Appeared in *Applied Intelligence*, 11: 79-108, 1999

Advances in *Shruti* — A neurally motivated model of relational knowledge representation and rapid inference using temporal synchrony

Lokendra Shastri
International Computer Science Institute
1947 Center Street
Berkeley, CA 94704
shastri@icsi.berkeley.edu

Abstract

We are capable of drawing a variety of inferences effortlessly, spontaneously, and with remarkable efficiency — as though these inferences are a *reflex* response of our cognitive apparatus. This remarkable human ability poses a challenge for cognitive science and computational neuroscience: How can a network of slow neuron-like elements represent a large body of systematic knowledge and perform a wide range of inferences with such speed? The connectionist model *Shruti* attempts to address this challenge by demonstrating how a neurally plausible network can encode a large body of semantic and episodic facts, systematic rules, and knowledge about entities and types, and yet perform a wide range of explanatory and predictive inferences within a few hundred milliseconds. Relational structures (frames, schemas) are represented in *Shruti* by clusters of cells, and inference in *Shruti* corresponds to a transient propagation of rhythmic activity over such cell-clusters wherein *dynamic bindings* are represented by the synchronous firing of appropriate cells. *Shruti* encodes mappings across relational structures using high-efficacy links that enable the propagation of rhythmic activity, and it encodes items in long-term memory as coincidence and conincidence-error detector circuits that become active in response to the occurrence (or non-occurrence) of appropriate coincidences in the on going flux of rhythmic activity. Finally, "understanding" in *Shruti* corresponds to reverberant and coherent activity along closed loops of neural circuitry. Over the past several years, *Shruti* has undergone several enhancements that have augmented its expressiveness and inferential power. This paper describes some of these extensions that enable *Shruti* to (i) deal with negation and inconsistent beliefs, (ii) encode evidential rules and facts, (iii) perform inferences requiring the dynamic instantiation of entities, and (iv) seek coherent explanations of observations.

Keywords: knowledge representation; inference; evidential reasoning; dynamic binding; temporal synchrony.

1 Introduction

We are capable of drawing a variety of inferences effortlessly and with remarkable efficiency. To wit, our ability to understand language in real-time — a task that requires the hearer to draw a number of inferences in order to establish referential and causal coherence, generate expectations, make predictions, and recognize the speaker's intent.[1] Nevertheless we can understand language at the rate of *several hundred words per*

[1]Empirical data suggests that inferences required to establish referential and causal coherence occur rapidly and automatically during text understanding (see e.g., McKoon and Ratcliff 1980; McKoon and Ratcliff 1981; Keenan, Baillet, and Brown 1984). The evidence for the automatic occurrence of *elaborative* or predictive inferences however, is mixed (see e.g., Kintsch 1988; Potts, Keenan, and Golding 1988).

minute. This rapid rate of language understanding suggests that we are capable of performing a wide range of inferences rapidly, spontaneously and without conscious effort — as though they are a *reflex* response of our cognitive apparatus. In view of this, such reasoning may be described as *reflexive* reasoning (Shastri, 1993).

This remarkable human ability 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 wide range of inferences with such speed?

In 1989, Ajjanagadde and Shastri proposed a structured connectionist[2] model *Shruti*[3] which attempted to address this challenge and demonstrated how a network of neuron-like elements could encode a large body of structured knowledge and perform a variety of inferences within a few hundred milliseconds (Ajjanagadde and Shastri, 1989, 1991; Ajjanagadde, 1990; Shastri and Ajjanagadde, 1990a, 1990b, 1993; Shastri, 1992). D.R. Mani made several contributions to the model (Mani and Shastri, 1993), wrote the first *Shruti* simulator, and developed a parallel implementation of *Shruti* on the CM-5 (Mani, 1995).

Shruti shows that reflexive inference can be the spontaneous and natural outcome of a neurally plausible system. In *Shruti* there is no separate interpreter or inference mechanism that manipulates and rewrites symbols. The network encoding of commonsense knowledge is a vivid *model* of the agent's environment and 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, finds coherent explanations and makes predictions. In particular, *Shruti* suggested that

- the encoding of relational knowledge (frames, predicates, etc.) is mediated by neural circuits composed of *focal* cell-clusters,
- inference involving relational knowledge corresponds to a transient propagation of *rhythmic* activity across such focal-clusters,
- a *binding* between a conceptual role and an entity filling that role in a given situation is represented within this rhythmic activity by the *synchronous* firing of appropriate cells.[4]
- a systematic mapping between relational structures — and other rule-like knowledge — is encoded by high-efficacy links between focal clusters that enable the propagation of rhythmic activity, and
- a fact in long-term memory (LTM) is encoded as a temporal pattern-matching circuit that detects coincidences and coincidence failures in the ongoing flux of rhythmic activity.

The possible role of *synchronous* activity in neural representations and binding had been suggested by several researchers (e.g., Milner, 1974; von der Malsburg, 1981; Sejnowski, 1981; Abeles, 1982; Crick, 1984; Damasio, 1989), but *Shruti* offered a detailed computational account of how such activity can be harnessed to solve problems in the representation and processing of high-level conceptual knowledge and inference. While the role of synchronous activity in the brain remains a matter of debate and controversy, a rich body of neurophysiological data suggests that such activity occurs in the brain and might even play a role in neural information processing (e.g., see Eckhorn, Bauer, Jordan, Brosch, Kruse, Munk, and Reitbock, 1988; Gray, Konig, Engel, and Singer, 1989; Murthy and Fetz, 1992; Abeles, Bergman, Margalit, and Vaadia, 1993; Singer, 1993; Singer and Gray, 1995; deCharms and Merzenich, 1996; Usher and Donnelly,

[2] For an overview of the structured connectionist approach and its merits see (Feldman and Ballard, 1982; Feldman, 1989; Shastri, 1995).

[3] http://www.icsi.berkeley.edu/∼shastri/shruti/index.html

[4] Several other solutions to the binding problem within a connectionist framework have been proposed (e.g., Lange and Dyer, 1989; Smolensky, 1990; Hölldobler, 1990; Barnden and Srinivas, 1991; Sun, 1992). The advantages of the synchrony approach over some of these approaches are discussed in (Shastri, 1996).

1998). Over the past few years, several models that also use synchrony to encode dynamic bindings during inference have been proposed (e.g., Park, Robertson, and Stenning, 1995; Sougne, 1996; Hayward and Diederich, 1997; Hummel and Holyoak, 1997).[5]

Over the past five years, *Shruti* has been augmented in a number of ways in collaborative work between the author, his students, and other collaborators (see below). These enhancements enable *Shruti* to:

1. Encode negated facts and rules and deal with inconsistent beliefs (with D.J. Grannes) (Shastri and Grannes, 1996).
2. Seek coherent explanations for observations.
3. Encode soft/evidential rules and facts (with D.J. Grannes and C. Wendelken).
4. A novel representation of types, instances, and the subtype and supertype relations.
5. Deal with rules that require the dynamic instantiation of entities during backward reasoning (with D.J. Grannes and J. Hobbs).
6. Represent rules with multiple antecedents as well as multiple consequents (with D.J. Grannes, J. Hobbs, and C. Wendelken).
7. Encode attribute-values of entities and types (with D.J. Grannes).
8. Exhibit priming effects for entities, types, facts, and rules and competition between multiple entities (with D.J. Grannes, B. Thompson, and C. Wendelken).
9. Support context-sensitive unification of entities (with B. Thompson, and C. Wendelken).
10. Tune network weights and rule-strengths via supervised learning (with D.J. Grannes, B. Thompson, and C. Wendelken).
11. Realize control and coordination mechanisms required for encoding parameterized schemas and reactive plans (Shastri, Grannes, Narayanan, and Feldman, 1999).

All of the above enhancements have been incorporated into the *Shruti* simulator by D.J. Grannes, B. Thompson, and C. Wendelken. In addition to the developments mentioned above, V. Ajjanagadde has worked on the problem of abductive reasoning (Ajjanagadde, 1991, 1993) and also pursued an alternate set of representational mechanisms (Ajjanagadde, 1997).

As an illustration of *Shruti*'s inferential ability consider the following narrative:

John fell in the hallway. Tom had cleaned it. He got hurt.

Upon being presented with the above narrative[6] *Shruti* reflexively infers the following:

---

[5]A large number of computational models that use temporal synchrony to solve problems in perceptual processing have been developed (e.g., Strong and Whitehead, 1989; Wang, Buhmann, and von der Malsburg, 1990; Horn and Usher, 1991; Grossberg and Somers, 1992; Hummel and Biederman, 1992; Niebur, and Koch, 1994). In contrast, Bienenstock (1995) has proposed a general model of computation in the neocortex based on the propagation of synchronous activity.

[6]Each sentence in the narrative is conveyed to *Shruti* as a set of *dynamic bindings* (see Section 2.6). Thus, the three sentences in the narrative are conveyed as the dynamic bindings (i) *(⟨fall-patient=John⟩, ⟨fall-location=Hallway⟩)*, (ii) *(⟨clean-agent=Tom⟩, ⟨clean-location=an-inanimate-entity⟩)*, and (iii) *(⟨ hurt-=a-male-human⟩)*. The sentences are presented in the order of their occurrence in the narrative. After each sentence is presented, the network is allowed to propagate activity for a fixed number of cycles. Note that *Shruti* itself does not perform any syntactic processing — in its present form, it only models reflexive inference.

1. Tom had mopped the floor.
2. The floor was wet.
3. John was walking in the hallway.
4. John slipped and fell because the floor was wet.
5. John got hurt because he fell.

Notice that *Shruti* draws inferences required to establish referential and causal coherence. It explains John's fall by making the plausible inference that John was walking in the hallway and he slipped because the floor was wet. It also infers that John got hurt because of the fall. Moreover, it determines that "it" in the second sentence refers to the hallway, and that "He" in the third sentence refers to John, and not to Tom.

Shruti draws these inferences based on commonsense knowledge such as:

1. When one cleans a place, one may mop the floor of that place.
2. Mopping a floor makes the floor wet.
3. If one walks on a wet floor one might slip.
4. If one slips one may fall.
5. If one falls, one may get hurt
6. Relevant type information such as: John and Tom are human, John and Tom are male, hallway is a place.

The above knowledge includes systematic rules that can be viewed as a system's *causal model* of the world. An important feature of such knowledge is that it is instantiation-independent — it is not tied to specific entities, rather, it applies to a range of entities of certain types. In formal terms, the expression of such knowledge involves *types*, *variables*, and *quantifiers*. Moreover such knowledge is evidential in nature, and typically, express likely possibilities rather than certainties. As we shall see, *Shruti* can encode such general evidential rules as well as particular facts in its long-term memory (LTM), and reflexively activate them in response to inputs to drawn inferences.

Shruti identifies a number of constraints on the representation and processing of relational knowledge and predicts the capacity of the active (working) memory underlying reflexive reasoning (Shastri, 1992; Shastri and Ajjanagadde, 1993). These constraints shed light on differences between reflexive and reflective processing. First, on the basis of neurophysiological data pertaining to the occurrence of synchronous activity in the γ band, *Shruti* predicts that a large number of facts (relational instances) can be active simultaneously and a large number of rules can fire in parallel during an episode of reflexive reasoning. However, *the number of distinct entities participating as role-fillers in these active facts and rules must remain very small* (see Section 5). Recent experimental findings as well as computational models lend support to this prediction (e.g., Lisman and Idiart, 1995; Jensen and Lisman, 1996; Luck and Vogel, 1997). Second, since the quality of synchronization degrades as activity propagates along a chain of cell clusters, *Shruti* predicts that as the depth of inference increases, binding information is gradually lost and systematic inference reduces to a mere spreading of activation. Thus *Shruti* predicts that reflexive reasoning has a limited inferential horizon. Third, *Shruti* predicts that only a small number of instances of any given relation can be active simultaneously.

The representational and inferential machinery of *Shruti* can be applied to other problems involving relational structures, systematic and context-sensitive mappings between such structures, and rapid interactions between persistent and dynamic structures. Some examples of such problems are parsing, the dynamic linking of syntactic and semantic structures during language processing, and model-based visual object recognition. In particular, Henderson (1994) has shown that constraints predicted by *Shruti* explain several properties of human syntactic processing including garden path effects and our limited ability to deal with center-embedded sentences (The cat that the dog chased climbed up a tree). The work on *Shruti* meshes well with the NTL project (Bailey, Chang, Feldman, and Narayanan, 1998) on language acquisition and provides neurally plausible solutions to several representational and computational requirements arising in the project.

Shruti has also been coupled with a *metacognitive* component (Cohen, Freeman, and Wolf, 1996) capable of attention shifting, making and testing assumptions, identifying conflicts, and evaluating uncertainty. Such a two-tier reflexive/metacognitive system has been used to model certain aspects of tactical decision making under stress.

This paper describes some of the enhancements to *Shruti* between circa 1993 and 1997. Section 2 discusses the basic elements of *Shruti*'s representational machinery. This include focal-clusters for the representation of generic relations, entities, and types, expression of dynamic bindings via temporal synchrony, the encoding of long-term facts, the representation of systematic mappings (or rules) between relational structures, and the organization of concepts and instances into hierarchical structures. Section 3 illustrates the functioning of *Shruti* with the help of a simple example of inference, and Section 4 discusses the encoding of more complex rules. The constraints on the reflexive processing of relational knowledge are discussed in Section 5 and Section 6 illustrates how conflicting information is handled with the help of a simple example. Section 7 provides pointers to work on the mapping of *Shruti* onto parallel hardware.

2 Representational machinery of *Shruti*

A description of *Shruti* requires the specification of its *structure* as well as a description of its *dynamic* behavior. All long-term (persistent) knowledge is encoded in *Shruti* via structured networks of nodes and links. Such long-term knowledge includes generic relations, instances, types, general rules, and specific facts. The dynamic aspects of *Shruti* involve the encoding of dynamic bindings and active facts via synchronous activity, the propagation of bindings by the propagation of synchronous activity along linked nodes, the activation of long-term facts in response to dynamic bindings, evidence combination via summation of activation, competition via inhibition, the development of coherence via reverberant activity along closed loops, and priming.

We begin our description of *Shruti* with an overview of its representational structure. Figure 1 illustrates some of the key elements of *Shruti*'s representational machinery. These elements are discussed in detail later, but a cursory examination of the network depicted in Figure 1 in conjunction with the following description of the encoded knowledge should provide the reader with an overview of *Shruti*'s structure and aid in understanding the details that follow.

The network fragment shown in Figure 1 depicts a partial encoding of the following rules, facts, instances, and types:

1. $\forall$*(x:Agent y:Agent z:Thing) give(x,y,z)* $\Rightarrow$ *own(y,z) [800,800];*
2. $\forall$*(x:Agent y:Thing) buy(x,y)* $\Rightarrow$ *own(x,y) [900,980];*
3. EF: *give(John, Mary, Book-17) [1000];*
4. TF: $\forall$*(x:Human y:Book) buy(x,y) [50];*

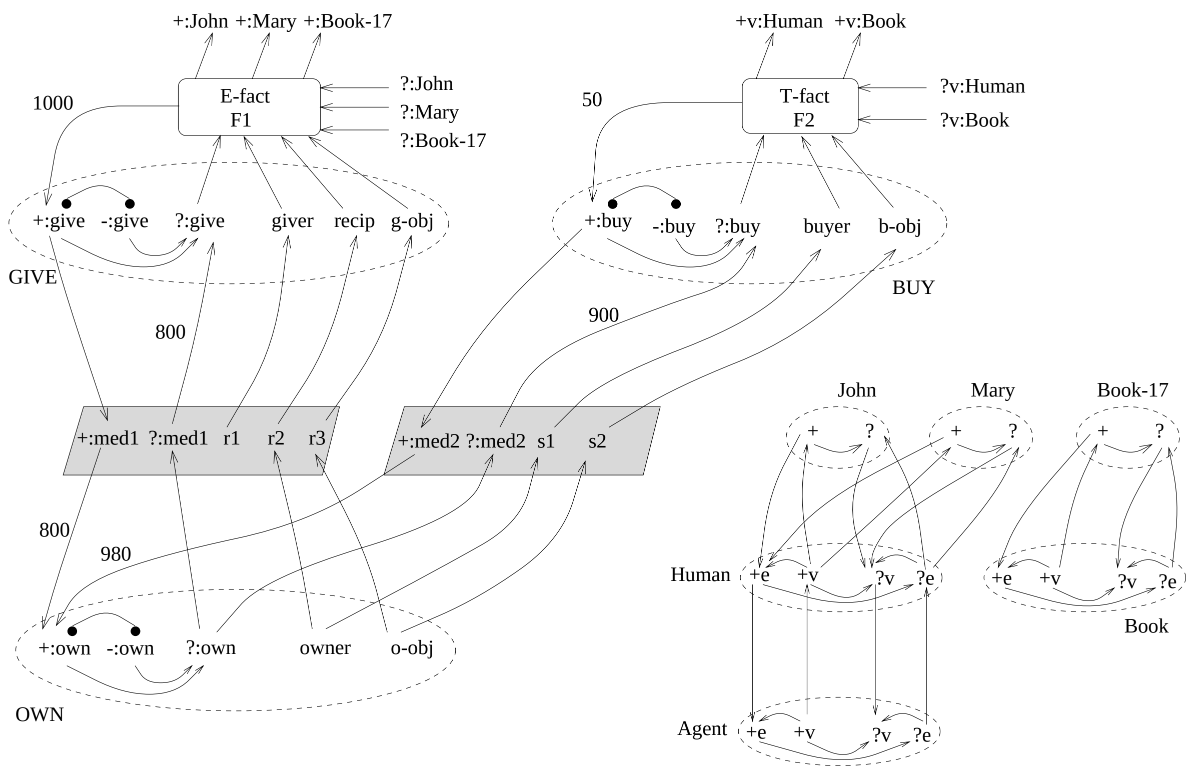

Figure 1: An overview of the representational machinery of *Shruti*. See text for details.

5. *is-a(John, Human);*
6. *is-a(Mary, Human);*
7. *is-a(Human, Agent);*
8. *is-a(Book-17, Book);*

Item (1) is a rule which captures a systematic relationship between giving and owning. It states that when an entity x of type *Agent*, gives entity z of type *Thing*, to an entity y of type *Agent*, then the latter comes to own it. Similarly, item (2) is a rule which states that whenever any entity of the type *Agent* buys something, it comes to own it. The pair of weights [a,b] associated with a rule have the following interpretation: a indicates the degree of evidential support for the antecedent being the probable cause (or explanation) of the consequent, and b indicates the degree of evidential support for the consequent being a probable effect of the antecedent.[7] Item (3) corresponds to a long-term "episodic" fact (or E-fact) which states that John gave Mary a specific book (Book-17). Item (4) is a long-term "taxon" fact (or T-fact) which states that the prior evidential support for a given (random) human buying a given (random) book is 50. Item (5) states that John is a human. Similarly, items (6–8).

Given the above knowledge, *Shruti* can rapidly draw inferences of the following sort within a few hundred milliseconds[8] (numbers in *[]* indicate strength of inference):

1. *own(Mary, Book-17) [784]*; Mary, owns a particular book (referred to as Book17).
2. ∃*x:Book own(Mary,x) [784]*; Mary owns a book.
3. ∃*(x:Agent y:Mary z:Book) give(x,y,z) [980]*; Some agent gave Mary a book.
4. ∃*(x:Agent y:Thing) own(x,y) [784]*; Some agent owns something.
5. *buy(Mary,Book-1) [41]*; Mary bought a particular book (referred to as Book-1).
6. *is-a(John, Agent)*; John is an agent.
7. *is-a(Mary, Agent)*; Mary is an agent.

We now examine some of the components of *Shruti*'s representational machinery in more detail.

2.1 Different node types and their computational behavior

Nodes are computational abstractions and correspond to *small ensembles of cells*. Moreover, a connection from a node A to a node B corresponds to several connections from cells in the A ensemble to cells in the B ensemble.

Shruti makes use of four node types: m-ρ-nodes, τ-and nodes, τ-or nodes (type 1) and τ-or nodes (type 2). This classification is based on the *computational* properties of nodes, and not on their functional or representational role. As we shall see, nodes serving different representational functions will often be of the same computational type. The computational behavior of the four node types are as follows:

[7] Weights in *Shruti* lie in the interval [0,1000]. The mapping of probabilities and evidential supports to weights in *Shruti* is non-linear and loosely defined. The initial weights can be set approximately, and subsequently fine tuned to model a given domain via supervised learning using a gradient-descent algorithm.

[8] The time required for drawing an inference is estimated by $c * \pi$, where c is the number of cycles of rhythmic activity it takes *Shruti* to draw an inference (see Section 3), and π is the period of rhythmicity. A plausible value of π is 25 milliseconds (see Section 5). These times do not take into account the time that would be taken up by perceptual, linguistic, and motor processes to process and respond to inputs.

m-ρ nodes: An m-ρ node becomes active and fires upon receiving above-threshold *synchronous* inputs. Here synchrony is defined relative to a window of temporal integration ω. Thus all inputs arriving at a node with a lead/lag of no more than ω are deemed to be synchronous. In particular, an m-ρ node A receiving an above-threshold periodic input from an m-ρ node B produces a periodic spike-train that is *in-phase* with B. Unlike the ρ-btu nodes introduced in (Shastri and Ajjanagadde, 1993) which fired in at most one phase per cycle, an m-ρ node can fire in multiple phases within the same period. Thus an m-ρ node A receiving above-threshold periodic inputs from m-ρ nodes B and C (where B and C may be firing in different phases) will respond by firing in phase with both B and C. The generalization of ρ-btu nodes to m-ρ was suggested by Strong (1993) in his commentary on (Shastri and Ajjanagadde, 1993) and has also been advocated by (Park, Robertson, and Stenning, 1995). Recall that each node corresponds to a cluster of cells. Hence the firing of node A in multiple phases means that some of the cells in A's cluster are firing in phase with B while some other cells in its cluster are firing in phase with C.

A scalar level (or strength of activity) is associated with the response of an m-ρ node.[9] This strength of activity is computed by the *activation combination function* (ECF) associated with the node. Some ECFs used in the past are *sum*, *max*, and *sigmoid*. For example, an m-ρ node with sum as its ECF computes the weighted sum of inputs arriving in each phase i, and if this sum exceeds the threshold, the node produces a response in phase i equal in strength to the weighted sum of inputs.

τ-and nodes: A τ-and node becomes active on receiving an uninterrupted and above-threshold input over an interval $\geq \pi_{max}$, where π_{max} is a system parameter. Computationally, this sort of input can be idealized as a pulse whose amplitude exceeds the threshold, and whose duration is greater than or equal to π_{max}. Physiologically, such an input may be identified with a high-frequency burst of spikes. Thus a τ-and node behaves like a *temporal and* node and becomes active upon receiving adequate and uninterrupted inputs over an interval π_{max}. Upon becoming active, such a node produces an output pulse of width $\geq \pi_{max}$. The level of output activation is determined by the ECF associated with the node for combining the weighted inputs arriving at the node. Some of the functions used in the past are: max, min, *sigmoid*, and *average*. More flexible combination functions are under investigation (Shastri and Wendelken, 1998; Wendelken and Shastri, in preparation).

τ-or node (type 1): A τ-or node (type 1) becomes active on receiving above-threshold *synchronous* inputs. Upon becoming active such a node produces an output pulse of width π_{max}. Thus a τ-or node (type 1) behaves like a *temporal or* node.

τ-or node (type 2): A τ-or node (type 2) becomes active on receiving inputs in two or more distinct phases during an interval π_{max}. Upon becoming active, such a node produces an output pulse of width π_{max}. Here a phase refers to any temporal interval of width ω.

The model also makes use of *inhibitory modifiers* that can block the flow of activation along a link — a spike propagating along an inhibitory modifier will block a synchronous pulse propagating along a link that the modifier impinges upon.

2.2 Focal-clusters and the Encoding of Relational Structures

Each generic relation (in general, a frame, a schema, or a predicate) is represented with the help of a *focal-cluster* depicted by a dotted ellipse. Such a focal-cluster for the relation *give* is depicted in Figure 2. Note that each *label* within the focal-cluster (e.g., *+:give*) denotes a node.

For the purpose of this example, it is assumed that *give* has three roles: *giver*, *recipient* and *give-object*. Each of these roles is encoded by a separate node labeled *giver*, *recip* and *g-obj*, respectively. The focal-cluster of *give* also includes an *enabler* node labeled *?:give* and two *collector* nodes labeled *+:give* and *–:give*. In general, the focal-cluster for an *n-place* generic relation contains n role nodes, one enabler (*?*)

[9] The response-level of a m-ρ node in a phase can be governed by the number of cells in the node's cluster firing in that phase.

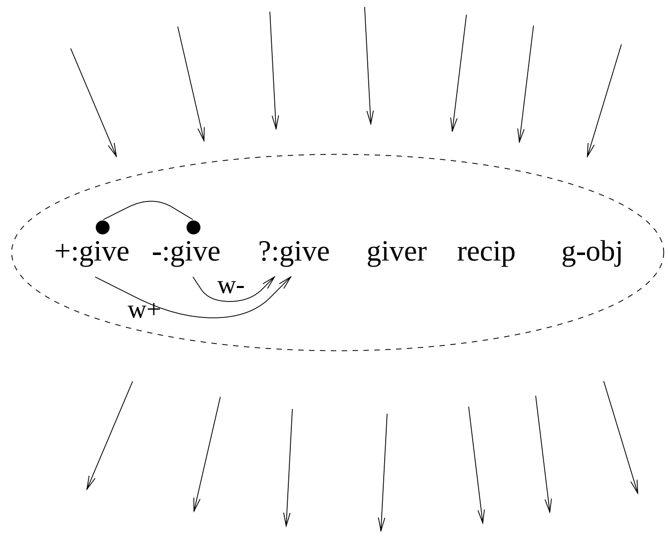

Figure 2: An idealized depiction of the focal-cluster for the generic relation *give*. Each label corresponds to a node and consists of a small but physically dispersed ensemble of cells. In the illustration, the relation *give* is assumed to have three roles: *giver*, *recipient* and *give-object*. Each of these roles is encoded by a distinct node labeled *giver*, *recip* and *g-obj*, respectively. The focal-cluster also includes an *enabler* node *?:give*, and two *collector* nodes *+:give* and *–:give*. See text for details.

node, one positive collector (+) node, and one negative collector (–) node.

As explained below, the *enabler* node "enables" the search for an explanation for the currently active instance of the generic relation, and the + and – *collector* nodes serve to "collect" evidence for and against, the currently active instance of the generic relation. Hence, the choice of terminology. The computational type of collector and enabler nodes is τ-and, and the computational type of role nodes is m-ρ.

The focal-cluster associated with a generic relation acts as an anchor for the complete encoding of a generic relation. All information pertaining to a relation converges on its focal-cluster, and all such information can be accessed by fanning out from this focal-cluster.

2.2.1 Semantic import of enabler and collector nodes

The enabler and collector nodes have the following significance: Assume that the roles of a generic relation *give* have been *dynamically bound* to some fillers, and thereby represent an active instance of *P* (as we will see, shortly, a dynamic binding between a role and an entity is expressed via the synchronous activity of role and entity nodes). The activation of the enabler *?:P* means that the system is seeking an explanation for (or trying to find support for) the currently active instance of *P*. The request for such an explanation might be generated internally by the reasoning system, or be communicated to it by some other subsystem (e.g., the planning module). In contrast, the activation of *+:P* means that the system is affirming the currently active instance of *P*. Such an affirmation could be the result of retrieval, inference, or some perceptual or linguistic process. Similarly, the activation of *–:P* means that the system is affirming the negation of the currently active instance of *P*.

The *level* of activation of *?:P* signifies the strength with which information about *P* is being sought. Similarly, the *level* of activation of *+:P* (*–:P*) signifies the degree of belief in the truth (falsity) of the currently active instance of *P*.

For example, if the roles *giver*, *recipient* and *object* are dynamically bound to John, Mary, and a-book, respectively, then the activation of *?:give* means that the system is asking whether the fact "John gave Mary a book" matches one of the events in the system's memory, or whether it can be inferred from what is known to the system. In contrast, the activation of *+:P* with the same role bindings means that the system is asserting the event "John gave Mary a book". If *–:P* is active instead of *+:P* it means that the system is asserting the negation of the event "John did give Mary a book" (i.e., it is asserting "It is not true that John gave Mary a book").

A related scheme for dealing with positive and negative propositions (i.e., unary relations) was described in (Cottrell, 1989), and an extension of this scheme to n-ary relations was suggested in (Cottrell, 1993). The suggested scheme required two focal-clusters for each relation *P* — one for positive knowledge about *P* (*+:P*), and another for negative knowledge about *P* (*–:P*). It also required a mechanism for comparing bindings across the focal-clusters *+:P* and *–:P* and ensuring that the same set of bindings were expressed in both the clusters. *Shruti*'s use of + and – collectors within a single focal-cluster, however, simplifies the representation of negative relational instances and makes it easy to detect contradictions.

2.2.2 Degrees of Belief: support, no information, and contradiction

The levels of activation of the + and – collectors of a generic relation measure the effective degree of support offered by active knowledge structures to the currently active instances of the relation. These levels of activation are the result of the activation incident on the collectors from the rest of the network and the mutual inhibition between the two collectors. The two activation levels encode a graded belief ranging continuously from "no" at one extreme (where only the – collector is active), to "yes" at the other extreme (where only the + collector is active), with "don't know" in between (where neither collector is very active). If both the collectors receive comparable and strong activation then both collectors can be in a high state of activity, in spite of the mutual inhibition between them. When this happens, a contradiction is detected. In the current implementation this is done by an additional node within each predicate cluster (not shown in Figure 2) that receives excitatory inputs from both the collectors.

2.2.3 Significance of collector to enabler connections

The links from the collectors to the enabler of a generic relation (see Figure 2) convert a dynamic assertion of a relational instance into a query about the assertion. This means that the system is always seeking support (or an explanation) for active assertions. It is as though the system constantly "evaluates" incoming knowledge in the context of existing knowledge and seeks to affirm or reject it based on what it knows.

The weights $w+$ and $w-$ on links from the + and the – collectors to enablers, respectively, consist of two components. One of the components is a measure of the system's overall propensity for seeking explanations. A system with a high value for this component can be viewed as being highly skeptical, while one with a low value can be viewed as being highly credulous. The other component is relation specific and is inversely related to the probability of occurrence associated with the positive and negative instance of this relation. Thus the more likely a fact, the more intense the search for its explanation. In view of this, $w+$ and $w-$ may be expressed as follows:

$$w+ = \frac{\alpha * \text{system's propensity for seeking explanations}}{\beta * \text{probability of occurrence of +P}}$$

$$w- = \frac{\alpha * \text{system's propensity for seeking explanations}}{\beta * \text{probability of occurrence of –P}}$$

where α and β are scalar values.

The ability of a system to evaluate incoming information also gives it the ability to detect inconsistencies between incoming information and prior knowledge. The scope of inconsistency detection is however, *local*. In other words, inconsistencies are detected only between two facts that are within a limited inferential distance from each other. This bound on inferential distance is governed by the constraint on the depth of inference discussed in Section 5. Observe that we are referring to a reflexive process of evaluation and not a deliberate search for contradictions. Thus while this process of detecting inconsistencies would be very fast and automatic, it would be subject to the constraints on reflexive reasoning which include bounds on the depth of inference.

2.2.4 Collector to enabler connections lead to reverberant activity

The links from the collectors of a generic relation to its enabler also create positive feedback loops of activation and thereby create stable coalitions of active cells under appropriate circumstances. Assume that the system is seeking an explanation about the currently active instance of *P*, and therefore, *?:P* is active. If the cognitive apparatus finds supports for this instance of *P* it will activate *+:P*. This will create a feedback loop — or a stable coalition — consisting of *?:P*, other ensembles participating in the explanation, *+:P* and finally *?:P*. Similarly, if the cognitive apparatus finds supports for the negation of this instance of *P* it will activate *–:P*. This will also create a feedback loop consisting of *?:P*, other ensembles participating in the explanation, *–:P* and finally *?:P*.

Within the *Shruti* framework, such reverberant activity is the network correlate of successful "retrieval" and "understanding." Moreover, such activity also leads to priming (see Section 2.5), and possibly, to the formation of episodic memories (see Shastri, 1997b, 1998).

2.3 Focal-clusters for Encoding Types and Instances

The encoding of types and instances, and the instance-of, superordinate, and subordinate relations is depicted in Figure 3 and described in brief below. A more detailed treatment appears in (Shastri, 1999b). The focal-cluster of an entity *I* consists of two nodes, *?:I* and *+:I*. In contrast, the focal-cluster of a type T consists of a pair of "*?*" nodes (*?e:T* and *?v:T*) and a pair of "+" nodes (*+e:T* and *+v:T*). All these nodes are of type m-ρ.

While the nodes *+v:T* and *?v:T* participate in the expression of knowledge (facts and attributes) involving the whole type *T*, the nodes *+e:T* and *?e:T* participate in the encoding of knowledge involving particular instances of type *T*. Thus the pair of *v* nodes and the pair of *e* nodes signify universal and existential quantification, respectively.

The *levels* of activation of *?:I*, *?v:T*, and *?e:T* nodes signify the strength with which information about entity *I*, type *T*, and an instance of type *T*, respectively, is being sought. Similarly, the *levels* of activation of *+:I*, *+v:T*, and *+e:T* signify the degree of belief that the entity *I*, the type *T*, and an instance of type *T*, respectively, play appropriate roles in the situation currently encoded by the system's state of activity.

As explained in Section 2.7, the closure between the "*?*" and "+" nodes is provided by facts.

Interconnections between nodes *within* the focal-cluster of an instance and a type lead to the following desirable functionality (in the following, *I* refers to an instance, *T* refers to a type): (i) The link from *+:I* to *?:I* causes an assertion about an instance to lead to a search for a possible explanation of the assertion. Similarly, the link from *+v:T* to *?v:T* causes an assertion about the whole type *T* to lead to a search for a possible explanation of the assertion. (ii) The link from *+v:T* to *+e:T* causes an assertion about the type to lead to the same assertion about an unspecified member of the type (e.g., "Humans are mortal" leads to "there exists a mortal human").[10] (iii) The link from *+e:T* to *?e:T* causes an assertion about an instance of the type to lead to a search for a possible instance that would verify the assertion (e.g., the assertion "A

[10] *Shruti* infers the existence of a mortal human given that all humans are mortal, though this is not entailed in classical logic.

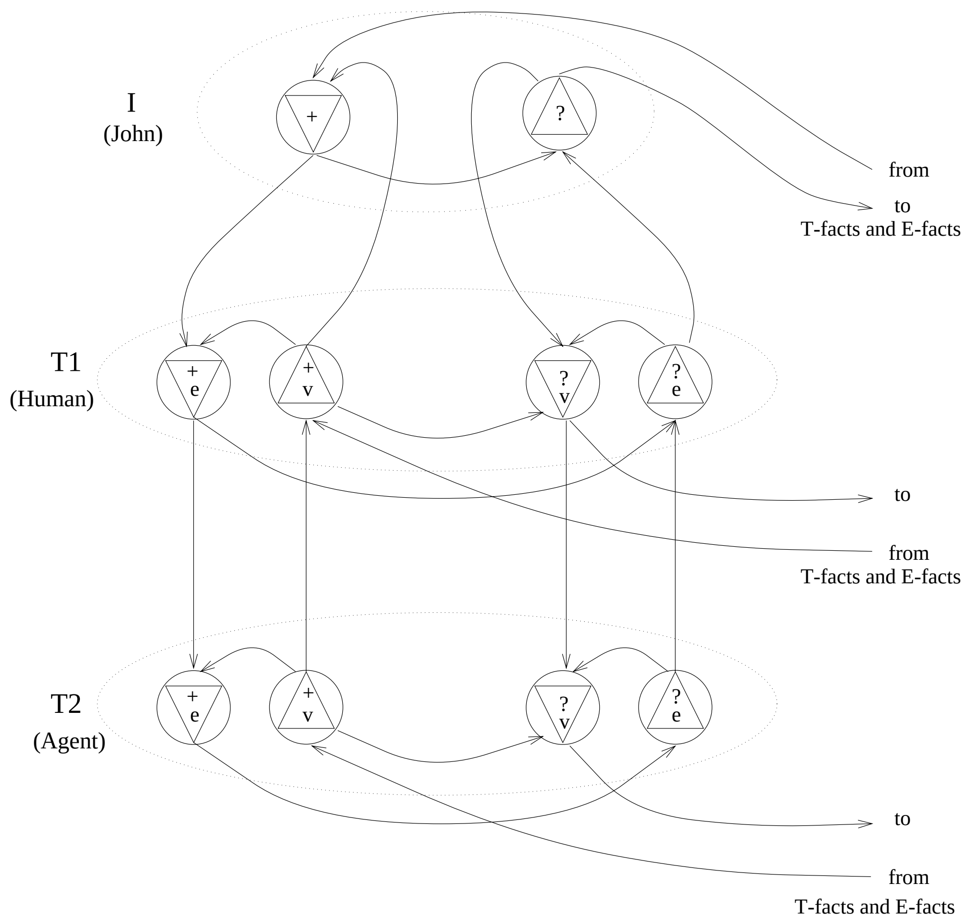

Figure 3: The encoding of types and (specific) instances. *I* is an instance of type *T1* and *T1* is a subtype of type *T2*. For example, *I*, *T1* and *T2* may be John, Human, and Agent, respectively. Each instance is expressed using a collector (+) and an enabler (?) node. Each type is expressed using a pair of collector nodes and a pair of enabler nodes. The nodes labeled "*+v*" and "*?v*" participate in the encoding of facts (and attributes) involving the whole type. The nodes labeled "*+e*" and "*?e*" participate in the encoding of facts and attributes involving an unspecified instance of the type. Here "*v*" and "*e*" may be interpreted as universal and existential quantifiers, respectively.

human is mortal" to the query "Is there is a human who is mortal?") (iv) The link from *?e:T* to *?v:T* causes a query (i.e., a search for an explanation) about a member of the type to lead to a query about the whole type (one way of determining whether "A human is mortal" is to find out whether "Humans are mortal"). (iv) Pathways formed by the above links lead to other behaviors. For example, given the path from *+v:T* to *?e:T*, any assertion about the whole type leads to a query or search for an explanation of the assertion applied to a given subtype/member of the type (e.g., "Humans are mortal" leads to the query "Is there a human who is mortal?").

2.3.1 Interconnection among focal-clusters of instances and types

Interconnections between nodes of the focal-cluster of an instance and that of a type lead to the following functionality:

- Because of the link from *+v:T1* to *+:I*, any assertion about the type *T1* leads to the same assertion about the instance *I* ("Humans are mortal" leads to "John is mortal").
- Because of the link from *+:I* to *+e:T1*, any assertion about *I* leads to the same assertion about a member of *T1* ("John is mortal" leads to "A human is mortal").
- Because of the link from *?:I* to *?v:T1*, a query about *I* leads to a query about the whole type *T1* (one way of determining whether "John is mortal" is to determine whether "Humans are mortal").
- Because of the link from *?e:T1* to *?:I*, a query about a member of *T1* leads to a query about *I* (one way of determining whether "A human is mortal" is to determine whether "John is mortal").

2.3.2 Interconnection between focal-clusters of types

Interconnections between the focal-clusters of sub- and supertypes lead to the following functionality.

- Because of the link from *+v:T2* to *+v:T1*, any assertion about the supertype *T2* leads to the same assertion about the subtype *T1* ("Agents can cause change" leads to "Humans can cause change").
- Because of the link from *+e:T1* to *+e:T2*, any assertion about a member of *T1* leads to the same assertion about a member of *T2* ("a mortal human exists" leads to "a mortal agent exists").
- Because of the link from *?v:T1* to *?v:T2*, a query about the whole type *T1* leads to a query about the whole type *T2* (one way of determining whether "Humans are mortal" is to determine whether "Agents are mortal").
- Because of the link from *?e:T2* to *?e:T1*, a query about a member of *T2* leads to a query about a member of *T1* (one way of determining whether "an agent is mortal" is to determine whether "a human is mortal").

Thus the interaction between the links within the encoding of a type and instance and the links between the encodings of different types and instances leads to a desirable set of type inferences.

2.4 Mutual exclusion and collapsing of phases in the type hierarchy

Instances in the type hierarchy can be part of one or more "phase-level mutual exclusion clusters" (ρ-mex clusters). Members of a ρ-mex cluster inhibit one another from firing in close temporal proximity of one another, and hence, only the member with the highest activation fires in a sustained manner in a given phase.

In other words, a ρ-mex cluster behaves like a phase-sensitive winner-take-all cluster. A similar ρ-mex can be formed by mutually exclusive types.

If the + node of an entity *I* is firing in multiple phases i and j, and if the + node of no other entity besides *I* is firing in phase j, then *Shruti* can collapse phases i and j. This corresponds to the synchronization of cells in the cluster *+:I*.

2.5 Priming

The type hierarchy also supports associative priming of links between enabler nodes. Such priming facilitates the reactivation of a type (or entity) as an answer to a query, if the type (or entity) had recently been activated as an answer to another query. With reference to Figure 3:

- The weight of the link from *?e:T2* node to *?e:T1* node will be primed (increased) if node *?e:T1* receives concurrent activity from the *+e:T1* node while it is receiving activity along the link from node *?e:T2*.
- The weight of the link from *?v:T1* node to *?v:T2* node will be primed if node *?v:T2* receives concurrent activity from the *+v:T2* node while it is receiving activity along the link from node *?v:T1*.
- The weight of the link from *?e:T1* node to *?:I* node would be increased (primed) if node *?:I* receives concurrent activity from the *+:I* node while it is receiving activity along the link from node *?e:T1*.
- The weight of the link from *?:I* node to *?v:T1* node will be primed if node *?v:T1* receives concurrent activity from the *+v:T1* node while it is receiving activity along the link from node *?:I*.

The priming can affect the response time as well as the answer found by the network. Since a primed entity becomes active sooner than an unprimed entity, a query whose answer involves primed entities is answered faster than a query whose response involves only unprimed entities (all else being equal). Furthermore, all else being equal, a primed entity would dominate an unprimed entity in a winner-take-all competition, and hence, in any situation wherein a primed and an unprimed entity compete to be a filler of a role the primed entity wins and emerges as the filler of the role.

Analogous priming of links also occurs for links connecting the enabler nodes of the consequents of a rule to the enabler nodes of the antecedents of the rule (via the mediator). This sort of priming is also selective in nature and occurs only if the enabler node at the destination of such a link receives concurrent activation from its + or – collector. Note that this sort of priming raises the link weights along the complete set of rules and facts involved in an explanation and makes it possible to rapidly recreate an explanation that has recently been computed by the network.

2.6 Encoding of dynamic bindings

The *dynamic* (or active) encoding of a relational instance corresponds to a *rhythmic* pattern of activity wherein bindings between roles and entities are represented by the *synchronous* firing of appropriate role and entity nodes. With reference to Figure 2, the rhythmic pattern of activity shown in Figure 4 is the dynamic representation of the relational instance *(give:* $\langle$ *giver=John* $\rangle$, $\langle$ *recipient=Mary* $\rangle$, $\langle$ *give-object=a-Book* $\rangle$*)* (i.e., the event, "John gave Mary a book"). Observe that the collector ensembles *+:John*, *+:Mary* and *+:a-Book* are firing in distinct phases. But *+:John* and the role *giver* are firing in synchrony, *+:Mary* and the role *recip* are firing in synchrony, and *+:a-Book* and the role *g-obj* are firing in synchrony. Since *+:give* is also firing, the system is essentially making an assertion. Note that as a result of the connections between the collector and enabler ensembles, the enabler ensembles *?:give*, *?:John*, *?:Mary*, and *?:a-Book* also start

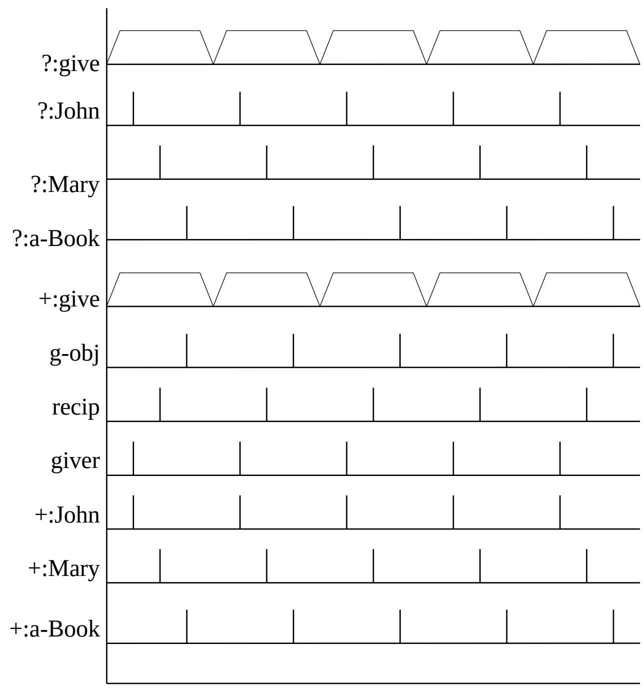

Figure 4: The rhythmic pattern of activation representing the dynamic bindings *give(John,Mary,a-Book)*. Bindings are expressed by the synchronous activation of bound role and entity nodes. Each spike in the illustration signifies the synchronous firing of cells in the ensemble of the appropriate node.

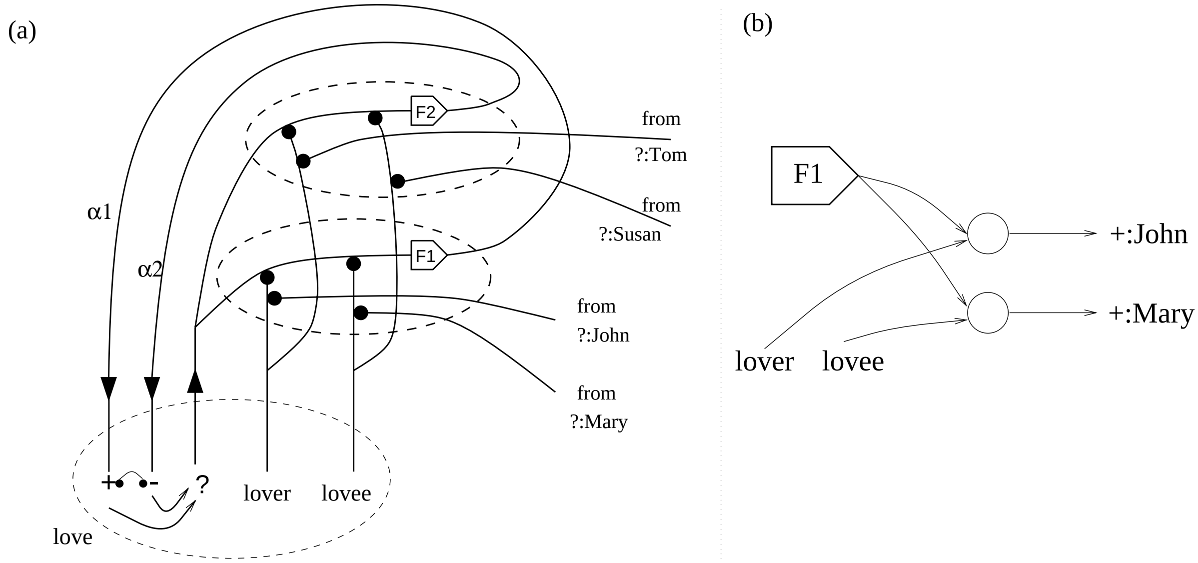

Figure 5: (**a**) The encoding of E-facts: *love(John,Mary)* and ¬*love(Tom,Susan)*. The pentagon shaped node serves as a "fact node" and is of type τ-and. The filled blobs impinging on links denote inhibitory modifiers. The firing of a role node without the *synchronous* firing of the associated filler node blocks the activation arriving at the fact node. In other words, if the role is bound in the dynamic instance, it must be bound to the same entity to which it is bound in the E-fact. Otherwise the E-fact will be blocked. (**b**) Links from the fact node back to collectors of role-fillers are shown for the fact *love(John,Mary)* to avoid clutter. Similar linkage exists between F2, *lover* and *lovee* and +*:Tom* and +*:Susan*. The circular nodes are m-ρ nodes with a high threshold which is satisfied only when both the role node and the fact node are firing. Consequently, a binder node fires in phase with the associated role node, if the fact node is firing. Weights such as $\alpha 1$ and $\alpha 2$ indicate the strength of belief in the respective facts.

firing (this is depicted in Figure 4). Activation also spreads in the type hierarchy, and we will discuss this in more detail in Section 6.

The dynamic representation of the query "Did John give Mary a book" would be similar except that only the role nodes *giver*, *recipient*, and *give-object*, the enabler nodes *?:give*, *?:John*, *?:Mary*, and *?:a-Book* would be active; the collector nodes would remain inactive.

The rhythmic activity underlying the dynamic representation of relational instances is expected to be highly variable and dynamic, but it is assumed that over short durations — ranging from a few hundred milliseconds to about a second — such activity may be viewed as being composed of k interleaved quasi-periodic activities where k equals the number of distinct entities that fill roles in the relational instance being processed. The period of this transient activity, π, is at least $k * \omega_{int}$ where ω_{int} is the *window of synchrony*, i.e., the amount by which two spikes can lead/lag and still be treated as being synchronous by appropriate neural circuits. As speculated in (Shastri and Ajjanagadde, 1993), the activity of role and entity cells engaged in dynamic bindings might correspond to γ band activity.

2.7 Encoding of facts and distributions in LTM: E-facts and T-facts

Currently *Shruti* encodes two types of relational "instances" (i.e., facts) in its LTM. These are *episodic* facts (E-Facts) and *taxon* facts (T-facts). In general, a large number of E-facts and T-facts can be associated with any relation.

E-facts correspond to "facts" in the usual sense of the word. They are specific instances of a generic relation (e.g., "John gave Mary a book on Tuesday"). In contrast, a T-fact corresponds to a distillation or statistical summary of various instances of the generic relation observed by the agent and can be viewed as coding the *prior support*[11] for a situation, conditioned by the type of *role-fillers* involved in the situation. Thus the T-fact ∀*(x:Human, y:Book) own(x,y) [50]* states that the prior support for the event "a human bought a book" is 50 and the T-fact ∀*(x:Californian, y:Car) buy(x,y) [950]* states that the prior support for the event "a Californian bought a car" is 950. Taken together, the T-facts associated with a relation convey distributional information about the instances of a relation conditioned by the types of role-fillers.

In a sense, while E-facts are analogous to episodic memories (Tulving, 1983), T-facts are analogous to associative memories and subsume certain kinds of semantic memory. The choice of terminology is motivated by the episodic memory versus taxon memory distinction suggested by O'Keefe and Nadel (1978).[12]

In general, E-facts and T-facts can have the form:

$\exists x_1{:}X_1, \ldots, x_r{:}X_r \; \forall y_1{:}Y_1, \ldots, y_s{:}Y_s \; (+/-)P(\ldots) \; [\alpha]$

where arguments of P are either entities or variables x_i and y_i. Universally quantified variables are assumed to be distinct. The bindings specified in a fact can be specific entities or entities constrained to be of a certain type.

Note that in addition to encoding fully instantiated relational instances such as "John gave Mary a book in the library on Tuesday", an E-fact can also encode partially instantiated and quantified assertions such as "John gave someone a book", "All permanent employees receive a bonus" and "there is an employee who is the manager of all employees". Moreover, E-facts can also be associated with a strength α which measures the agent's belief in the fact.

Though E-facts and T-facts share the same surface form, they have different semantic imports, and this is reflected in their responses to ongoing activity. An E-fact E_1 associated with a generic relation P becomes active whenever all the dynamic bindings specified in an active instance of P match those encoded in E_1. Thus an E-fact is sensitive to mismatches between the bindings it encodes and active dynamic bindings. In contrast, a T-fact associated with P is only sensitive to similarities between its bindings and those specified by active instance of P. Thus a T-fact associated with P ignores binding errors and produces a response that is proportional to the number of matches between its bindings and those of an active instance of P.

The distinction between E-facts and T-facts is motivated in part by the distinct circuitry required to memorize such facts. In particular, while T-facts can be modeled easily as an associative memory that is responsive to similarity, E-facts require specialized circuitry to enable them to detect binding-matches as well as binding-errors.

2.7.1 Encoding E-facts: Memory as a temporal pattern matcher

An E-fact should rapidly detect matches and mismatches between the bindings encoded by it and the dynamic bindings expressed in the system's state of activity. Given that dynamic bindings are represented by synchronous activity, it follows that the encoding of an E-fact should include coincidence-match circuits (for detecting binding matches) and coincidence-failure detector circuits (for detecting binding mismatches).

[11] The prior support may be viewed as being monotonically related to prior probability.

[12] The classification of memory is an ongoing areas of research and conventional classifications such as declarative/procedural, episodic/semantic, have proved problematic. As explained later, the distinction between E-facts and T-facts being made here is guided by computational considerations. Whereas, E-facts are sensitive to binding matches as well as binding-errors, T-facts are sensitive only to binding matches.

Figure 5 illustrates the encoding of E-facts *love(John,Mary)* and ¬*love(Tom,Susan)* used in the *Shruti* model.[13] Each E-fact is encoded using a distinct fact node of type τ-and (drawn as a pentagon). The τ-and node encoding a fact receives a link from the *enabler* node of the associated generic relation and sends a link to the + or – collector of the generic relation depending on whether the fact encodes a positive or a negative assertion. The link from the enabler to the fact node is modified by inhibitory links from role nodes of the associated generic relation. If a role is bound to an entity, the modifier input from this role node is in turn modified by an inhibitory link from the *?* node of the appropriate entity. The weight on the links from a "fact" node to the collector node encodes the strength of the fact.

In addition to connections that link a role node and the *?* node of the role-filler to the fact node, there also exist connections from the fact node and a role node to the + node of the role-filler. These connections are mediated by *binder* nodes. For every binding specified in an E-fact, there is a binder node of type m-ρ which receives one link from the fact node and another from the role node, and sends a link to the + node of the role-filler. The threshold of a binder node is satisfied when it receives inputs from *both* the role node and the fact node. Consequently, whenever both the fact and role nodes are active, a binder node fires in phase with the associated role node and propagates activity to the + node of the entity filling the role.

Given the query *love(John,Mary)*? the E-fact node F1 will become active and activate *+:love*, *+:John* and *+:Mary* nodes indicating a "yes" answer to the question. Similarly, given the query *love(Tom,Susan)*?, the E-fact node F2 will become active and activate *–:love*, *+:Tom* and *+:Susan* nodes indicating a "no" answer to the query. Finally, given the query *love(John,Susan)?*, neither *+:love* nor *–:love* would become active, indicating that the system can neither affirm nor deny whether John loves Susan (the nodes *+:John* and *+:Susan* will also not receive any activation).

Types can also serve as role-fillers in E-facts (e.g., Dog in "Dogs chase cats") as so can unspecified instances of a type (e.g., a dog in "a dog bit John"). Such E-facts are encoded by using the appropriate nodes in the focal-cluster for "Dog". In general, if an existing instance, *I*, is a role-filler in a fact, then *?:I* provides the input to the fact cluster and *+:I* receives inputs from the binder node in the fact cluster. If the whole type *T* is a role-filler in a fact, then *?v:T* provides the input to the fact cluster and *+v:T* receives inputs from the binder node in the fact cluster. If an unspecified instance of type *T* is a role-filler in a long-term fact, then a new instance of type *T* is created and its "*?*" and "+" nodes are used to encode the fact.

2.7.2 Encoding T-facts

The encoding of the T-fact ∀ *(x:Human, y:Book) buy(x,y) [50]* is depicted in Figure 6. The circular nodes are m-ρ nodes with a high threshold, and hence, they fire upon receiving synchronous activity from both their inputs. The output produced by the triangular node is proportional to the ratio of active input links to the total number of input links. The link from the enabler node of *buy* serves to enable the functioning of the triangular node. The weight on the link from the triangular node to the MAX node denotes the strength of the T-fact. For each generic relation such as *buy* there is a MAX node corresponding to positive T-facts about the relation, and another MAX node corresponding to all the negative T-facts about the relation. Observe that the encoding responds to binding matches and does not attempt to detect any binding errors. As in the case of E-facts, there also exist *binder* nodes that connect a T-fact node and a role node to the + node of the entity filling the role in the T-fact.

Typically, the role-fillers of T-facts are types. If type *T* is a role-filler in a T-fact, then *?v:T* provides the input to the fact cluster and *+v:T* receives inputs from the binder node in the fact cluster. In general, if an existing instance, *I*, is a role-filler, then *?:I* provides the input to the fact cluster and *+:I* receives inputs from

[13] The encoding described here solves the representational problem of encoding E-facts, but is not amenable to learning. An alternate coding that satisfies the representational requirements and at the same time is also amenable to rapid learning is described in (Shastri, 1999a).

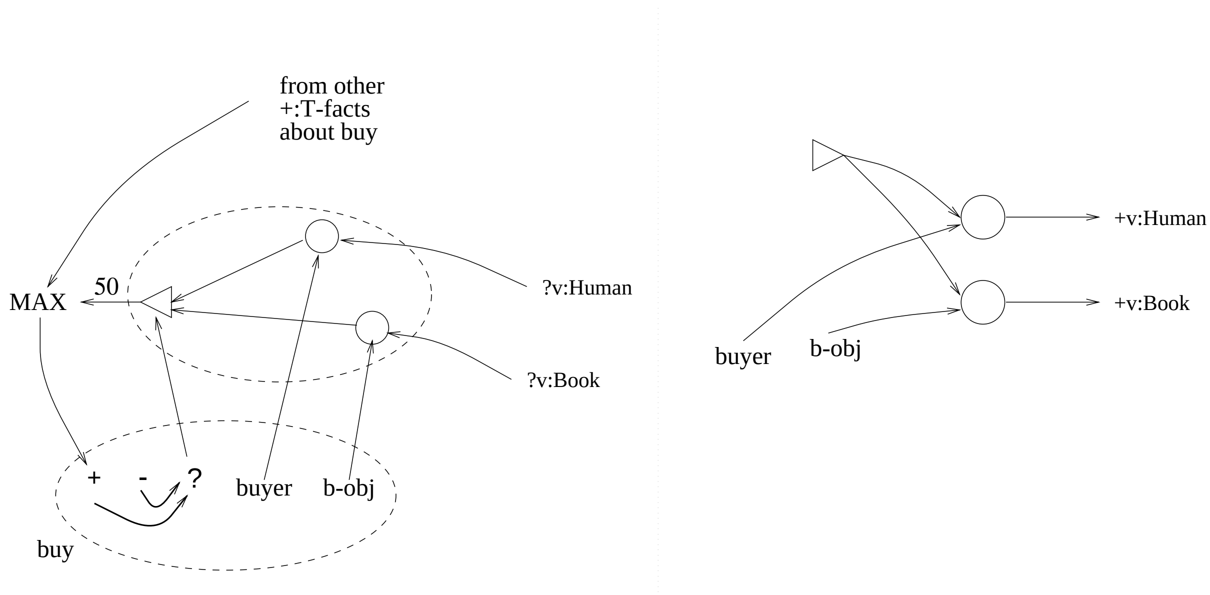

Figure 6: The encoding of T-fact: ∀*(x:Human, y:Book) buy(x,y) [50]*. The circular nodes have a high threshold and fire upon receiving coincident activity along both their incoming links. The triangular node is a τ-or node that produces an output in proportion to the ratio of active links to the total number of incoming links. The weight 50 on the link from the triangular node to the MAX nodes encodes the strength of the T-fact. Negative T-facts are encoded in the same manner, except the output of the MAX node corresponding to negative T-facts impinges on the – collector instead of the + collector. As in the case of E-facts, binder nodes link the T-fact node and a role node to the + node of the entity filling the role in the T-fact.

the binder node in the fact cluster. If an unspecified instance of type *T* is a role-filler in a long-term fact, then a new instance of type *T* is created and its "?" and "+" nodes are used to encode the fact.

2.8 Encoding of rules:

Figures 7 and 8 illustrate the encoding of the following rule in greater detail:

$\forall$ *x:Agent y:Agent z:Thing give(x,y,z)* $\Rightarrow$ *own(y,z)* [w1,w2]

which says that whenever any entity of type *Agent* gives something to another entity (also of type *Agent*), then the latter comes to own the thing that was given. The pair of weights [w1, w2] has the following interpretation: $w1$ indicates the degree of evidential support for *give* being the probable cause (or explanation) of *own*, and $w2$ indicates the degree of evidential support for *own* being a probable effect of *give*. These strengths are defined on a non-linear scale ranging from 0 to 1000. As pointed out earlier, the mapping of probabilities and evidential supports to weights in *Shruti* is non-linear. In general, the weights can be set approximately, and subsequently, fine tuned using gradient-descent based supervised learning algorithms to model a given domain.

A rule is encoded via a focal-cluster that mediates the flow of activity and bindings between the antecedent and the consequent of the rule. We refer to this cluster as the *mediator* of the rule.[14] The mediator consists of a collector node (+) and an enabler node (?) and as many *role-instantiation* nodes as there are distinct variables in the antecedent. Every role in the antecedent maps to exactly one role-instantiation node in the mediator, with co-referenced roles mapping to the same role-instantiation node. Role-instantiation nodes are abstractions of small neural circuits and are depicted as square nodes (see Figure 7).

The encoding of a rule establishes links between nodes in the antecedent, consequent, and mediator clusters as follows:

- The roles of the consequent relations are linked to the roles of the antecedent relations via appropriate role-instantiation nodes in the mediator. This linking reflects the correspondence between antecedent and consequent roles specified by the rule.

- The *enablers* of the consequent relations are connected to the *enablers* of the antecedent relations via the enabler of the mediator.

- The appropriate (+/–) collectors of the antecedent relations are linked to the appropriate (+/–) collectors of the consequent relations via the collector of the mediator. A collector to collector link originates at the + (–) collector of an antecedent relation if the relation appears in its positive (negated) form in the antecedent. The link terminates at the + (–) collector of the consequent relation if the predicate appears in a positive (negated) form in the consequent.

- The weights on the links capture the *evidential* nature of the rule. Thus the two weights in Figure 7 can be interpreted informally as follows:

  w1: $\forall$ *x:Agent y:Agent z:Thing Support(give(x,y,z)* $|$ *own(y,z))*.

  w2: $\forall$ *x:Agent y:Agent z:Thing Support(own(y,z)* $|$ *give(x,y,z))*.

  It is possible to interpret the evidential weights in a number of ways. This includes probabilistic as well fuzzy interpretations. A detailed discussion of various possibilities, and their relevance to modeling different sorts of knowledge, is beyond the scope of this paper (see Shastri and Wendelken, 1998; Wendelken and Shastri, in preparation).

[14] The design of the mediator was motivated, in part, by discussions the author had with Jerry Hobbs.

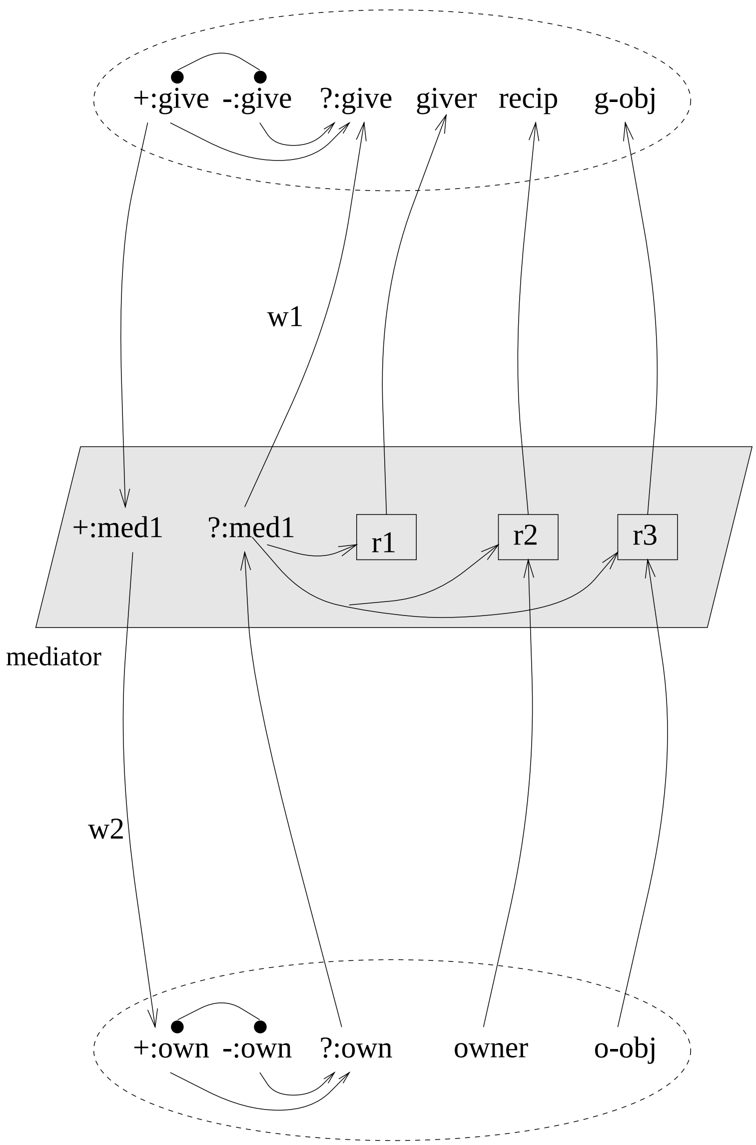

Figure 7: A simplified version of the encoding of the rule: $\forall$ *x:Agent y:Agent z:Thing give(x,y,z)* $\Rightarrow$ *own(y,z) [w1,w2]*. Enabler to enabler, collector to collector, and role to role connections between the antecedent and consequent predicates are mediated by a mediator cluster that reflects the structure of the antecedent. The square nodes are abstractions of a neural circuit. See text for details.

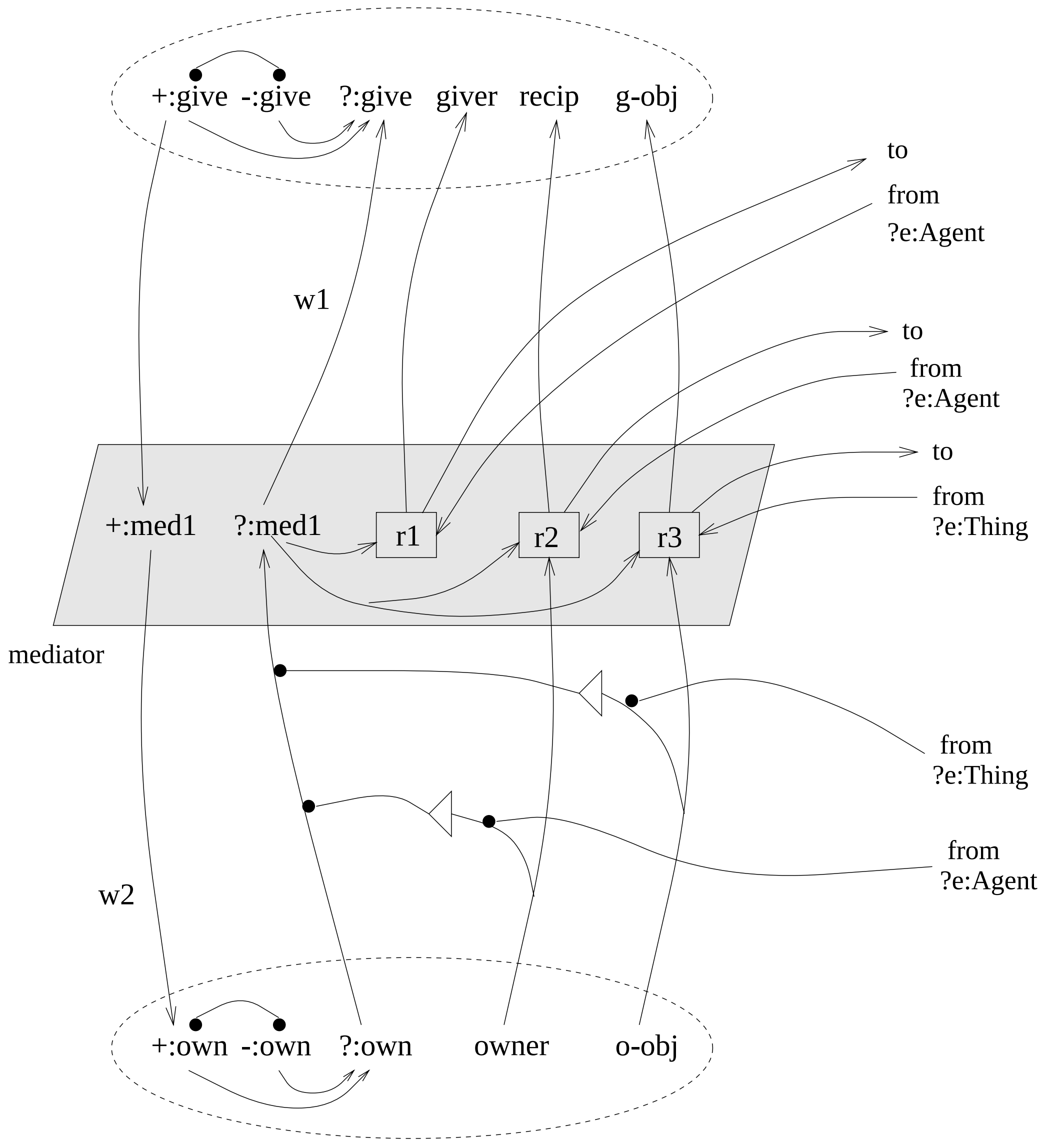

Figure 8: A more detailed encoding of the rule: ∀ *x:Agent y:Agent z:Thing give(x,y,z)* ⇒ *own(y,z)*. Additional circuitry required for enforcing type restrictions is shown along with the connections between the focal-clusters of the antecedent and consequent predicates. The triangular nodes are type 1 τ-or nodes. One such node enforces the restriction that the role fillers of *o-obj* must be of the type *Thing* in order for this rule to fire in the backward direction. The τ-or node blocks the firing of the rule if the *b-obj* is bound to a type other than *Thing*. The circuitry for enforcing restriction that the role-filler of *owner* be of the type *Agent* is analogous. The square nodes are abstractions of a neural circuit. See text for details about the functioning of these nodes.

- Type restrictions on role-fillers are encoded using type 1 τ-or nodes (see Figure 8). The restriction that the role-filler of *o-obj* must be of the type *Thing* is encoded by linking the output of the *o-obj* node to a type 1 τ-or node which will fire whenever *o-obj* fires, and block the link from ?:give to the mediator enabler, unless the *o-obj* node fires in synchrony with *?e:Thing*. Recall that the synchronous firing of *o-obj* with *?e:Thing* signifies that *o-obj* role is bound to an instance of type *Thing*. Thus during a search for an explanation, this rule fires only if the role *o-obj* is either unbound, or bound to an entity of type *Thing*. The circuit for enforcing the restriction that the role-filler of *owner* be of the type *Agent* is analogous.

- Circuitry for enforcing other special conditions in rules (e.g., repeated variables in the consequent) is described in (Shastri and Ajjanagadde, 1993).

The role-instantiation node is an abstraction of a neural circuit with the following functionality. A role-instantiation node receives one link from the mediator enabler node and zero or more links from consequent role nodes. If a role-instantiation node receives activation from the mediator enabler *and* one or more consequent role nodes, it simply propagates the activity onward to the connected antecedent role nodes. If on the other hand, the role-instantiation node receives activity *only* from the mediator enabler, it sends activity to the *?e* node of the type specified in the rule as the type restriction for this role. This causes the *?e* node of this type to become active in an unoccupied phase.[15] The *?e* node of the type conveys activity in this phase to the role-instantiation node which in turn propagates this activity to connected antecedent role nodes. This interaction between the mediator and the type representation essentially leads to activity that corresponds to the following: "Is there any entity of the specified type that could be a role-filler for the given role?" With reference to Figure 8, consider *r1*, the role-instantiation node corresponding to the role *giver*. This node is not connected to any role of *own*, and hence, whenever this rule fires, *r1* will receive activation only from the mediator *enabler*. Consequently, it will send activation to the *?e:Agent* node in the type hierarchy. This will lead to the activation of *?e:Agent* node in an unoccupied phase. The *?e: Agent* node will now activate *r1* in this phase, and *r1* will in turn propagate this activity to the role *giver* of *give*.

With reference to Figure 1, note that each rule is associated with its own mediator structure. Furthermore, the rule weights indicate that buying is more likely to explain ownership than giving (900 versus 800) and that you are more likely to own an object if you buy it, than if you were given it (980 versus 800).

3 An example of inference

Figure 9 depicts a schematic response of selected nodes in Figure 1 to the query "Does Mary own a book?" ($\exists$*x:Book own(Mary,x)?*).[16]

This query is posed by activating *?:Mary* and *?e:Book* nodes, the role nodes *owner* and *o-obj*, and the enabler *?:own*, as shown in Figure 9. Observe that *?:Mary* and *owner* are firing in synchrony and so are *?e:Book* and *o-obj*. We will refer to the phases of activation of *?:Mary* and *?e:Book* as ρ_1 and ρ_2, respectively.

Activation from the focal-cluster for *own* reaches the mediator structure of rules (1) and (2). Consequently, nodes *r2* and *r3* in the mediator for rule (1) become active in phases ρ_1 and ρ_2, respectively. Similarly, nodes *s1* and *s2* in the mediator of rule (2) become active in phases ρ_1 and ρ_2, respectively. At the same time, the activation from ?:own activates the enablers *?:med1* and *?:med2* in the mediators of rules (1)

[15] In the current implementation of *Shruti* this is done artificially (in software). In a detailed and neurally plausible implementation, this will be the result of inhibitory interactions between instance nodes in the type hierarchy. A similar form of node activation in an unoccupied phase is suggested by Ajjanagadde (1990) for realizing function terms.

[16] In general, the form of queries in *Shruti* is similar to that of facts except that whereas repeated universally quantified variables can occur in queries, existentially quantified variables are assumed to be distinct.

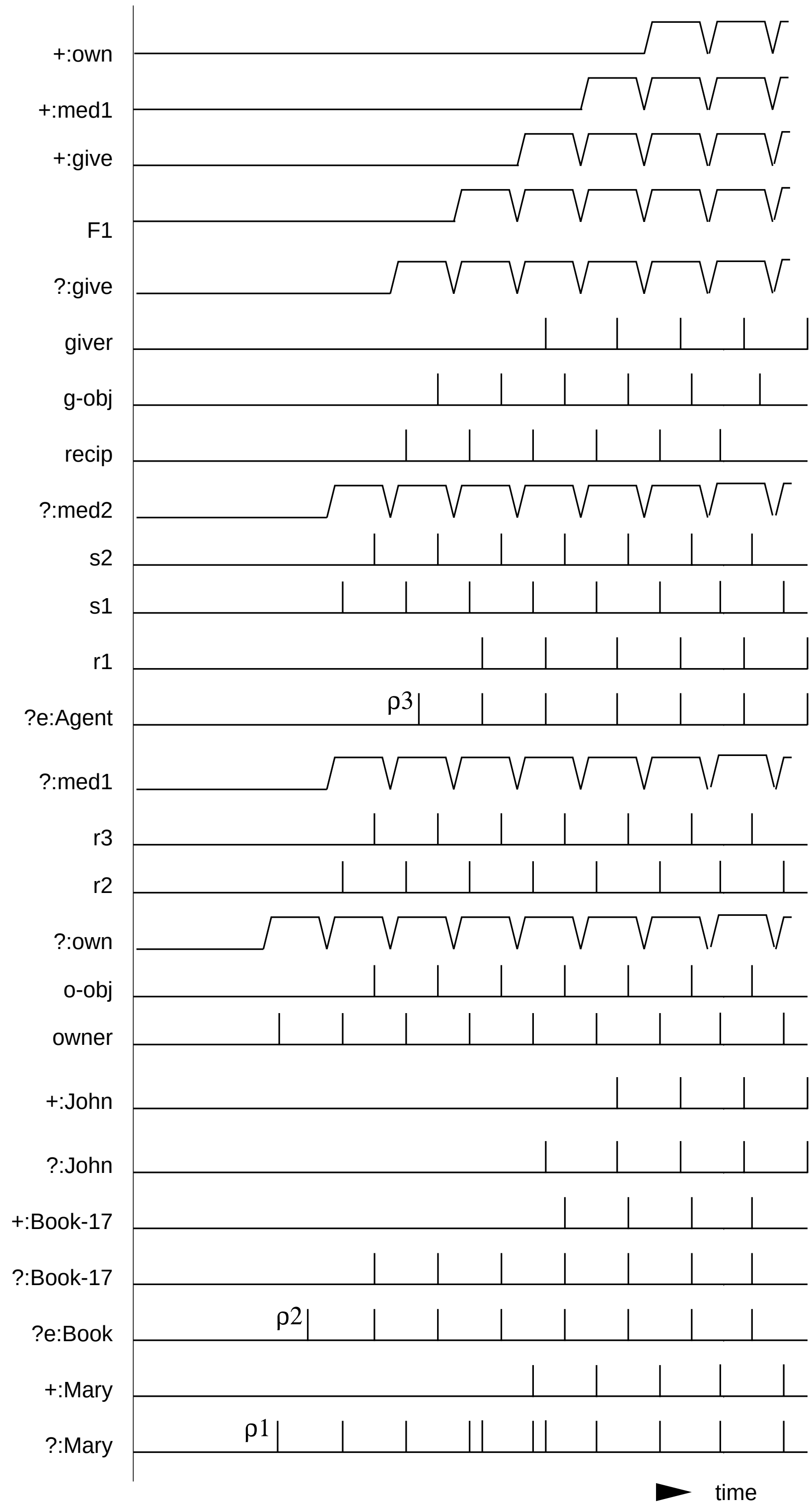

Figure 9: Schematic response of selected nodes in Figure 1 to the query $\exists$*x:Book own(Mary,x)?* (Does Mary own a book?) The trace does not show the strength (or level) of activity of nodes. See text for details.

and (2). Since *r1* does not receive any activation from any role in its consequent's focal-cluster (i.e., from *own*), it activates the node *?e:Agent* in the type hierarchy which becomes active in a free phase (say $\rho 3$).

The activation from nodes *r1*, *r2* and *r3* reach the roles *giver*, *recip* and *g-obj* in the *give* focal-cluster, respectively. Similarly, activation from nodes *s1* and *s2* reach the roles *buyer* and *b-obj* in the *buy* focal-cluster, respectively. Thus after a few cycles, roles *recip* and *buyer* become active in ρ_1, roles *g-obj* and *b-obj* become active in ρ_2, and role *give* becomes active in a new phase ρ_3 along with the node *?e:Agent* in the type structure.

In essence, the system has created new bindings for *give* and *buy* wherein *giver* is bound to an undetermined entity of type *Agent*, *recipient* is bound to *Mary*, *g-obj* is bound to *a book*, *buyer* is bound to *Mary* and *b-obj* is bound to *a book*. These bindings together with the activation of the enabler nodes *?:give* and *?:own* encode two new queries: "Did *some agent* give Mary a book?", and "Did Mary buy a book?".

Concurrently, activation travels in the type hierarchy from *?e:Agent* to *?e:Person*, etc., and thereafter to *?:John*, *?:Mary*, etc. As a result of this propagation, the queries about "giving" and "buying" gets mapped to a family of queries with different role-fillers (e.g., "Did a human give Mary a book?", "Did John give Mary a book?", "Did John give Mary Book-17?", "Did Mary buy all books" etc.

The τ-and node associated with the E-fact *give(John, Mary, Book-17)* now becomes active as a result of matching the query *give(John, Mary, Book-17)?* and causes *+:give* to become active. This in turn causes the mediator collector, *+:med1*, to become active and transmit activity to *+:own* (the level of activity attained by *+:give* and *+:own* is 980 and 784, respectively).

The activation of *+:own*, along with the firing of *owner* and *o-obj* nodes in synchrony with various entity nodes means that the network state is asserting, among other things, "Mary owns a book". This corresponds to an affirmative answer to the query. Moreover, the activation *+own* creates a reverberant loop of activity involving the clusters *own*, *med1*, *give*, the fact node F1, and the entities *John*, *Mary*, and *Book-17*. This state of reverberant activity signifies that the network has found a coherent explanation (or answer) to the query "Does Mary own a book", namely, John gave Mary Book-17, so Mary owns a book.

Observe that reasoning is the spontaneous and natural outcome of the network's behavior. The network does not apply syntactic rules of inference such as *modus-ponens*. There is no separate interpreter that manipulates and rewrites symbols. The network encoding 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.

3.1 Parallelism and the significance of structure

The encoding of rules by the explicit encoding of the inferential dependency between predicates and predicate roles, in conjunction with the use of temporal synchrony provides an efficient mechanism for propagating dynamic bindings and performing reasoning. Conceptually, the proposed encoding of rules creates a directed *inferential dependency graph*: Each predicate role is represented by a node in this graph and each rule is represented by links between nodes denoting the roles 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. Specifically, a large number of rules can fire in parallel and the time taken to draw an inference just equals $l\pi$, where l is the *length* of the chain of inference and π is the period of activity. In other words, *Shruti* supports parallelism at the knowledge-level and the time it takes to draw an inference is only proportional to the depth of inference.

3.2 On the expected nature of rhythmic activity

Let us assume that rhythmic activity of the sort suggested by *Shruti* underlies the representation and propagation of dynamic bindings in the brain. What sort of activity should we then expect to find in the brain during an episode of reflexive reasoning? As discussed in (Shastri and Ajjanagadde, 1993), the answer to this question would vary dramatically depending upon our expectations about the nature of representations employed by the brain.

If one believes in fully distributed representations and assumes that entities are represented as patterns of activity over *large* populations of cell, one would expect a large number of cells to participate in rhythmic activity during an episode of reflexive reasoning. On the other hand, if one believes in a more compact representation[17] of the type espoused in *Shruti*, one would expect a *relatively* small number of cells to do so.

In view of the above, consider the observation that periodically firing cells form a small tail in the distribution, and the spike intervals of a majority of cells yield a distribution that is more Poisson than periodic (Freeman, 1993). What should one conclude from this observation? Does this observation support the biological plausibility of *Shruti* or does it undermine it?

If one believes in fully distributed representations, one might be compelled to conclude that rhythmic activity does not underlie the representation of dynamic bindings. But if one believes in relatively compact representations, one would find strong evidence in support of the hypothesis that rhythmic activity underlies dynamic bindings; since only a very small fraction of cells would be involved in rhythmic activity at any point in time, the small tail constitutes just the right sort of evidence.

Let us consider a thought experiment to illustrate the nature of rhythmic activity in higher-order cortical areas entailed by a system such as *Shruti*. Assume that the system is in a ‘quiescent’ state, i.e., it is not receiving any stimulus and it is not engaged in any systematic thought. At this time the nodes in the system would be firing with some background rate, perhaps Poisson. Now assume that the dynamic fact ‘John bought a Rolls Royce’ is injected into the system. Consequently, we would expect two trains of rhythmic activity to propagate in the system. One train would originate at the John and buyer clusters and rapidly expand to include other clusters representing owner, person, wealthy, etc. A second train of activity would originate at the Rolls-Royce and buy-object clusters and expand to include other clusters such as car and own-object. This rhythmic activity might last a second or so, after which the synchronization might break down, though the active nodes may continue firing above the background rate for some time.

Even if the reflexive reasoning resulting from the input “John bought a Rolls Royce” were to activate several hundred relations, types, and features, the total number of cells engaged in synchronous activity during this episode of reasoning would remain relatively small – perhaps no more than 10^5. Furthermore these cells would be physically distributed in the area(s) where conceptual knowledge is represented. This estimate is extremely crude and speculative, but it does convey the essential point, namely, even if we knew which areas of the brain encode conceptual knowledge and even if we recorded from cells in these areas, we would see an extremely small fraction of cells participating in synchronous activity during any given episode of reflexive reasoning.

Finally, consider an extended *Shruti*-like cognitive system. Such a system would be capable of responding to continuous stimuli and shifting its focus of attention. The dynamics of such a system would be far more complex than the simple oscillatory patterns depicted in Figure 9. In such a system, the frequency of oscillations will vary constantly since the period will decrease whenever entities drop out of the focus of attention, and increase whenever entities enter the focus of attention. Furthermore, different modalities in the system may fire at different frequencies and have their own phase distribution. Therefore, the rhythmic activity observed in such a system will be far more complex than the activity portrayed in Figure 9.

[17] Recall that each node in *Shruti* corresponds to a small, but physically dispersed, cluster of cells. If we assume that such a cluster contains 100-1000 cells, it follows that the focal-cluster of a relation or entity might contain about 1000–10000 cells.

4 Encoding of more complex rules

Figure 10 shows the encoding of a rule with multiple antecedents and consequents:

$\forall$ *x:T1 y:T2 z:T3 P1(x,y) [w1,w2]* $\wedge\neg$ *P2(y,z) [w3,w4]* $\Rightarrow\neg$ *Q1(x,w) [w5,w6]* $\wedge$ *Q2(x,z) [w7,w8]*.

The enabler nodes of all consequents converge to the enabler node of the mediator, which then diverges to the enabler nodes of all the antecedents. Similarly, the collector nodes of all the antecedents converge to the collector node of the mediator, which then diverges to the collector nodes of all the consequents. Also note that co-reference of roles in the consequent is captured by the convergence of appropriate consequent role nodes to the same role-instantiation node. Similarly, the co-reference of role nodes in the antecedent is captured by the same role-instantiation node being connected to the appropriate antecedent role nodes.

The weights depicted in Figure 10 can be interpreted informally as follows:

w1: $\forall$ *x:Agent y:Agent z:Thing Support(P1(x,y)* $\mid$ *cons).*
w2: $\forall$ *x:Agent y:Agent z:Thing Support(ant* $\mid$ *P1(x,y)),*
w3: $\forall$ *x:Agent y:Agent z:Thing Support(P2(x,y)* $\mid$ *cons).*
w4: $\forall$ *x:Agent y:Agent z:Thing Support(ant* $\mid\neg$ *P2(y,z)),*
w5: $\forall$ *x:Agent y:Agent z:Thing Support(cons* $\mid$ *Q1(x,w)),*
w6: $\forall$ *x:Agent y:Agent z:Thing Support(*$\neg$ *Q1(x,w)* $\mid$ *ant).*
w7: $\forall$ *x:Agent y:Agent z:Thing Support(cons* $\mid$ *Q2(x,z)),*
w8: $\forall$ *x:Agent y:Agent z:Thing Support(Q2(x,w)* $\mid$ *ant).*

In the above, *ant* refers to the situation describing the antecedent as a whole, and *cons* refers to the situation describing the consequent as a whole.

In general rules have the following form:

$$\exists x_1{:}X_1,\ldots,x_p{:}X_p\ \forall y_1{:}Y_1,\ldots,y_r{:}Y_r\ ECF_{ant}\ P_1(\ldots)[a_1,b_1]\wedge\cdots\wedge P_n(\ldots)[a_n,b_n]\Rightarrow$$

$$\exists u_1,\ldots,u_t\ ECF_{cons}\ Q_1(\ldots)[c_1,d_1]\wedge Q_m(\ldots)[c_m,d_m]$$

wherein P_i are positive or negative literals, an argument of P_i can be an entity or one of the variables x_i and y_i. An argument of Q can be an entity or one of the variables x_i, y_i, and u_i. X_is and Y_is are types and specify restrictions on the bindings of variables, a_i, b_i, c_i, and d_i are weights, and ECF_{ant} and ECF_{cons} are evidence combination functions for combining activations impinging on the mediator collector from antecedent collector nodes, and the mediator enabler from the consequent enabler nodes, respectively.

4.1 Evidence combination

The enabler node of a rule's mediator combines incoming evidence from various consequents of a rule to compute the net support for the consequent as a whole. Similarly, the collector node of a rule's mediator combines incoming evidence from the various antecedents of a rule to compute the net support for the antecedent as a whole. A variety of evidence combination functions (ECFs) can be deployed at these nodes. In the past we have used functions such as: max, min, *sigmoid*, and *average*, but we are currently experimenting with more flexible evidence combination functions including a family of functions such as *soft-and*, *soft-min*, *average*, *soft-max*, and *soft-or* functions (Shastri and Wendelken, 1998; Wendelken and Shastri, in preparation).

4.2 Some other features of rules

Rules in *Shruti* can refer to attribute values of entities and also include comparison operations between attribute values of entities. Thus a rule may state:

$\forall$*(x:Object) closing-in(x,Self)* $\wedge$ *(av-extract(Speed,x)* $>$ *av-extract(Speed,Self))* $\Rightarrow$ *monitor(x) [800,1000];*

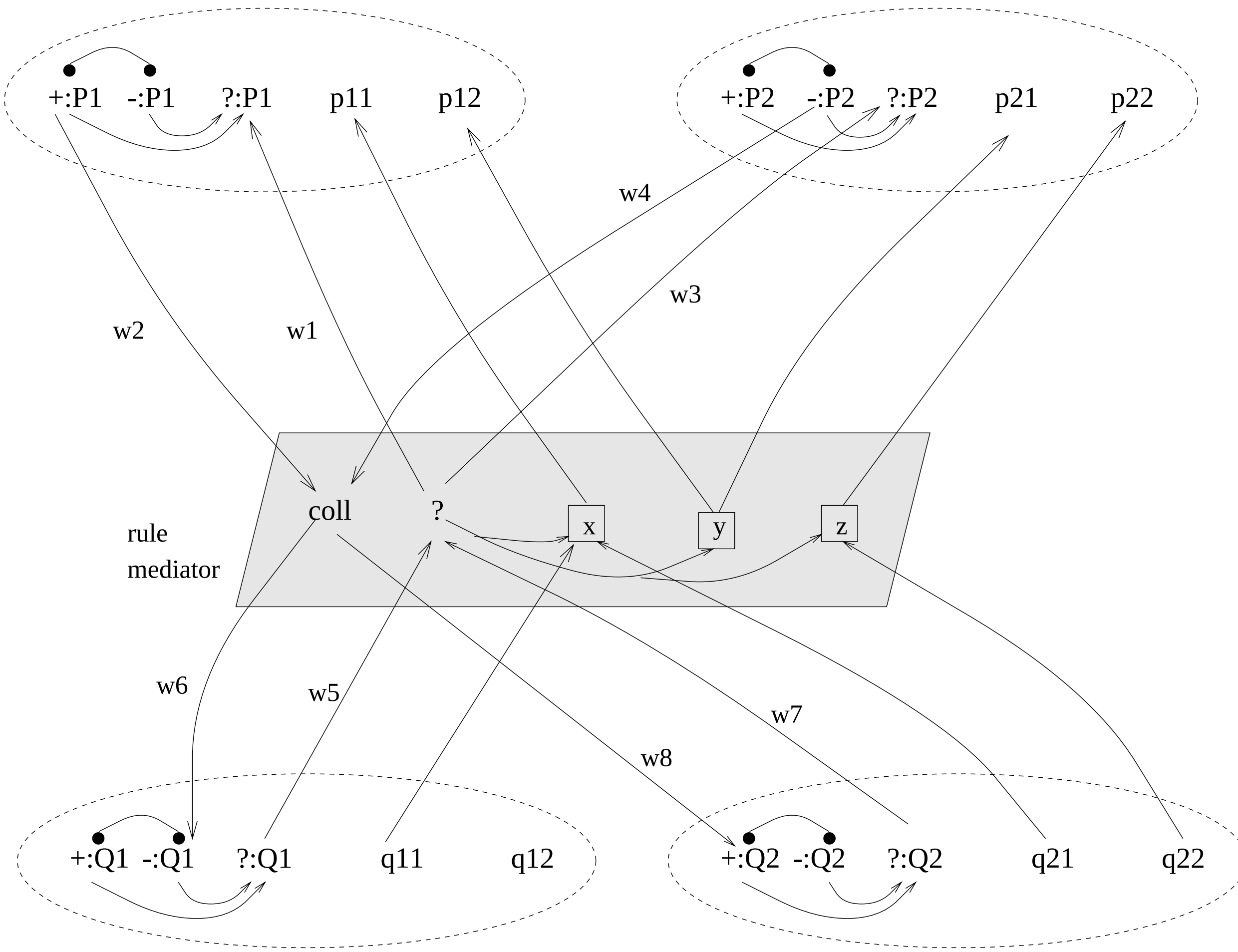

Figure 10: Encoding of the rule: $\forall$ *x:T1 y:T2 z:T3 P1(x,y) [w1,w2]* $\wedge\neg$ *P2(y,z) [w3,w4]* $\Rightarrow$ $\neg$ *Q1(x,w) [w5,w6]* $\wedge$ *Q2(x,z) [w7,w8]*. Circuitry for enforcing type restrictions on role-fillers, and links from the role-instantiation nodes to the type hierarchy have not been shown. The output level of the mediator enabler and collector would be determined by a suitable activation combination function.

the above says that the organism must monitor something that is closing in on it and is moving faster than it. Here *av-extract* is a connectionist structure that can extract values of specified attributes such as *Speed*. *Shruti* also supports the setting of attribute values from inside of rules.

Rules can also include exception conditions. Thus one can say, if you buy an object you own it, unless the object has been stolen. That is,

$\forall$ *(x:Agent y:Thing) buy(x,y)* $\Rightarrow$ *own(x,y) [a,b] unless stolen(y) [c,d];*

In the above, the weights *[a,b]* are as before. Weight *c* indicates how much energy is directed into checking the exception and *d* indicates how strongly the belief in the truth of the exception inhibits the firing of the rule.

An exception condition introduced via an unless clause and a negative condition introduced in the antecedent have different significance. While ignorance about an exception condition does not affect the total support for the antecedent, ignorance about a condition in the antecedent does. Imagine that John buys a car, and it is not known whether or not the car had been stolen. In this condition, the above rule would lead to the inference "John owns the car" with a strength of about *b*. If however, the above rule were to be replaced by:

$\forall$ *(x:Agent y:Thing) buy(x,y)* $\wedge\neg$ *stolen(y)* $\Rightarrow$ *own(x,y) [a,b]*

the strength of inference in a similar situation would typically be much less than *b*, because only one of the two antecedents is satisfied (the actual strength of inference would depend on the ECF used at the collector node of the rule mediator).

4.3 Multiple instantiations of relations

The simultaneous activation of more than one instance of the same relation can lead to cross-talk between the active instances of the relation. Consider the the simultaneous activation of two instances of the relation *love*: "John loves Mary" and "Tom loves Susan". Unless the system is able to separate the bindings pertaining to the two instances, the state of activity can give rise to spurious blends such as "John loves Susan" and "Tom loves Mary". This problem was described as the multiple-instantiation problem in (Shastri and Ajjanagadde, 1993).

Shruti allows a bounded number of instances of each relation to be active at the same time without cross-talk.[18] The ability to support multiple instances of a relation, in turn, enables *Shruti* to compute inferences involving *bounded* recursion. For example, *Shruti* can encode and reason with rules such as:

1. $\forall$*(x:Animate, y:Animate) sibling(x,y)* $\Rightarrow$ *sibling(y,x) [1000,1000];*

2. $\forall$*(x:Animate, y:Animate, z: Animate) older(x,y)* $\wedge$ *older(y,z)* $\Rightarrow$ *older(x,z) [1000,1000];*

3. $\forall$*(x:Animate, y:Animate, z:Animate) loves(x,y)* $\wedge$ *loves(y,z)* $\wedge\neg$ *equal(x=z)* $\Rightarrow$ *jealous(x,z) [a,b];*

However, the tight bound on the number of instances of a relation that can be active simultaneously (see the following section), places limits on the prolific use of such rules during reflexive reasoning. Note that the use of rule (1) and rule (2) requires 2 and 3 instances of the same relation, respectively, to be active at the same time.

In order to represent multiple active instances of a relation, the representation of a relation is augmented and *k* focal-clusters are associated with each relation instead of one (here *k* is a parameter). Each relation also has an associated "switching" network which prevents cross-talk between different active instances of

[18] In general, a large number of E-facts and T-facts associated with a relation can become active simultaneously. Since each active relational instance can match several facts associated with that relation, the number of active facts can be much higher than the number of active relational instances. For example, the relational instance encoding the query $\exists$*x:T P(a,x)* will match all facts such as *P(a,b), P(a,c),* etc., where *b* and *c* are instances of the type *T*.

the relation. A description of a connectionist realization of such a switch appears in (Mani and Shastri, 1993). More recently, we have augmented the representation of a bank of focal-clusters such that multiple instances of a relation can dynamically collapse into a single instance if the entities participating in the two instances collapse into a single entity.

An alternate solution to the multiple-instantiation problem, using two levels of synchronous activity was described in (Shastri and Ajjanagadde, 1990b). Therein it was proposed that fine grained synchronization can be used to bind roles and entities, and coarse grained synchronization can be used to bind all the role-entity bindings pertaining to a single relational instance. A related solution using period doubling is described in (Sougne, 1998).

5 Constraints and predictions

Shruti identifies a number of representational and processing constraints on reflexive processing in addition to the constraint on the form of rules discussed above. These relate to the capacity of the "working memory" underlying reflexive processing and the bound on the depth of reasoning.

5.1 Working memory underlying reflexive processing

Dynamic bindings, and hence, active relational instances (i.e., active facts) are represented in *Shruti* as a rhythmic pattern of activity over nodes in the LTM network. In functional terms, this transient state of activation holds information temporarily during an episode of reflexive reasoning and corresponds to the *working memory underlying reflexive reasoning* (WMRR). Note that WMRR is just the state of activity of the LTM network and not a separate buffer.

Shruti predicts that the capacity of WMRR is very large but at the same time it is constrained in critical ways. Most proposals characterizing the capacity of the working memory underlying cognitive processing have not paid adequate attention to the structure of items in the working memory and their role in processing. For example, proposals such as Just and Carpenter (1992) characterize working memory capacity in terms of "total activation". In contrast, the constraints on working memory capacity predicted by *Shruti* depend not on total activation but rather on the maximum number of *distinct* entities that can participate in dynamic bindings simultaneously, and the maximum number of instances of any given relation that can be active simultaneously.

5.1.1 Bound on the number of distinct entities referenced in 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 role-fillers in relational instances (facts) active in the WMRR cannot exceed $\lfloor \pi_{max}/\omega \rfloor$, where π_{max} is the maximum delay between two consecutive firings of cell-clusters involved in synchronous firing and ω equals the width of the window of synchrony — i.e., the allowable lead/lag between the firing of synchronous cell-clusters. Note that the activation of an entity together with all its active super-concepts counts as only *one* entity. Thus the synchronous activation of Agent, Human, and John clusters counts as the activation of a single entity.

Assuming that synchronous activity underlying dynamic bindings occurs in the γ band, a neurally plausible value of π_{max} is about 33 milliseconds. Similarly, a plausible estimate of ω suggested by a statistical analysis of synchronous activity (Gray, Engel, Koenig, and Singer, 1991) is about 6 milliseconds. These estimates of π_{max} and ω lead to the following prediction: As long as the number of distinct entities serving as role-fillers in active facts is five or less, ω will be more than 6 milliseconds, and hence, there will be very little cross-talk among dynamic bindings. However, if more than five entities occur as role-fillers in active facts, the effective ω would have to shrink to accommodate all the active entities. As ω shrinks, the possibility of cross-talk between dynamic bindings would increase until eventually, the cross-talk would become excessive and disrupt the system's ability to perform systematic reasoning. Given the noise and variation indicated by the data on synchronous activity (Gray et al. 1991), it appears unlikely that ω can be less than 3 milliseconds. Hence we predict that an *upper bound* on the number of distinct entities that can be referenced by active facts is less than 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.

Given the remarkable match between the bound on the number of entities that may be referenced by the facts active in the WMRR on the one hand, and 7 ± 2, the robust measure of short-term memory capacity suggested by (Miller 1956) on the other, Ajjanagadde and Shastri had predicted that temporal synchrony may also underlie other short-term and dynamic representations (see Ajjanagadde and Shastri, 1991 pp. 131; Shastri, 1992 pp. 162; Shastri and Ajjanagadde, 1993 pp. 443). In a similar spirit, Horn and Usher (1991) had observed that limits on the number of synchronous patterns that can be co-active might explain the limits on attention capacity (pp. 42). Recent experimental findings as well as computational models lend support to this prediction (see Lisman and Idiart, 1995; Jensen and Lisman, 1996; Luck and Vogel, 1997).

Note that the active facts represented in the WMRR during an episode of reflexive reasoning should not be confused with the small number of short-term facts an agent may *overtly* keep track of during *reflective* processing and problem solving. WMRR should also not be equated with the short-term memory implicated in various memory span tasks (Baddeley, 1986). In our view, in addition to the overt working memory, there exist as many "working memories" as there are major processes or modalities in the brain since a "working memory" is nothing but the state of activity of a network. Some recent findings seem to support this view (Duncan, Martens, and Ward, 1997).

5.1.2 Bound on the multiple instantiation of relations

The capacity of WMRR is also limited by the constraint that only a small number of instances of each relation may be *active* at the same time. Recall that the total number of relational instances active throughout the network can be very high.

The cost of maintaining k active instances of a relation turns out to be significant in terms of space and time. For example, the number of nodes required to encode a rule is proportional to the square of k. Furthermore, the worst case time required for propagating multiple instances of a relation also increases by a factor of k. In view of the additional space and time costs associated with maintaining multiple instances, and given the necessity of keeping these resources within bounds in the context of reflexive processing, we believe that the value of k during reflexive reasoning is quite small. The computational requirements of tasks such as parsing and inference suggest that a multiple-instantiation bound of less than three would be overly limiting. At the same time, a system with a bound of four would require almost twice as many nodes as a system with a bound of three. This suggests that a bound greater than three for all relations, across the board, may be too expensive. Consequently, we predict that in general, relations have a multiple-instantiation limit of three, and only select relations that participate in rules requiring multiple instances of the relation may have a bound greater than three.

The multiple-instantiation bound implies that it is expensive to apply the *abstract* notion of transitivity in a reflexive manner. The abstract assertion that the relation P is transitive requires a rule such as $\forall x, y, z P(x, y) \wedge P(y, z) \Rightarrow P(x, z)$ and each firing of this rule involves three instances of P. But given that transitivity plays an important role in common sense reasoning — to wit, reasoning about sub and super-categories, part-of relationships, greater than, less than, etc. — the limit on reasoning with transitivity might seem overly restrictive. However, this is not the case. As argued in (Shastri and Ajjanagadde, 1993), as far as query answering is concerned, humans seem to be good at dealing with the transitivity of only 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. For example, if the sequence in which letters appear in the English alphabet is encoded explicitly in the system (e.g., via a chain of links), then given a pair of letters (i, j), the system can readily determine whether or not i comes before j, by propagating activity along the chain of links that encode the ordering of letters in the alphabet. For a more complex example, consider the organization of concepts in an *is-a* hierarchy. The specialized structure described in Section 2.3 converts the problem of computing the transitive closure of *is-a* relations 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 spreading activation along links in the structure.

5.1.3 Unbounded number of rule firings

Shruti demonstrates that a large number of rules — even those containing variables — may fire in parallel as long as no relation is instantiated more than ≈ 3 times and the number of distinct entities referenced by the active facts remains below 10. This may be contrasted with Newell's suggestion (1980) that while "productions" (i.e., rules) without variables can be executed in parallel, productions with variables may have to be executed in a serial fashion. Thus *Shruti* suggests that neurally plausible architectures can support a high degree of parallelism — even when dealing with complex knowledge involving variables.

5.2 Bound on the depth of the chain of reasoning

Consider the propagation of synchronous activity along a chain of role ensembles during an episode of reflexive reasoning. Two things might happen as activity propagates along the chain of role ensembles. 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 the propagation delay of links 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 role'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 role bindings and more and more to an associative spread of activation. For example, the propagation of activity along a chain of rules such as: $P_1(x, y, z) \Rightarrow P_2(x, y, z) \Rightarrow ...P_n(x, y, z)$ due to a dynamic fact $P_1(a, b, c)$ may lead to a state of activation where all one can say about P_n is this: there is an instance of P_n which involves 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.

5.3 Limits on reflexive processing — some evidence from parsing

Henderson (Henderson, 1994) has developed an on-line parser for English using a *Shruti*-like architecture whose speed is independent of the size of the grammar and which can recover the structure of arbitrary long sentences as long as the dynamic state required to parse the sentence does not exceed the capacity of the parser's working memory. The parser models a range of linguistic phenomena and shows that the constraints on the parser's working memory help explain several properties of human parsing involving long distance dependencies, garden path effects and our limited ability to deal with center-embedding. This suggests that the working memory constraints resulting from *Shruti* have implications for other rapid processing phenomena besides reasoning.

6 A simple example involving conflicting evidence

An agent with limited resources must sometimes act with only limited attentional focus and often under time pressure. This means that an agent may sometimes overlook relevant information and act in an erroneous manner. Extended evaluation or an appropriate cue, however, might make the necessary information available and lead to a correct response. Several interesting aspects of such a situation are captured in the following scenario which we have referred to as the *Post Office Example* (Shastri and Grannes, 1996): John runs into Mary on the street. "Where are you going?" asks John. "To the post office," replies Mary. "But isn't today Presidents' Day?" remarks John. "Oops! I forgot that today was a federal holiday," says Mary after a momentary pause and heads back.

Clearly, Mary had sufficient knowledge to infer that "today" was a postal holiday. But the fact that she was going to the post office indicates that she had assumed that the post office was open. So in a sense, Mary held inconsistent beliefs. John's question served as a trigger and brought the relevant information to the surface and made Mary realize her mistake. A cognitively plausible model should be capable of modeling such situations.

The agent's knowledge is modeled as follows (refer to Figure 11):

1. *presidents-day(day)* $\Rightarrow$ *federal-holiday(day) [1000,1000]*,
2. *3rd-Mon-Feb(day)* $\Rightarrow$ *presidents-day(day) [1000,1000]*,
3. *3rd-Mon-Feb(16-Feb-98) [1000]*,
4. $\neg$*3rd-Mon-Feb(20-Feb-98) [1000]*,
5. *weekday(day)* $\wedge$ *post-office(x)* $\Rightarrow$ *open(x,day) [900,800]*
6. *weekend(day)* $\wedge$ *post-office(x)* $\Rightarrow$ $\neg$*open(x,day)[900,1000]*,
7. *federal-holiday(day)* $\wedge$ *post-office(x)* $\Rightarrow$ $\neg$*open(x,day) [200,1000]*, and
8. *post-office(PO) [1000]*.

The significance of items (1), (5), (6), and (7) is fairly obvious. Item (2) specifies that third Mondays in February are Presidents' Days. Ideally *3rd-Mon-Feb* would be realized as a mental process (or a schema). We are indirectly simulating such a procedure by assuming that such a mental process is accessed via the relation *3rd-Mon-Feb* in order to determine whether the day bound to its role is a third Monday in February. In this example, this mental "calendar" consists of two facts stated in items (3) and (4). Item (8) states that PO is a particular post office. Items (1), (2), (6), and (7) are categorical rules about the domain, but item (5) corresponds to a default rule. We assume that "Today" is a concept which is bound each day to the appropriate date and to "weekday" or "weekend" depending on the day. These bindings are assumed to be available as facts in the agent's memory.

Imagine it is 16-Feb-98, which is Presidents' Day, and Mary is planning a trip to the post office (PO). Her "go-to-post-office" schema has the precondition that the post office must be open so it poses the query *open(PO,Today)?* Assume that after posing the query the schema monitors the activity of the collectors *+:open* and *–:open* and accepts an answer based on the criterion: *Accept a "yes" ("no") answer if the + (–) collector stays ahead and exceeds a threshold,* θ_{accept}*, for some minimum length of time,* Δ_t. Once the schema accepts an answer, it terminates the query and proceeds with its execution.

Since "Today" is bound to *16-Feb-98*, the fact *weekday(16-Feb-98)* is present in Mary's memory. When the schema asks the query *open(PO,Today)?*, the default rule about post offices remaining open on weekdays becomes active first and activates *+:open* (refer to Figures 11 and 12). If we assume θ_{accept} to be 0.5, the

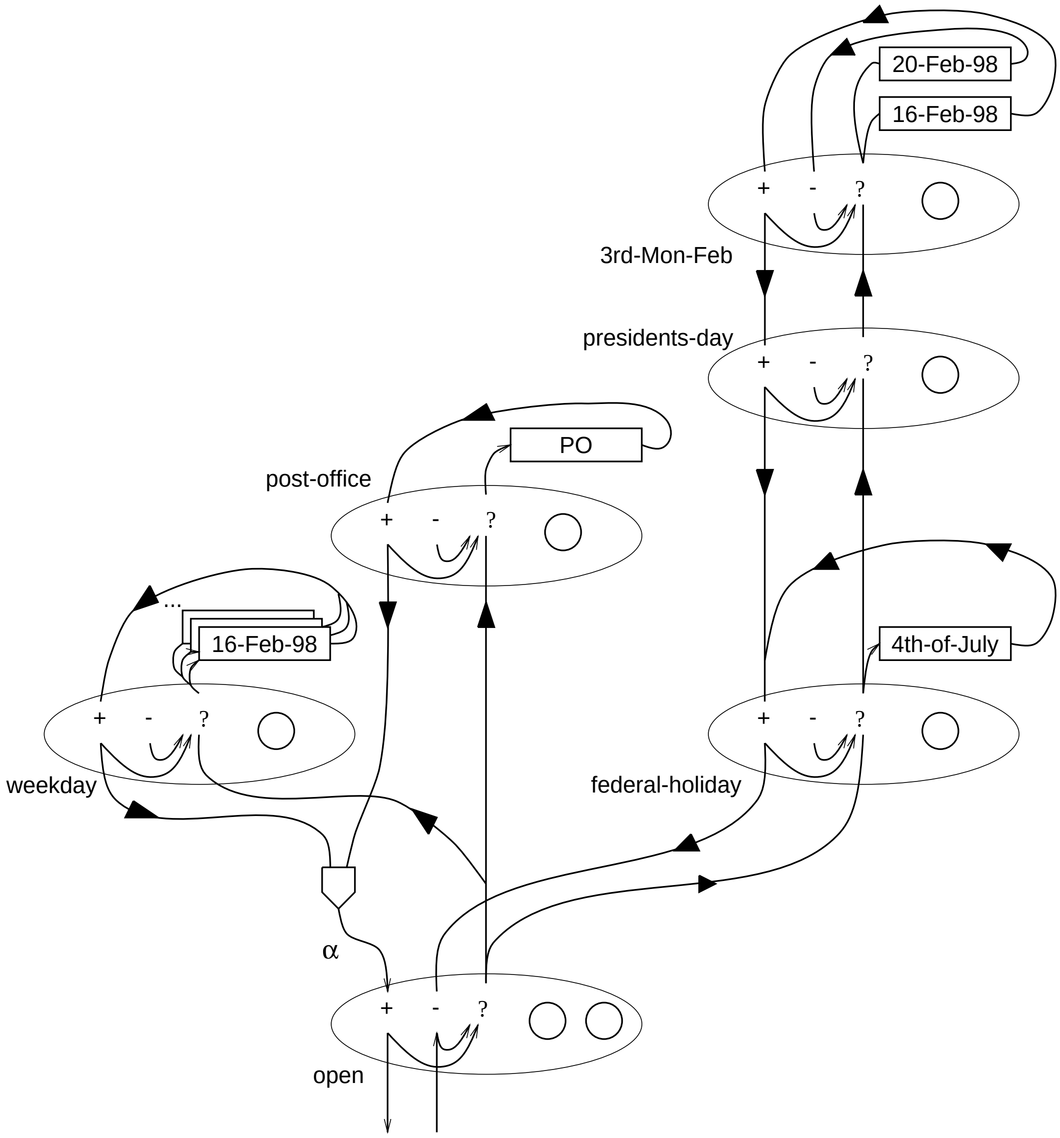

Figure 11: The network representation of the Post Office Example. Rule mediators, links between roles, detailed encoding of facts, the relation *weekend* and the encoding of rule (6) is not shown. Rules (5) and (7) are multiple antecedent rules.

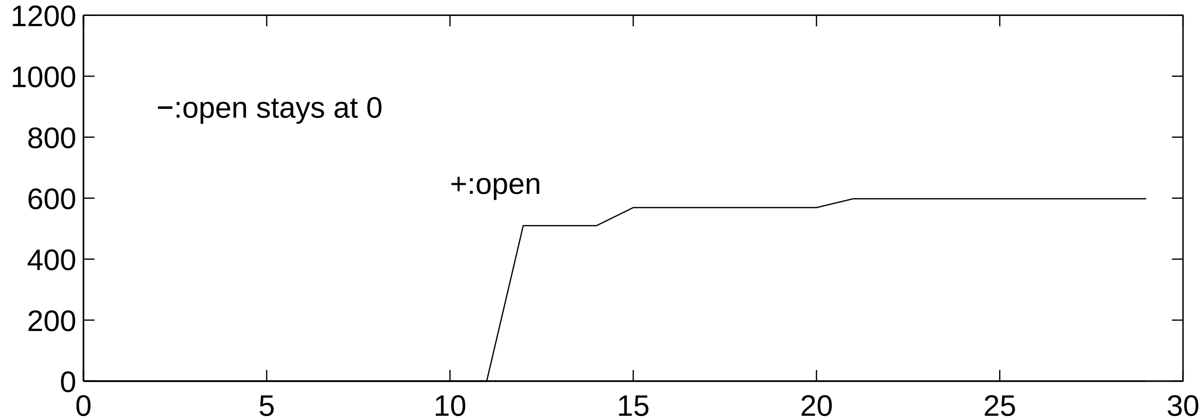

Figure 12: The activation trace for the query *open(PO,Today)?*; where today is 16-Feb-98. The vertical axis denotes activation level and has a scale factor of 1000. The horizontal axis denotes number of simulation steps.

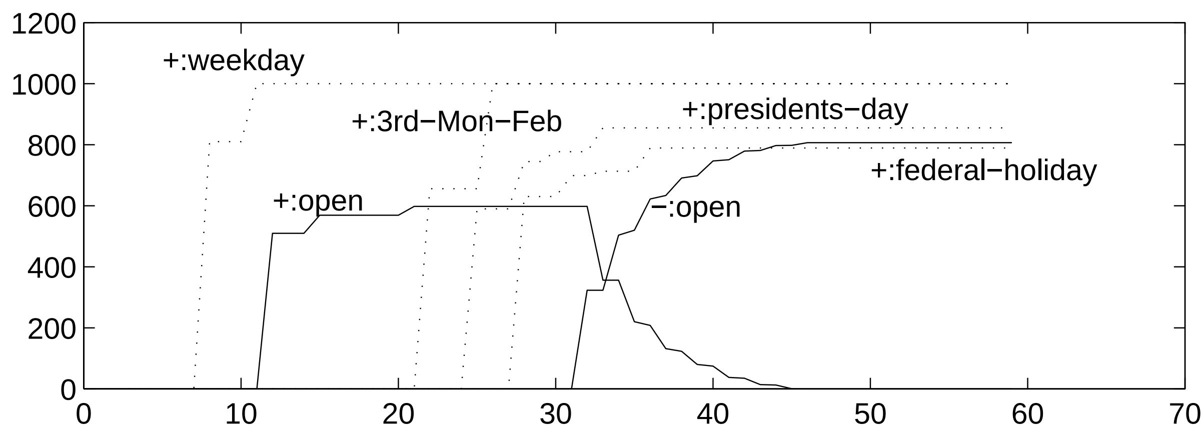

Figure 13: The activation trace for the query *open(PO,Today)?* — today being 20-Feb-98 — allowed to run its full course.

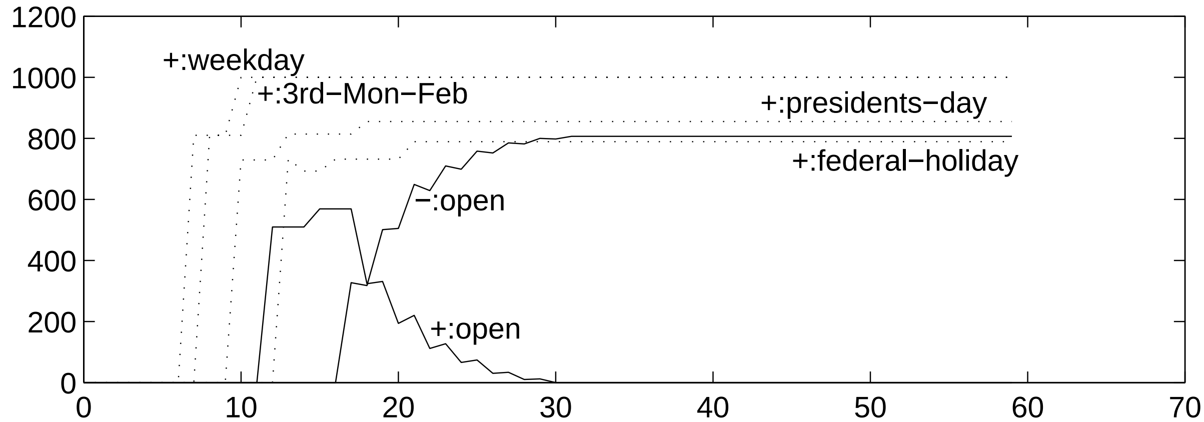

Figure 14: The activation trace for the query "Isn't today Presidents' Day?" posed to Mary on 20-Feb-98 long after her "go-to-post-office" schema has posed the query *open(PO,Today)?* and accepted a no answer.

activation of *+:open* exceeds θ_{accept} after 12 cycles and stays above threshold for about 20 cycles. During this time, *–:open* does not receive any activation and stays at 0. If we assume that Δ_t is 10 cycles, the schema will accept *+:open* as an answer and withdraw the query. So Mary will set off to the post office.[19]

Had the query remained active, the inference process would have eventually inferred that the post office is not open today. The result of the inferential process, if the query *open(PO,Today)?* had not been terminated by the schema, is shown in Figure 13. The dark lines show the activation of the collectors of *open* while the dotted lines show the activations of the collectors of some other relevant relations. First it is inferred that today is a weekday. Next it is inferred that today is the third Monday in February. As a result, the inference that today is Presidents' Day, and hence, a federal holiday, follows. This in turn leads to the inference that the post office is not open today.

Subsequently, John asks Mary: "Isn't today Presidents' Day?". This causes the language process to activate *?:Presidents-day* and bind the role of Presidents' Day to *16-Feb-98*. This leads to the activation of *?:3rd-Mon-Feb* and then *+c:3rd-Mon-Feb* (via the fact *3rd-Mon-Feb(16-Feb-98)*). The activation from *+c:3rd-Mon-Feb* works its way back and activates *–:open*. Since this activation is due to categorical rules (rules ii, i, and vii), it is stronger than that arriving at *+:open* from the default rule (item 5). The mutual inhibition between the highly activated *–:open* and the moderately activated *+:open* results in the suppression of *+:open*, making Mary realize that the post office is not open (see Figure 14).

7 Mapping *Shruti* onto Parallel Machines

There are several aspects of *Shruti* that lead to an efficient knowledge representation and reasoning system when it is mapped onto parallel machines. These include some basic features of structured connectionism and constraints on rules and derivations suggested by *Shruti*.

Shruti is a massively parallel model but makes use of nodes that perform simple computations and communicate via simple scalar messages. Secondly, unlike many neural network models, *Shruti* is *sparsely connected*. The most important source of *Shruti*'s efficiency, however, is that it is a *limited* inference system (see Section 5). The constraints imposed by *Shruti* translate into bounds on system resources and, in turn,

[19]The values of θ_{accept} and Δ_t cited above are the ones used in the simulation.

lead to a fast and efficient parallel implementation. Since the number of distinct entities that can be active during an episode of reasoning is bounded, the number of bindings contained in any instance of a relation is quite small. Consequently, the amount of information that must be communicated between relations is also quite small. Moreover, the depth of inference is bounded. This constrains the spread of activation in the network and further limits resource requirements.

There is an excellent match between the inter-node communication requirements of *Shruti* and the active message facility provided by the CM-5 for low-latency interprocessor communication of short messages. Given that nodes in our model are required to discriminate among only a small number of distinct phases, the "message" carried by a spike train can be encoded using just a few bits. Consequently, information pertaining to a whole cluster of cells can be encoded within a single CM-5 active message. Indeed, we were able to pack all the information that one relation cluster has to communicate to another relation cluster during rule-firing into a single active message!

The mapping of *Shruti* (circa 1993) onto the CM-5 is described in (Mani 1995; Shastri and Mani, 1997). The resulting system can encode knowledge bases with over 500,000 (randomly generated) rules and facts, and yet respond to a range of queries requiring derivations of depth five in under 250 milliseconds. Even queries with derivation depths of eight are answered in well under a second. The possibility of mapping *Shruti* onto a cluster/network of workstations and personal computers is also being investigated.

8 Conclusion

Shruti is a neurally plausible model of reflexive reasoning. It demonstrates that systematic inference with respect to a large body of general as well as specific knowledge can be the spontaneous and natural outcome of a neurally plausible system. *Shruti*'s encoding of commonsense knowledge corresponds to a vivid *model* of its environment and when nodes in *Shruti* are activated to reflect a given state of affairs in the environment, it spontaneously simulates the behavior of the external world and, in doing so, finds coherent explanations and makes predictions.

Shruti derives its representational and inferential power from blending structured representations with temporal synchrony. It shows that given the appropriate structure, networks that can propagate and detect synchronous activity can solve many seemingly difficult problems in the representation and processing of high-level conceptual knowledge.

Shruti 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 it is hoped that they will be explored further by experimental psychologists.

In this paper we described several enhancements of *Shruti*. These enable it to deal with negation and inconsistent beliefs, encode evidential rules and facts, perform inferences requiring the dynamic instantiation of entities, and seek coherent explanations of observations. A novel representation of type hierarchies incorporated into *Shruti* has also been described. All the features of *Shruti* presented in this article have been implemented and verified using simulations. Other enhancements listed in Section 1 will be described in detail in forthcoming publications. These include a systematic treatment of evidence combination using a family of soft evidence combination functions; priming of entities, types, facts, and rules; competition between multiple entities and context-sensitive unification of entities; and use of supervised learning to modify rule and fact strengths in order to improve the network's domain model.

A large number of issues remain open. Several of these concern learning, in particular, the learning of new relations and new rules (mappings). In (Shastri, 1997a) its is shown that a recurrent network can learn rules involving variables and semantic restrictions using gradient-descent learning. While this work serves as a proof of concept, it does not address issues of scaling and catastrophic interference. There is growing interest in the learning of structures in neural networks (e.g., Goller and Kuchler, 1996; Frasconi, Gori, and Sperduti, 1998). Some of this work has its antecedents in the work on Recursive Auto-Associative Memories (Pollack, 1990). It remains to be seen how well these techniques will scale up when applied to the domain of commonsense where the size of the "knowledge base" could easily run into a million items. Several researchers are pursuing solutions to the problem of learning structure in the context of language acquisition (e.g., Regier, 1996; Bailey, Chang, Feldman, and Narayanan, 1998; Gasser, 1998). The results of this research are expected to provide important insights into the acquisition and processing of conceptual knowledge.

In ongoing work, M.C. Cohen, B. Thompson, C. Wendelken and the author are augmenting *Shruti* to integrate the propagation of *belief* with the propagation of *utility*. The augmented model will encode beliefs, as it does now, and also encode utilities associated with the occurrences of certain world states (which states are desirable and which are not). The integrated system would be capable of using this knowledge about utility to direct its search for explanations, and predictions, and focus its activation along paths that promise to have a high utility.

The encoding of E-facts in *Shruti* solves the representational problem of encoding E-facts, but is not amenable to learning. An alternate coding that satisfies the representational requirements and at the same time is also amenable to rapid learning has been developed (Shastri, 1997b, 1998). It is shown that a dynamic representation of a relational instance (fact) can be transformed rapidly into a persistent structure within a system whose architecture and circuitry is similar to that of the hippocampal formation. In future work, we plan to integrate this model of episodic memory (*Smriti*) with *Shruti*.

We also plan to integrate *Shruti* with other neurally plausible components capable of performing syntactic processing and providing a linkage between the syntactic and semantic tiers of language. Finally, we plan to pursue a more physiologically realistic implementation of *Shruti*. It is hoped that such a simulation would enable us to investigate the effect of jitter and noise found in cortical discharges (e.g., Softky and Koch, 1992) on the propagation of dynamic bindings. This would help refine the constraints on reflexive reasoning suggested by *Shruti* and perhaps, lead to further insights into the nature of relational information processing in the mind/brain.

Acknowledgments

Work on *Shruti* is partially funded by NSF grants SBR-9720398; IRI 88-05465; ARO grants DAA29-84-9-0027 and DAAL03-89-C-0031; ONR grant N00014-93-1-1149; and subcontracts from Cognitive Technologies Inc. related to ONR contract N00014-95-C-0182 and ARI contract DASW01-97-C-0038. Thanks to fellow travellers – Venkat Ajjanagadde, Marvin Cohen, Jerry Feldman, Dean Grannes, Jerry Hobbs, George Lakoff, D.R. Mani, Srini Narayanan, Bryan Thompson, Carter Wendelken, and members of the NTL group.

References

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

Abeles, M., Bergman, H., Margalit, E., & Vaadia, E. (1993) Spatiotemporal Firing Patterns in the Frontal Cortex of Behaving Monkeys. *Journal of Neurophysiology*, 70(4):1629–1638.

Ajjanagadde, V. (1990) Reasoning with function symbols in a connectionist network. In *Proceedings of the Twelfth Conference of the Cognitive Science Society*, Cambridge, MA. pp. 285–292.

Ajjanagadde, V. (1991) Abductive reasoning in connectionist networks: Incorporating variables, background knowledge, and structured explanada, Technical Report WSI 91-6, Wilhelm-Schickard Institute, University of Tubingen, Tubingen, Germany.

Ajjanagadde, V. (1993) Incorporating background knowledge and structured explanada in abductive reasoning: A framework, *IEEE Transactions on Systems, man, and Cybernetics,* 23, 650-654.

Ajjanagadde, V. (1997) Rule-Based Reasoning in Connectionist Networks. Ph.D. Dissertation, Department of Computer Science, University of Minnesota, 1997.

Ajjanagadde, V. & Shastri, L. (1989) Efficient inference with multi-place predicates and variables in a connectionist. In *Proceedings of the Eleventh Conference of the Cognitive Science Society,* Ann-Arbor, MI, pp. 396-403.

Ajjanagadde, V. & Shastri, L. (1991) Rules and variables in neural nets, *Neural Computation* 3, 121–134.

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

Bailey, D., Chang, N., Feldman, J., & Narayanan, S. (1998) Extending Embodied Lexical Development. In *Proceedings of the Twentieth Conference of the Cognitive Science Society*, Madison, WI. pp. 84–89.

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

Bienenstock, E. (1995) A Model of Neocortex. *Network: Computation in Neural Systems* 6, 179–224.

Cohen, M.S., Freeman, J.T., & Wolf, S. (1996) Meta-Recognition in Time Stressed Decision Making: Recognizing, Critiquing, and Correcting. *Human Factors* (38,2), pp. 206-219.

Cottrell, G.W. (1989) *A connectionist approach to word sense disambiguation.* Morgan Kaufmann/Pitman.

Cottrell, G.W. (1993) From symbols to neurons: Are we yet there? *Behavioral and Brain Science*, 16:3 p.454.

Crick, F. (1984) Function of the thalamic reticular complex: The searchlight hypothesis. *Proceedings of the National Academy of Sciences*, 81:4586–4590.

Damasio, A. R. (1989) The brain binds entities and events by multiregional activation from convergence zones. *Neural Computation* 1:123-32.

deCharms, C. & Merzenich, M. (1996) Primary cortical representation of sounds by the coordination of action-potential timing. *Nature* 381:610–613.

Duncan, J., Martens, S., & Ward, R. (1997) Restricted attentional capacity within but not between sensory modalities. *Nature* 387:808–810.

Eckhorn, R., Bauer, R., Jordan, W. Brosch, Kruse, W., Munk, M. & Reitbock, H.J. (1988) Coherent oscillations: A mechanism for feature linking in the visual cortex? Multiple electrode and correlation analysis in the cat. *Biological Cybernetics* 60:121-30.

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

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

Frasconi, P., Gori, M., & Sperduti, A. (1998) A general framework for adaptive processing of data structures, *IEEE Transactions on Neural Networks*, vol. 9, no. 5, pp. 768-786, September 1998.

Freeman, W.J. (1993) Deconstruction of neural data yields biologically implausible periodic oscillation. *Behavioral and Brain Sciences*, 16:3 p. 458-459.

Gasser, M. & Colunga, E. (1998) Where Do Relations Come From? Indiana University Cognitive Science Program, Technical Report 221, January 1998.

Goller, C. & Küchler, A. (1996) Learning Task-Dependent Distributed Representations by Backpropagation Through Structure. Proceedings of the International Conference on Neural Networks, June 1996. 347-352, IEEE press.

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

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 and W. Singer. Weinheim. VSH Publishers.

Grossberg, S. & Somers, D. (1992) Synchronized oscillations for binding spatially distributed feature codes into coherent spatial patterns. In *Neural networks for vision and Image Processing* ed. G. Carpenter and S. Grossberg. MIT Press.

Hayward, R. & Diederich, J. (1997) Realising Explanation Based Generalisation. In *Connectionist Systems for Knowledge Representation and Deduction* ed. F. Maire, R. Hayward and J. Diederich. Queensland University of Technology, Neurocomputing Research Center.

Henderson, J. (1994) Connectionist Syntactic Parsing Using Temporal Variable Binding. *Journal of Psycholinguistic Research*, 23 (5) p. 353–379.

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.

Hummel, J. E., & Biederman, I. (1992) Dynamic Binding in a neural network for shape recognition. *Psychological Review*, 99, 480-517.

Hummel, J. E., & Holyoak, K.J. (1997) Distributed representations of structure: a theory of analogical access and mapping. *Psychological Review*, 104, 427-466.

Jensen, O. & Lisman, J. E. (1996) Novel lists of 7 ± 2 known items can be reliably stored in an oscillatory short-term memory network: Interaction with long term memory. *Learning and Memory*, 3, 257-263.

Just, M.A. & Carpenter, P.A. (1992) A Capacity Theory of Comprehension: Individual Differences in Working Memory. *Psychological Review*, Vol. 99, No. 1, 122-149.

Horn, D. & Usher, M. (1991) Parallel Activation of Memories in an Oscillatory Neural Network. *Neural Computation* 3, 31–43.

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

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

Lisman, J. E., & Idiart, M. A. P. (1995) Storage of 7 ± 2 Short-Term Memories in Oscillatory Subcycles. *Science*, 267, 1512-1515.

Luck, S. J. & Vogel, E. K., (1997) The capacity of visual working memory for features and conjunctions. *Nature* 390:279-281.

Mani, D.R. (1995) The Design and Implementation of Massively Parallel Knowledge Representation and Reasoning Systems: A Connectionist Approach, Ph.D. Dissertation, Department of Computer and Information Science, University of Pennsylvania, 1995.

Mani, D.R. and Shastri, L. (1993) Reflexive Reasoning with Multiple-Instantiation in a Connectionist Reasoning System with a Typed Hierarchy, *Connection Science*, Vol. 5, No. 3 & 4, p. 205–242.

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.

Milner, P. (1974): A model for visual shape recognition, *Psychological Review*, 81, pp. 521-535.

Murthy, V.N. & Fetz, E.E. (1992) Coherent 25 to 35 Hz oscillations in the sensorimotor cortex of awake behaving monkeys. *Proceedings of the National Academy of Sciences, USA*, 89, 5670–5674.

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

Niebur, E. & Koch, C. (1994) A Model for the Neuronal Implementation of Selective Visual Attention Based on Temporal Correlation Among Neurons. *Journal of Computational Neuroscience* 1:141-158.

item []O'Keefe, R.C. & Nadel, L. (1978). *The hippocampus as a cognitive map.* Oxford University Press. London.

Park, N.S., Robertson, D. & Stenning, K. (1995) An extension of the temporal synchrony approach to dynamic variable binding in a connectionist inference system. *Knowledge-Based Systems*, 8 (6), 345–358.

Pollack, J. (1990) Recursive Distributed Representations. *Artificial Intelligence*, 46, 1, 77–105.

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.

Regier, T. (1996). *The Human Semantic Potential: Spatial Language and Constrained Connectionism*, Cambridge, MA: MIT Press.

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

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) Neurally motivated constraints on the working memory capacity of a production system for parallel processing. In *Proceedings of the Fourteenth Conference of the Cognitive Science Society*, Bloomington, IN. pp. 159-164.

Shastri, L. (1993) A Computational Model of Tractable Reasoning – Taking Inspiration from Cognition. *Proceedings of IJCAI-93, the Thirteenth International Joint Conference on Artificial Intelligence*. France, pp. 202-207.

Shastri, L. (1995) Structured Connectionist Model, In *The Handbook of Brain Theory and Neural Networks*, ed. M. Arbib, MIT Press.

Shastri, L. (1996) Temporal Synchrony, Dynamic Bindings, and *Shruti*: a representational but non-classical model of reflexive reasoning, *Behavioral and Brain Sciences* Vol. 19, No. 2, 331-337.

Shastri, L. (1997a) Exploiting temporal binding to learn relational rules within a connectionist network. TR-97-003, International Computer Science Institute, Berkeley. May 1997.

Shastri, L. (1997b) A Model of Rapid Memory Formation in the Hippocampal System, In *Proceedings of the Nineteenth Annual Conference of the Cognitive Science Society*, 680-685, Stanford University, CA. August 1997.

Shastri, L. (1999a) From Transient Patterns to Persistent Structures. In preparation.

Shastri, L. (1999b) Types and Quantifiers in Shruti — a connectionist model of rapid reasoning and relational processing. To appear in *Hybrid Neural Symbolic Integration*, S. Wermter and R. Sun (Eds.), Lecture Notes in Computer Science, Lecture Notes in Artificial Intelligence Springer-Verlag, Berlin.

Shastri, L. & Ajjanagadde, V. (1990a) An optimally Efficient Limited Inference System. In *Proceedings of AAAI-90*. Boston, MA. pp. 563-570.

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

Shastri, L. & Ajjanagadde V. (1993) From simple associations to systematic reasoning: A connectionist encoding of rules, variables and dynamic bindings using temporal synchrony. *Behavioral and Brain Sciences*, 16:3 p. 417-494.

Shastri, L. & Grannes, D.J. (1996) A connectionist treatment of negation and inconsistency. In *Proceedings of the Eighteenth Conference of the Cognitive Science Society*, San Diego, CA.

Shastri, L. Grannes, D.J., Narayanan, S. & Feldman, J.A. (1999) A Connectionist Encoding of Schemas and Reactive Plans. To appear in *Hybrid Information Processing in Adaptive Autonomous vehicles*, G.K. Kraetzschmar and G. Palm (Eds.), Lecture Notes in Computer Science, Lecture Notes in Artificial Intelligence Springer-Verlag, Berlin.

Shastri, L. & Mani, D.R. (1997) Massively parallel knowledge representation and reasoning: Taking a cue from the brain. In *Parallel Processing for Artificial Intelligence 3*, ed. Geller, J., Kitano, H. and Suttner, C. Elsevier Science. 1997.

Shastri, L. & Wendelken, C. (1998) Soft Computing in *Shruti* — A neurally plausible model of reflexive reasoning and relational information processing. Submitted.

Singer, W. (1993) Synchronization of cortical activity and its putative role in information processing and learning. *Annual Review of Physiology* 55:349-74.

Singer, W. & Gray, C.M. (1995) Visual feature integration and the temporal correlation hypothesis. *Annual Review of Neuroscience* 18:555-586

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

Sougne, J. (1996). A connectionist model of reflective reasoning using temporal properties of nodes firing. In *Proceedings of the Eighteen conference of the Cognitive Science Society*, San Diego, 1996. Lawrence Erlbaum Associates

Sougne, J. (1998) Connectionism and the problem of multiple-instantiation *Trends in Cognitive Sciences*, 2, 183-189

Strong, G.W. (1993) Phase logic is biologically relevant logic. *Behavioral and Brain Sciences*, 16, 472-473.

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

Sun, R. (1992) On variable binding in connectionist networks. *Connection Science*, 4(2):93-124.

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

Usher, M. & Donnelly, N. (1998) Visual synchrony affects binding and segmentation in perception. *Nature* 394:179–182.

Valiant, L. (1994). *Circuits of the mind*. New York : Oxford University Press.

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, Germany.

Wang, D., Buhmann, J, & von der Malsburg, C. (1990) Pattern segmentation in associative memory. *Neural Computation*. 2, 94–106.

Wendelken, C. & Shastri, L. (in preparation) Evidential reasoning in structured connectionist networks.