearly not a *rule following* system. At the same time it is not *rule described* or *rule governed* in the sense a falling apple is. As Hatfield (1991) argues, the system is best described as being *rule instantiating*.

3.5 From mechanisms to systems

The mechanisms proposed in the previous sections provide the building blocks for a connectionist system that can represent and reason with knowledge involving n-ary predicates and variables. These mechanisms may interact in different ways to realize different sorts of reasoning behavior. For example, they can lead to a forward reasoning system that can perform *predictive* inferences. Our discussion in the previous sections was in the context of such a system.

The proposed mechanisms may also be used to create a backward reasoning system that behaves as follows: If the system's state of activation is initialized to represent a query, it attempts to answer the query based on the knowledge encoded in its LTKB. A backward reasoning system may be generalized to perform *explanatory* inferences. If the state of such a system is initialized to represent an input 'situation', it will automatically attempt to explain this situation on the basis of knowledge in its LTKB and a 'minimal' set of assumptions.

With the aid of additional mechanisms it is possible to design a system that performs both predictive and explanatory inferences. Such a system would make predictions based on incoming information and at the same time, seek explanations for, and test the consistency of, this information.

4 A backward reasoning system

In this section we describe a *backward* reasoning system based on the representational mechanisms described in Section 3. The system encodes facts and rules in its LTKB and answers queries based on this knowledge. For example, if the system encodes rules $\forall x, y, z\ [give(x, y, z) \Rightarrow own(y, z)]$ and $\forall u, v\ [own(u, v) \Rightarrow can\text{-}sell(u, v)]$, and the long-term fact 'John *bought* Porsche7' it will respond 'yes' to queries such as 'Does John *own* Porsche7', or 'Can John *sell* something?'. The time taken to respond 'yes' to a query is only proportional to the *length* of the shortest derivation of the query and is independent of the size of the LTKB.

In subsequent sections we describe several extensions of the backward reasoning system. In Section 5 we show how the system may be combined with an *IS-A* hierarchy that encodes entities, types (categories), and the super/sub-concept relations between them. The augmented system allows the occurrence of types, non-specific instances of types, as well as entities in rules, facts, and queries. This in turn makes it easier to encode the appropriateness aspect of rules. An extension of the system to perform abduction is described in (Ajjanagadde 1991).

4.1 The backward reasoning system – a functional specification

The reasoning system can encode rules of the form:[17]

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

The arguments of P_i's are elements of $\{x_1, x_2, ...x_m\}$. An argument of Q is either an element of $\{x_1, x_2, ...x_m\}$, or an element of $\{z_1, z_2, ...z_l\}$, or a constant. It is required that any variable occurring in multiple argument positions in the antecedent of a rule must also appear in the consequent.[18] The significance of this constraint is discussed in Section 4.9. Additional examples of rules are:

$\forall x, y, t \; [omnipresent(x) \Rightarrow \; present(x, y, t)]$
; 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.
$\forall x \; [triangle(x) \Rightarrow number\text{-}of\text{-}sides(x, 3)]$
$\forall x, y \; [sibling(x, y) \wedge born\text{-}together(x, y) \Rightarrow twins(x, y)]$

Facts are assumed to be partial or complete instantiations of predicates. In other words, facts are atomic formulae of the form $P(t_1, t_2 ... t_k)$ where t_i's are either constants or distinct existentially quantified variables. Some examples of facts are:

$give(John, Mary, Book1)$;	John gave Mary Book1.
$give(x, Susan, Ball2)$;	Someone gave Susan Ball2.
$buy(John, x)$;	John bought something.
$own(Mary, Ball1)$;	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.

A query has the same form as a fact. 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?' Table 1 lists some queries, their interpretation(s), and their answer(s).

****** Table 1 goes about here ******

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. We will also restrict ourselves to *yes-no* queries at first. Subsequent sections will provide the relevant details.

4.2 Encoding rules and facts in the long-term memory

Fig. 12 depicts the encoding of 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 facts: *give(John, Mary, Book1)*, *buy(John, x)*, and *own(Mary, Ball1)*.

********** Figure 12 goes about here **********

As stated in Section 3, a constant (i.e., an entity) is represented by a ρ-btu node and an n-ary predicate is represented by a pair of τ-and nodes and n ρ-btu nodes. One of the τ-and nodes is referred to as the *enabler* and the other as the *collector*. An *enabler* is drawn pointing upwards and is named *e:[predicate-name]*. A *collector* is drawn pointing downwards and is named *c:[predicate-name]*. The *enabler*, *e:P*, of a predicate P has to be activated whenever the system is queried about P. As we shall see, such a query may be posed by an external process or generated internally by the system during an episode of reasoning. On the other hand, the system activates the *collector*, *c:P*, of a predicate P whenever the current dynamic bindings of the arguments of P are consistent with the long-term knowledge encoded in the system.

Each fact is encoded using a distinct τ-and node that is interconnected with the appropriate enabler, collector, argument and entity nodes as described in Section 3.3.

A rule is encoded by connecting i) the collector of the antecedent predicate to the collector of the consequent predicate, ii) the enabler of the consequent predicate to the enabler of the antecedent predicate, and iii) 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. 12). Notice that the links are directed from the arguments of the consequent predicate to the arguments of the antecedent predicate. The direction of links is *reversed* because the system performs backward reasoning.

4.3 Posing a query: specifying dynamic bindings

A query is a (partially) specified predicate instance of the form $P(t_1, ..., t_n)?$, where t_is are either constants (entities) or existentially quantified variables. Therefore, posing a query to the system involves specifying the query predicate and the argument bindings specified in the query. We will assume that only one external process communicates with the reasoning system. The possibility of communication among several modules is discussed in Section 10.4 (also see Sections 10.1–10.3). Let us choose an *arbitrary* point in time — say, t_0 — as our point of reference for initiating the query. The argument bindings specified in the query are communicated to the network as follows:

- Let the argument bindings involve m distinct entities: $c_1, ..., c_m$. With each c_i, associate a delay δ_i such that no two delays are within ω of one another and the longest delay is less than $\pi - \omega$. Here ω is the width of the window of synchrony, and π lies in the interval $[\pi_{min}, \pi_{max}]$.

- The argument bindings of an entity c_i are indicated to the system by providing an oscillatory spike train of periodicity π starting at $t_0 + \delta_i$, to c_i and all arguments of the query predicate bound to c_i. As a result, a distinct phase is associated with each distinct entity introduced

in the query and argument bindings are represented by the synchronous activation of the appropriate entity and argument nodes.

- The query predicate is specified by activating *e:P*, the enabler of the query predicate P, with a pulse train of width and periodicity π starting at time t_0.

Observe that posing a query simply involves activating the enabler node of the query predicate and the arguments and fillers specified in the query. There is no central controller that monitors and regulates the behavior of individual nodes at each step of processing.

4.4 The inference process for *yes-no* queries

Once a query is posed to the system, it's state of activation evolves automatically and produces an answer to the query. The activation of the collector node of the query predicate indicates that the answer to the query is *yes*. The time taken by the system to produce a *yes* answer equals $2\pi(l+1)$, where π is the period of oscillation of nodes and l equals the *length* of the *shortest* derivation of the query.[19] If the collector node of the query predicate does not receive any activation within $2(d+1)$ periods of oscillations — where d equals the diameter of the inferential dependency graph, the answer to the query is *don't know*. If we make the *closed world assumption*[20] then a don't know answer can be viewed as a 'no' answer.

We illustrate the inference process with the help of an example (refer to Fig. 12). Consider the query *can-sell(Mary,Book1)?* (i.e., 'Can Mary sell Book1?'). This query is posed by providing inputs to the entities *Mary* and *Book1*, the arguments *p-seller*, *cs-obj* and the enabler *e:can-sell* as shown in Fig. 13. Observe that *Mary* and *p-seller* receive synchronous activation and so do *Book1* and *cs-obj*. Let us refer to the phase of activity of *Mary* and *Book1* as phase-1 and phase-2 respectively.

********** Figure 13 goes about here **********

As a result of the inputs, *Mary* and *p-seller* fire synchronously in phase-1, while *Book1* and *cs-obj* fire synchronously in phase-2 of every period of oscillation The node *e:can-sell* also oscillates and generates 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. Thereafter, the nodes *Mary, owner,* and *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*. At this point the system has essentially, created two dynamic bindings — *owner = Mary* and *own-object = Book1*. Given that *e:own* is also active, the system's state of activity now also encodes the internally generated query *own(Mary,Book1)?* (i.e., 'Does Mary own Book1?')!

The fact node associated with the fact *own(Mary,Ball1)* does not match the query and remains inactive. Recall that fact nodes are τ-and nodes and hence, become active only upon receiving an uninterrupted pulse train (see Section 3.3). Since *Ball1* is not firing, the inhibitory activation from the argument node *owner* interrupts the activation going from *e:own* to the fact node and prevents it from becoming active.

The activation from *owner* and *o-obj* reaches the arguments *recip* and *g-obj* of *give*, and *buyer* and *b-obj* of *buy* respectively. Thus beginning with the third period, arguments *recip* and *buyer* become active in phase-1 while arguments *g-obj* and *b-obj* become active in phase-2. In essence, the system has created new bindings for the arguments of predicates *can-sell* and *buy*. Given that the nodes *e:buy* and *e:give* are also active, the system's state of activity now encodes two additional queries: *give(x,Mary,Book1)?* and *buy(Mary,Book1)?*

The fact node representing the fact *buy(John,x)* does not become active because the activation from *e:buy* is interrupted by the inhibitory activations from the arguments *buyer* and *b-obj*. (Notice that *John* is not active). The fact node $F1$, associated with the fact *give(John,Mary,Book1)* however, does become active as a result of the uninterrupted activation it receives from *e:give*. Observe that the argument *giver* is not firing and the inhibitory inputs from the arguments *recip* and *g-obj* are blocked by the synchronous inputs from *Mary* and *Book1*, respectively. The activation from $F1$ 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 the query predicate *can-sell* becomes active, resulting in an affirmative answer to the query.

4.5 Encoding rules with constants and repeated variables in the consequent

In this section we describe how rules containing constants (entities) and/or existentially quantified variables in the consequent are encoded. Consider the rule:

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

********** Figure 14 goes about here **********

The encoding of the rule $R2$ is shown in Fig. 14. It uses a new type of node which we refer to as a τ-or node (node $g1$, in Fig. 14). Such a node behaves like a temporal *or* node and becomes active on receiving any input above its threshold and generates an oscillatory response with a period and pulse width equal to π_{max}, the maximum period at which the ρ-btu nodes can sustain synchronous activity.

Node $g1$ projects inhibitory modifiers to links between argument and enabler nodes that can block the firing of the rule. The node $g1$ ensures that the rule participates in an inference only if all the conditions implicit in the consequent of the rule are met. The first condition concerns the occurrence of existentially quantified variables in the consequent of a rule. Observe that such a rule only warrants the inference that there exists *some* filler of an existentially quantified argument, and hence, cannot be used to infer that a specific entity fills such an argument. Therefore, if an existentially quantified variable in the consequent of a rule gets bound in the reasoning process, the rule cannot be used to infer the consequent. With reference to $R2$, the desired behavior is achieved by the link from the existentially quantified (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 to any filler.

The second condition concerns the occurrence of entities in the consequent of a rule. $R2$ cannot be used if its fifth argument is bound to any entity other than a. In general, a rule that has an entity in its consequent cannot be used if the corresponding argument gets bound to any other entity during the reasoning process. In the encoding of $R2$, 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 entity other than a, $g1$ will become active and block the firing of the rule.

If the same variable occurs in multiple argument positions in the consequent of a rule, it means that this variable should either remain unbound or get bound to the same entity. This constraint can be encoded by introducing a node that receives inputs from all the arguments that correspond to the same variable

and becomes active and inhibits the firing of the rule unless all such arguments are firing in synchrony. Observe that due to the temporal encoding, arguments bound to the same entity will fire in the same phase and hence, *a node need only check that the inputs from appropriate argument nodes are in synchrony to determine that the arguments are bound to the same entity.* Consider the network fragment shown in Fig. 15 that depicts the encoding of the rule $\forall x, y\ P(x) \Rightarrow Q(x, x, y, a)$. 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 rule is inhibited unless the appropriate arguments are bound to the same entity.

********** Figure 15 goes about here **********

4.6 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. 16 illustrates the encoding of the multiple antecedent rule $\forall x, yP(x, y) \wedge Q(y, x) \Rightarrow R(x, y)$. The τ-and node labeled $g3$ has a threshold of 2.

********** Figure 16 goes about here **********

4.7 Answering *wh*-queries

As stated in Section 4.1, a query with unbound arguments can be interpreted either as a *yes-no* query or a *wh*-query. To answer a *yes-no* query the system need only determine whether there exist *some* instantiations of the unbound arguments. To answer a *wh*-query, however, the system must also determine the instantiations of unbound arguments for which the query is true. We describe how the proposed system can be extended to do so.

Consider the proof of the query $can\text{-}sell(Mary, x)$ with respect to the network shown in Fig. 12. The *yes-no* version of this query will be answered in the affirmative and 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?', simply consists of the entities bound to the arguments $g\text{-}obj$ and $b\text{-}obj$, respectively, of the two active facts. Observe that the arguments $g\text{-}obj$ and $b\text{-}obj$ are precisely the arguments that map to the unbound argument $cs\text{-}obj$ of $can\text{-}sell$ via the rules encoded in the system. The system can extract this information by the same binding propagation mechanism it uses to map bound arguments. A straightforward way of doing so is to posit an *answer extraction* stage that occurs 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 activate the facts that lead to a *yes* answer, namely, $own(Mary, Ball1)$ and $give(John, Mary, Book1)$. The answer extraction stage follows and picks out the entities $Ball1$ and $Book1$ as the answers.

********** Figure 17 goes about here **********

In order to support answer extraction, the representation of a fact is augmented as shown in Fig. 17. The representation of a fact involving an n-ary predicate is modified to include $n + 1$

additional nodes: for each of the n arguments there is a two input ρ-btu node with a threshold of two. We 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 remains so even after the inputs are withdrawn (for several periods of oscillations). This node, which we will refer to as a *latch* node, receives an *Answer* input and an input from the associated fact node.

At the end of the first stage, the fact nodes corresponding to all the relevant facts would have become active. The output of these 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 (i.e., the collector of the query predicate becomes active), the desired *unbound* arguments of the query predicate are activated in a distinct phase. The activation of these arguments eventually leads to the activation of the appropriate arguments in the facts relevant to answering 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 an output that activates the associated entities in phase with the appropriate query arguments. The answer to the *wh*-query — i.e., the entities that fill the argument a_i of the query — will be precisely those entities 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.

4.8 Admitting function terms

The expressiveness and reasoning power of the system can be extended by allowing restricted occurrences of function-terms in rules and facts. Function-terms introduce new entities during the reasoning process. But given that entities are represented as a phase in the pattern of activity, an entity introduced by a function-term can be represented by an additional phase in the rhythmic activity. Thus the reference to $mother\text{-}of(Tom)$ during an episode of reasoning should lead to activity in a distinct phase. This phase would represent the 'mother of Tom' and any arguments bound to the 'mother of Tom' would now fire in this phase. A provisional solution along these lines is described in (Ajjanagadde 1990).

4.9 Constraints on the form of rules

The encoding of rules described thus far enforces i) the correspondence between the arguments of the antecedent and consequent predicates in a rule and ii) equality among arguments in the consequent of a rule. In certain cases, however, it is difficult for the 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 unspecified. Consequently, the system cannot enforce the condition implicit in the rule 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. Performing such an equality test is complicated in a system that allows multiple predicates in the antecedent of rules and the chaining of inference. Consider the rule $\forall x, y \; P(x, y) \wedge R(x, y) \Rightarrow Q(y)$, and the query $Q(a)?$ The predicates P and R may be derivable

from other predicates by a long sequence of rule application. Hence to derive the query $Q(a)$?, the system may have to test the equality of arbitrary pairs of argument-fillers in a potentially large number of facts distributed across the LTKB. It is conjectured that such non-local and exhaustive equality testing cannot be done effectively in any model that uses only a linear number of nodes in the size of the LTKB, and time that is independent of the size of the LTKB.

Contrast the above 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 automatically enforced by the propagation of bindings and is expressed in the dynamically generated query at P. 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 reasoning process. The above entails that for the system to respond to a query,

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

The above constraint is required in a backward reasoning system but not in a forward reasoning system. In a forward reasoning system, the rule $\forall x, y, z\ P(x, y) \wedge Q(y, z) \Rightarrow R(x, z)$ would be encoded as shown in Fig. 18. The τ-or node with a threshold of 2 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 ensures that the rule fires only if the second and first arguments of P and Q respectively, are bound to the same filler. In the case of forward reasoning, a rule that has variables occurring in multiple argument positions in its *consequent* can participate in the reasoning process provided such variables also appear in its antecedent and get bound during the reasoning process. The restrictions mentioned above on the form of rules excludes certain inferences. We discuss these exclusions and their implications in Section 8.2.

********** Figure 18 goes about here **********

5 Integrating the rule-based reasoner with an *IS-A* hierarchy

The rule-based reasoner described in the previous section can be integrated with an *IS-A* hierarchy representing entities, types (categories), the instance-of relations between entities and types, and the super/sub-concept relations between types. For convenience, we will refer to the instance-of, super-concept, and sub-concept relations collectively as the *IS-A* relation. The augmented system allows the occurrence of types as well as entities in rules, facts, and queries. Consequently, the system can store and retrieve long-term facts such as 'Cats prey on birds' and 'John bought a Porsche' that refer to types ('Cat' and 'Bird') as well as non-specific instances of types ('a Porsche'). The system can also combine rule-based reasoning with type inheritance. For example, it can infer 'John owns a car' and 'Tweety is scared of Sylvester' (the latter assumes the existence of the rule 'If x preys on y then y is scared of x' and the *IS-A* relations 'Sylvester is a Cat' and 'Tweety is a Bird'). Combining the reasoning system with an *IS-A* hierarchy also

facilitates the representation of the *appropriateness* aspect of a rule. Recall that appropriateness concerns the applicability of the systematic aspect of a rule in a given situation, depending on the types of the argument fillers involved in that situation. As we shall see, the augmented system allows knowledge in the *IS-A* hierarchy to interact with the encoding of the systematic aspects of a rule in order to enforce type restrictions and type preferences on argument-fillers.

The integration of the reasoner with the *IS-A* hierarchy described below is a first cut at enriching the representation of rules. We only model the instance-of, sub-concept, and super-concept relations and suppress several issues such as a richer notion of semantic distance, frequency and category size effects, and prototypicality (e.g., see Lakoff 1987).

********** Figure 19 goes about here **********

Fig. 19 provides an overview of the encoding and reasoning in the integrated reasoning system. The rule-base part of the network in Fig. 19 encodes the rule $\forall x, y\ [preys\text{-}on(x, y) \Rightarrow scared\text{-}of(y, x)]$, and the facts $\forall$*x:Cat, y:Bird preys-on(x,y)* and $\exists$*x:Cat* $\forall$*y:Bird loves(x,y)*.

The first fact says 'cats prey on birds' and is equivalent to *preys-on(Cat,Bird)*. The second fact states 'there exists a cat that loves all birds'. The type hierarchy in Fig. 19 encodes the *IS-A* relationships: *is-a(Bird,Animal)*, *is-a(Cat,Animal)*, *is-a(Robin,Bird)*, *is-a(Canary,Bird)*, *is-a(Tweety,Canary)*, *is-a(Chirpy,Robin)*, and *is-a(Sylvester,Cat)*. Facts involving typed variables are encoded in the following manner: A typed, universally quantified variable is treated as being equivalent to its type. Thus $\forall$*x:Cat, y:Bird preys-on(x,y)* is encoded as *preys-on(Cat,Bird)*. A typed existentially quantified variable is encoded using a unique sub-concept of the associated type. Thus in Fig. 19, $\exists$ *x:Cat* $\forall$*y:Bird loves(x,y)* is encoded as *loves(Cat-1,Bird)* where *Cat-1* is some unique instance of *Cat*. In its current form, the system only deals with facts and queries wherein all existential quantifiers occur outside the scope of universal quantifiers.

For now let us assume that i) each concept[21] (type or entity) in the *IS-A* hierarchy is encoded as a ρ-btu node, ii) each *IS-A* relationship, say *is-a(A,B)*, is encoded using two links — a bottom-up link from *A* to *B* and a top-down link from *B* to *A*, and iii) the top-down and bottom-up links can be enabled selectively by *built-in* and automatic control mechanisms. How this is realized is explained in Section 5.2.

********** Figure 20 goes about here **********

The time course of activation for the query: *scared-of(Tweety,Sylvester)?* (Is Tweety scared of Sylvester?) is given in Fig. 20. The query is posed by turning on *e:scared-of* and activating the nodes *Tweety* and *Sylvester* in synchrony with the first and second arguments of *scared-of*, respectively. The bottom-up links emanating from *Tweety* and *Sylvester* are also enabled. The activation spreads along the *IS-A* hierarchy and eventually, *Bird* and *Cat* start firing in synchrony with *Tweety* and *Sylvester*, respectively. At the same time, the activation propagates in the rule-base. As a result, the initial query *scared-of(Tweety,Sylvester)?* is transformed into the query *preys-on(Cat,Bird)?* which matches the stored fact *preys-on(Cat,Bird)* and leads to the activation of *c:preys-on*. In turn *c:scared-of* becomes active and signals an affirmative answer.

The advantages of expressing certain rules as facts

Although the reasoning system described in Section 4 can *use* rules to draw inferences, it cannot *retrieve* the rules *per se*; for knowledge to be retrievable, it must be in the form of a fact.

Hence integrating the rule-based reasoner with an *IS-A* hierarchy has added significance because it allows certain rule-like knowledge to be expressed as facts thereby also making it retrievable in addition to being usable during inference. Consider 'Cats prey on birds'. The rule-based reasoner can only express this as the rule: $\forall$ *x,y Cat(x)* $\wedge$ *Bird(y)* $\Rightarrow$ *preys-on(x,y)* and use it to answer queries such as *preys-on(Sylvester,Tweety)?*. It however, cannot answer queries such as *preys-on(Cat,Bird)?* that can be answered by the integrated system.

5.1 Some technical problems

There are two technical problems that must be solved in order to integrate the *IS-A* hierarchy and the rule-based component. First, the encoding of the *IS-A* hierarchy should be capable of representing *multiple instantiations* of a concept. For example, in the query discussed above, the concept *Animal* would receive activation originating at *Tweety* and well as *Sylvester*. We would like the network's state of activation to represent both 'the animal Tweety' and 'the animal Sylvester'. This however, cannot happen if concepts are represented by a single ρ-btu node because the node *Animal* cannot fire in synchrony with both *Tweety* and *Sylvester* at the same time. Second, the encoding must provide *built-in* mechanisms for *automatically* controlling the direction of activation in the *IS-A* hierarchy so as to correctly deal with queries containing existentially and universally quantified variables. The correct treatment of quantified variables — assuming that all *IS-A* links are indefeasible, i.e., without exceptions[22] — requires that activation originating from a concept C that is either an entity or the type corresponding to a universally quantified variable in the query, should propagate upwards to all the ancestors of C. The upward propagation checks if the relevant fact is universally true of some super-concept of C. The activation originating from a concept C that appears as an existentially quantified variable in the query should propagate to the ancestors of C, the descendants of C, as well as the ancestors of the descendants of C.[23] A possible solution to these problems has been proposed in (Mani & Shastri 1991) and is outlined below.

5.2 Encoding of the *IS-A* hierarchy

Each concept C represented in the *IS-A* hierarchy, is encoded by a group of nodes called the *concept cluster* for C. Such a cluster is shown in the middle of Fig. 21. The concept cluster for C has k_1 *banks* of ρ-btu nodes, where k_1 is the *multiple instantiation constant* and refers to the number of dynamic instantiations a concept can accommodate. In general, the value of k_1 may vary from concept to concept but for ease of exposition we will assume that it is the same for all concepts. In Fig. 21 k_1 is 3. Each *bank* of concept C consists of three ρ-btu nodes: C_i, $C_{i\uparrow}$, $C_{i\downarrow}$. Each C_i can represent a distinct (dynamic) instantiation of C. The *relay* nodes $C_{i\uparrow}$ and $C_{i\downarrow}$ control the *direction* of the propagation of activation from C_i. The nodes $C_{i\uparrow}$ and $C_{i\downarrow}$ have a threshold of 2. Note that C_i is connected to $C_{i\uparrow}$ and $C_{i\downarrow}$ and $C_{i\downarrow}$ is linked to $C_{i\uparrow}$.

********** Figure 21 goes about here **********

Every concept C is associated with two subnetworks — the *top-down* switch and the *bottom-up* switch. These switches are identical in structure and automatically control the flow of activation to the concept cluster. A switch has k_1 outputs. $Output_i$ ($1 \leq i \leq k_1$) from the bottom-up

switch connects to C_i and $C_{i\uparrow}$ while the $output_i$ from the top-down switch goes to nodes C_i and $C_{i\downarrow}$. The bottom-up switch has $k_1 n_{sub}$ inputs while the top-down switch has $k_1 n_{sup}$ inputs, where n_{sub} and n_{sup} are the number of sub- and super-concepts of C, respectively. There are also links from the C_i nodes to both the switches. The interaction between the switches and the concept cluster brings about efficient and *automatic* dynamic allocation of banks in the concept cluster by ensuring that (i) activation gets channeled to the concept cluster banks only if any 'free' banks are available and (ii) each instantiation occupies only one bank.

********** Figure 22 goes about here **********

The architecture of the switch (with $k_1 = 3$) is illustrated in Fig. 22. The k_1 ρ-btu nodes, $S_1, \ldots, S_{k_1}$, with their associated τ-or nodes form the switch. Inputs to the switch make two connections — one excitatory and one inhibitory — to each of $S_2, \ldots, S_{k_1}$. As a result of these excitatory-inhibitory connections, nodes $S_1, \ldots, S_{k_1}$ are initially disabled and cannot respond to incoming activation. Any input activation only effects node S_1 since the switch inputs directly connect to S_1. S_1 becomes active in response to the first available input and continues to fire in phase with the input as long as the input persists. As S_1 becomes active, the τ-or node associated with S_1 turns on and enables S_2. However, inhibitory feedback from C_1 ensures that S_2 is *not* enabled in the phase in which C_1 is firing. Thus S_2 can start firing only in a phase *other than* ρ. Once S_2 starts firing, S_3 gets enabled and so on.

Note that C_i could receive input in *two* phases — one from its bottom-up switch for C, and another from its top-down switch. C_i being a ρ-btu node, fires in only one of these phases. At any stage, if C_i, $1 \leq i \leq k_1$, picks up activation channeled by the *other* switch, feedback from C_i into the τ-or node associated with S_i causes S_{i+1} to become enabled even though S_i may not be firing. The net result is that as instantiations occur in the concept cluster, the ρ-btu nodes in the switch get enabled, in turn, from left to right in distinct phases.

********** Figure 23 goes about here **********

An *IS-A* relation of the form *is-a(A,B)* is represented as shown in Fig. 23 by: (i) connecting the $A_{i\uparrow}, i = 1, \ldots, k_1$ nodes to the bottom-up switch for *B*; (ii) connecting the $B_{i\downarrow}, i = 1, \ldots, k_1$ nodes to the top-down switch for *A*.

Consider a concept C in the *IS-A* hierarchy. Suppose C_i receives activation from the bottom-up switch in phase ρ. In response, C_i starts firing in synchrony with this activation. The $C_{i\uparrow}$ node now receives *two* inputs in phase ρ (one from the bottom-up switch and another from C_i; see Fig. 21). Since it has a threshold of 2, $C_{i\uparrow}$ also starts firing in phase ρ. This causes activation in phase ρ to eventually spread to the super-concept of C. Hence any upward traveling activation continues to travel upward – which is the required behavior when C is associated with a universal typed variable. Similarly, when C_i receives activation from the top-down switch in phase ρ, both C_i and $C_{i\downarrow}$ become active in phase ρ. $C_{i\uparrow}$ soon follows suit because of the link from $C_{i\downarrow}$ to $C_{i\uparrow}$. Thus eventually the whole i^{th} bank starts firing in phase ρ. This built-in mechanism allows a concept associated with an existential typed variable to eventually spread its activation to its ancestors, descendants and ancestors of descendants. The switching mechanism introduces an extra delay in the propagation of activation along *IS-A* links and typically, the switch takes three steps to channel the activation. In the worst — and also the least likely — case the switch may take up to eight steps to propagate activation.

The time taken to perform inferences in the integrated system is also independent of the size of the LTKB and is proportional only to the length of the shortest derivation of the query. The time taken to perform a predictive inference is approximately $l_1\pi + 3l_2\pi$, where l_1 and l_2 are the lengths of the shortest chain of rules, and the shortest chain of *IS-A* links, respectively, that must be traversed in order to perform the inference. The time required to answer a *yes-no* query is approximately $max(l_1\pi, 3l_2\pi) + l_1\pi + 2\pi$.[24]

5.3 Typed variables in queries

Consider a query $P(\ldots, x, \ldots)?$, where the j^{th} argument is specified as a typed variable x. If x is universally quantified, i.e., the query is of the form $\forall x : C\ P(\ldots, x, \ldots)$, C_i and $C_{i\uparrow}$ are activated in phase with the j^{th} argument node of P (the subscript i refers to one of the banks of C). If x is existentially quantified, i.e., the query is of the form $\exists x : C\ P(\ldots, x, \ldots)$, C_i and $C_{i\downarrow}$ are activated in phase with the j^{th} argument node of P. As before (Section 4.3) an untyped variable in a query is not activated. Simple queries of the type *IS-A*(C, D)? are posed by simply activating the nodes C_i and $C_{i\uparrow}$ and observing whether one or more D_i's become active.

5.4 Encoding appropriateness as type restrictions on argument fillers

The *IS-A* hierarchy can be used to impose type restrictions on variables occurring in rules. This allows the system to encode context dependent rules that are sensitive to the types of the argument-fillers involved in particular situations. Fig. 24 shown the encoding of the following rule in a *forward* reasoning system: $\forall x : animate, y : solid\text{-}obj \quad walk\text{-}into(x, y) \Rightarrow hurt(x)$ (i.e., 'If an *animate* agent walks into a *solid* object, the agent gets hurt'). The types associated with variables specify the admissible types (categories) of fillers, and the rule is expected to fire only if the fillers bound to the arguments are of the appropriate type. The encoding makes use of τ-or nodes that automatically check whether the filler of an argument is of the appropriate type. Thus the τ-or node a in Fig. 24 would become active if and only if the first argument of *walk-into* is firing in synchrony with *animate* indicating that the filler of the argument is of type *animate*. Similarly, the τ-or node b would become active if and only if the second argument of *walk-into* is firing in synchrony with *solid-object* indicating that the filler of this argument is of type *solid-object*. The activation of nodes a and b would enable the propagation of activity from the antecedent to the consequent predicate. In a forward reasoner, typed variables are allowed only in the antecedent of the rule.

********** Figure 24 goes about here **********

In the backward reasoner, typed variables are allowed only in the consequent of a rule. The encoding of a typed universally quantified variable in the consequent is similar to the encoding of an entity in the consequent of a rule explained in Section 4.5 (see Fig. 14). Instead of originating at an entity, the inhibitory link originates at the concept representing the type of the universally quantified variable. The encoding of a typed existentially quantified variable is similar to that of a typed universally quantified variable except that the inhibitory link originates from a unique subconcept of the associated concept. For details refer to (Mani & Shastri 1991).

The rule: $\forall$ *x:animate, y:solid-obj walk-into(x,y)* $\Rightarrow$ *hurt(x)* is logically equivalent to the rule: $\forall$ *x,y animate(x)* $\wedge$ *solid-obj(y)* $\wedge$ *walk-into(x,y)* $\Rightarrow$ *hurt(x)*. Thus it would appear that the *IS-A* hierarchy is not essential for encoding type restrictions on rules. Note however, that while the former variant has only *one* predicate in the antecedent, the latter has *three*. This increase in the number of antecedent predicates can be very costly, especially in a forward reasoning system capable of supporting multiple dynamic predicate instantiations (Mani & Shastri 1992). In such a system, the number of nodes required to encode a rule is proportional to k_2^m, where k_2 is the bound on the number of times a predicate may be instantiated dynamically during reasoning (see Section 6), and m equals the number of predicates in the antecedent of the rule. Thus it is critical that m be very small. The *IS-A* hierarchy plays a crucial role in reducing

the value of m by allowing restrictions on predicate arguments to be expressed as type restrictions.[25]

5.5 Encoding soft and defeasible rules

The proposed solution to the binding problem can be generalized to soft/defeasible rules. Observe that the strength of dynamic bindings may be represented by the degree of synchronization between an argument and a filler (this possibility was suggested by Jed Harris). Such a scheme becomes plausible if each argument is encoded by an ensemble of nodes (see Section 7.3). For then the degree of coherence in the phase of firing of nodes within an argument ensemble can indicate the strength of the binding the argument is participating in. In the limiting case, a highly dispersed activity in an argument ensemble may mean that the argument is bound to one of the active entities, although it is not clear which (Shastri 1992).[26]

In addition to specifying a mechanism for representing the strength of dynamic bindings and rule firing, we also need to specify the basis for computing these strengths. It is customary to view the strength of a rule as a numerical quantity associated with a rule (e.g., *certainty factors* in MYCIN (Buchanan & Shortliffe 1984)). Such an 'atomic' and uninterpreted view of the strength of a rule is inadequate for modeling rules involving *n*-ary predicates. Our approach involves defining the 'strength' of a rule (and similarly, the strength of a binding) to be a *dynamic* quantity that depends upon the types/features of the entities bound to arguments in the rule at the time of rule application. Such a strength measure also generalizes the notion of type *restrictions* on rules to type *preferences* on rules. Thus instead of rules of the form: $\forall x : animate,\ y : solid\text{-}obj\ walk\text{-}into(x, y) \Rightarrow hurt(x)$, the system can encode rules of the form:

$\forall x, y\ walk\text{-}into(x, y) \Rightarrow$ [with strength $\sigma(type(x), type(y))$] $\Rightarrow hurt(x)$

where the strength of the rule may vary from one situation to another as a function, σ, of the types of the argument-fillers in a given situation. Observe that the value of $\sigma(t_i, t_j)$ need not be known for *all* types t_i and t_j in the *IS-A* hierarchy, and may be inherited. For example, if $\sigma(t_1, t_2)$ is not known but $\sigma(t_m, t_n)$ is, and t_1 and t_2 are subtypes of t_m and t_n respectively, then $\sigma(t_m, t_n)$ can be used in place of $\sigma(t_1, t_2)$. This is analogous to property inheritance in an *IS-A* hierarchy, where property values may be attached to just a few concepts and the property values of the rest of the concepts inferred via inheritance. The proposed treatment would allow the system to incorporate exceptional and default information during reasoning. This relates to Shastri's (1988a,b) work on a connectionist semantic network (see Section 9.2).

6 Representing multiple dynamic instantiations of a predicate

The representation of dynamic bindings described thus far cannot simultaneously represent multiple dynamic facts about the *same* predicate. The proposed representation can be extended to do so by generalizing the scheme for encoding multiple instantiations of concepts outlined in Section 5.2. The extension assumes that during an episode of reflexive reasoning, each predicate can be instantiated only a bounded number of times. In general, this bound may vary across predicates and some critical predicates may have a marginally higher bound. For ease of exposition however, we will assume that this bound is the same for all predicates and refer to it as

k_2. The ability to handle multiple instantiations of the same predicate allows the system to deal with more complex inferential dependencies including circularities and *bounded* recursion. The system can make use of rules such as $\forall x, y\ sibling(x, y) \Rightarrow sibling(y, x)$. A forward reasoning system can use a rule such as $\forall x, y, z\ greater(x, y) \wedge greater(y, z) \Rightarrow greater(x, z)$ and infer 'a is greater than c' on being told 'a is greater than b' and 'b is greater than c'.

Since up to k_2 dynamic instantiations of a predicate may have to be represented simultaneously, the representation of an n-ary predicate is augmented so that each predicate is represented by k_2 *banks* of nodes, with each *bank* containing a *collector*, an *enabler*, and n argument nodes. For a given predicate P, the *enabler* of the i-th bank *e:*P_i will be active whenever the i-th bank has been instantiated with some dynamic binding. The *collector c:*P_i of the i-th bank will be activated whenever the dynamic bindings in the i-th bank are consistent with the knowledge encoded in the system. Fig. 25 depicts the encoding of two binary predicates P and Q and a ternary predicate R.

********** Figure 25 goes about here **********

Given that a predicate is represented using multiple banks of predicate and argument nodes, the connections between arguments of the antecedent and consequent predicates of a rule have to be mediated by a 'switching' mechanism similar to the one described in Section 5.2. The switch automatically channels input instantiations into available banks of its associated predicate. It also ensures that each distinct instantiation occupies only one bank irrespective of the number of consequent predicates that may be communicating this instantiation to the switch.

With the inclusion of the switch in the backward reasoning system, the number of nodes required to represent a predicate and a long-term fact becomes proportional to k_2 and the number of nodes required to encode a rule becomes proportional to k_2^2. Furthermore, the time required for propagating multiple instantiations of a predicate increases by a factor of k_2. Thus there is significant space and time costs associated with multiple instantiation of predicates. The complete realization of the switch and its interconnection is described in (Mani & Shastri 92).

7 Biological plausibility

In this section we cite recent neurophysiological data that suggests that synchronous and rhythmic activity occurs in the brain and the time course of such activity is consistent with the requirements of reflexive reasoning. The data also provides evidence in support of the hypothesis that the cat visual system solves the dynamic binding problem using temporal synchrony.

7.1 Neurophysiological support

There is considerable evidence for the existence of rhythmic activity in the animal brain. Synchronous activity has been documented for some time in the olfactory bulb, hippocampus, and the visual cortex (Freeman 1981; MacVicar & Dudek 1980; Gerstein 1970; Toyama, Kimura & Tanaka 1981). The most compelling evidence for such activity, however, comes from findings of synchronous oscillations in the visual cortex of anaesthetized cats responding to moving visual stimuli (Eckhorn, Bauer, Jordan, Brosch, Kruse, Munk & Reitboeck 1988; Eckhorn, Reitboeck,

Arndt & Dicke 1990; Engel, Koenig, Kreiter, Gray & Singer 1991; Gray & Singer 1989; Gray, Koenig, Engel & Singer 1989; Gray, Engel, Koenig & Singer 1991). These findings are based on the correlational analysis of local field potentials, multi-unit recordings, as well as single unit recordings. Recently, similar activity has also been reported in an awake and behaving monkey (Kreiter & Singer 1992).[27] Relevant aspects of the experimental findings are summarized below.

- Synchronous oscillations have been observed at frequencies ranging from 30 – 80 Hz. A typical frequency being around 50 Hz.
- Synchronization of neural activity can occur within a few (sometimes even one) periods of oscillations (Gray et al. 1991).
- In most cases synchronization occurs with a lag or lead of *less than* 3 msec although in some cases it even occurs with precise phase-locking (Gray et al. 1991).
- Synchronization of neural activity occurs i) between local cortical cells (Eckhorn et al. 1988, Gray & Singer 1989), ii) among distant cells in the same cortical area (Gray et al. 1989), iii) among cells in two different cortical areas — for example, areas 17 and 18 (Eckhorn et al. 1988) and areas 17 and PMLS (Engel et al. 1991), and iv) among cells across the two hemispheres (Engel et al. 1991).
- Once achieved, synchrony may last several hundred msec (Gray et al. 1991).

The synchronous activity observed in the brain is a complex and dynamic phenomenon. The frequency and degree of phase-locking varies considerably over time and the synchronization is most robust when viewed as a property of groups of neurons. The nature of synchronous activity assumed by our model is an idealization of such a complex phenomenon (but see Sections 7.3 and 10.1–10.4).

Temporal synchrony and dynamic bindings in the cat visual cortex

Neurophysiological findings also support the hypothesis that the dynamic binding of visual features pertaining to a single object may be realized by the synchronous activity of cells encoding these features (see Eckhorn et al. 1990; Engel et al. 1991). In one experiment, multi-unit responses were recorded from four different sites that had overlapping receptive fields but different orientation preferences — 157^o, 67^o, 22^o, and 90^o, respectively. A vertical light bar resulted in a synchronized response at sites 1 and 3 while a light bar oriented at 67^o lead to a synchronized response at sites 2 and 4. A combined stimulus with the two light bars superimposed lead to activity at all the four sites, but while the activity at sites 1 and 3 was synchronized and that at sites 2 and 4 was synchronized, there was no correlation in the activity across these pair of sites (Engel et al. 1991). Experimental evidence such as the above suggests that the synchronous activity in orientation specific cells may be the brain's way of encoding that these cells are participating in the representation of a *single* object. This is analogous to the situation in Fig. 9 wherein the synchronous activity of the nodes *recip, owner* and *cs-seller* in phase with *Mary* is the system's way of encoding that all these roles are being filled by the *same object*, Mary.

7.2 Some neurally plausible values of system parameters

The neurophysiological data cited above also provides a basis for making coarse but neurally plausible estimates of some system parameters. The data indicates that plausible estimates of π_{min} and π_{max} may be 12 and 33 msec, respectively, and a typical value of π may be 20 msec. The degree of synchronization varies from episode to episode but a conservative estimate of ω, the width of the window of synchrony, may be derived on the basis of the cumulative histogram of the phase difference (lead or lag) observed in a large number of synchronous events. The standard deviation of the phase differences was 2.6 msec in one data set and 3 msec in another (Gray et al. 1991). Thus a plausible estimate of ω may be about 6 msec. Given that the activity of nodes can get synchronized within a few cycles — sometimes even within one, and given the estimates of π_{min} and π_{max}, it is plausible that synchronous activity can propagate from one ρ-btu node to another in about 50 msec. The data also suggests that synchronous activity lasts long enough to support episodes of reflexive reasoning requiring several steps.

7.3 Propagation of synchronous activity — a provisional model

Our system requires the propagation of synchronous activity over interconnected nodes in spite of non-zero and noisy propagation delays. The neurophysiological evidence cited in the previous sections suggests that such propagation occurs in the cortex. The exact neural mechanisms underlying the propagation of such activity, however, remain to be determined. On the basis of anatomical and physiological data (1982), theoretical analysis, and simulation results (1991), Abeles has argued that synchronous activity can propagate over chains of neurons connected in a many-to-many fashion (synfire chains) with a small and stable 'jitter' even if random fluctuations are taken into account. Bienenstock (1991) has examined how synfire chains may arise in networks as a result of learning. Below we outline a provisional model (Mandelbaum & Shastri 1990) which demonstrates that synchronized activity can propagate in spite of noisy propagation delays. The model is meant to demonstrate the feasibility of such propagation and should not be viewed as a detailed neural model.

We assume that each argument in the reasoning system is represented by an ensemble of n nodes rather than just a single node. Connections between arguments are encoded by connecting nodes in the appropriate ensembles: if ensemble A connects to ensemble B, then each node in A is randomly connected to m nodes in B ($m \leq n$). Thus on an average, each node in B receives inputs from m nodes in A (see Fig. 26) and has a threshold comparable to m. The propagation delay between nodes in two different ensembles is assumed to be noisy and is modeled as a Gaussian distribution. If ensemble A is connected to ensemble B and nodes in ensemble A are firing in synchrony, then we desire that within a few periods of oscillation nodes in ensemble B start firing in synchrony with nodes in ensemble A.

********** Figure 26 goes about here **********

Nodes within an ensemble are also sparsely interconnected, with each node receiving inputs from a few neighboring nodes within the ensemble. Synchronization within an ensemble is realized as a result of the interaction between the feedback received by a node from its neighbors *within* the ensemble. The model makes use of the negative slope of the threshold-time characteristic during the *relative refractory period* (RRP) to modulate the timing of the spike generated

by a node. Observe that a higher excitation can hasten, while a lower excitation can delay, the firing of a node. At the same time, the lag between the firing times of nodes in two connected ensembles due to the mean propagation delay is partially offset by the interaction between the various parameters of the system enumerated below.

********** Figure 27 goes about here **********

The threshold-time characteristic of a node and the distribution of the arrival times of input spikes from a preceding ensemble are illustrated in Fig. 27. After firing, a node enters an absolute refractory period (ARP) in which its threshold is essentially infinite. The ARP is followed by a decaying RRP during which the threshold decays to its resting value. During the RRP the threshold-time characteristic is approximated as a straight line of gradient *g* (a linear approximation is not critical). The incoming spikes from a preceding ensemble arrive during a node's ARP and the early part of its RRP. It is assumed that immediate neighbors within an ensemble can rapidly communicate their potential to each other.

A node's potential is the combined result of the inter-ensemble and intra-ensemble interactions and in the period between spikes is modeled as: $P_i(t + \Delta t) = P_i(t) + In_i(t) + \alpha * \sum_j [P_i(t) - P_j(t)]$
$P_i(t)$ is the potential of node i at time t. The change in potential is caused by $In_i(t)$, the input arriving at node i from nodes in the preceding ensemble as well as the difference in the potential of i and that of its immediate neighbors j. In the simulation, j ranged over *six* immediate neighbors of i. If nodes i and j are immediate neighbors and i is firing ahead of j, then we want i to hasten the firing of j by sending it an excitatory signal and j to delay the firing i by sending it an inhibitory signal. Doing so would raise the potential of j, causing it to fire early, and lower the potential of i, causing it to fire later in the *next* cycle. Thus i and j will tend to get synchronized.[28]

********** Figure 28 goes about here **********

The results of a sample simulation are shown in Fig. 28. The diagram shows the cycle-by-cycle distribution of the firing times of nodes within a 'driven' ensemble that is being driven by a 'driver' ensemble whose nodes are oscillating in phase-lock. Δt was chosen to be a 0.001 time units (i.e., all calculations were done every $1/1000$ of a time unit), where a unit of time may be assumed to be 1 msec. Other simulation parameters were as follows: i) n, the number of nodes in ensemble equals 64, ii) m, the inter-ensemble connectivity equals 20, iii) g, the slope of the threshold during the RRP equals 0.032, iv) α, the 'coupling' factor between immediate neighbors within an ensemble equals 0.07, v) d, the average inter-ensemble propagation delay equals 4.5 time units, vi) s, the standard deviation of inter-ensemble propagation delay equals 1.5 time units, and vii) π, the expected period of oscillation equals 10.5 time units.

As shown in Fig. 28, despite noisy propagation delays, the *maximum* lag in the firing of nodes in the 'driven' ensemble becomes less than 3 msec and the *mean* lag becomes less than 1 msec within 2 cycles. By the end of 7 cycles the maximum and mean lags reduce to 1 and 0.2 msec, respectively.

8 Psychological Implications

In this section we examine the psychological implications of our system, specially in view of the biologically motivated estimates of the system parameters discussed in Section 7.3.

8.1 A neurally plausible model of reflexive reasoning

The proposed system can encode specific as well as general instantiation independent knowledge and perform a broad range of reasoning with efficiency. The system makes use of very simple nodes and yet its node requirement is only linear in the size of the LTKB (the size being measured in terms of the number of predicates, facts, rules, concepts, and *IS-A* relations). Thus the system illustrates how a large LTKB may be encoded using only a fraction of 10^{12} nodes.

The system demonstrates that a class of forward and backward reasoning can be performed very rapidly, in time independent of the size of the LTKB. Below we set the values of appropriate system parameters to neurally plausible values identified in Section 7.3 and indicate the time the system takes to perform certain inferences.[29] These times are at best, broad indicators of the time we expect the internal reasoning process to take. Also note that they do not include the time that would be taken by perceptual, linguistic, and motor processes to process and respond to inputs.

Some typical retrieval and inference timings

Let us choose π to be 20 msec and assume that ρ-btu nodes can synchronize within two periods of oscillations. The system takes 320 msec to infer 'John is jealous of Tom' after being given the dynamic facts 'John loves Susan' and 'Susan loves Tom' (this assumes the rule 'if x loves y and y loves z then x is jealous of z). The system takes 260 msec to infer 'John is a sibling of Jack' given 'Jack is a sibling of John'. Similarly, the system takes 320 msecs to infer 'Susan *owns* a car' after its internal state is initialized to represent 'Susan *bought* a Rolls-Royce'. If the system's LTKB includes the long-term fact 'John bought a Rolls-Royce', the system takes 140 msec, 420 msec, and 740 msec, respectively, to answer 'yes' to the queries 'Did John buy a Rolls-Royce', 'Does John own a car?' and 'Can John sell a car?'

Thus our system demonstrates that a class of reasoning can occur rapidly, both in the forward (predictive) mode as well as backward (query answering) mode. The above times are independent of the size of the LTKB and do not increase when additional rules, facts, and *IS-A* relationships are added to the LTKB. If anything, these times may *decrease* if one of the additional rules is a composite rule and short-circuits an existing inferential path. For example, if a new rule 'if x buys y then x can sell y' were to be added to the LTKB, the system would answer the query 'Can John sell a car?' in 420 msec instead of 740 msec.

Variations in inference and retrieval times

Consider two ρ-btu nodes A and B such that A is connected to B (although we are referring to individual nodes, the following comment would also apply if A and B were ensembles of nodes). It seems reasonable to assume that the number of cycles required for B to synchronize

with A will depend on the synaptic efficacy of the link from A to B. This suggests that the time taken by the propagation of bindings — and hence, rule firing — will vary depending on the weights on the links between the argument nodes of the antecedent and consequent predicates. Rules whose associated links have high weights will fire and propagate bindings faster than rules whose associated links have lower weights. It also follows that different facts will take different times to be retrieved depending on the weights of the links connecting the appropriate arguments and filler concepts (refer to Fig. 10). Note that the inhibitory signal from an argument will continue to block the activation of a fact node until the signals from the filler concepts and the argument get synchronized. Similarly, during the processing of *wh*-queries, the time it would take for the filler concepts to synchronize with the binder units will depend on the weights of the links from the binder nodes to the concept nodes (refer to Fig. 17). Thus the retrieval of argument fillers will be faster if the weights on the appropriate links are high.[30] Observe that the variation in times refers to the variations in *the time it takes for nodes to synchronize* and not the time it takes for nodes to become active. This suggests that the time course of systematic inferences and associative priming may be quite different.

8.2 Nature of reflexive reasoning

Our model suggests several constraints on the nature of reflexive reasoning. These have to do with i) the capacity of the working memory underlying reflexive reasoning, ii) the form of rules that may participate in such reasoning, and iii) the depth of the chain of reasoning.

The working memory underlying reflexive reasoning

Dynamic bindings, and hence, dynamic facts are represented in the system as a rhythmic pattern of activity over nodes in the LTKB network. In *functional terms*, this transient state of activation can be viewed as a limited capacity dynamic *working memory* that temporarily holds information during an episode of reflexive reasoning. Let us refer to this working memory as the WMRR.

Our system predicts that the capacity of the WMRR is very large, and at the same time, it is very limited! The number of dynamic facts that can simultaneously be present in the WMRR is very high and is given by $k_2 p$, where k_2 is the predicate multiple instantiation constant introduced in Section 6, and p is the number of predicates represented in the system. The number of concepts that may be active simultaneously is also very high and equals $k_1 c$, where c is the number of concepts in the *IS-A* hierarchy and k_1 is the multiple instantiation constant for concepts introduced in Section 5.2. But as we discuss below there are two constraints that limit the number of dynamic facts that may *actually* be present in the WMRR at any given time.

Working memory, medium-term memory, and *overt* short-term memory

Before moving on let us make two observations. First, the dynamic facts represented in the WMRR during an episode of reasoning should not be confused with the small number of short-term facts that an agent may *overtly* keep track of during *reflective* processing and problem solving. In particular, the WMRR should not be confused with the (overt) short-term memory implicated in various memory span tasks (for a review see Baddeley 1986). Second, our reasoning

system implies that a large number of dynamic facts will be produced as intermediate results during reasoning and would be represented in the WMRR. These facts, however, are only *potentially* relevant and would remain *covert* and decay in a short time unless they turn out to be relevant in answering a 'query' or providing an explanation. We expect that only a small number of dynamic facts would turn out to be relevant, and any that do, would 'enter' a medium-term memory where they would be available for a much longer time (see Section 10.5). Some of these facts may also enter the overt short-term memory. Note that this short-term memory need not be a physically distinct 'module'. It may simply consist of facts/entities in the WMRR that are currently receiving an attentional spotlight (cf: Crick 1984; Crick & Koch 1990).

A bound on the number of distinct entities referenced in the WMRR

During an episode of reasoning, each entity involved in dynamic bindings occupies a distinct phase in the rhythmic pattern of activity. Hence the number of distinct entities that can occur as argument-fillers in the dynamic facts represented in the WMRR cannot exceed $\lfloor \pi_{max}/\omega \rfloor$, where π_{max} is the maximum period (corresponding to the lowest frequency) at which ρ-btu nodes can sustain synchronous oscillations and ω equals the width of the window of synchrony. Thus the WMRR may represent a large number of facts, as long as these facts refer to only a small number of distinct entities. Note that the activation of an entity together with all its active super-concepts counts as only *one* entity.

In Section 7.2 we pointed out that a neurally plausible value of π_{max} is about 33 msec and a conservative estimate of ω is around 6 msec. This suggests that as long as the number of entities referenced by the dynamic facts in the WMRR is five or less, there will essentially be no cross-talk among the dynamic facts. If more entities occur as argument-fillers in dynamic facts, the window of synchrony ω would have to shrink in order to accommodate all the entities. For example, ω would have to shrink to about 5 msec in order to accommodate 7 entities. As ω shrinks, the possibility of cross-talk between dynamic bindings would increase and eventually disrupt the reasoning process. The exact bound on the number of distinct entities that may fill arguments in dynamic facts would depend on the smallest feasible value of ω. Given the noise and variation indicated by the data on synchronous activity cited in Section 7.1, it appears unlikely that ω can be less than 3 msec. Hence we predict that a neurally plausible *upper bound* on the number of distinct entities that can be referenced by the dynamic facts represented in the WMRR is about 10. This prediction is consistent with our belief that most cognitive tasks performed without deliberate thought tend to involve only a small number of distinct entities at a time (though of course, these entities may occur in multiple situations and relationships).

It is remarkable that the bound on the number of entities that may be referenced by the dynamic facts in the WMRR relates so well to 7 ± 2, the robust measure of short-term memory capacity (Miller 1956). This unexpected coincidence merits further investigation as it suggests that temporal synchrony may also underlie other short-term and dynamic representations. Similar limitations of the human dynamic binding mechanism are also illustrated in experimental work on the attribute binding problem (Stenning, Shepard & Levy 1988).

The bound on the number of distinct entities referenced in the WMRR is independent of similar bounds on the working memories of other subsystems. As we discuss in Section 10.4,

dynamic structures in the working memory of other subsystems may refer to different sets of entities using phase distributions local to those subsystems.

A bound on the multiple instantiation of predicates

The capacity of the WMRR is also limited by the constraint that it may only contain a small number of dynamic facts pertaining to each predicate. This constraint stems from the high cost of maintaining multiple instantiations of a predicate. As stated in Section 6, in a backward reasoning system, if k_2 denotes the bound on the number of times a predicate may be instantiated during an episode of reasoning, then the number of nodes required to represent a predicate and the associated long-term facts is proportional to k_2, and the number of nodes required to encode a rule is proportional to k_2^2. Thus a backward reasoning system that can represent three dynamic instantiations of each predicate will have anywhere from three to nine times as many nodes as a system that can only represent one instantiation per predicate. In a forward reasoning system the cost is even higher and the number of nodes required to encode a rule is k_2^m, where m is the number of antecedents in the rule. The time required for propagating multiple instantiations of a predicate also increases by a factor of k_2. In view of the significant space and time costs associated with multiple instantiation and the necessity of keeping these resources within bounds in the context of reflexive reasoning, we predict that k_2 is quite small, perhaps no more than 3. As observed in Section 6, k_2 need not be the same for all predicates and it is possible that some critical predicates may have a slightly higher k_2.[31]

Form of rules that may participate in reflexive reasoning

In Section 4.9 we pointed out that when answering queries based on the long-term knowledge encoded in the LTKB, our reflexive reasoning system cannot use rules that contain variables occurring in multiple argument positions in the *antecedent* unless such variables also appear in the consequent and get bound during the query answering process. A similar constraint applies to forward (predictive) reasoning: When making predictions based on given dynamic facts, a system cannot use a rule that contains variables occurring in multiple argument positions in the *consequent*, unless such variables also appear in the antecedent and get bound during the reasoning process. These constraints predict that certain queries cannot be answered in a reflexive manner even though the corresponding predictions can be made reflexively. For example, consider an agent whose LTKB includes the rule 'if x loves y and y loves z then x is jealous of z', and the long-term facts 'John loves Mary' and 'Mary loves Tom'. Our system predicts that if this agent is asked 'Is John jealous of Tom?', she will be unable to answer the query in a *reflexive* manner. Note that the antecedent of the rule includes a repeated variable, y, that does not occur in the consequent. Hence our system predicts that answering this question will require deliberate and conscious processing (unless the relevant long-term facts are active in the WMRR for some reason at the time the query is posed). However, an agent who has the above rule about love and jealousy in its LTKB would be able infer 'John is jealous of Tom' in a reflexive manner, on being 'told' 'John loves Mary' and 'Mary loves Tom'. This because such an inference involves forward (predictive) reasoning.

As another example of the predictions made by the constraint, assume that our agent's conception of kinship relations is one wherein the maternal/paternal distinction at the grandparent level is not primary. Let us also assume that the agent's maternal-grandfather is George. The constraint predicts that the agent cannot answer 'yes' to the query 'Is George your maternal grandfather?' in a reflexive manner even though he/she may be able to answer the question 'Is George your grandfather?' in a reflexive manner (this example is due to Jerry Feldman). The basis of this prediction is as follows: If 'maternal-grandfather' is not a primary kinship relation then it must be computed using an appropriate rule. Given the nature of the maternal-grandfather relationship, any rule that does so would violate the repeated variable restriction.[32]

The restrictions imposed on the reasoning system also imply that it is not possible to apply the *abstract* notion of transitivity in a reflexive manner when answering queries. Observe that we need to state $\forall x, y, z P(x, y) \wedge P(y, z) \Rightarrow P(x, z)$ in order to assert that the relation P is transitive and the rule has the variable y occurring twice in the antecedent but not even once in the consequent. Given that transitivity plays an important role in common sense reasoning — to wit, reasoning about sub and super-categories, part-of relationships, greater and less than — the inability to handle transitivity might appear to be overly limiting. However, this is not the case. We believe that as far as query answering is concerned, humans are only good at dealing with the transitivity of a *small number of relations*. In these cases, the transitivity of the appropriate relations is encoded *explicitly* and the computation of transitivity does not require the use of an abstract transitivity rule. The organization of concepts in an *IS-A* hierarchy using *IS-A* links to capture the sub-class/super-class relationship is an excellent case in point. The use of *IS-A* links converts the problem of computing the transitive closure from one of applying the transitivity rule $\forall$ *x,y,z IS-A(x,y)* $\wedge$ *IS-A(y,z)* $\Rightarrow$ *IS-A(x,z)*, to one of propagating activation along links.

Bound on the depth of the chain of reasoning

Two things might happen as activity propagates along a chain of argument ensembles during an episode of reflexive reasoning. First, the lag in the firing times of successive ensembles may gradually build up due to the propagation delay introduced at each level in the chain. Second, the dispersion within each ensemble may gradually increase due to the variations in propagation delays and the noise inherent in synaptic and neuronal processes. While the increased lag along successive ensembles will lead to a 'phase shift' and hence, binding confusions, the increased dispersion of activity within successive ensembles will lead to a gradual *loss of binding information*. Increased dispersion would mean less phase specificity, and hence, more *uncertainty* about the argument's filler. Due to the increase in dispersion along the chain of reasoning, the propagation of activity will correspond less and less to a propagation of argument bindings and more and more to an associative spread of activation. For example, the propagation of activity along the chain of rules: $P_1(x, y, z) \Rightarrow P_2(x, y, z) \Rightarrow \ldots P_n(x, y, z)$ resulting from the input $P_1(a, b, c)$ may lead to a state of activation where all one can say about P_n is that there is an instance of P_n involving the entities $a, b,$ and c, but it is not clear which entity fills which role of P_n.

In view of the above, it follows that the depth to which an agent may reason during reflexive reasoning is bounded. Thus an agent may be unable to make a prediction (or answer a query)

— even when the prediction (or answer) logically follows from the knowledge encoded in the LTKB — if the length of the derivation leading to the prediction (or the answer) exceeds this bound. It should be possible to relate the bound on the depth of reflexive reasoning to specific physiological parameters, but at this time we are not aware of the relevant data upon which to base such a prediction. We would welcome pointers to appropriate data.

8.3 Nature of inputs to the reflexive reasoner

Our system demonstrates that rule-like knowledge may be used effectively during reflexive reasoning provided it is integrated into the LTKB and wired into the inferential dependency graph. It also demonstrates that reflexive reasoning can effectively deal with small dynamic input in the form of facts.[33] We suspect that the ability of any reflexive reasoning system to deal with novel rule-like information will be extremely limited; if the input contains rule-like information that is not already present in the LTKB, the agent may have to revert to a reflective mode of reasoning in order to use this information. This may partially explain why human agents find it difficult to perform syllogistic reasoning without deliberate and conscious effort even though, in a *formal sense*, such reasoning is simpler than some of the reasoning tasks we can perform in a reflexive manner. In syllogistic reasoning, the 'input' has the form of rules and the reflexive reasoner may be unable to use them unless they are already part of the LTKB.

8.4 The reflexive reasoning system and production systems

As may be evident, there exists a correspondence between a production system and the reflexive reasoning system described in this paper — the declarative memory corresponds to long-term facts, productions correspond to rules, and the working memory corresponds to the WMRR. Thus our system can be viewed as a parallel production system.

Estimates of the working memory capacity of production system models range from very small (about seven elements) to essentially unlimited. Our work points out that the working memory of a reflexive processor can contain a very large number of *elements* (dynamic facts in the case of the reasoning system) as long as i) the elements do not refer to more than (about) 10 *entities* and ii) the elements do not involve the same relation (predicate) more than (about) 3 times. The proposed system also demonstrates that a large number of rules — even those containing variables — may fire in parallel as long as any predicate is not instantiated more than (about) 3 times (cf: Newell's suggestion (1980) that while productions without variables can be executed in parallel, productions with variables may have to be executed in a serial fashion).

A number of cognitive models are based on the production system formalism; two of the most comprehensive being ACT* (Anderson 1983) and SOAR (Newell 1990). Neurally plausible realizations of these models, however, have not been proposed. Although several aspects of ACT* such as its use of levels of activation and weighted links have neural underpinnings, it has not been shown how certain critical aspects of the model may be realized in a neurally plausible manner. For example, ACT* represents productions with variables, but Anderson does not provide a neurally plausible explanation of how bindings are propagated and how nodes determine whether two bindings are the same. In his exposition of SOAR, Newell had analyzed

the time course of neural processes to estimate how long various SOAR operations should take, but he had not suggested how a system such as SOAR may be realized in a neurally plausible manner (see p. 440 Newell, 1990). Although a complete mapping of comprehensive systems such as SOAR and ACT* to a neurally plausible architecture still remains an open problem, our system could provide a basis for doing so. In this context, the biologically motivated constraints on the capacity of the WMRR indicated by our system seem particularly significant.

8.5 Reflexive reasoning and text understanding

Several problems will have to be addressed in order to integrate the proposed reasoning system with a comprehensive cognitive system. Some of these problems are discussed in Section 10 and include i) interactions between the reflexive reasoning system and the medium-term memory, ii) how medium-term memory is mapped into long-term memory, iii) how the set of entities in the WMRR changes in a fluid manner, and iv) how distinct modules performing different reflexive processes (e.g., a parser and a reasoner) communicate with one another.

The problem of text understanding is particularly relevant because there exists a rich body of empirical data on the role of inferences based on long-term knowledge during language understanding. The data strongly suggests that certain types of inferences, for example, inferences that help establish referential and causal coherence, do occur very rapidly and automatically during text understanding (see e.g., Kintsch 1974; Carpenter & Just 1977; McKoon & Ratcliff 1980; McKoon & Ratcliff 1981; Keenan, Baillet, and Brown 1984). The evidence for the automatic occurrence of *elaborative* inferences however, is mixed (see e.g., Singer & Ferreira 1983; McKoon & Ratcliff 1986; Kintsch 1988; Potts, Keenan, and Golding 1988). Elaborative inferences predict highly likely consequences of events mentioned in the discourse and correspond to forward reasoning in our system. However, as Potts et al. (1988) point out, available experimental evidence does not rule out the possibility that elaborative inferences are performed during reading. The experiments involve two sentence texts and it is likely that the subjects do not have any inherent interest in making predictive inferences. It may turn out that subjects do make such inferences when reading longer texts.

Our system suggests that reflexive reasoning *can* occur in backward as well as forward direction (although as pointed out in Section 8.2 there are critical differences in the form of rules that may participate in the two types of reasoning). This suggests that agents may perform inferences required for establishing referential and causal coherence as well as predictive inferences in a reflexive manner. The system's prediction can be resolved with the observed data if we assume that the results of predictive inferences only last for a short time (say a few hundred msec) and then disperse *unless* subsequent input (text) indicates that these inferences are *significant* and/or *relevant* to the discourse. Only those inferred facts that turn out to be relevant get encoded in the medium-term memory and become available for a longer time.

The extensive body of empirical data on the role of long-term knowledge and inferences in reading will inform future work on our model of reflexive reasoning. At the same time, we hope that the constraints on the form of rules and the capacity of the working memory underlying reflexive reasoning that have emerged from our work will help experimental psychologists in formulating and testing novel hypotheses about the role of reflexive reasoning in reading.

8.6 Reflexive reasoning and the fan effect

Our initial hypothesis as well as our system's behavior suggests that the time taken by reflexive reasoning is independent of the size of the LTKB. This conflicts with the fan effect (Anderson 1983; Reder & Ross 1983). In broad terms, this effect refers to the following phenomenon: The more facts associated with a particular concept, the slower the recognition of any one of the facts. We hypothesize that this effect applies only to medium-term knowledge and not to long-term knowledge (where we use 'long-term' in the sense discussed in Section 1.1). Consider the nature of the task that leads to the fan effect. An agent studies a set of facts until he can recall them. Subsequently, the agent is asked to recognize and make consistency judgments about the learned material and his reaction times are recorded. It is observed that the time taken to recognize a fact increases with the number of facts studied by the agent involving the same concept(s). Observe however, that the fan effect concerns an *arbitrary collection of facts that the agent studied prior to the experiment*. We hypothesize that these facts only get encoded in the agent's medium-term memory and do not get assimilated into the agent's LTKB. Thus the fan effect is *not* about facts in the LTKB, rather it is about facts in the medium-term memory.

9 Related work

In spite of the apparent significance of reflexive reasoning there have been very few attempts at modeling such rapid inference with reference to a large body of knowledge. Some past exceptions are Fahlman's work on NETL (1979) and Shastri's work on a connectionist semantic memory (1988a) (also see Geller & Du, 1991). Both these models primarily deal only with inheritance and classification within an *IS-A* hierarchy. Hölldobler (1990) and Ullman and van Gelder (1988) have proposed parallel systems for performing more powerful logical inferences, but these systems have unrealistic space requirements. The number of nodes in Hölldobler's system is quadratic in the the size of the LTKB and the number of processors required by Ullman and van Gelder is even higher.[34] A significant amount of work has been done by researchers in knowledge representation and reasoning to identify classes of inference that can be performed efficiently (e.g., see Frisch & Allen 1982; Levesque & Brachman 1985; Levesque 1988; McAllester 1990; Bylander, Allemang, Tanner, & Josephson 1991; Kautz & Selman 1991). The results however, have largely been negative. The few positive results reported do not provide insights into the problem of reflexive reasoning because they assume a weak notion of efficiency (polynomial time), restrict inference in implausible ways (e.g., by excluding chaining of rules), and/or deal with overly limited expressiveness (e.g., only propositional calculus).

9.1 Relation between NETL and the proposed system

It was pointed out in Section 3 that as an abstract computational mechanism, temporal synchrony is related to the notion of marker passing. It was also mentioned that Fahlman had proposed the design of a parallel marker passing machine (NETL) that could solve a class of inheritance and recognition problems efficiently. But as discussed in Section 3, NETL was not neurally plausible. In view of the correspondence between temporal synchrony and marker passing, our

system offers a neurally plausible realization of marker passing. It is important to underscore the significance of this realization. First, nothing is stored at a node in order to mark it with a marker. Instead, the *time* of firing of a node *relative* to other nodes and the *coincidence* between the time of firing of a node and that of other nodes has the *effect* of marking a node with a particular marker. Furthermore, a node does not have to match anything akin to markers. It simply has to detect whether appropriate inputs are coincident. Second, the system does not require a central controller. Once a query is posed to the system by activating appropriate nodes, it computes the solution without an external controller directing the activity of every node at every step of processing. The system's ability to do so stems from the distributed control mechanisms that are an integral part of the representation. Some examples of such *built-in* mechanisms that automatically control the propagation of activation are the $C_{\uparrow}$, and $C_{\downarrow}$ *relay* nodes in concept clusters (Section 5.2), and the *switch* networks associated with concepts and predicates that automatically direct the flow of activation to unused banks (Sections 5.2 and 6). Third, our realization of marker passing quantifies the capacity of the working memory underlying reflexive processing in terms of biological parameters. As we have seen, these constraints have psychological significance.

In addition to demonstrating that a marker passing system can be realized in a neurally plausible manner, our system also shows that a richer class of representation and reasoning problems can be performed using temporal synchrony — and hence, marker passing — than what was realized in NETL. If we set aside the issue of exceptional knowledge (see below), NETL represented an *IS-A* hierarchy and *n*-ary *facts*, where terms in a fact could be types or instances in the *IS-A* hierarchy. NETL however, did not represent *rules* involving *n*-ary predicates. NETL derived *inherited* facts by replacing terms in a fact by their subtypes or instances (this characterization accounts for NETL's ability to perform simple (unary) inheritance as well as relational inheritance), but it did not combine inheritance with rule-based reasoning. Consider the example of relational inheritance where "preys-on(Sylvester,Tweety)" is derived from "preys-on(Cat,Bird)". Observe that this only involves substituting Sylvester for Cat and Tweety for Bird based on the *IS-A* relations "IS-A(Sylvester, Cat)" and "IS-A(Tweety,Bird)". This form of reasoning is weaker than that performed by our system. Our reasoning system can also encode rules such as $\forall x, y\ preys\text{-}on(x, y) \Rightarrow scared\text{-}of(y, x)$ and given "preys-on(Cat,Bird)", it can not only infer "preys-on(Sylvester,Tweety)" but also "scared-of(Tweety,Sylvester)".

The presence of a central controller allowed NETL to compute and enumerate results of queries involving an arbitrary sequence of set intersection and set union operations. NETL's central controller could decompose a query into the required sequence of intersection and union operations and instruct NETL to perform these operations in the proper sequence. This is something our reflexive reasoning system does not (and is not intended to) do.

NETL also allowed exceptions in the *IS-A* hierarchy, but its treatment of exceptions suffered from serious semantic problems (see Fahlman, Touretzky & von Roggen 1981; and Touretzky 1986). In Sections 5.4 and 5.5 we described how rules with type restrictions are encoded in our system and explained how this encoding may be extended to deal with type preferences so that the *appropriateness* — or strength — of a rule firing in a specific situation may depend on the types of the entities involved in that situation. The ability to encode evidential rules will allow our system to incorporate exceptional and default information in an *IS-A* hierarchy (see below).

9.2 CSN — A connectionist semantic memory

Shastri (1988a,b) developed CSN, a connectionist semantic network that could solve a class of inheritance and classification problems in time proportional to the *depth* of the conceptual hierarchy. CSN computed its solutions in accordance with an evidential formalization and dealt with exceptional and conflicting information in a principled manner. It found the *most likely* answers to inheritance and recognition queries by combining the information encoded in the semantic network. CSN operated without a central network controller that regulated the activity of its nodes at each step of processing. This was the result of using distributed mechanisms (e.g., relay nodes) for controlling the flow of activity. A complete integration of a CSN-like system and the proposed reasoning system should lead to a system capable of dealing with evidential and conflicting rules and facts in a principled manner.

9.3 Some connectionist approaches to the dynamic binding problem

Feldman (1982) addressed the problem of dynamically associating any element of a group of N entities with any element of another group of N entities using an interconnection network. He showed how it was possible to achieve the association task with an interconnection network having only $4N^{3/2}$ nodes. The work however did not address how such a representation could be incorporated within a reasoning system where bindings need to be propagated.

Touretzky and Hinton (1988) developed DCPS, a distributed connectionist production system, to address the problem of rule-based reasoning within a connectionist framework. The ability of DCPS to maintain and propagate dynamic bindings is, however, quite limited. First, DCPS can only deal with rules that have a *single* variable. Second, DCPS is serial at the knowledge level because each step in its reasoning process involves selecting and applying a *single* rule. Thus in terms of efficiency, DCPS is similar to a traditional (serial) production system and must deal with the combinatorics of search. Third, it assumes that there is only one candidate rule that can fire at each step of processing. Hence it is not a viable model of reflexive reasoning.

Smolensky (1990) describes a representation of dynamic bindings using tensor products. Arguments and fillers are viewed as n and m dimensional vectors, respectively, and a binding is viewed as the $n * m$ dimensional vector obtained by taking the tensor product of the appropriate argument and filler vectors. The system encodes arguments and fillers as patterns over pools of n argument and m filler nodes and argument bindings over a network of $n * m$ nodes. The system can only encode $n * m$ bindings without cross-talk, although a greater number of bindings can be stored if some cross-talk is acceptable. Dolan and Smolensky (1989) describe TPPS, a production system based on the tensor product encoding of dynamic bindings. However, like DCPS, TPPS is also serial at the knowledge level and allows only one rule to fire at a time.

The primary cause of knowledge level serialism in DCPS and TPPS is that these systems represent arguments and fillers as patterns of activity over *common* pools of nodes. This severely limits the number of arguments, fillers, and dynamic bindings that may be represented at the same time. In contrast, the compact encoding of predicates, arguments, and concepts in our system allows it to represent and propagate a large number of dynamic bindings simultaneously.

Another system that uses a compact encoding and supports knowledge level parallelism is ROBIN (Lange & Dyer 1989). This system was designed to address the problem of natural

language understanding — in particular, the problem of ambiguity resolution using evidential knowledge. ROBIN and our system have several features in common, for example, ROBIN can also maintain a large number of dynamic bindings and encode 'rules' having multiple variables. There are also important differences in the two systems. ROBIN permanently allocates a unique numerical *signature* to each constant in the domain and represents dynamic bindings by propagating the signature of the appropriate constant to the argument(s) to which it is bound. The use of signatures allows ROBIN to deal with a large number of entities during an episode of reasoning. There is, however, a potential problem with the use of signatures: if each entity has a unique signature, then signatures can end up being high precision quantities. For example, assigning a distinct signature to 50,000 concepts will require a precision of 16 bits. Hence propagating bindings would require nodes to propagate and compare high precision analog values. This problem may be circumvented by representing signatures as n-bit vectors and encoding arguments as clusters of n nodes communicating via bundles of links (see Section 9.4).

The temporal synchrony approach can be compared to the signature based approach as follows: Although the total number of entities is very large, the number of entities involved in a particular reasoning episode is small. Hence instead of assigning a distinct signature to every entity, it suffices to assign distinct signatures to only entities that are participating in an episode of reasoning. Furthermore, this assignment need exist only for the duration of a reasoning episode. One can interpret the relative phase in which a node is firing as such a *transient* signature of the node. The discussion in Section 8.2 about working memory and medium-term memory (also Section 10) suggests how an augmented system including a medium-term memory may engage in tasks involving more than 10 or so entities.

Barnden & Srinivas (1991) have proposed 'Conposit', a connectionist production system. In Conposit, patterns are associated by virtue of the *relative position* of registers containing these patterns, as well as the *similarity* between patterns. Argument bindings are propagated by a connectionist interpreter that reads the contents of registers and updates them. We believe that Conposit may be an appropriate architecture for modeling complex reflective processes, but it may not be best suited for modeling reflexive reasoning.

Another solution to the binding problem is based on frequency modulation whereby dynamic bindings may be encoded by having the appropriate nodes fire with the same frequency (Tomabechi & Kitano 1989).

9.4 Using patterns for propagating bindings

An important aspect of the proposed reasoning system is the organization of n-ary rules into a directed graph wherein the inferential dependencies between antecedent and consequent predicates together with the correspondence between the predicate arguments are represented explicitly. As we have seen, this encoding in conjunction with the temporal representation of dynamic bindings leads to an efficient reasoning system. But the above encoding of rules is significant in its own right. One may take this framework for organizing rules and obtain other organizationally isomorphic connectionist systems by using alternate techniques (e.g., frequency encoding) for representing dynamic bindings. These systems, however, will differ in the size of the resulting network, constraints on the nature of reasoning, reasoning speed, and biological plausibility. To

illustrate how the suggested organization of rules and arguments may be combined with alternate techniques for propagating dynamic bindings, we use the proposed encoding of rules in conjunction with what may be referred to as the *pattern-containment* approach.[35]

In the pattern-containment approach we assume that each argument is represented by a cluster of n nodes and inferential links between arguments are represented by connecting the nodes in the associated argument clusters. An n-dimensional pattern of activity is associated with each concept (i.e., an instance or a type), and a dynamic binding between a concept and an argument is represented by inducing the pattern of activation associated with the concept in the appropriate argument cluster. The propagation of dynamic bindings in the system occurs by the propagation (replication) of patterns of activity along connected argument clusters.

It is instructive to compare the pattern-containment approach with the temporal synchrony approach. The key question is: 'What is the significance of the pattern of activity that is associated with a concept and propagated across argument clusters?' One possibility is that each n dimensional pattern encodes the signature associated with some concept (Lange & Dyer 1989). As we pointed out earlier, the value of n would depend on N, the number of distinct concepts represented in the system. If we assume that concepts are assigned *arbitrary* patterns as signatures, n would equal $log_2 N$. Alternately the pattern of activity could encode all the micro-features of a concept (Hinton, 1981; Rumelhart & McClelland, 1986). Such a pattern, however, would have to be even larger. Both these interpretations of patterns make suboptimal use of computational resources: Each argument cluster has to be large enough to encode the full signature of a concept or all the micro-features associated with a concept. Also individual bindings have to be propagated by propagating large patterns of activity. An attractive alternative would be to assume that the patterns associated with concepts during the propagation of bindings are some sort of *reduced descriptions*. We suggest that the temporal synchrony approach does exactly this — albeit in an unusual manner. During the propagation of bindings, the relative phase of firing of an active concept acts as a highly reduced description of that concept.

The use of temporal synchrony enables our system to do with one node and one link, what the pattern-containment approach does using n nodes and links. The temporal approach also leads to a simple encoding of long-term facts. In contrast, the realization of a long-term fact in the pattern-containment approach will be more complex since it must support m — where m is the arity of the fact predicate — n-bit comparisons to check whether the dynamic bindings match the static bindings encoded in the fact. In Section 7.3, we suggested that single (idealized) nodes in our system would have to be mapped to ensembles of nodes and single (idealized) links would have to be mapped to a group of links. This mapping however, was required to deal with noise in the system and the pattern-containment approach will also have to be augmented in order to deal with noise.

10 Discussion

We have presented a neurally plausible model for knowledge representation and reflexive reasoning. The model supports the long-term encoding of general instantiation independent structures as well as specific situations involving *n*-ary relations. It also supports the representation of dynamic information and its interaction with long-term knowledge. Everything presented in the

paper, except for the treatment of soft rules (Section 5.5) has been simulated. The proposed model makes several specific predictions about the nature of reflexive reasoning and the capacity of the working memory underlying reflexive reasoning. These predictions are verifiable and we hope that they will be explored by experimental psychologists. The proposed representational mechanisms are quite general and should be applicable to other problems in cognition whose formulation requires the expressive power of *n*-ary predicates and whose solution requires rapid and systematic interactions between long-term and dynamic structures. These include problems in high-level vision, other problems in language processing such as syntactic processing, and reactive planning. Below we discuss some problems that need to be addressed if the representational mechanisms proposed here are to be applied in an extended setting.

10.1 Where do phases originate?

In a sense, the 'source' of rhythmic activity in the proposed reasoning system is clearly identifiable: The process that poses a query to the system provides staggered oscillatory inputs to entities mentioned in the query and thereby activates them in distinct phases. In a composite perceptual/linguistic/reasoning system, however, such a separation in the phase of the firing of distinct entities must occur intrinsically. For example, the utterance 'John gave Mary Book1' should automatically result in the representations of 'John', 'Mary', and 'Book1' firing in different phases and synchronously with *giver*, *recipient*, and *give-obj*, respectively.

The problem of automatic phase separation and consequent segmentation and feature binding has been addressed by several researchers. For example, Horn et al. (1991) demonstrate how an input pattern containing a red square and a blue circle, can result in the firing of nodes representing the features 'red' and 'square' in one phase, and the nodes representing the features 'blue' and 'circle' in a different phase. The model, however, does not work if there are more than two objects. An internal attentional mechanism similar to the 'searchlight' proposed by Crick (1984) may be required for dealing with more elaborate situations.

In the case of linguistic input, we believe that the *initial* phase separation in the firing of each constituent is the outcome of the parsing process. The parser module expresses the result of the parsing process — primarily, the bindings between syntactic arguments and constituents — by forcing appropriate nodes to fire in, and out of, synchrony. This is illustrated in a parser for English designed by Henderson (1991) using the proposed model for reflexive reasoning.

10.2 Who reads the synchronous firing of nodes?

There is no homunculus in our system that 'reads' the synchronous activity to detect dynamic bindings. Instead, the synchronous activity is 'read' by various long-term structures in the system which do so by simply detecting coincidence (or the lack of it) among their inputs. For example, long-term facts 'read' the rhythmic activity as it propagates past them and become active whenever the dynamic bindings are appropriate. Similarly, τ-or nodes enforce type restrictions (e.g., the node a in Fig. 24) by enabling the firing of a rule whenever the appropriate argument and type nodes are firing in phase. We have also designed a connectionist mechanism that automatically extracts answers to *wh*-queries and relays them to an output device (McKendall

1991). We associate a code or a 'name' with each concept. This name has no internal significance and is meant solely for communicating with the system's environment. The mechanism channels the 'names' of concepts that constitute an answer to an output buffer in an interleaved fashion. For example, the patterns for 'Ball1' and 'Book1' would alternate in the output buffer after the *wh*-query *own(Mary,?)?* is posed with reference to the network in Fig. 12.

10.3 How are phases recycled?

The constraint that computations must involve only a small number of entities at any given time seems reasonable if we restrict ourselves to a single episode of reasoning, understanding a few sentences, or observing a simple scene. But what happens when the agent is participating in a dialog or scanning a complex scene where the *total* number of significant entities exceeds the number of distinct phases that can coexist. In such situations the set of entities in 'focus' must keep changing constantly with entities shifting in and out of focus in a dynamic manner. Identifying the mechanisms that underlie such internal shifts of attention and cause the system's oscillatory activity to evolve smoothly so that new entities start firing in a phase while entities presently firing in a phase gradually 'release' their phase, remains a challenging open problem (but see Crick & Koch 1990). In this context one must also note that the notion of an entity is itself very fluid. In certain situations, John may be an appropriate entity. In other situations, John's face or perhaps even John's nose may be the appropriate entity.

The notion of the release of phases has a natural interpretation in the parsing system described in (Henderson 1991). The parser is incremental and its output is a sequence of derivation steps that leads to the parse. The entities in the parser are non-terminals of the grammar, and hence, each active non-terminal must fire in a distinct phase. Under appropriate conditions during the parsing process — for example, when a non-terminal ceases to be on the right frontier of the phrase structure — the phase associated with a non-terminal can be 'released' and hence, become available for non-terminals introduced by subsequent words in the input. This allows the parser to recover the structure of arbitrary long sentences as long as the dynamic state required to parse the sentence does not exceed the bounds on the number of phases and the number of instantiations per predicate.

10.4 Generalizing the use of synchronous oscillations

Thus far we have assumed that the scope of phase distribution is the entire system. We must however, consider the possibility where the system is composed of several modules (say the perceptual, linguistic, or reasoning modules). If we combine the requirements of all these modules, it becomes obvious that ten or so phases will be inadequate for representing all the entities that must remain active at any given time. Thus a temporal coding of dynamic bindings is not viable if a single phase distribution must extend across all the modules. Therefore it becomes crucial that each module have its own phase distribution so that each module may maintain bindings involving ten or so entities. This however, poses a problem: how should modules communicate with each other in a consistent manner? Consider a system whose visual module is seeing 'John' and whose conceptual module is thinking something about John. How

should the visual and conceptual modules share information about John even though the phase and frequency of the nodes encoding John in the two systems may be different? Aaronson (1991) describes a connectionist interface that allows two phase-based modules, each with its own phase structure, to exchange binding information.

10.5 Memorizing facts: converting dynamic bindings to static patterns

In the proposed system, dynamic information is represented as transient rhythmic activity while long-term memory is encoded using 'hard-wired' interconnections between nodes. We have not discussed how appropriate dynamic information may be converted into, and recorded as, synaptically encoded long-term structures. A specific problem concerns the conversion of dynamic bindings corresponding to a novel (but salient) fact into a *medium-term* fact by converting the set of dynamic bindings into a set of static bindings that last longer than a few hundred msec (perhaps, even days or weeks). This problem has been addressed in (Geib 1990) by using recruitment learning (Wickelgren 1979; Feldman 1982; Shastri 1988a) in conjunction with a fast weight change process abstractly modeled after long-term potentiation (Lynch 1986). The proposed solution allows a one-shot conversion of dynamic facts into a structurally encoded fact in the presence of a 'learn' signal. It is envisaged that subsequently, such medium-term structures can be converted into long-term structures by other processes (Marr 1971; Squire 1987; Squire & Zola-Morgan 1991). The notion of fast synapses proposed by von der Malsburg (1981) may also play an intermediate role in sustaining memories that must last beyond a few hundred msec.

10.6 Learning rules

The problem of learning the representation of rules in a system that uses a temporal representation is no more difficult than the problem of learning structured representation in connectionist networks. Instead of being triggered by 'simple' co-activation, learning must now be triggered by synchronous activation. Recently, Mozer, Zemel & Behrman (1991) have demonstrated how backpropagation style learning may be generalized to networks of nodes that are essentially like ρ-btu nodes. We are addressing the problem of learning in the context of pre-existing predicates and concepts where it is desired that the co-occurrence of events should lead to the formation of appropriate connections between predicate arguments. A special case involves assuming generic interconnections between predicate arguments, and viewing rule-learning as learning the correct type restrictions/preferences on argument fillers. This may be achieved by modifying weights on links between the type hierarchy and the rule component. (refer to Sections 5.4 and 5.5).

End Notes

1. For example, see Kintsch 1974; Bobrow & Collins 1975; Charniak 1976; Just & Carpenter 1977; Schank & Abelson 1977; Fahlman 1979; Lehnert & Ringle 1982; Dyer 1983; Wilensky 1983; Allen 1987; Norvig 1989; and Corriveau 1991.

2. That reflexive reasoning occurs spontaneously and without conscious effort does not imply that the agent cannot become aware and conscious of the *result* of such reasoning. To wit,

an agent's 'yes' response to the question 'Does John own a car?' given 'John bought a Rolls-Royce'. In many situations, however, the result of reflexive reasoning may only be manifest in the 'mental state' of the agent. For example, during reading the effect of such reasoning may be manifest primarily in the agent's sense of 'understanding', 'coherence' (or lack thereof), 'disbelief', 'humor', etc.

3. The reflexive/reflective distinction we make here in the context of reasoning, shares a number of features with the automatic/controlled distinction proposed by Schneider and Shiffrin (Schneider & Shiffrin 1977; Shiffrin & Schneider 1977) (also see Posner & Snyder 1975). Like automatic processing, reflexive reasoning is parallel, fast, occurs spontaneously, and the agent is unaware of the reasoning process per se. However, the working memory underlying reflexive reasoning has specific capacity limitations (see Section 8.2). In formulating the problem of reflexive reasoning and developing a detailed computational model for it, we have generalized the notion of automatic processing by bringing into its fold the more conceptual task of systematic reasoning.

4. If we assume that information is encoded in the firing rate of a neuron then the amount of information that can be conveyed in a 'message' would depend on ΔF, the range over which the firing frequency of a presynaptic neuron can vary, and ΔT, the window of time over which a postsynaptic neuron can 'sample' the incident spike train. ΔT is essentially how long a neuron can 'remember' a spike and depends on the time course of the postsynaptic potential and the ensuing changes in the membrane potential of the postsynaptic neuron. A plausible value of ΔF may be about 200. This means that in order to decode a message containing 2 bits of information, ΔT has to be about 15 msec, and to decode a 3-bit message it must be 35 about msec.

   One could argue that neurons may be capable of communicating more complex messages by using *variations* in interspike delays to encode information (for e.g., see Strehler & Lestienne 1986). However, Thorpe and Imbert (1989) have argued that in the context of rapid processing, the firing rate of neurons relative to the time available to neurons to respond to their inputs implies that a presynaptic neuron can only communicate one or two spikes to a postsynaptic neuron before the latter must produce an output. Thus the information communicated in a message remains limited even if interspike delays are used as temporal codes. This does not imply that *networks* of neurons cannot represent and process complex structures. Clearly they can. The interesting question is how?

5. This observation does not presuppose any particular encoding scheme and applies to 'localist', 'distributed', as well as 'hybrid' schemes of representation. The point is purely numerical — any encoding scheme that requires n^2 nodes to represent a LTKB of size n will require 10^{16} nodes to represent a LTKB of size 10^8.

6. This hypothesis does not conflict with the *fan effect* (Anderson 1983). See Section 8.6.

7. The rules used in this and other examples are only meant to illustrate the dynamic binding problem, and are not intended to be a detailed characterization of common sense knowl-

edge. For example, the rule relating 'giving' and 'owning' is an oversimplification and does not capture the richness and complexity of the actual notions of giving and owning.

8. While 'systematicity' has a broader connotation (e.g., see Fodor & Pylyshyn 1988), we use it here to refer specifically to the correspondence between predicate arguments stipulated by rules.

9. The symbol $\forall$ is the universal quantifier which may informally be interpreted to mean 'for all', and the symbol $\Rightarrow$ is the logical connective 'implies'. Thus the statement $\forall u, v\ [buy(u, v) \Rightarrow own(u, v)]$ asserts that 'for any assignment of values to u and v, if u buys v then u owns v'.

10. A similar formation of 'static' bindings occurs in any learning network with hidden nodes. Observe that a hidden node at level l learns to respond systematically to the activity of nodes at levels $l-1$ and below, and in doing so the network *learns* new bindings between representations at level l and $l-1$. These bindings, however, are *static* and the time it takes for such 'bindings' to get established is many orders of magnitude greater than the time within which dynamic bindings must be established.

11. Feature binding can be achieved by creating sets of features such that features belonging to the same entity are placed in the same set. In terms of expressive power, *unary* predicates suffice to solve this problem. For example, the grouping of features belonging to a 'red smooth square' and a 'blue dotted circle' can be expressed using unary predicates such as: $red(obj1) \wedge smooth(obj1) \wedge square(obj1)$ and $blue(obj2) \wedge dotted(obj2) \wedge circle(obj2)$.

12. We first described our proposed model in 1990 (Shastri & Ajjanagadde 1990). An earlier version using a central clock was reported in (Ajjanagadde & Shastri 1989).

13. As stated in note 11 *unary* predicates suffice to solve the feature binding problem and the expressive power of the models cited above is limited to *unary*-predicates (see Hummel & Biederman 1991). The greater expressive power provided by n-ary predicates would eventually be required by more sophisticated models of visual processing.

14. There are other variants of marker passing (e.g., see Charniak 1983; Hirst 1987; Hendler 1987; and Norvig 1989) where 'markers' are even more complex messages containing a marker bit, a strength measure, backpointers to the original and immediate source of the marker, and sometimes a flag that indicates which types of links the marker will propagate along. The marker passing system has to process the information contained in markers, extract paths traced by markers, and evaluate the relevance of these paths. In view of this, such marker passing systems are not relevant to our discussion.

15. We can generalize the behavior of a ρ-btu node to account for weighted links by assuming that a node will fire if and only if the weighted sum of synchronous inputs is greater than or equal to n (see Section 5.5 and 8.1).

16. In the idealized model each argument is encoded as a single ρ-btu node and hence, it is reasonable to assume that a node may fire in response to a single input. The thresholds of nodes in the ensemble based model will be higher and will depend on the average inter-ensemble connections per node.

17. A constant refers to a specific entity in the domain, the symbol $\exists$ is the existential quantifier which may be interpreted to mean 'there exists'. Recall that the symbol $\forall$ is the universal quantifier which may be interpreted to mean 'for all'. Thus the statement: $\forall x\ [person(x) \Rightarrow \exists z\ mother(z, x)]$ asserts that 'for every person x there exists some z such that z is the mother of x. The symbol $\wedge$ is the logical connective 'and'.

18. The system can encode first-order, function-free Horn Clauses with the added restriction that any variable occurring in multiple argument positions in the antecedent of a rule must also appear in the consequent. Horn Clauses form the basis of PROLOG, a programming language used extensively in artificial intelligence (e.g., see Genesereth & Nilsson 1987).

19. This time consists of i) $l\pi$, the time taken by the activation originating at the enabler of the query predicate to reach the enabler of the predicate(s) that are relevant to the derivation of the query, ii) π, the time taken by the relevant fact(s) to become active, iii) π, the time taken by the active fact(s) to activate the relevant collector(s), and iv) $l\pi$, the time taken by the activation to travel from the collector of the relevant predicate(s) to the collector of the query predicate.

20. The closed world assumption simply means that any fact F that is neither in the knowledge base nor deducible from the knowledge base, may be assumed to be false.

21. Here we are using 'concept' to refer only to the entities and types encoded in the hierarchy. This is not to suggest that predicates such as 'give' and 'own' that are not represented in the *IS-A* hierarchy are not concepts in the broader sense of the word.

22. In our formulation each *IS-A* link is strict and only property values are exceptional. This approach for dealing with exceptional and defeasible information in *IS-A* hierarchies is explained in (Shastri 1988a).

23. This is required because a fact is true of *some* entity of type C if one or more of the following holds: i) The fact is universally true of a super-concept of C, ii) the fact is true of *some* sub-concept/instance of C, or iii) the fact is universally true of a super-concept of a sub-concept/instance of C. The latter is required if concepts in the *IS-A* hierarchy can have multiple parents.

24. These times are approximate because the time required for propagation along the *IS-A* hierarchy and the rules may overlap and hence, the actual time may be less. For example, the time to perform a predictive inference may also only be $max(l_1\pi, 3l_2\pi)$. It is also possible for the actual time to be greater because in the worst case, it may take up to eight cycles instead of three to traverse an *IS-A* link.

25. m, the number of antecedent predicates in a rule, can also be reduced by introducing ancillary predicates. For example, the rule $\forall x, y, z\ P(x, y, z) \wedge Q(x, y, z) \wedge R(x, y, z) \Rightarrow S(x, y, z)$ may be replaced by two rules each of which has only two antecedent predicates: $\forall x, y, z\ \ P(x, y, z) \wedge Q(x, y, z) \ \Rightarrow\ S1(x, y, z)$ and $\forall x, y, z\ \ S1(x, y, z) \wedge R(x, y, z) \ \Rightarrow\ S(x, y, z)$. The benefit of reducing m in this manner has to be weighed against the cost of introducing an additional predicate in the system. But the savings outweigh the costs if such a predicate helps in reducing the m value of several rules.

26. The reasoning system uses of the phase of activation to encode binding information. Therefore, in principle, the *amplitude* of activation could be used to represent the 'strength' of dynamic bindings and rule firings. Note however, that the amplitude of a node's output is encoded by the spiking frequency and the use of varying frequency to encode rule strengths will interfere with the encoding of dynamic bindings.

27. While the occurrence of synchronous activity is less controversial, the occurrence of synchronized oscillations in the animal brain and its representational significance is still a matter of controversy. More evidence is needed to firmly establish the role of oscillatory activity in neural information processing. Some researchers have reported difficulty in demonstrating oscillatory activity in the primate visual system using static stimuli (e.g., Rolls 1991; Tovee & Rolls 1992). In this context, however, it must be recognized that a very small fraction of neurons would be expected to participate in an episode of synchronous activity. Furthermore, the grouping of neurons will be dynamic and vary considerably from one episode of reasoning to another. Hence synchronous oscillations would be very difficult to detect.

28. A more detailed model of such coupling has since been developed (Mandelbaum 1991).

29. These timings were obtained by analyzing the simulations of the reflexive reasoning system carried out using a simulation system developed by D.R. Mani.

30. The above behavior generalizes the notion of a 'strength' associated with concepts (cf: Anderson 1983) and extends it to rules, *IS-A* relations, facts, and even individual static bindings in the LTKB.

31. The cost of realizing multiple instantiation of concepts is considerably lower than that of realizing the multiple instantiation of predicates. Thus the value of k_1 can be higher than 3. Observe however, that k_1 need be no more than $\lfloor \pi_{max}/\omega \rfloor$.

32. There are several ways of encoding the relevant kinship knowledge. All these, however, pose the same problem — the antecedent of one of the rules contains a repeated variable that does not occur in the consequent. One possible encoding of the relevant knowledge is given below (note that 'Self' refers to the agent and the rest of the names have been chosen arbitrarily to complete the example). The long-term facts are: $grandfather(George, Self)$, $mother(Susan, Self)$, and $father(George, Susan)$. The rule is: $\forall x, y, z\ \ grandfather(x, y) \ \wedge father(x, z) \ \wedge mother(z, y) \ \Rightarrow\ maternal\text{-}grandfather(x, y)$.

33. In addition to the constraints on the WMRR, the number of dynamic facts that can be communicated to an agent at one time will be bounded by the rather limited capacity of the *overt* short-term memory.

34. Ullman and van Gelder treat the number of nodes required to encode the LTKB as a fixed cost, and hence, do not refer to its size in computing the space complexity of their system. If the size of the LTKB is taken into account, the number of processors required by their system turns out to be a high degree polynomial.

35. The relation between our approach and the pattern-containment approach was pointed out by Geoff Hinton.

Acknowledgements

Thanks to Moshe Abeles, John Barnden, Gary Cottrell, Mike Dyer, Jerry Feldman, George Gerstein, Pat Hayes, Geoff Hinton, Christopher von der Malsburg, Jordan Pollack, Terry Sejnowski, Paul Smolensky, Simon Thorpe, Dave Touretzky, and several anonymous referees for their comments and suggestions. Thanks to D.R. Mani for his work on the *IS-A* hierarchy interface and the multiple instantiation problem. Also for simulating the model and generating figures. This research was supported by NSF grants IRI 88-05465, ARO grants DAA29-84-9-0027 and DAAL03-89-C-0031, and the DFG grant Schr 275/7-1.

References

Aaronson, J., (1991) Dynamic Fact Communication Mechanism: A Connectionist Interface. *Proceedings of the Thirteenth Conference of the Cognitive Science Society*. Lawrence Erlbaum.

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

Abeles, M. (1991) Neural Codes for Higher Brain Function. In *Information Processing by the Brain: Views and Hypotheses from a Physiological-Cognitive Perspective*. ed. H. J. Markowitsch. Hans Huber.

Ajjanagadde, V. G. (1990) Reasoning with function symbols in a connectionist system, *Proceedings of the Twelfth Conference of the Cognitive Science*. Lawrence Erlbaum.

Ajjanagadde, V. G. (1991) Abductive reasoning in connectionist networks: Incorporating variables, background knowledge, and structured explananda. Technical report WSI-91-7. Wilhelm-Schickard-Institute, University of Tuebingen, Germany.

Allen, J. (1987) *Natural Language Understanding*. Benjamin Cummins.

Anderson, J. R. (1983) *The Architecture of Cognition*. Harvard University Press.

Baddeley, A. (1986) *Working Memory*. Clarendon Press.

Barnden, J., & Srinivas, K. (1991) Encoding Techniques for Complex Information Structures in Connectionist Systems. *Connection Science*, Vol. 3, No. 3, 269-315.

Bienenstock, E. (1991) Notes on the Growth of a "Composition Machine". Presented at the Interdisciplinary Workshop on Compositionality in Cognition and Neural Networks. Abbaye de Royaumont. May 1991.

Bobrow, D. & Collins, A. eds. (1975) *Representation and Understanding*. Academic Press.

Buchanan, B. G., & Shortliffe E. F. (1984) *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*. Addison-Wesley.

Bylander, T., Allemang, D., Tanner, M. C. & Josephson, J. R. (1991) The computational complexity of abduction. *Artificial Intelligence*, 47(1-3), 25–60.

Carpenter, P. A. & Just, M. A. (1977) Reading Comprehension as Eyes See It. In: *Cognitive Processes in Comprehension*. ed. M. A. Just & P. A. Carpenter. Lawrence Erlbaum.

Charniak, E. (1976) Inference and Knowledge (I and II). In *Computational Semantics*, ed. E. Charniak & Y. Wilks. North-Holland.

Charniak, E. (1983) Passing Markers: A Theory of Contextual Inference in Language Comprehension. *Cognitive Science*, 7, 171-190.

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

Corriveau, J. (1991) Time-Constrained Memory for Reader-Based Text Comprehension. Technical Report CSRI-246. Computer Science Research Institute, University of Toronto.

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

Crick, F. & Koch, C. (1990) Towards a neurobiological theory of consciousness. *Seminars in Neurosciences*, 2, 263-275.

Damasio, A. R. (1989) Time-locked multiregional retroactivation: A systems-level proposal for the neural substrates of recall and recognition. *Cognition*, 33, 25-62.

Dolan, C. P. & Smolensky, P. (1989) Tensor product production system: a modular architecture and representation. *Connection Science*, 1, 53–68.

Dyer, M. (1983) *In-Depth Understanding — A Computer Model of Integrated Processing for Narrative Comprehension*. MIT Press.

Eckhorn, R., Bauer, R., Jordan, W., Brosch, M., Kruse, W., Munk, M., & Reitboeck, H.J. (1988) Coherent oscillations: A mechanism of feature linking in the visual cortex? Multiple electrode and correlation analysis in the cat. *Biol. Cybernet.* **60** 121-130.

Eckhorn, R., Reitboeck, H.J., Arndt, M., & Dicke, P. (1990) Feature linking via Synchronization among Distributed Assemblies: Simulations of Results from Cat Visual Cortex. *Neural Computation* **2**, 293-307.

Engel, A. K., Koenig, P., Kreiter, A. K., Gray, C. M., & Singer, W., (1991) Temporal Coding by Coherent Oscillations as a Potential Solution to the Binding Problem: Physiological Evidence. In *Nonlinear Dynamics and Neural Networks*, ed. H. G. Schuster & W. Singer. Weinheim.

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

Fahlman, S. E., (1981) Representing Implicit knowledge. In *Parallel Models of Associative Memory* ed. G. E. Hinton & J. A. Anderson. Lawrence Erlbaum.

Fahlman, S. E., Touretzky, D. S. & Walter van R., (1981) Cancellation in a parallel semantic network. In *The Proceedings of IJCAI-81*. Morgan Kaufman.

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

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

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

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

Freeman, W.J. (1981) A physiological hypothesis of perception. In *Perspectives in Biology and Medicine*, 24(4), 561–592. Summer 1981.

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

Geib, C. (1990) A connectionist model of medium-term memory. Term Report. Fall, 1990. Department of Computer and Information Science, University of Pennsylvania.

Geller, J. & Du, C. (1991) Parallel implementation of a class reasoner. *Journal of Theoretical Artificial Intelligence*, 3, 109-127.

Genesereth, M.R & Nilsson, N.J. (1987) Logical Foundations of Artificial Intelligence. Morgan Kaufman.

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.

Gray, C. M., Koenig, P., Engel, A. K., & Singer, W. (1989) Oscillatory responses in cat visual cortex exhibit inter-columnar synchronization which reflects global stimulus properties. *Nature*. Vol. 338, 334-337.

Gray, C. M. & Singer, W. (1989) Stimulus-specific neuronal oscillations in orientation specific columns of cat visual cortex. *Proceedings of the National Academy of Science*, Vol. 86, pp. 1698-1702.

Gray, C. M., Engel, A. K., Koenig, P., & Singer, W. (1991) Properties of Synchronous Oscillatory Neuronal Interactions in Cat Striate Cortex. In *Nonlinear Dynamics and Neural Networks*, ed. H. G. Schuster & W. Singer. Weinheim.

Guha, R.V. & Lenat, D.B. (1990) Cyc: A Mid-Term report, *AI Magazine*, Volume 11, Number 3, 1990.

Hatfield, H. (1991) Representation and rule-instantiation in connectionist networks. In *Connectionism and the philosophy of mind*, ed. T. Horgan & J. Tienson. Kluwer Academic.

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

Henderson, J. (1991) *A connectionist model of real-time syntactic parsing in bounded memory*. Dissertation proposal. Department of Computer and Information Science, University of Pennsylvania. (Unpublished report).

Hendler, J. (1987) *Integrating Marker-Passing and Problem Solving: A Spreading Activation approach to Improved Choice in Planning*. Lawrence Erlbaum.

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

Hirst, G. (1987) *Semantic Interpretation and Ambiguity*. Cambridge University Press.

Hölldobler, S. (1990). CHCL: A Connectionist Inference System for Horn Logic based on the Connection Method and Using Limited Resources. *TR-90-042*, International Computer Science Institute, Berkeley, CA.

Horn, D., Sagi, D., & Usher, M. (1991) Segmentation, Binding, and Illusory Conjunctions. *Neural Computation*, Vol. 3, No. 4, 510-525.

Hummel, J. E., & Biederman, I. (1991) Dynamic Binding in a neural network for shape recognition. *Psychological Review* (in press).

Just, M. A., & Carpenter, P. A. eds. (1977). *Cognitive Processes in Comprehension*. Lawrence Erlbaum.

Kautz, H. A. & Selman, B., (1991) Hard problems for Simple Default Logics. *Artificial Intelligence*, 47(1-3), 243-279.

Keenan, J. M., Baillet, S. D., & Brown, P. (1984) The Effects of Causal Cohesion on Comprehension and Memory. *Journal of Verbal Learning and Verbal Behavior*, 23, 115-126.

Kintsch, W. ed. (1974) *The Representation of Meaning in Memory*. Lawrence Erlbaum.

Kintsch, W. (1988) The Role of Knowledge Discourse Comprehension: A Construction-Integration Model. *Psychological Review*, Vol. 95, 163-182.

Kreiter, A. K. & Singer, W., (1992) Oscillatory Neuronal Responses in the Visual Cortex of the Awake Macaque Monkey. *European Journal of Neuroscience*, Vol. 4, 369-375.

Lange, T. E., & Dyer, M. G. (1989) High-level Inferencing in a Connectionist Network. *Connection Science*, Vol. 1, No. 2, 181-217.

Lakoff, G. (1987) *Women, Fire, and Dangerous Things. What Categories Reveal about the Mind*. Chicago University Press.

Lehnert, W.G. & Ringle, M.H. ed. (1982) *Strategies for Natural Language Processing.* Lawrence Erlbaum.

Levesque, H. J. (1988) Logic and the complexity of reasoning. *Journal of Philosophical Logic*, 17, pp 335-389.

Levesque, H.J. & Brachman, R. (1985) A fundamental tradeoff in knowledge representation and reasoning, In *Readings in Knowledge Representation*, ed. R. Brachman & H.J. Levesque. Morgan Kaufman.

Lynch, G. (1986) *Synapses, Circuits, and the Beginnings of Memory*. MIT Press.

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

von der Malsburg, C. (1981) The correlation theory of brain function. Internal Report 81-2. Department of Neurobiology, Max-Planck-Institute for Biophysical Chemistry, Gottingen, FRG. 1981.

von der Malsburg, C. (1986) Am I thinking assemblies? In *Brain Theory*, ed. G. Palm & A. Aertsen. Springer-Verlag.

von der Malsburg, C. & Schneider (1986) A Neural Cocktail-Party Processor, *Biological Cybernetics*, 54, 29-40.

Mandelbaum, R., (1991) A robust model for temporal synchronisation of distant nodes: Description and Simulation. Term Project: CIS 629, Spring 1991. University of Pennsylvania. (Unpublished report).

Mandelbaum, R. & Shastri, L., (1990) A robust model for temporal synchronisation of distant nodes. (Unpublished report).

Mani, D. R. & Shastri, L. (1991) Combining a Connectionist Type Hierarchy with a Connectionist Rule-Based Reasoner, *Proceedings of the Thirteenth Conference of the Cognitive Science Society*. Lawrence Erlbaum.

Mani, D. R. & Shastri, L., (1992) A connectionist solution to the multiple instantiation problem using temporal synchrony. To appear in *Proceedings of the Fourteenth Conference of the Cognitive Science Society*. Lawrence Erlbaum.

Marr, D. (1971) Simple memory: a theory for archicortex. *Philosophical Transactions of the Royal Society*, B 262: 23-81.

McAllester, D.A. (1990) Automatic recognition of tractability in inference relations. Memo 1215, MIT Artificial Intelligence Laboratory, February 1990.

McKendall, T. (1991) A Design for an Answer Extraction and Display Scheme for a Connectionist Rule-Based Reasoner. Report on work done for NSF REU grant IRI 88-05465. (Unpublished report).

McKoon, G., & Ratcliff, R. (1980) The Comprehension Processes and Memory Structures Involved in Anaphoric Reference. *Journal of Verbal Learning and Verbal Behavior*, 19, 668-682.

McKoon, G., & Ratcliff, R. (1981) The Comprehension Processes and Memory Structures Involved in Instrumental Inference. *Journal of Verbal Learning and Verbal Behavior*, 20, 671-682.

McKoon, G., & Ratcliff, R. (1986) Inferences About Predictable Events. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, Vol. 12, 82-91.

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), pp. 81-97.

Minsky, M., (1975) A Framework for Representing Knowledge. In *The Psychology of Computer Vision*, ed. P. M. Winston. McGraw Hill.

McCarthy, J. (1988) Epistemological challenges for connectionism. Commentary to 'Proper treatment of Connectionism' by Smolensky, P. *Behavioral and Brain Sciences*, 11:1.

Mozer, M. C., Zemel, R. S., & Behrman, M. (1991) Learning to segment images using dynamic feature binding. *Technical Report CU-CS-540-91*, University of Colorado at Boulder.

Newell A. (1980) Harpy, production systems and human cognition. In *Perception and production of fluent speech*, ed. R. Cole. Lawrence Erlbaum.

Newell, A. (1990) *Unified Theories of Cognition*. Harvard University Press.

Newell, A & Simon, H.A. (1972) *Human problem solving*. Prentice-Hall.

Norvig, P. (1989) Marker Passing as a Weak Method for Text Inferencing. *Cognitive Science*, 113:569-620.

Posner, M.I. & Snyder, C.R.R. (1975) Attention and Cognitive Control. In *Information Processing and Cognition. The Loyola Symposium*, ed. R.L. Solso. Lawrence Erlbaum.

Potts, G. R., Keenan, J. M., & Golding, J. M. (1988) Assessing the Occurrence of Elaborative Inferences: Lexical Decision versus Naming. *Journal of Memory and Language*, 27, 399-415.

Quillian, M.R. (1968). Semantic Memory. In *Semantic Information Processing*, ed. M. Minsky. MIT Press.

Reder, L. M., & Ross, B. H. (1983) Integrated Knowledge in Different Tasks: The Role of Retrieval Strategy on Fan Effects. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, Vol. 9, 55-72.

Rolls, E.T. (1991) Neural organisation of higher visual functions. *Current Opinion in Neurobiology*, 1:274-278.

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

Schank, R. C., & Abelson, R. P. (1977) *Scripts, Plans, Goals and Understanding*. Lawrence Erlbaum.

Schneider, W & Shiffrin, R.M. (1977) Controlled and automatic human information processing: I. Detection, Search, and Attention. *Psychological Review*, 84, 1-66.

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

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

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. (1990) Connectionism and the Computational Effectiveness of Reasoning. *Theoretical Linguistics*, Vol. 16, No. 1, 65–87, 1990.

Shastri, L. (1991) Relevance of Connectionism to AI: A representation and reasoning perspective. In *Advances in Connectionist and Neural Computation Theory*, vol. 1, ed. J. Barnden and J. Pollack. Ablex.

Shastri, L. (1992) A realization of preference rules using temporal synchrony. (In preparation).

Shastri, L. & Ajjanagadde, V. G. (1990). A connectionist representation of rules, variables and dynamic bidings. Technical Report MS-CIS-90-05, Department of Computer and Information Science, Univ. of Pennsylvania.

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

Shiffrin, R.M. & Schneider, W. (1977) Controlled and Automatic Human Information Processing: II. Perceptual Learning, Automatic Attending, and a General Theory. *Psychological Review*, 84, 127-190.

Singer, M., & Ferreira, F. (1983) Inferring Consequences in Story Comprehension. *Journal of Verbal Learning and Verbal Behavior*, 22, 437-448.

Smolensky, P. (1990) Tensor product variable binding and the representation of symbolic structure in connectionist systems. *Artificial Intelligence*, 46 (1-2), 159–216.

Squire, L. R. (1987) *Memory and Brain.* Oxford University Press.

Squire, L. R. & Zola-Morgan, S. (1991) The medial Temporal Lobe Memory System, *Science*, 253, 1380–1386. 253, p. 1380–1386.

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

Strehler, B.L. & Lestienne, R., (1986) Evidence on precise time-coded symbols and memory of patterns in monkey cortical neuronal spike trains. *Proceedings of the National Academy of Science*, 83, 9812-9816.

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

Sumida, R. A. & Dyer, M. G. (1989) Storing and Generalizing Multiple Instances while Maintaining Knowledge-Level Parallelism. In *Proceedings of IJCAI-89.* Morgan Kaufman.

Thorpe, S.J. and Imbert, M. (1989) Biological constraints on connectionist modelling. In *Connectionism in Perspective*, ed. R. Pfeiffer, Z. Schreter, F. Fogelman-Souile, & L. Steels. Elsevier.

Tomabechi, H. & Kitano, H. (1989) Beyond PDP: The Frequency Modulation Neural Network Approach. In *Proceedings of IJCAI-89*. Morgan Kaufman.

Touretzky, D. S., (1986) *The Mathematics of Inheritance Systems*. Morgan Kaufman/Pitman.

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

Tovee, M.J. & Rolls, E.T. (1992) Oscillatory activity is not evident in the primate temporal visual cortex with static stimuli. *Neuroreport*, 3:369-371.

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

Triesman, A. & Gelade, G. (1980) A feature integration theory of attention. *Cognitive Psychology*, 12, 97–136.

Tulving, E. (1983) *Elements of Episodic Memory*. Oxford University Press.

Ullman, J. D. & Gelder, A. V. (1988) Parallel Complexity of Logical Query Programs. *Algorithmica*, 3, 5-42.

Wickelgren, W. A., (1979) Chunking and Consolidation: A Theoretical Synthesis of Semantic Networks, Configuring in Conditioning, S-R Versus Cognitive Learning, Normal Forgetting, the Amnesic Syndrome, and the Hippocampal Arousal System. *Psychological Review*, 86 (1): 44-60.

Wilensky, R. (1983) *Planning and Understanding: A Computational Approach to Human Reasoning*. Addison-Wesley.

Figure Captions

Fig. 1. Encoding static bindings using dedicated nodes and links. $give23$ 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. Later in Section 7.4 we will discuss how nodes may be replaced by an ensemble of nodes.

Fig. 3. Rhythmic pattern of activation representing the dynamic bindings $(giver = John,$ $recipient = Mary,$ $give\text{-}object = Book1)$. These bindings constitute the fact $give(John,$ $Mary,$ $Book1)$. The binding between an argument and a filler is represented by the in-phase firing of associated nodes.

Fig. 4. Representation of the dynamic binding $(giver = John)$ that constitutes the partially instantiated fact 'John gave someone something'.

Fig. 5. Pattern of activation representing the dynamic bindings ($giver = John$, $recipient =$ $Mary$, $give\text{-}object = Book1$, $owner = Mary$, $own\text{-}object = Book1$, $potential\text{-}seller =$ $Mary$, $can\text{-}sell\text{-}object = Book1$). These bindings constitute the facts $give(John,$ $Mary,$ $Book1)$, $own(Mary,$ $Book1)$, and $can\text{-}sell(Mary,$ $Book1)$. The transient representation of an entity is simply a *phase* 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 entities* involved in the bindings. In this example three distinct phases are required. The bindings between $Mary$ and the arguments $recipient$, $owner$, and $potential\text{-}seller$ are represented by the in-phase firing of the appropriate argument nodes with $Mary$.

Fig. 6. 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. 7. Initial pattern of activation representing the bindings *(giver=John, recipient=Mary, give-object=Book1)*

Fig. 8. Pattern of activation after one period of oscillation (with reference to the state of activation in Fig. 7). This state represents the dynamic bindings: *(giver=John, recipient=Mary, give-object=Book1, owner=Mary, own-object=Book1)*. The system has essentially inferred the fact $own(Mary, Book1)$.

Fig. 9. Pattern of activation after two periods of oscillation (with reference to the state of activation in Fig. 7). This state represents the dynamic bindings: *(giver=John, recipient=Mary, give-object=Book1, owner=Mary, own-object=Book1, potential-seller=Mary, can-sell-object=Book1)*. The system has essentially inferred the facts *own(Mary, Book1)* and *can-sell(Mary, Book1)*.

Fig. 10. Encoding of a long-term fact. The interconnections shown here encode the *static* bindings *(giver=John, recipient=Mary, give-object=Book1)* that constitute the long-term fact *give(John,Mary,Book1)*. The pentagon shaped nodes are τ-and nodes. A τ-and node becomes active if it receives an uninterrupted input for the duration of the period of oscillation. The activation of *e:give* represents an externally or internally generated query asking whether the

dynamic bindings indicated by the pattern of activity of argument nodes match any long-term fact encoded in the LTKB. The activation of *c:give* represents an assertion by the system that these dynamic bindings match the knowledge encoded in the LTKB.

Fig. 11. Encoding of the partially instantiated long-term fact $give(John, Mary, x)$, i.e., 'John gave Mary something'. The input from $g\text{-}obj$ does not receive an inhibitory input from any filler.

Fig. 12. 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: $give(John,\ Mary,\ Book1)$, $buy(John,\ x)$, and $own(Mary,\ Book2)$. The links between arguments are in the reverse direction because the rules are wired for 'backward reasoning'.

Fig. 13. Activation trace for the query $can\text{-}sell(Mary, Book1)$? (Can Mary sell Book1?). The query is posed by providing an oscillatory input to *e:can-sell,* $Mary$, $Book1$, *p-seller* and $cs\text{-}obj$ as shown. The activation of *c:can-sell* indicates a yes answer.

Fig. 14. Encoding rules with existentially quantified variables and constants in the consequent. The network encodes the rule $\forall x1, x2, y\ [P(x1, x2) \Rightarrow \exists z\ Q(x1, x2, y, z, a)]$. This rule must not fire during the processing of a query, if either the existentially bound argument z gets bound, or the last argument gets bound to a constant other than a. The node $g1$ is a τ-or node. It projects inhibitory modifiers that block the firing of the rule if the above condition is violated.

Fig. 15. 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 occurring variable is unbound, or 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 τ-or node $g1$. The firing of $g1$ blocks the firing of the rule whenever the first and second arguments of Q get bound to different constants. (The encoding also enforces the constraint that last argument of Q should not be bound to any constant other than a.

Fig. 16. 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. Multiple antecedent rules are encoded using an additional τ-and node whose threshold equals the number of predicates in the antecedent. This node becomes active on receiving inputs from the *collector* nodes of all the antecedent predicates.

Fig. 17. Augmented representation of a long-term fact in order to support answer extraction. For each argument of the associated predicate there exists a ρ-btu node with a threshold of two. The node shown as a filled-in pentagon behaves like a τ-and node except that once activated, it stays active for some time – say about 20π – even after the inputs are withdrawn.

Fig. 18. Encoding a rule with repeated variables in the antecedent within a forward reasoning system. The figure shows the encoding of the rule $\forall x, y, z\ P(x, y) \wedge Q(y, z) \Rightarrow R(x, z)$. This rule should fire only if the two arguments in the antecedent corresponding to variable y get bound to the same constant. The τ-or node with a threshold of 2 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. These nodes have suitable thresholds.

Fig. 19. Interaction between a rule-based reasoner and an *IS-A* hierarchy. The rule component encodes the rule $\forall x, y\ preys\text{-}on(x, y) \Rightarrow scared\text{-}of(y, x)$ and the facts $\forall$*x:Cat, y:Bird preys-*

on(x,y) and $\exists$*x:Cat* $\forall$*y:Bird* *loves(x,y)*. The first fact is equivalent to *preys-on(Cat,Bird)* and states that 'cats prey on birds'. The second fact states that there is a cat that loves all birds'.

Fig. 20. Activation trace for the query $scared$-of$(Tweety, Sylvester)$?, i.e., 'Is Tweety scared of Sylvester?'

Fig. 21. Structure of the concept cluster for C and its interaction with the bottom-up and top-down switches. The cluster has three banks of nodes and is capable of storing upto 3 distinct instances of the concept (in other words, the multiple instantiation constant k_1 equals 3). The $\uparrow$ and $\downarrow$ relay nodes have a threshold of 2.

Fig. 22. Architecture of a switch that mediates the flow of activation into concept clusters. The depicted switch assumes that the associated cluster can represent upto 3 instances. The switch provides a built-in and distributed control mechanism for automatically allocating banks within a concept cluster. Each distinct incoming instantiation is directed to a distinct bank provided a bank is available.

Fig. 23. Encoding of the *IS-A* relation *is-a(A,B)*. A bundle of k_1 links is shown as a single link.

Fig. 24. Encoding rules with typed variables. The network fragment encodes the rule: $\forall x : animate, y : solid\text{-}obj \;\; walk\text{-}into(x, y) \Rightarrow hurt(x)$. The numbers associated with nodes denote thresholds (only thresholds other than 1 have been indicated explicitly). The τ-or node a (b) become active if and only if the first (second) argument node of $walk\text{-}into$ fires in synchrony with the concept *animate* (*solid-obj*). Once active, these nodes enable the propagation of binding to the predicate *hurt*. Thus type restrictions are enforced using temporal synchrony.

Fig. 25. The encoding of predicates for accommodating multiple instantiations. P and Q are binary predicates and R is a ternary predicate. The encoding assumes that any predicate may be instantiated at most three times (i.e., the multiple instantiation constant $k_2 = 3$). An n-ary predicate is represented by k_2 banks of nodes. The connections suggest that there are two rules, one of the form $P() \Rightarrow Q()$ and the other of the form $P() \Rightarrow R()$ (the argument correspondence is not shown). The connections between antecedent and consequent predicates of a rule are mediated by a 'switching' mechanism similar to the one described in Fig. 22. The switch for P automatically channels incoming instantiations of P into available banks of P. The switch has k_2 output 'cables' where each cable consists of output links to a predicate bank of P. The inputs to the switch are cables from banks of predicates that are in the consequent of rules in which P occurs in the antecedent.

Fig. 26. Individual ρ-btu nodes are replaced by an ensemble of such nodes. A connection between a pair of individual ρ-btu nodes is replaced by a number of random inter-ensemble connections. Nodes within an ensemble can communicate with their immediate neighbors in the ensemble and the intra-ensemble propagation delays are assumed to be much smaller than the inter-ensemble propagation delays.

Fig. 27. The time course of a node's threshold. After generating a spike a node enters an absolute refractory period (ARP). The ARP is followed by a relative refractory period (RRP). After the RRP a node's threshold reverts to its normal level. The distribution of the arrival times of signals from a connected ensemble is depicted by the shaded region. The noisy propagation delays are modeled as a Gaussian with mean d and standard deviation s.

Fig. 28. The cycle-by-cycle distribution of the firing times of nodes within a 'driven' ensemble being driven by a 'driver' ensemble whose nodes are firing in synchrony. The left hand 'wall'

of the isometric diagram displays the standard deviation and mean of the node firing times with reference to the ideal firing time. The nodes in the driven ensemble become synchronized in spite of noisy propagation delays. The *maximum* lag in the firing times of nodes in the 'driven' ensemble becomes less than 3 msec and the *mean* lag becomes less than 1 msec within 2 cycles. By the end of 7 cycles the maximum and mean lags reduce to 1 and 0.2 msec, respectively.