Computer Science Department Technical Report
Artificial Intelligence Laboratory
University of California
Los Angeles, CA 90024-1596

HIGH-LEVEL INFERENCING IN A CONNECTIONIST NETWORK

Trent E. Lange
Michael Dyer

November 1989
CSD-890063

High-Level Inferencing In A Connectionist Network†*

Trent E. Lange
Michael G. Dyer
Artificial Intelligence Laboratory
Computer Science Department
University of California, Los Angeles

ABSTRACT

Connectionist models have had problems representing and applying general knowledge *rules* that specifically require variables. This *variable binding problem* has barred them from performing the high-level inferencing necessary for planning, reasoning, and natural language understanding. This paper describes ROBIN, a structured neural network model capable of high-level inferencing requiring variable bindings and rule application. Variable bindings are handled by *signatures* — activation patterns which uniquely identify the concept bound to a role. Signatures allow multiple role-bindings to be propagated across the network in parallel for rule application and dynamic inference path instantiation. Signatures are integrated within a connectionist semantic network structure whose constraint-relaxation process selects between those newly-instantiated inferences. This allows ROBIN to handle an area of high-level inferencing difficult even for symbolic models, that of resolving multiple constraints from context to select the best interpretation from among several alternative and possibly ambiguous inference paths.

1. INTRODUCTION

Critical to cognitive abilities such as natural language understanding and planning is the need to perform *high-level inferencing* to make explanations and predictions from what is known about the world. Connectionist models have been unable to perform high-level inferencing because of their difficulties with representing and applying general knowledge rules. They have so far been unable to solve this *variable binding problem*, i.e. the ability to maintain multiple variable bindings and modify them by rule application. It has recently been argued that these deficits strictly limit the usefulness of connectionist networks for modelling high-level cognitive tasks [Fodor & Pylyshyn, 1988].

This paper describes a structured connectionist model capable of variable binding and rule application. ROBIN (ROle Binding and Inferencing Network) performs high-level inferencing over structured connections of nodes that encode world knowledge in semantic networks similar to those of other models.

---

†To appear in *Connection Science, Vol. 1 (2), 1989*.

*This research has been supported in part by a contract with the JTF program of the DOD and grants from the W.M. Keck and ITA Foundations. The model has been implemented on an Apollo DN4000, donated by an Apollo Computer Inc. equipment grant to the UCLA interdisciplinary Cognitive Science Research Program. The authors can be contacted by e-mail at lange@cs.ucla.edu or dyer@cs.ucla.edu. We would like to thank Jack Hodges and the anonymous referees of *Connection Science* for their valuable comments on previous drafts of this paper.

However, ROBIN has additional node-pathway structure to handle variables and dynamic role-binding. With this structure, the model is able to maintain multiple role-bindings and propagate them along paths defined by the knowledge base's general knowledge rules, thus performing inferencing.

Although the ability to maintain variables and apply general knowledge rules is necessary for high-level inferencing, it alone is not sufficient. This is because one of the most difficult parts of the high-level inferencing problem is that of *selecting* the best interpretation from among multiple alternative and potentially ambiguous inference paths. The connectionist semantic network within which ROBIN's variable binding network structure is integrated allows a solution to this problem, since its smooth constraint satisfaction process allows automatic selection of the most-highly activated path as the network's interpretation.

2. HIGH-LEVEL INFERENCING

High-level inferencing is the ability to use previous knowledge and rules about the world to build new beliefs about what is true. In natural language understanding, for example, a reader must often make multiple inferences to understand the motives of actors and to causally connect actions that are unrelated on the basis of surface semantics alone. Complicating the inference process is the fact that language is often ambiguous on both the lexical and conceptual levels. Consider the phrase:

P1: *"John put the pot inside the dishwasher"*

Most people will conclude that John transferred a Cooking-Pot inside of a dishwasher in an attempt to get it clean. This conclusion is an example of a high-level inference. However, suppose P1 is followed by:

P2: *"because the police were coming."*

Suddenly, the interpretation selected for the word *"pot"* in P1 changes to Marijuana, and John's Transfer-Inside action becomes a plan for hiding the Marijuana from the police. This reinterpretation requires the inferences shown in Table 1 to understand the most probable *causal relationship* between the actions of phrase P1 and P2 (collectively called the **Hiding Pot** episode).

To understand episodes such as **Hiding Pot**, a system must minimally be able to dynamically make such chains of inferences (by applying general knowledge rules) and temporarily maintain them (with a variable-binding mechanism). For example, a system must know about the general concept (or *frame*) of an actor transferring himself to a location ("coming"). To represent the initial knowledge given by phrase P2 of **Hiding Pot**, the system must be able to temporarily maintain a particular *instantiation* of this Transfer-Self frame in which the Actor role (a variable) is bound to Police and the Location role is bound to the location of John. The system must also have the general knowledge that when an actor transfers himself to a location, he ends up in the proximity of that location, which might be represented as the rule:

```
R1:  [Actor X Transfer-Self Location Y]
        == results-in ==>  [Actor X Proximity-Of Object Y]
```

Applying this rule to the instantiation of the police Transfer-Self would allow the system to make inference I5 in Table 1, that the police will be in the proximity of John and his marijuana. Another piece of

I1:	If the police see John's marijuana, then they will know that he possesses an illegal object (since marijuana is an illegal substance).
I2:	If the police know that John is in possession of an illegal object, then they will arrest him, since possessing an illegal object is a crime.
I3:	John does not want to get arrested.
I4:	John has the goal of stopping the police from seeing his marijuana.
I5:	The police coming results in them being in the proximity of John and his marijuana.
I6:	The police being in the proximity of John's marijuana enables them to see it.
I7:	John's putting the marijuana inside the dishwasher results in the marijuana being inside the dishwasher.
I8:	The marijuana is inside an opaque object (the dishwasher).
I9:	Since the marijuana is inside an opaque object, the police cannot see it, thus satisfying John's goal.

Table 1: Inferences needed to understand the sentence *"John put the pot inside the dishwasher because the police were coming."* (**Hiding Pot**)

knowledge that the system must have is that an actor must be in the proximity of an object in order to see it, which might be represented as the rule:

```
R2:   [Actor X Proximity-Of Object Y]
         == precondition-for ==> [Actor X See-Object Object Y]
```

If this rule is applied to the new piece of knowledge that the Police will be in the proximity of John, then the system would be able to infer that there is the potential for them to see John and his marijuana (I6). The rest of the inferences in Table 1 to understand **Hiding Pot** are the result of the application of similar rules and knowledge about the world.

Unfortunately, even the ability to maintain variable bindings and apply general knowledge rules of the above sort is often insufficient for language understanding and other high-level cognitive tasks. This is because language is often ambiguous, as **Hiding Pot** illustrates, with several possible interpretations that must be chosen between. One of the fundamental problems in high-level inferencing is thus that of *frame selection*. When should a system make inferences from a given frame instantiation? And when conflicting rules apply to a given frame instantiation, which should be selected? Only a system that can handle these problems will be able to address the following critical tasks:

Word-Sense Disambiguation: Choosing the meaning of a word in a given piece of text. In P1, the word *"pot"* refers to a Cooking-Pot, but when P2 is presented, the evidence is that the interpretation should change to Marijuana.

Inferencing: Applying causal knowledge to understand the results of actions and the motives of actors. There is nothing in **Hiding Pot** that explicitly states that the police might see the pot (I6), or even that the police will be in proximity of it and John (I5). Nor is it explicitly stated what the police will do if they see he possesses Marijuana (I1, I2). Each of these assumptions must be inferred from phrases P1 and P2.

Concept Refinement: Instantiating an applicable specific frame from a more general one. In P1, the fact that the pot was inside a dishwasher tells us more than the simple knowledge that it was inside a container. In contrast, the salient point in **Hiding Pot** is that it is inside of an opaque object (I8), which allows us to infer that the police will not be able to see it (I9).

Plan/Goal Analysis: Recognizing the plan an actor is using to fulfill his goals. In P1, it appears that John put the pot into the dishwasher as part of the $Dishwasher-Cleaning script to satisfy his goal of getting it clean. In **Hiding Pot**, however, it appears that it is part of his plan to satisfy his sub-goal of hiding it from the police (I4), which is part of his overall goal to avoid arrest (I3).

Frame selection is complicated by the effect of additional context, which often causes *reinterpretation* to competing frames. The contextual evidence in **Hiding Pot** can conflict even more, and the explanation change again, if, for example, the next phrase is:

P3: *"They were coming over for dinner."*

As a result of P3, the word *"pot"* might be reinterpreted back to Cooking-Pot. These examples clearly point out two sub-problems of frame selection, those of *frame commitment* and *reinterpretation*. When should a system commit to one interpretation over another? And if it does commit to one interpretation, how does new context cause that interpretation to change?

3. PREVIOUS APPROACHES

Symbolic artificial intelligence (AI) systems have so far been the only types of models capable of performing high-level inferencing. A good example is BORIS [Dyer, 1983], a natural language understanding program for modelling in-depth understanding of relatively long and complex stories. BORIS had a symbolic knowledge base containing knowledge structures representing various actions, plans, goals, emotional affects, and methods for avoiding planning failures. When reading in a story, BORIS would fire rules from its knowledge base to perform inferencing and form an internal representation of the story, about which it could then answer questions. Other models that have successfully approached complex parts of the language understanding process have all had similar types of knowledge representation and rule-firing capabilities.

Connectionist networks, however, have significant potential advantages over traditional symbolic approaches to the interpretation process. Their conceptual knowledge is stored entirely in an interconnected network of simple nodes whose activations are calculated based on their previous activation and that of the nodes to which they are connected. As a result, a major portion of the understanding process is controlled by a simple spreading-activation mechanism, instead of by large collections of brittle and sometimes ad-hoc rules.

3.1. Distributed Connectionist Networks

Distributed connectionist models, such as those of [McClelland & Kawamoto, 1986] and [Touretzky & Hinton, 1988] use massively parallel networks of simple processing elements which represent knowledge as patterns of activation across a large number of nodes. Distributed connectionist models have many desirable properties, such as their learning rules that allow stochastic category generalization, their ability to perform noise-resistant associative retrieval, and their robustness to damage. Unfortunately, no distributed connectionist model has exhibited the ability to make high-level inferences even near the complexity required to understand episodes such as **Hiding Pot**. The primary reason for these limitations is their difficulties in handling variables and applying general knowledge rules. Distributed networks, furthermore, are *sequential at the knowledge level* (i.e can select and fire only one rule at a time) [Dyer, 1990] and lack the *representation of structure* needed to handle complex conceptual relationships [Feldman, 1989].

3.2. Structured Connectionist Networks

Structured connectionist models, such as those of [Cottrell & Small, 1983], [Waltz & Pollack, 1985], and [Shastri, 1988], also use massively parallel networks of simple processing elements. Structured networks, sometimes known as localist networks, represent knowledge by simple nodes and their weighted interconnections, with each node standing for a distinct concept. Activation on a conceptual node represents the amount of *evidence* available for that concept in the current context.

Unlike distributed connectionist networks, the connectivity of structured networks implicitly represents structural relationships between concepts. More importantly, structured models are *parallel at the knowledge level*, with multiple inference paths pursued simultaneously in networks whose interpretation is based on nodal activation levels. Disambiguation is achieved automatically as related concepts under consideration provide evidence and feedback to one another. Furthermore, the complicated and expensive backtracking rules of symbolic approaches are not needed when context forces a change in interpretation, since new context simply provides more activation evidence for the previously less-active interpretation.

The main problem with previous structured models is that the evidential activation on their conceptual nodes gives no clue as to *where* that evidence came from. Because of this, structured models have so far had no more success than distributed connectionist models at representing and utilizing dynamic variables and role-bindings — and thus remain unsuited to tasks requiring high-level inferencing.

4. OVERVIEW OF ROBIN'S KNOWLEDGE

ROBIN is a structured connectionist model that has all of the advantages of previous structured approaches but, in addition, resolves many of the problems of dynamic role-binding, inferencing, and frame selection. ROBIN's networks consist entirely of connectionist nodes [Feldman & Ballard, 1982] that perform simple computations on their inputs: summation, summation with thresholding and decay, or maximization. Connections between nodes are weighted, and either excitatory or inhibitory.

ROBIN uses structured connections of nodes to encode a semantic knowledge base of related frames [Minsky, 1975]. Each frame has one or more roles, with each role having expectations and logical constraints on its fillers. Every frame can be related to one or more other frames, with pathways between corresponding roles (representing general knowledge rules) for inferencing. There is no information in the knowledge base about the specific episodes (such as **Hiding Pot**) that the networks will be used to understand.

As in nearly all structured models, ROBIN's knowledge base is hand-built. The knowledge base, made up of the conceptual frames and rules needed for a given domain, is used to construct the actual networks' structure before any processing begins. After the network has been constructed, nodes in the network are clamped to represent the surface role-bindings from an episode (such as from phrases P1 and P2 of **Hiding Pot**). Activation representing role-bindings and evidence for individual concepts then spreads from the nodes representing one frame to the nodes representing related frames, thus automatically instantiating other frames and performing the processes of inferencing and frame selection.

```
(FRAME Inside-Of
      State  (Roles   (Object    (Physical-Object  0.05))
                      (Location  (Container-Object 0.50))
                      (Planner   (Human            0.05)))
             (Phrase
                 (<S_"is inside of"_DO> 1.0 (Object   Subject)
                                            (Location Direct-Object))
             (Result-Of
                 (Transfer-Inside       1.0 (Object   Object)
                                            (Location Location)
                                            (Planner  Actor))
             (Refinements
                 (Inside-Of-Stove       1.0 (Object   Object)
                                            (Location Location)
                                            (Planner  Planner))
                 (Inside-Of-Dishwasher  1.0 (Object   Object)
                                            (Location Location)
                                            (Planner  Planner))
                 (Inside-Of-Opaque      1.0 (Object   Object)
                                            (Location Location)
                                            (Planner  Planner))))
```

Figure 1. Simplified definition of the frame representing the state Inside-Of. The weights (numbers) from each of the concepts correspond to how much evidence there exists for Inside-Of given that the concept is active.

An example of how concepts are statically defined in ROBIN's general semantic knowledge bases is shown in Figure 1. The figure shows a simplified definition of the state frame Inside-Of, which represents the knowledge that an object is inside of a container. Inside-Of has three roles: an Object that is inside of something, a Location that the object is inside of, and a Planner that caused the state to be reached. The lexical phrase <Subject "is inside of" Direct-Object> directly accesses Inside-Of, as in *"the roast is inside of the stove."*

4.1. Logical Binding Constraints on Roles

Every role has *logical binding constraints* that tell which types of concepts may be bound to it. For instance, only a Stove or something that *is-a* Stove can be bound to the Location role of Inside-Of-Stove. This constraint is needed because Inside-Of-Stove is by definition a refinement of Inside-Of which allows the possible inference that the Object is being cooked. Similarly, Inside-Of-Dishwasher and Inside-Of-Opaque each have the binding constraints that their Location be something that *is-a* Dishwasher or Opaque-Object, respectively. The binding constraints defined in Figure 1 for state Inside-Of are that the Object must be some kind of Physical-Object, that the Location must be some sort of Container-Object, and that the Planner (if any) must be a Human. A role's binding constraint also serves as its *prototypical filler,* i.e. the concept that serves as the role's default binding.

4.2. Rules as Relations Between Frames

The relations that each frame has to other frames define the network's general knowledge rules and alternative inference paths. For example, in Figure 1, Inside-Of is related to four other frames. The first frame that it is related to is the action Transfer-Inside, which it is a result-of, since transferring an object inside of something results in that thing being inside of it. The Figure 1 also displays the links between corresponding roles; showing, for example, that the Object of Inside-Of can be inferred to be the same as the Object of Transfer-Inside. Defining Inside-Of's relation to Transfer-Inside in this way is equivalent to defining it in the form of a rule such as:

```
R3:   [Actor X Transfer-Inside Object Y Location Z]
        == results-in ==> [Object Y Inside-Of Location Z]
```

Finally, Figure 1 specifies that there are three potential refinement frames (Inside-Of-Stove, Inside-Of-Dishwasher, and Inside-Of-Opaque) which compete for selection as *the* refinement interpretation of a given instantiation of Inside-Of. These refinements are themselves related to the frames representing the probable reasons for the object being inside of the location. For example, in **Hiding Pot**, it initially appears that it is important that the pot is inside of a dishwasher (Inside-Of-Dishwasher), so that it could be cleaned. However, the final inference is that the salient property is that it is inside of something that is opaque (Inside-Of-Opaque), so that it will be hidden from sight. They are thus *mutually exclusive* parts of any one interpretation.

4.3. Connection Weights

Whenever a frame or concept is activated in a given context or episode, it provides a certain amount of *evidence* that the frames it is related to are activated. For example, if somebody has performed a Transfer-Inside into a container, then there is quite strong evidence that something is now Inside-Of that container. The relative levels of these amounts of evidence are built into the *connection weights* of the networks constructed from the knowledge base.

In general, weights are chosen on the basis of how much evidence the activity of the related frame (F_r) provides for the activity of frame being defined (F_d). Specifically, the connection weight from F_r to F_d is equal to the probability that F_d is active given the knowledge that F_r is active, or:

$$w_{Fr \rightarrow Fd} = P(F_d \mid F_r)$$

This method of selecting connection weights between concepts is similar to that used in the structured evidential reasoning networks of [Shastri, 1988]. Unfortunately, it is usually impossible to calculate a precise probability of one action or fact given another in the uncertain domains of natural language understanding and planning. The above weight "formula" is therefore used as a rule of thumb when creating the connection weights.

The numbers in Figure 1 specify the basic connection weights from related frames to Inside-Of and its roles. For example, if something is inside of a stove (Inside-Of-Stove), then the network can definitely infer that it is Inside-Of something, so the connection weight from Inside-Of-Stove to Inside-Of is set at a maximum (1.0 in Figure 1). If a Container-Object is active, on the other hand, there would be substantial, though not definite, evidence that something is Inside-Of something else (since there are often things inside of mentioned containers, but not always). The weight from it to Inside-Of reflects this (0.50). Finally, the fact that a Physical-Object is active in an episode provides only limited evidence for it being Inside-Of something, so a very small weight is given (0.05). The actual weight values chosen are clearly arbitrary. What is important is that they be in a *range* reflecting the amount of evidence the concepts provide for their related concepts in a certain knowledge base.

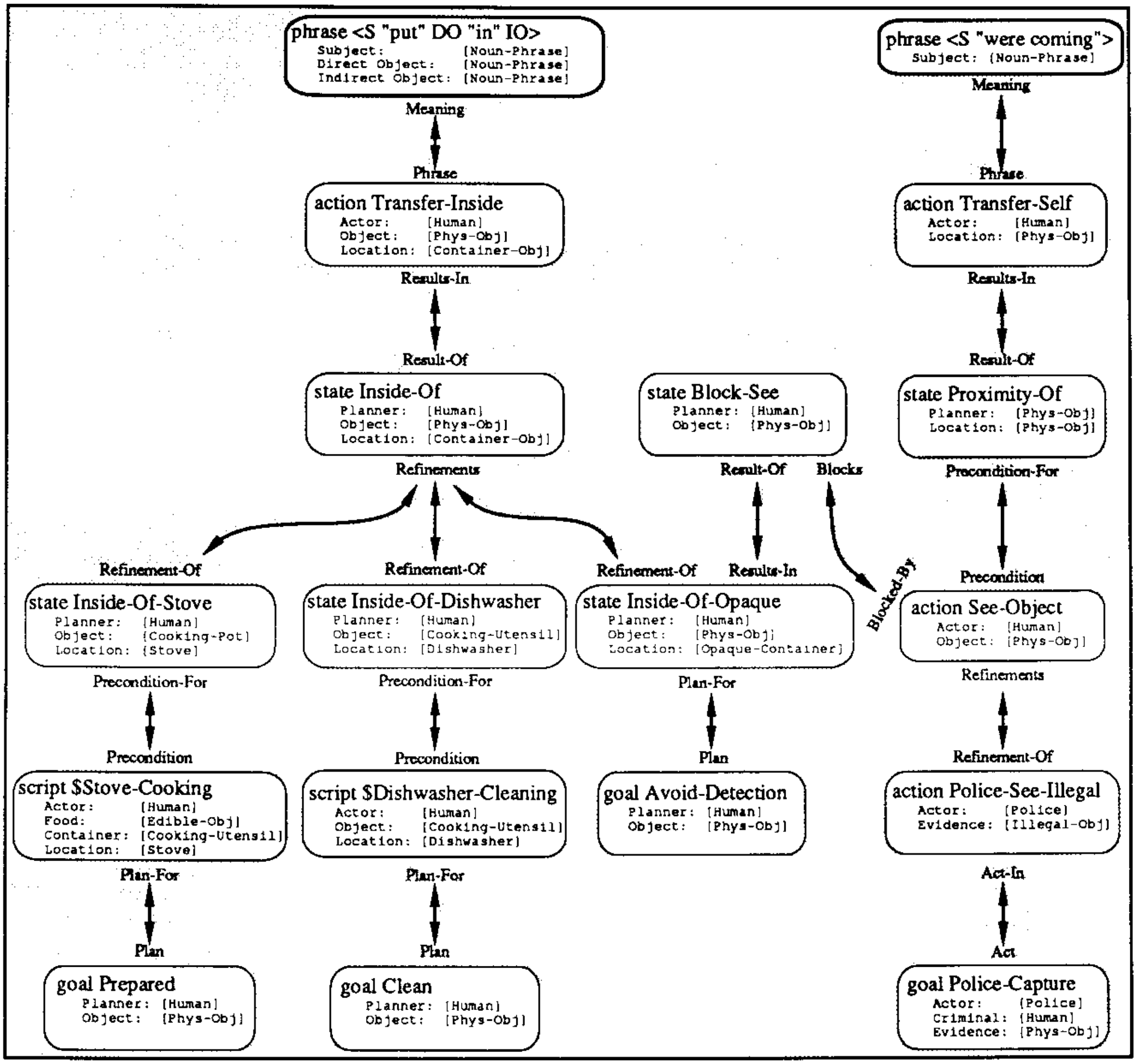

Figure 2: Overview of a relevant portion of a knowledge-base defined in ROBIN. Bracketed objects to the right of a frame's role (e.g. [Container-Obj] in the Location role of Inside-Of) represent its logical binding constraints. The symbolic frames and their relations (links) defined in the knowledge base are not actual nodes and links in the network. They are instead used to initially *construct* a portion of the network which represents them, and in which the node and link names do not affect the actual spread of activation in any way.

4.4. Overall Knowledge Bases

Individual frame definitions combine to describe ROBIN's knowledge base of concepts and rules for inferencing. Figure 2 shows an overview of a relevant portion of a knowledge base consisting of the causal dependencies relating actions, plans, goals, and scripts [Schank & Abelson, 1977]. As can be seen, rules R1-R3 are encoded by the relations between frames shown in the figure, as are a number of the other general knowledge rules necessary to understand **Hiding Pot** and related episodes.

As Figure 2 shows, every relation from one frame to another has an inverse relation. Just as a state of Inside-Of can be inferred to be a result-of a given Transfer-Inside action, one can infer that a Transfer-Inside action results-in a given state of Inside-Of. The connection weights may be different, however: if there is a Stove active in an episode, then there is definitely an Appliance (so a weight to Appliance of 1.0), but if there is an Appliance active, then it is not necessarily a Stove (so a smallish weight to Stove of ~0.2).

5. ROLE-BINDING AND INFERENCING WITH *SIGNATURES*

ROBIN's knowledge bases of frames and their relations are used to initially construct the purely connectionist networks over which inferencing is performed. As in most other structured models, there is a single node in the network for each frame or role concept. Relations between concepts are represented by weighted connections between the nodes. Activation on a conceptual node is *evidential*, corresponding to the amount of evidence available for the concept and the likelihood that it is selected in the current context.

Simply representing the amount of evidence available for a concept, however, is not sufficient for complex inferencing tasks. A solution to the variable binding problem requires that some means exist for *identifying* a concept that is being dynamically bound to a role. Furthermore, the network's structure must allow these role-bindings to propagate across node pathways that encode the knowledge base's rules, thus dynamically instantiating inference paths representing the input.

5.1. Variable Binding With Signatures

The variable and role-binding problem is handled in ROBIN by network structure holding *signatures* — activation patterns which uniquely identify the concept bound to a role [Lange & Dyer, 1988]. Every concept in the network has a *signature node* that outputs its signature, a constant activation value different from all other signatures. A dynamic binding exists when a role or variable node's *binding node* has an activation matching the activation of the bound concept's signature.

In Figure 3a, the *virtual binding* of the Actor role node (of action Transfer-Inside) to John is represented by the fact that its binding node (the solid black circle) has the same activation (3.1) as John's signature node. The same binding node could, at another time, hold a different virtual binding, simply by having the activation of another concept's signature (as in Figure 3b, where it is bound to Police). The complete Transfer-Inside frame is represented in the network by the group of nodes that include the conceptual node Transfer-Inside, a conceptual node for each of its roles (only the Actor role shown), and the binding nodes for each of its roles.

5.2. Structure of the Network

The most important feature of signature activation is that it propagates across paths of binding nodes to generate candidate inferences. Figure 4 illustrates the structure of the network that automatically accomplishes this.

The conceptual nodes and connections on the bottom plane of Figure 4 (i.e. Transfer-Inside and its Object role) are part of the normal semantic network constructed from the knowledge base of Figure 2. Nodes and connections for the Actor, Planner, and Location roles are not shown. The connections between nodes on this

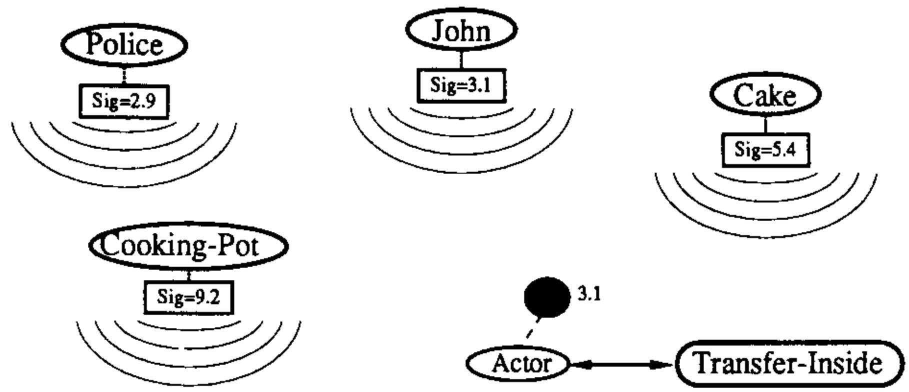

Figure 3a. Several concepts (ovals) and their uniquely-identifying signature nodes (rectangles) are shown, along with the Actor role of the Transfer-Inside frame. The Actor role has a *virtual binding* to John because its binding node (black circle) has the same activation (3.1) as John's signature.

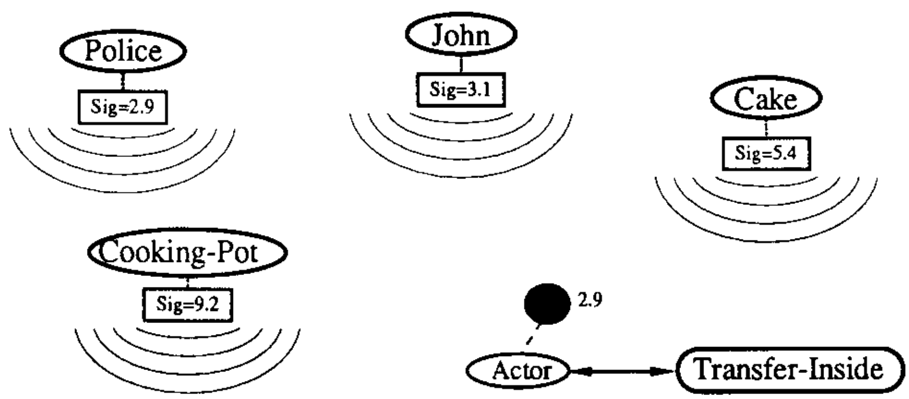

Figure 3b. The same binding node holding a different virtual binding, this time to Police.

bottom plane are specified by the frame definitions of the knowledge base. For example, the weighted connection from node Transfer-Inside to node Inside-Of represents the result-of relation defined in Figure 1. As in other structured models, activation propagating across this structure of the network is *evidential*.

The top plane of Figure 4, on the other hand, consists of the network's binding and signature nodes, over which *signature* activation (representing dynamic role-bindings) spreads. Each role has several binding nodes (two of which are shown). There are no connections from signature nodes to binding nodes, but there are *unit-weighted* connections between *corresponding* binding nodes over which inferences can be made. For example, the filler of Inside-Of's Object role can be inferred to be the same as the filler of Transfer-Inside's Object (as defined in Figure 1). There is therefore a connection from the left binding node of Transfer-Inside's Object to the left binding node of Inside-Of's Object. A similar link goes between the right binding nodes, as well as one-to-one connections from the (unseen) binding nodes of Transfer-Inside's other roles (Actor and Location) to the binding nodes of Inside-Of's corresponding roles (Planner and Location, respectively).

5.3. Activation Functions

There are different activation functions for the conceptual nodes of the bottom evidential layer and the binding nodes of the top signature layer.

5.3.1. Activation of Concept Nodes

The activation function of the network's conceptual nodes is equal to the weighted sum of their inputs plus their previous activation times a decay rate, or:

$$a_c(t+1) = \sum_i w_{ic} o_i(t) + a_c(t)(1-\Theta)$$

where $a_c(t)$ is the activation of conceptual node c at cycle t, w_{ic} is the incoming weight from node i to node c, $o_i(t)$ is the output of node i at cycle t, and Θ is the activation decay rate of conceptual nodes when they are receiving no input. The output function of the conceptual nodes is a simple linear threshold:

$$o_c(t) = \begin{cases} a_c(t) & a_c(t) \geq \theta \\ 0 & \text{otherwise} \end{cases}$$

where θ is the output threshold of all of the conceptual nodes. Generally, the activation values of the conceptual nodes in the network range from 0 to about 1. The output threshold θ is set quite low (usually around 0.05), and so is not much of a factor in the spread of activation. The net effect of the activation and output functions of the conceptual nodes is to allow them to "weigh" evidence from their related concepts, with nodes in paths between multiple sources of activation (i.e. in part of an inference chain between two phrases) tending to reinforce each other.

5.3.2. Activation of Binding Nodes

The activation and output functions of the binding nodes are equal to the *maximum* of their unit-weighted inputs, or:

$$a_b(t+1) = \mathrm{MAX}\,(w_{1b}o_1(t), w_{2b}o_2(t), \ldots w_{nb}o_n(t)), \quad o_b(t) = a_b(t)$$

where $a_b(t)$ is the activation of binding node b at cycle t, and $o_1 \ldots o_n$ are the outputs of all binding nodes that have incoming links to binding node b. Since all of the w_{ib} to binding nodes have unit weight, this causes the activation of a binding node to take on the activation of any of its active incoming binding nodes — and therefore allows signatures to be propagated without alteration.

5.4. Propagation of Signatures for Inferencing

Initially there is no activation on any of the conceptual or binding nodes in the network. To initiate the inferencing process, a phrase and its role-bindings are presented to the network by hand-clamping the proper roles' binding nodes to the signatures of the concepts bound in the phrase[1]. Thus, for phrase P1

[1]ROBIN does not currently address the problem of performing the original syntactic role-binding assignments, i.e. that *"pot"* is bound to the Object role of the phrase. Rather, ROBIN's networks are given these initial bindings and use them for high-level inferencing.

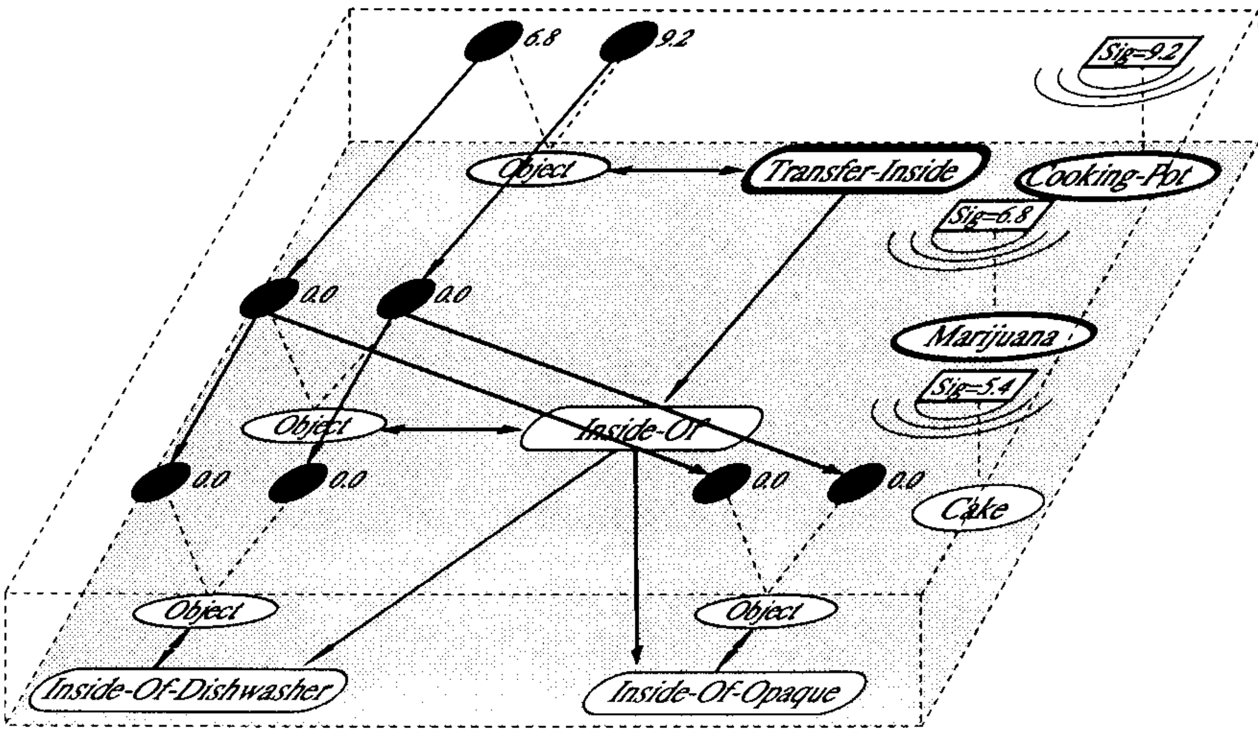

Figure 4. Simplified ROBIN network segment showing the parallel paths over which evidential activation (bottom plane) and signature activation (top plane) are spread for inferencing. For simplicity, the nodes and connections representing Inside-Of-Stove and the rest of the semantic network not shown. The figure shows the initial activation and clamping for phrase P1 of **Hiding Pot**. Signature nodes (outlined rectangles) and binding nodes (solid black circles) are in the top planes. Thickness of conceptual node boundaries (ovals) in the bottom plane represents their levels of evidential activation. (Node names do not affect the spread of activation in any way. They are simply used to initially set up the network's structure and to aid in analysis.)

("John put the pot inside the dishwasher"), the binding nodes of Transfer-Inside's Actor, Object, and Location roles are clamped to the signatures of the concepts meant by the words "John", "pot", and "dishwasher", respectively. For example, the binding nodes of Transfer-Inside's Object are clamped to the activations 6.8 and 9.2, which are the signatures for objects Marijuana and Cooking-Pot, respectively, representing the candidate bindings from the word "pot" (Figure 4)[1].

At the same time, the lexical concept nodes for each of the words in the phrase are clamped to a high level of evidential activation. In the case of phrase P1, this clamping directly ends up providing evidential activation for concepts John (from lexical node "John"), Transfer-Inside (from phrase <S "put" DO "in" IO>), Cooking-Pot and Marijuana (from lexical node "pot"), and Dishwasher (from lexical node "dishwasher").

---

[1]An alternative input, such as *"George put the cake inside the oven"*, would be done simply by clamping the signatures of its bindings (i.e. George, Cake, and Oven) instead. A completely different set of inferences would then ensue. This is unlike previous localist models, where all combinations of possible instantiations had to be hard-wired into the network.

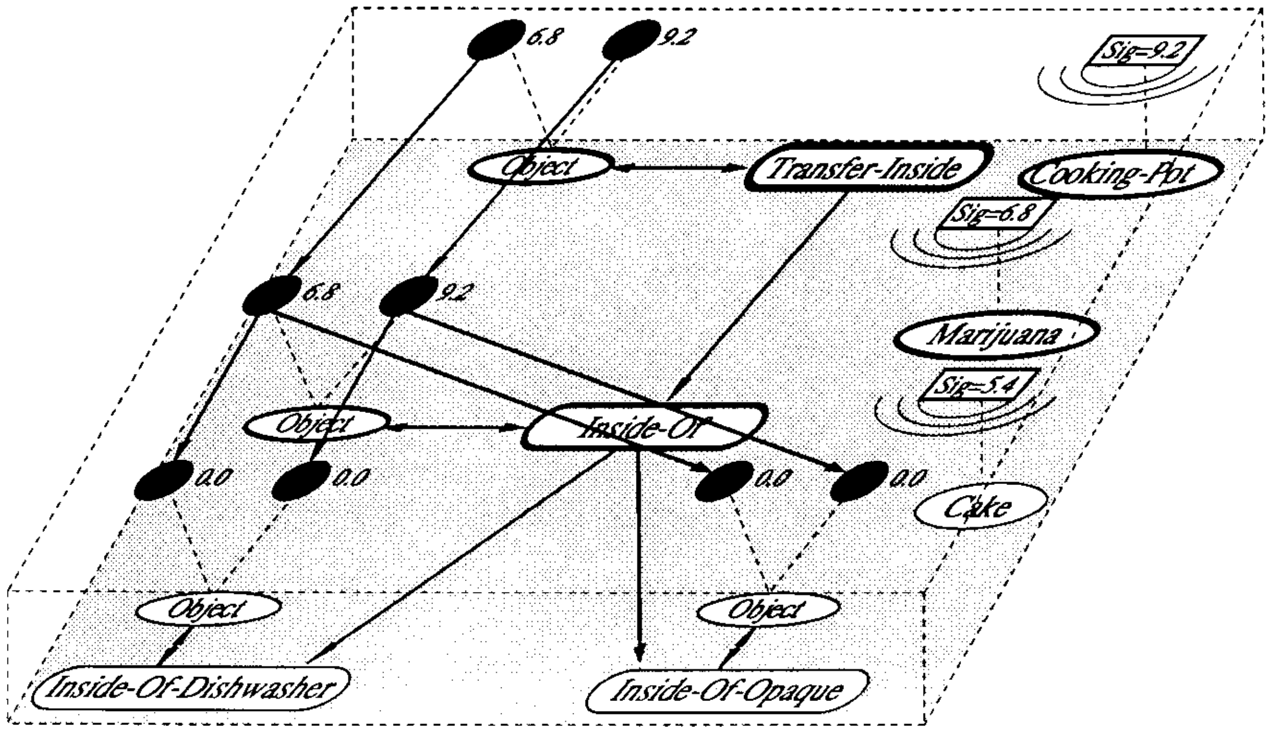

Figure 5. Evidential and signature activations in **Hiding Pot** after reaching Inside-Of.

With the original role-bindings thus input to the network, activation starts to spread from the initial clamped activation values. In Figure 5, Inside-Of receives evidential activation from Transfer-Inside, representing the strong evidence that something is now inside of something else. Concurrently, the signature activations on the binding nodes of Transfer-Inside's Object propagate to the corresponding binding nodes of Inside-Of's Object (Figure 5), since each of the binding nodes calculates its activation as the maximum of its inputs.

For example, Inside-Of's left Object binding node has only one input connection, that from the corresponding left Object binding node of Transfer-Inside. Since the connection has a unit weight and the left Object binding node of Transfer-Inside has an activation of 6.8, the activation of Inside-Of's left Object binding node also becomes 6.8 (Marijuana's signature). The potential binding of Cooking-Pot (signature 9.2) to Inside-Of's right Object binding node propagates at the same time, as do the bindings of Inside-Of's Planner role to the signature of John and its Location role to the signature of Dishwasher.

By propagating signature activations from the binding nodes of Transfer-Inside to the binding nodes of Inside-Of, the network has thus made its first inference. Because of the signatures now on Inside-Of's binding nodes, the network not only represents that something is inside of something else, but also represents the inference of *exactly which thing is inside the other* (I7 in Table 1).

From the activations of this newly inferred piece of knowledge, the network continues to make subsequent inferences. Evidential activation next spreads from Inside-Of to its refinements Inside-Of-Dishwasher and Inside-Of-Opaque. At the same time, the signatures on the binding nodes of Inside-Of's roles propagate to Inside-Of-Dishwasher and Inside-Of-Opaque's corresponding binding nodes, instantiating them (Figure 6). The network thus makes the inference that the reason for the Marijuana or a Cooking-Pot being inside of the Dishwasher was either because it is a dishwasher, or because it is opaque. From there, the signature

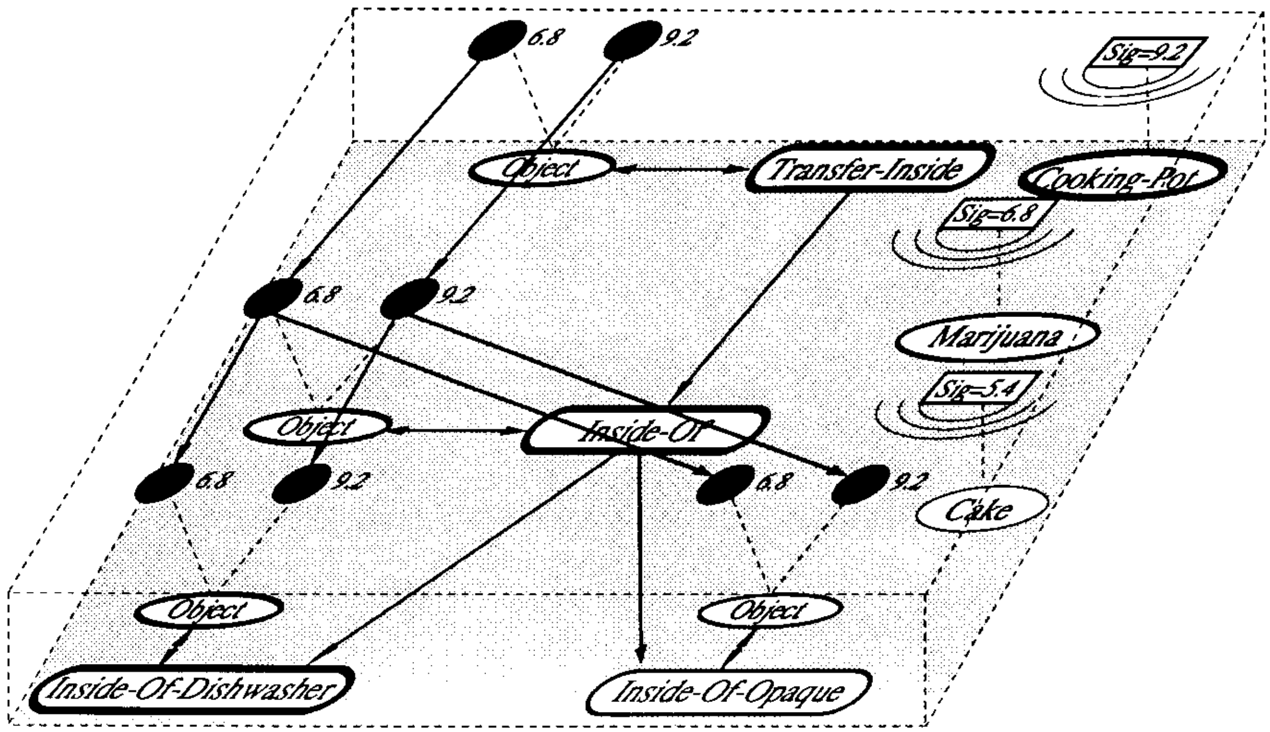

Figure 6. Activation after quiescence has been reached in processing for phrase P1. Cooking-Pot and Inside-Of-Dishwasher have higher (evidential) activations than Marijuana and Inside-Of-Opaque, as is illustrated by their thicker ovals.

and evidential activations continue to propagate through other parts of the network structure constructed from the definitions of Figure 2, instantiating the chains of related concepts down to the Clean goal, with some activation reaching goal Avoid-Detection, state Block-See, and so on.

Signatures thus propagate without change over the inference binding paths of the network constructed by the definitions of the knowledge base. As a result, ROBIN is able to dynamically instantiate inference paths and distinguish each of their inferred role-bindings.

5.5. Discussion of Signatures

There are a couple of points that is important to make about signatures. The first is that their actual or relative activation values do not affect the network's processing. The signatures of Marijuana and Cooking-Pot were arbitrarily chosen to be 6.8 and 9.2 when the network was constructed. However, they could just as easily have been chosen to be any other (even the reverse) values. It is only necessary that each signature be different from all others — and thus uniquely identify the concept bound to a role.

The second point is that a signature *can* happen to have the same activation value as the evidential activation on a conceptual node. The reason for this is that the paths over which evidential and signature activation spread are parallel to and completely separate from each other (i.e. along the bottom and top planes in Figure 6). It is therefore irrelevant whether or not a conceptual node coincidentally has an activation that is the same as some concept's signature. The activation on a conceptual node is always interpreted as the amount of evidence for that concept in the current context, while the activation on a binding node is always interpreted as a signature representing a role-binding.

6. FRAME SELECTION WITH EVIDENTIAL ACTIVATION

The ability to maintain variable bindings and propagate them throughout the network is critical for high-level inferencing. However, being able to dynamically generate inference paths alone is not sufficient for cognitive tasks such as natural language understanding and planning. The problem is that there are generally *multiple* alternative inference paths possible, only one of which best explains the input. Choosing the single most-plausible interpretation in a given context is one of the most difficult problems in high-level inferencing, that of *frame selection* [Lytinen, 1984] [Lange & Dyer, 1989].

An example of the need for frame selection can be seen in Figure 6, where both Inside-Of-Dishwasher and Inside-Of-Opaque have been instantiated with signatures inferred from phrase P1 ("John put the pot inside the dishwasher"). The inference path representing P1 has therefore already split into two alternatives: one *candidate path* that includes Inside-Of-Dishwasher, which is part of the interpretation that John is trying to clean the pot, and another candidate path including Inside-Of-Opaque, which is part of the interpretation that he is trying to hide it.

The network must somehow be able to weigh the evidence for each of these two alternative refinements of frame Inside-Of so that the most plausible of the two inference paths can be selected. With only phrase P1, the evidence appears to be that Inside-Of-Dishwasher is the best interpretation, but when P2 ("because the police were coming") is presented, it appears that the Inside-Of-Opaque path is the most likely.

Besides the problem of selecting the concept refinement of frame Inside-Of, **Hiding Pot** also requires word-sense disambiguation to select the appropriate meaning of the word *"pot"*. The same contextual evidence that causes Inside-Of-Dishwasher to be selected in P1 should cause Cooking-Pot interpretation of *"pot"* to become chosen, while the switch to Inside-Of-Opaque with evidence from P2 should cause a reinterpretation to Marijuana.

6.1. Selection by the Evidential Semantic Network

Deciding between the competing inference paths instantiated by signature activation is the function of the evidential portions of ROBIN's networks (such as the conceptual nodes on the bottom layer of Figure 6). The activations of the conceptual frame nodes are always approximately proportional to the amount of evidence available for them from their bindings and their related frames. The inference path selected as the interpretation in any given context is therefore simply *the most highly-activated path of frame nodes* and their bindings[1].

Thus, if the conceptual node for Inside-Of-Dishwasher has a higher level of (evidential) activation at stability than the node for Inside-Of-Opaque, then it will be selected as the refinement of Inside-Of and become part of the winning inference path. On the other hand, if Inside-Of-Opaque has the higher level of activation, then it will be selected.

---

[1]The network's "decision" or "selection" is actually simply the interpretation that the human modeler gives to the levels of activation present in it, as in all connectionist models.

6.2. Selection of Ambiguous Role-Bindings

Besides being able to select the most-highly activated inference path, the network must be also able to decide between ambiguous role-bindings. *All* meanings of an ambiguous word are bound to a role with signature activation, as was shown in Figure 4. Each role has several binding nodes, so that multiple candidate bindings for a given role may be propagated through the network at once. In general, this requires that there be as many binding nodes per role as there are possible meanings of the most ambiguous word in the network. In the **Hiding Pot** network, for example, there are actually three (though Figures 4 through 6 show only two) — since "pot" can mean Marijuana, Cooking-Pot or Planting-Pot. When a word bound to a role is unambiguous (like "dishwasher"), the extra binding nodes simply remain inactive.

Though having enough binding nodes per role to allow simultaneous propagation of ambiguous bindings increases the size of the network by a small constant factor, it is crucial for resolving ambiguities in context. The network's interpretation of which binding is selected at any given time is the one whose *conceptual node has greater evidential activation.* Because all candidate bindings propagate at once, with none being discarded until processing is completed, ROBIN is able *to handle meaning reinterpretations without resorting to backtracking.*

6.3. Structure of the Evidential Network

The structure of the conceptual layer and the activation function of the conceptual nodes (Section 5.3.1) is constructed so that the activation of each conceptual node is always approximately proportional to the amount of evidence available for it. For example, when only P1 ("John put the pot inside the dishwasher") is presented, there is more evidence and evidential activation for Cooking-Pot and Inside-Of-Dishwasher than for Marijuana and Inside-Of-Opaque. When the rest of **Hiding Pot** is presented, however, the balance of evidence — and thus evidential activation — shifts to Marijuana and Inside-Of-Opaque.

In general, a *candidate frame* is likely to be active if one or more of its instantiating frames are active. For example, if there was a Transfer-Inside of an object to a location, then there is strong evidence that the object is Inside-Of that location. The higher the activation of the instantiating frame, and the stronger the connection weight from it to the candidate frame, the more likely that the candidate frame is active. Similarly, a candidate frame can receive evidential activation from all of the frames that are directly related to it, and from the evidential activation of each of its conceptual roles.

Simply having frames receive evidential activation through direct connections from their related frames, however, would cause serious problems. Candidate frames that have potential relations to a large number of frames would always win out over candidates that have a smaller number of related frames. The activation of Inside-Of, for example, would always dominate Transfer-Inside, simply because Inside-Of has a very large number of potential refinement frames. In reality, however, those refinements are mutually exclusive, and only one will be chosen as *the* refinement of a given instantiation of Inside-Of. Thus, the only refinement relation that actually provides evidence for Inside-Of at a given cycle is the one that is most active.

Because of this, connections from related frames pass through an *input branch* node for their relation before they are received by the candidate frame. This is shown in Figure 7, which displays the connections between nodes on a portion of the evidential network centering around frame Inside-Of. For example, the weighted connections from Inside-Of-Dishwasher, Inside-Of-Opaque, and Inside-Of-Stove go into Inside-Of's refinements branch rather than directly into Inside-Of. Relation input branches calculate their acti-

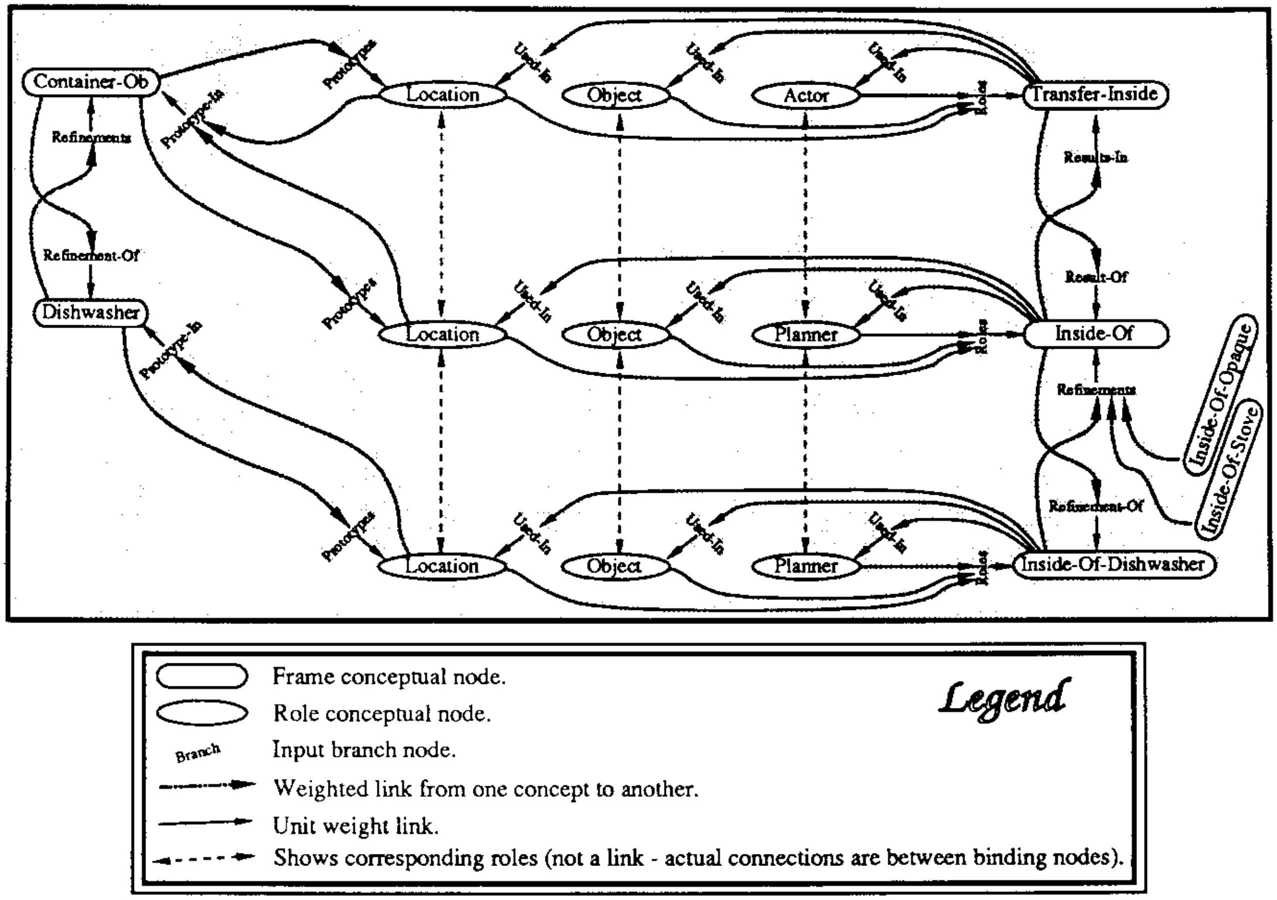

Figure 7. Detailed view of the evidential portion of the network for frame Inside-Of and part of frames Transfer-Inside and Inside-Of-Dishwasher. As usual, the node names are used only for initial construction of the network, and do not affect the spread of activation.

vation as the maximum of their inputs — so that only the currently selected (i.e. maximally activated) interpretation provides evidence for the frame[1].

Similarly, connections providing evidence from the activation of a frame's roles pass through its Roles input branch (as do the Location, Object, and Planner roles of Inside-Of in Figure 7). If none of the roles of a given frame are active, then that frame should receive no evidence from its roles. If all of them are active, then the frame should get maximum role evidence. Ratios in between should provide proportionate evidence. The amount of this evidence should not vary with the number of roles that a frame has: a frame with only a single role should receive just as much activation evidence if its one role is active as a frame with many roles that are all active. Hence the frame's Roles input branch calculates its activation as the *average* of its inputs.

---

[1]Input branches are analogous to the input *sites* on [Cottrell & Small, 1982]'s "case" units, which were used to make sure that the each case unit only received activation from its maximally activated prototypical filler and predicate.

Role nodes, like frame nodes, have several input branches. A role, however, gets evidence only from its competing prototypical fillers (the Prototypes branch) and from the frames that it is used in (the Used-In branch). Each branch calculates its activation as the maximum of its inputs, like the relation input branch for frames. With these evidential connections, the Location role of Inside-Of-Dishwasher, for example, will become activated if either its frame (Inside-Of-Dishwasher) or its prototypical filler (Dishwasher) is activated (Figure 7).

6.4. Activation Control

A major issue for all structured connectionist networks is controlling the spread of activation. Other spreading-activation models have usually addressed this problem by using direct inhibitory connections between competing concepts (e.g. [Waltz & Pollack, 1985]). For inferencing tasks, however, the inhibitory connections that these networks use are usually semantically unjustifiable and combinatorially explosive. The biggest problem, however, is that they are *winner-take-all networks*, acting to kill the activations of input interpretations that do not win the competition. This becomes a problem when a new context arises that makes an alternative interpretation more plausible. With the activations of the alternative interpretations killed by the inhibition from the false winner, it is exceedingly difficult for the activation from the new context to revive the correct one. The automatic backtracking capabilities of the networks are thus sabotaged.

ROBIN, on the other hand, has no inhibitory links between competing concepts. It instead uses a group of nodes which act as a global inhibition mechanism. These *global inhibition* nodes (Figure 8) serve to inhibit by equal proportions (short-circuit) all concepts in the network when their average activation becomes too high [Lange, 1990]. The concepts in the network are thus free to keep an activation level relative to the amount of evidence in their favor. Global inhibition nodes are similar to the "regulator units" of [Touretzky & Hinton, 1988], except that their regulator units are *subtractive inhibitory*, subtracting a constant amount of activation from all nodes and implementing a winner-take-all network, while ROBIN's global inhibition nodes are *short-circuiting inhibitory*, controlling the spread of activation, but leaving *relative* values of evidential activation unchanged.

The activation function of conceptual nodes shown in Section 5.3.1 is incomplete, having left out the short-circuiting inhibition term. The actual activation function for the conceptual nodes in the network is:

$$a_c(t+1) = \frac{\sum_i w_{ic} o_i(t) + a_c(t)(1-\Theta)}{a_g(t)}$$

where the other terms are as before, and $a_g(t)$ is the activation at cycle t of the conceptual global inhibitor node.

Because ROBIN's short-circuiting global inhibition mechanism allows all concepts in the network to hold a level of evidential activation relative to the amount of evidence in their favor (as opposed to driving the "losers" down to 0 using a winner-take-all network), ROBIN is able to easily perform reinterpretation. When new context that favors an alternative interpretation over a previous one enters the network, it boosts the new interpretation's relative level of evidential activation — often being enough to cause the new interpretation to become most highly-activated. This occurs in **Hiding Pot**, in which the evidence from P1 ("John put the pot inside the dishwasher") initially favors Cooking-Pot, but in which later evidence from the context of P2 ("the police were coming") causes a reinterpretation to Marijuana.

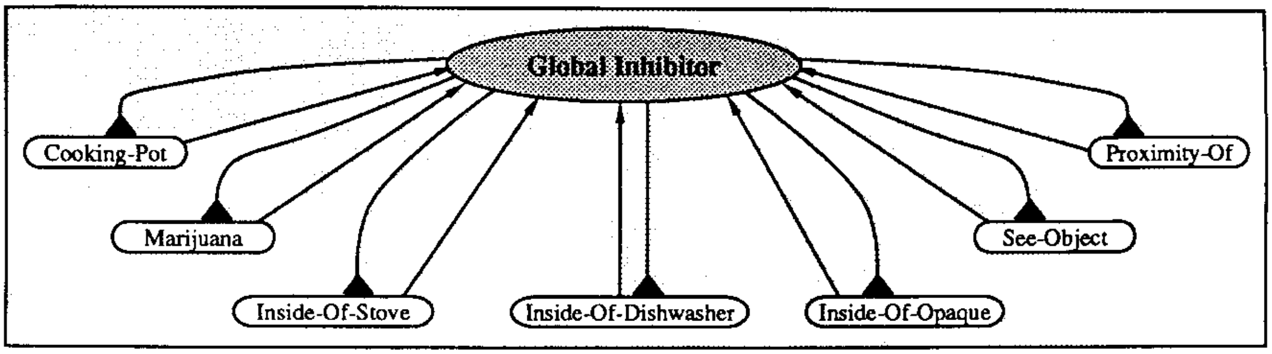

Figure 8. A ROBIN global inhibition node. Inputs to global inhibitor nodes from concept nodes are unit weighted, while outputs to concept nodes are "short-circuiting" inhibitory. The higher the total incoming activation to the global inhibitor, the greater the activation that all the concepts are divided by.

6.5. Spread of Activation in Hiding Pot

With the basic structure of signature propagation and the evidential network described, we can now explains what occurs as the rest activation spreads in **Hiding Pot**. The full knowledge base needed to understand **Hiding Pot** (Figure 2) is embedded in the network using the signature structure of Figure 4 integrated with the evidential network structure of Figure 7.

As described previously, both evidential and signature activation originally spread from the clamped activations representing the input of phrase P1 ("John put the pot inside the dishwasher") (Figures 4-6). Those activations continue to propagate along the chain of related concepts down to instantiate the Clean goal, with some activation reaching goal Avoid-Detection.

At this point, Cooking-Pot receives more evidential activation than Marijuana (Figure 6) by connections from the highly stereotypical usage of the dishwasher for the Clean goal. Inside-Of-Dishwasher has more activation than Inside-Of-Opaque for the same reason. Because of this, the network's "decision" between the two candidate bindings would be that it was a Cooking-Pot that was Inside-Of the Dishwasher, and that Inside-Of-Dishwasher was the correct refinement of Inside-Of.

To continue processing of **Hiding Pot**, phrase P2 ("because the police were coming") is next presented to the network by clamping lexical nodes "police" and <S "were coming"> to high evidential activation values and by clamping one of the binding nodes of Transfer-Self's Actor to the signature of Police (2.9 in Figure 3b). Evidential and signature activation then spreads along the path from Transfer-Self to both goals Police-Capture and Avoid-Detection, until the activation of the network finally settles.

The result is illustrated in the network overview of Figure 9, which shows the inferences made and the activation of the frames after processing of **Hiding Pot**. The network has made inferences I1-I9 of Table 1, with most being shown in the figure. For example, I8 (the inference that the Marijuana is inside of an opaque object) is represented by the instantiation of state Inside-Of-Opaque. The role-bindings of the frames shown were instantiated dynamically with signature activation.

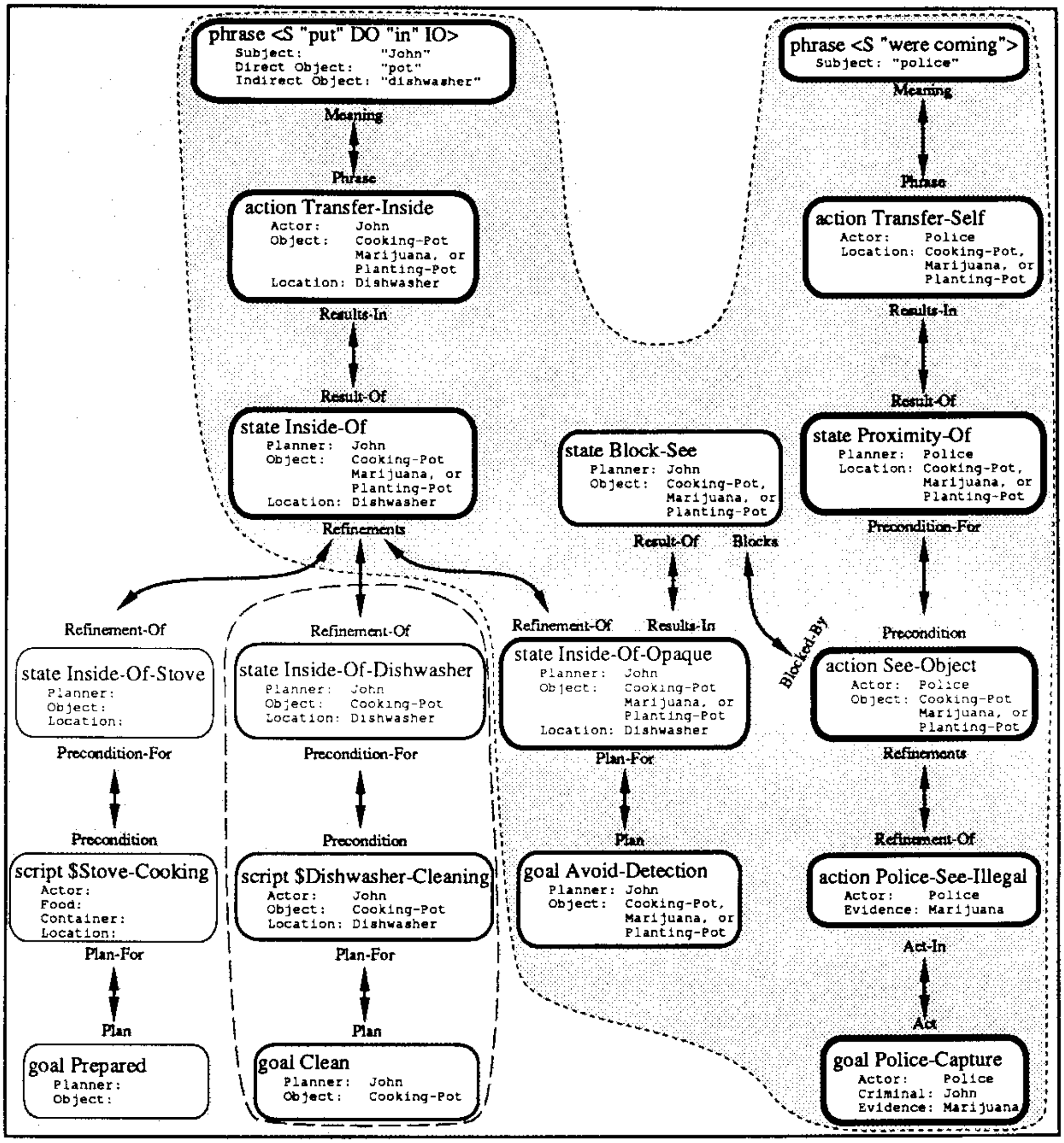

Figure 9. Overview of a small portion of a ROBIN semantic network (actually embedded in network structure such as in Figures 4 and 7) showing inferences dynamically made after clamping of the inputs for phrases P1 and P2 of **Hiding Pot**. Thickness of frame boundaries shows the amount of *evidential* activation on the frames' conceptual nodes. Role fillers shown are the ones dynamically instantiated by propagation of *signature* activation over the role's binding nodes. Darkly shaded area indicates the most highly-activated path of frames representing the most probable plan/goal analysis of the input. Dashed area shows the discarded dishwasher-cleaning interpretation. Frames outside of both areas show a very small portion of the rest of the network. These frames received no evidential or signature activation from either phrase.

Because of reinforcement and feedback from the inference paths generated by the Police's Transfer-Self, Marijuana ends up with a greater level of evidential activation than Cooking-Pot or Planting-Pot. Marijuana is thus selected over the Cooking-Pot and Planting-Pot *bindings* throughout the network. Inside-Of-Opaque also ends up with more evidential activation than Inside-Of-Dishwasher, so Inside-Of-Opaque is selected as the refinement of Inside-Of. The final interpretation of **Hiding Pot** is the most highly-activated evidential path of frames and their bindings inside the darkly shaded area.

6.6. Evidential vs Signature Activation

It is important to emphasize the differences between ROBIN's evidential and signature forms of activation. Both are simply activation from a computational point of view, but they propagate across separate pathways and fulfill different functions.

Evidential Activation:

1) *Previous work* — Similar to the activation of other structured models.
2) *Function* — Activation on a conceptual node represents the amount of evidence available for a node and the likelihood that its concept is selected in the current context.
3) *Node pathways* — Evidential activation spreads along weighted link pathways between related frames.
4) *Dynamic structure* — Evidential activation decides among candidate structures; i.e. Marijuana is more highly-activated than Cooking-Pot at the end of **Hiding Pot**, so is selected as the currently most plausible role-binding throughout the inference path in Figure 9.

Signature Activation:

1) *Previous work* — First introduced in ROBIN.
2) *Function* — Signature activation on a binding node is part of a unique pattern of activation representing a dynamic, virtual binding to the signature's concept.
3) *Node pathways* — Signature activation spreads along role-binding paths (having unit-valued weights that preserve their activation) between corresponding roles of related frames.
4) *Dynamic structure* — Signature activation represents a potential (candidate) dynamically instantiated structure; i.e., that either Marijuana or Cooking-Pot is Inside-Of a Dishwasher.

7. LOGICAL BINDING CONSTRAINTS

Although the propagation of signatures allows ROBIN to dynamically generate inferences and evidential activation allows the one with the most evidence in a given context to be selected, there is still the potential problem of crosstalk from *logically unrelated inferences.* An example of this is the following sentence:

"John ate some rice before he went to church." **(Church Service)**

The most probable interpretation of **Church Service** is that John had rice for breakfast before he went to attend services at his church ($Church-Service). Without considering logical binding constraints, however, the node for the $Wedding script would likely become the most highly-activated, because of the

Frame	Binding Constraints	Used In
Inside-Of-Stove	A Cooking-Pot is inside of a Stove	$Stove-Cooking
Inside-Of-Dishwasher	A Utensil is inside of a Dishwasher	$Dishwasher-Cleaning
Inside-Of-Opaque	A Physical-Object is inside of an Opaque-Object	Avoid-Detection

Figure 10. Three of the competing refinements of state Inside-Of.

combined activity of Church and Rice. The $Church-Service script would lose out, because it would only receive evidence from Church.

Clearly there is a need to inhibit the spread of activation for inferencing when frames' logical binding constraints are violated. In **Church Service**, Rice should not provide any evidence for $Wedding, because it was being eaten and not thrown.

Returning to the processing of **Hiding Pot** and examining the possible refinements of Inside-Of (Figure 10), we find that Inside-Of-Stove would not have received enough evidential activation to compete with Inside-Of-Dishwasher or Inside-Of-Opaque. They both receive strong activation from their Locations' fillers, Dishwasher and Opaque-Object, and so easily outstrip it. This is as it should be, since John could have been either trying to clean the pot or hide it, but (assuming he is rational) there is no way that John was trying to cook it by putting it into the dishwasher.

But what if Stove or $Stove-Cooking happened to be strongly activated from previous processing? If Stove or $Stove-Cooking is highly activated, then it is quite conceivable that Inside-Of-Stove could end up with more activation than Inside-Of-Dishwasher and Inside-Of-Opaque. It would thus be chosen as the refinement of Inside-Of, and the network would arrive at the ludicrous decision that John was trying to cook something in the pot when he put it into the dishwasher.

These examples illustrate the necessity to have *logical binding constraints control the spread of activation.* Frames whose binding constraints (as defined by the knowledge base, as in Figure 1) are violated, and hence cannot possibly be part of the inference path, should not receive either evidential or signature activation.

7.1. Overview of the Frame Selection Process

To solve the problem of crosstalk from unrelated inferences, the structure of the network insures that a frame competing for selection is ruled out completely when its logical binding constraints are violated. ROBIN instantiates, with signature and evidential activation, only those candidate frames whose binding constraints are met.

Gating and constraint-matching built into the network's structure assure that only legal interpretations receive evidence for selection, thus avoiding crosstalk from logically unrelated inferences (such as eating the Rice in **Church Service**, or priming of Stove in **Hiding Pot**). The accumulation of evidential activation solely from *applicable* context allows one of the legal candidates to be selected as the most likely interpretation.

Though *all inferencing is accomplished solely by the spread of activation* through the structure of the network, the complete frame selection process performed by this propagation can be viewed as a four-part process:

1) *Choosing candidate frames:* When the role bindings of a frame match the logical binding constraints on the roles of a related frame, then that related frame becomes a *candidate frame* for instantiation. Related frames are rejected when their logical binding constraints are violated.

2) *Propagating signature bindings to candidate frames:* Candidate frames receive signature activation (representing role-bindings) from their instantiating frame. New candidate inferences can then propagate from each of the candidate frames to explore their respective inference paths.

3) *Propagating evidential activation to candidate frames:* Candidate frames receive weighted evidential activation from their instantiating frame. Candidates whose binding constraints are only partially matched receive proportionately less evidential activation than if their constraints were matched perfectly.

4) *Selection between candidate instantiation frames:* At any given time, the candidate frame with the most evidential activation represents the preferred interpretation. Commitments may change if new context gives more evidence to a competing frame.

As an example of how the frame selection process proceeds in ROBIN, consider Figure 11a, which shows frame Inside-Of and its three refinements (from Figure 10) during processing for phrase P1. At this point, evidential activation and signature role-bindings have reached Inside-Of (as in Figure 5), so the candidates for its concept refinement must now be chosen. Inside-Of-Stove is rejected since a Dishwasher does not match the Stove constraint on its Location role. It therefore receives no signature or evidential activation. Inside-Of-Dishwasher, however, is chosen as a candidate refinement frame, since its constraints are matched. Inside-Of-Opaque is also chosen as a candidate, since a Dishwasher *is-a* Opaque-Object.

The result can be seen in Figure 11b, where both candidates have been instantiated. Marijuana and Planting-Pot violate the constraints on the Object of Inside-Of-Dishwasher (neither of them is cleaned in a dishwasher), so only Cooking-Pot is allowed through. The chosen candidate will be the frame with the greatest evidential activation. After activation has settled for **Hiding Pot**, Inside-Of-Opaque has the greater evidential activation (thicker oval), and is selected as the refinement-of Inside-Of, serving as the plan for hiding his Marijuana from the police.

7.2. Gating of Signatures and Evidential Activation

The previously described frame selection process of the network is performed by inhibitory gating on those links which allow propagation of signature and evidential activation from one frame to another. Activation is only allowed to pass from a frame to one of its related frames when its role-bindings match the candidate frame's logical binding constraints.

The nodes and connections providing this inhibitory gating are shown in Figure 12. Each frame has a separate *candidacy* node (outlined triangles on bottom plane) for every frame that it is related to. This node will be active (with an activation of 1.0) if the related frame is a candidate for interpretation, i.e. if each of the frame's role-bindings match the candidate's binding constraints. However, if any of the frame's binding constraints are violated, then its candidacy node will be inactive (activation of 0.0). A description of the structure of the network performing these constraint-matching calculations appears in the next section.

Figure 12 shows the separate candidacy node for each of Inside-Of's refinement frames. When the activation of the network reaches the state of Figure 11a, with Inside-Of's Object bound both to Marijuana

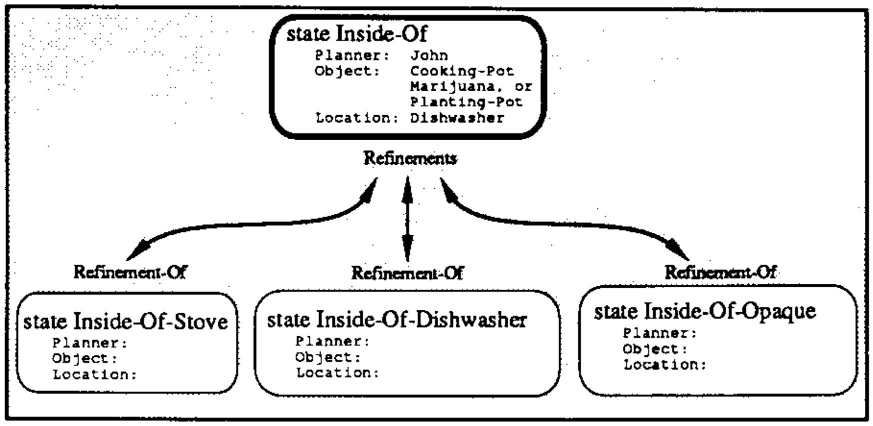

Figure 11a. An example of the frame selection problem — overview of bindings instantiated with signature activation (Figure 5).

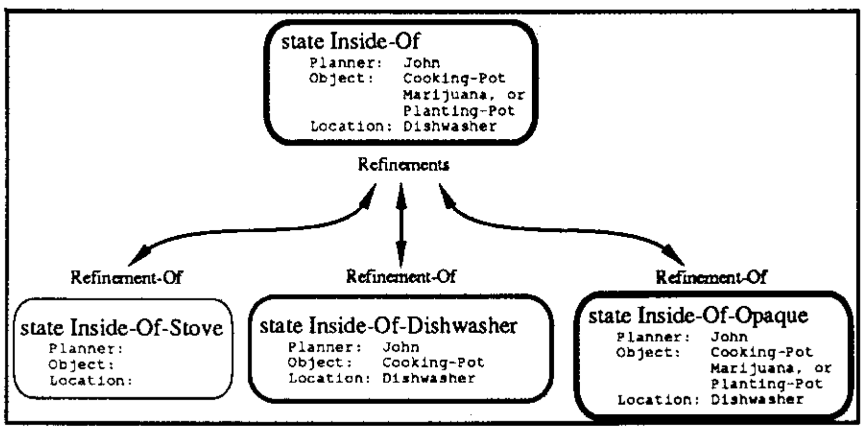

Figure 11b. Overview after Inside-Of-Dishwasher and Inside-Of-Opaque become candidate refinements of Inside-Of (Figure 6).

(signature 6.8) and Cooking-Pot (signature 9.2), the associated candidacy nodes of Inside-Of-Dishwasher and Inside-Of-Opaque become active (activation 1.0 in Figure 12) to choose them as candidate refinement frames. Inside-Of-Stove's candidacy node, however, remains inactive, since Inside-Of's Location violates its Stove constraint (a Dishwasher is not a Stove).

A further constraint on the propagation of each *individual* signature is that the binding it represents must match the role's logical constraints. Therefore, each separate binding path has a *binding constraint (BC)* node (solid black triangle) which calculates whether the object bound to the binding node matches the candidate's constraints.

For a signature to pass to the corresponding binding unit on a related frame, *both* the frame's candidacy node and the binding constraint node on the individual signature path *must be active.* As shown in Figure 12, each binding propagation link is gated by both its BC node and its frame's candidacy node. The link is conjunctive, as in the sigma-pi units described in [Rumelhart *et al.*, 1986].

The weighted link that allows evidential activation to pass through to the frame's conceptual node from the related frame's conceptual node is also gated by its candidacy node (Figure 12). Because of this, a re-

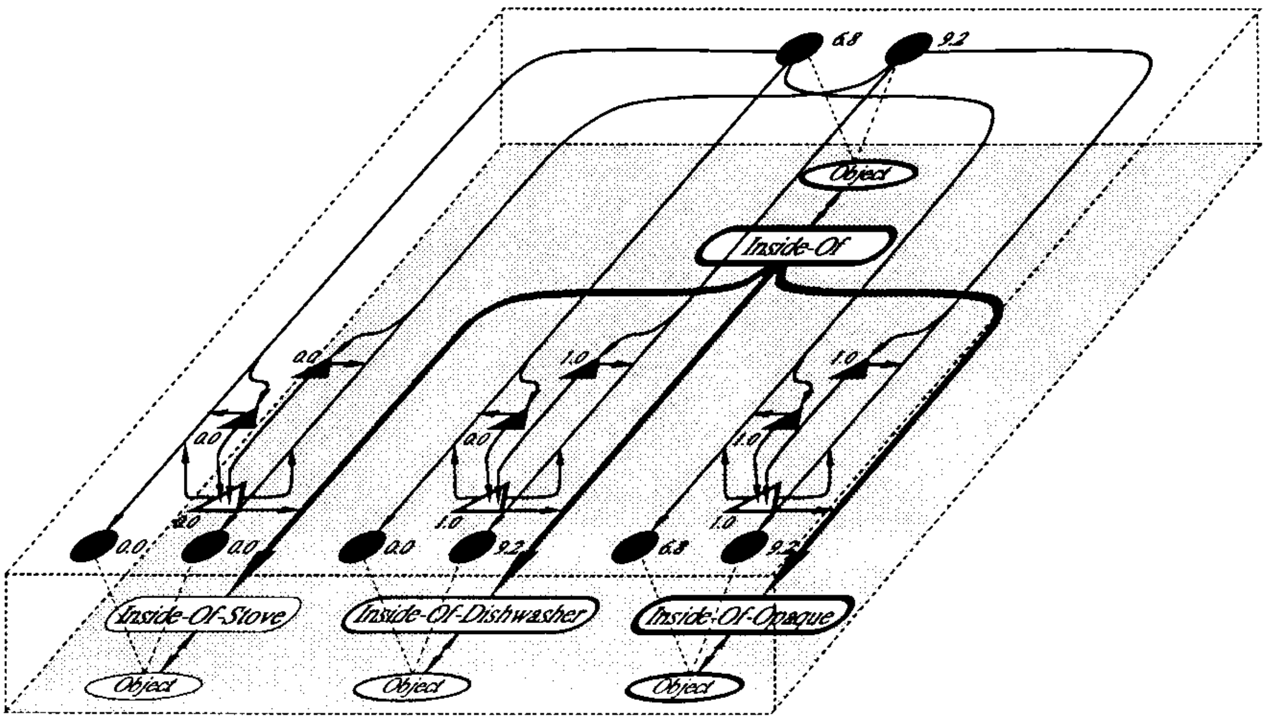

Figure 12. Paths from binding nodes of Inside-Of's Object to the corresponding binding nodes of its refinement frames are gated by connections from their frame *candidacy* nodes (outlined triangles on bottom plane). The weighted links allowing evidential activation to pass from Inside-Of to its refinement frames are also gated by their candidacy nodes. Individual binding paths are also gated by connections from their *binding constraint* nodes (solid triangles on top level). All connections to gated links must be active for activation to pass through.

lated frame whose role-bindings do not match the frame's logical binding constraints will not provide evidence for it.

In Figure 12, both Marijuana and Cooking-Pot match the Physical-Object constraint on Inside-Of-Opaque's Object role, so both of its BC nodes are active (1.0). Since their candidacy and BC gates have an activation of 1.0, the signatures pass through the conjunctive link to instantiate Inside-Of-Opaque's Object binding nodes, as shown. Evidential activation is allowed to pass through.

The same is true of the signature representing Cooking-Pot (9.2) for Inside-Of-Dishwasher (on its right signature path). Marijuana (signature 6.8), however, is not cleaned in a dishwasher, so its BC node (on the left signature path) stays inactive (0.0). The gate on that path therefore remains closed, and Marijuana is not inferred as a possible Object to be cleaned in Inside-Of-Dishwasher. Inside-Of-Dishwasher's overall binding constraints are matched, however (since Cooking-Pot matched the constraints), so its candidacy node is active and evidential activation is allowed to pass through.

Finally, because Inside-Of-Stove's binding constraints were violated (a Dishwasher is not a Stove, the constraint on its Location), its candidacy node has an activation of 0.0. It therefore receives none of Inside-Of's evidential activation or signature bindings, and so it remains uninstantiated — without possibility of being considered as the current refinement of Inside-Of.

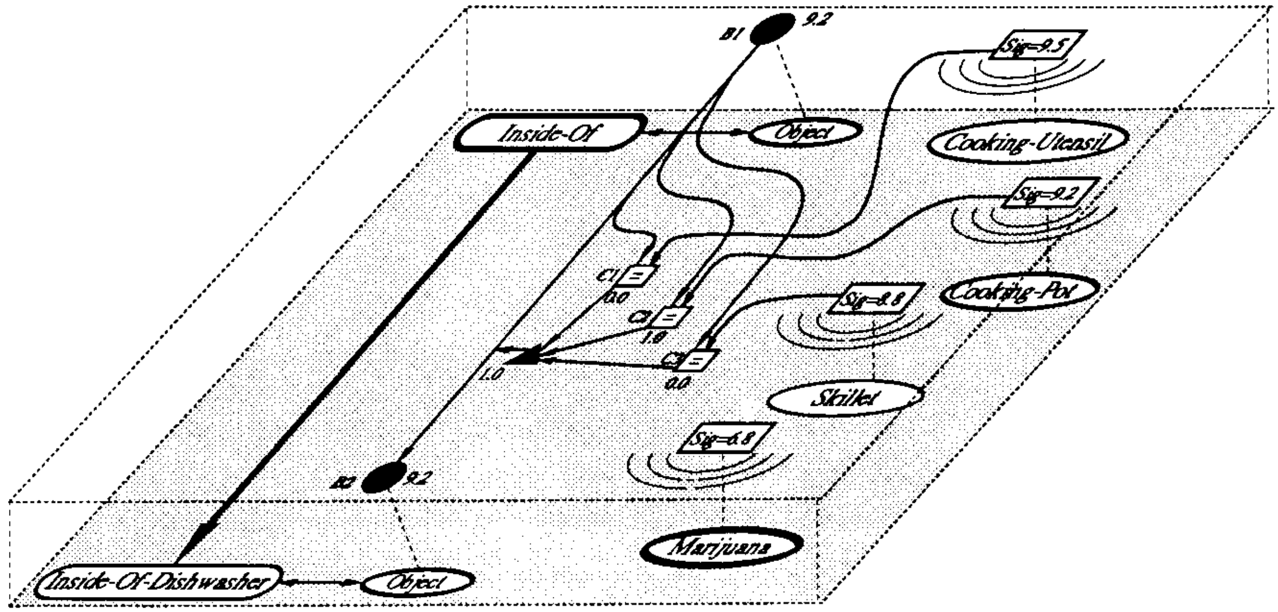

Figure 13. Figure illustrating how the binding constraint node (solid black triangle) for the path from one of the binding nodes of Inside-Of's Object role to the corresponding binding node of Inside-Of-Dishwasher's Object calculates its activation. The rectangular nodes with an "=" inside (C1, C2, and C3 on the top plane) are *comparator nodes*, which have an activation of 1.0 iff the signature activations from both incoming links are equal, 0.0 otherwise.

7.3. Binding Constraint Structure

The binding constraint nodes calculate whether or not a signature matches the logical binding constraints on a role. The original knowledge base definitions that construct the network (i.e. Figure 1) define a type (or types) of concept(s) that may be bound to each individual role. Examples are that Inside-Of-Dishwasher's Location role can only be filled with something that *is-a* Dishwasher, and that its Object role must be a Cooking-Utensil or something that *is-a* Cooking-Untensil, such as a Skillet or a Cooking-Pot (since they are the things that are cleaned in dishwashers).

In order for the network to decide whether a signature matches the binding constraints on a role, it must be compared to each of the signatures that form the role's *legal* bindings. Figure 13 shows how this is done for the BC node on the path from binding node B1 (bound to Cooking-Pot, signature 9.2) of Inside-Of's Object role to the corresponding binding node of Inside-Of-Dishwasher's Object.

In Figure 13, there are three concepts that Inside-Of-Dishwasher's Object can be bound to: (1) Cooking-Utensil, (2) Cooking-Pot, and (3) Skillet (because both Cooking-Pot and Skillet *is-a* Cooking-Utensil). To calculate whether B1 is bound to one of these three, there are three *comparator* nodes, C1, C2, and C3 (squares with an '=' inside). Each comparator node has two inputs to match: one candidate signature from the binding node, and one signature from the logical constraint. A comparator node will be active (activation 1.0) if the two input activations (signatures) are the same, inactive (activation 0.0) otherwise.

Comparator node C1 has an input from binding node B1 and the signature node of Cooking-Utensil, so directly calculates whether or not B1's signature matches that of Cooking-Utensil. It does not in Figure 13, so C1 is inactive (activation 0.0). Comparator C3 also fails in this case, since it compares B1 to the signature of Skillet. Comparator C2, on the other hand, compares B1's activation to the signature of Cooking-Pot. Cooking-Pot is indeed the concept that is bound to binding node B1, so both of C2's inputs have the same activation (9.2), and C2 produces an activation of 1.0. This activation propagates to the BC node that gates the path from B1 to B2. With this BC node active (and Inside-Of-Dishwasher's candidacy node also active), the signature of binding node B1 is clear to propagate past the BC gate to B2.

An equivalent structure (of comparator nodes) calculates the activation of the BC node for the other binding node of Inside-Of-Dishwasher's Object. However, since Marijuana is the signature on the corresponding binding node (Figure 12), none of its corresponding (C1-C3) comparators will find that it matches Cooking-Utensil, Cooking-Pot, or Skillet, and so will all remain inactive. The path's BC node will have an activation of 0.0, thus blocking the (logically impossible) signature of Marijuana from propagating to Inside-Of-Dishwasher's Object. Consequently, Cooking-Pot is inferred as an object that might be cleaned, but Marijuana is not.

Besides calculating when individual bindings match the constraints on a role and should be allowed to propagate on through, the network must also be able to calculate when the role-bindings of a related frame match *all* of a frame's logical binding constraints. As Figure 12 suggests, the individual BC nodes drive this calculation by having inputs into the frame's overall candidacy node. The details of the network's structure of simple connectionist nodes that perform this calculation are not necessary for the purposes of this paper, but can be found in [Lange, 1990], along with a further description of the comparator and BC nodes.

8. A DETAILED EXAMPLE

Figure 14 shows the evidential activation levels of the meanings of the word "pot" as activation is being spread in **Hiding Pot**. The word "pot" is first interpreted as a Cooking-Pot because of the strong evidence from inferred $Dishwasher-Cleaning context. However, evidential activation accumulates for Marijuana as the longer inference path involving the police and illegality is activated. Figure 15 contains a description of what is occuring as signature and evidential activation is propagated throughout the network. At stability (iteration 136), Marijuana has greater evidential activation than Cooking-Pot, so is selected as the binding throughout the selected plan/goal analysis inference chain of Figure 9. Note that evidential activation remains on Cooking-Pot and Planting-Pot, available for possible reinterpretation given new input, such as phrase P3 ("they were coming over for dinner").

Another observation is that even though phrase P2 ("because the police were coming") never stated where the Police were coming to, the network has inferred that they came to the Location of the Marijuana (the Transfer-Self in Figure 9). This role-binding has been inferred because John was trying to hide it from them, and has been instantiated with signature activation propagated along the paths of corresponding binding roles all the way from Transfer-Inside.

Most of the logical constraints on the role-bindings of the frames in Figure 9 were met, as shown in the final overview of the results. As described previously, Inside-Of-Stove never received any activation from the inference paths since its binding constraints were violated. Its candidacy node from Inside-Of, and the candidacy nodes of many other frames not shown, such as Inside-Of-Planting-Pot (i.e. *"the flower is in the pot"*, part of $Grow-Potted-Plant) and Fireman-See-Fire (i.e. *"the fireman saw the fire"*, part of $Fight-Fire), remained inactive because of these violations. Other frames, such as goal Prepared and $Smoke-

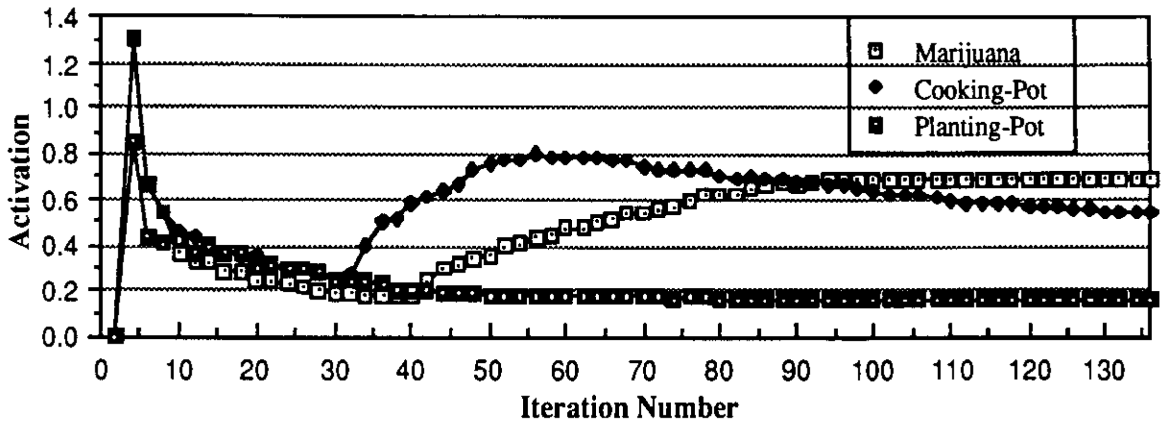

Figure 14. Evidential activations of the meanings of *"pot"* as activation spreads in **Hiding Pot**.

Iteration	Description
0	No activation in the network.
1-2	The conceptual node on the evidential layer for phrase <S "put" DO "in" IO> is clamped, and the binding nodes of its roles clamped to the signatures for phrase P1. The conceptual node for phrase <S "were coming"> is also clamped, and the binding nodes of its roles clamped to the signatures for phrase P2. Lexical nodes for "John", "pot", "dishwasher", and "police" are also clamped.
3-6	Marijuana, Cooking-Pot, and Planting-Pot all receive evidential activation from lexical node "pot".
7-30	Evidential activation of Marijuana, Cooking-Pot, and Planting-Pot all decay. Signatures instantiate Transfer-Inside and Inside-Of frames.
31-40	Evidential activation of Cooking-Pot climbs with feedback from newly-activated $Dishwasher-Cleaning script frames, starting with Inside-Of-Dishwasher. Marijuana and Planting-Pot continue to decay.
41-55	Evidential activation of Marijuana starts to climb because of feedback from the longer path involving the police and illegality.
56-135	Activation of Cooking-Pot decays with lack of new evidence; Marijuana continues to grow with further feedback from police and hiding inference path.
136	Network has completely stabilized. Marijuana is more highly activated than Cooking-Pot or Planting-Pot, so is chosen as the binding throughout the network. Instantiations and levels of activation of the frames in the network is as in Figure 9.

Figure 15. Table describing levels of evidential activation over time for competing concepts Marijuana, Cooking-Pot, and Planting-Pot.

Food (as in *"smoke the bacon in the smokehouse"*), never received activation because none of their related frames became instantiated with signatures.

9. CURRENT STATUS AND FUTURE WORK

ROBIN has been fully implemented in the DESCARTES connectionist simulator, which allows the flexible simulation of structured heterogeneous networks [Lange *et al.*, 1989a]. ROBIN's inferencing, plan/goal analysis, schema instantiation, disambiguation, and reinterpretation abilities have been successfully

"John smoked the pot."	"pot" = Marijuana, "smoke" = $Burn-And-Inhale
"Bill smoked the meat."	"smoke" = $Smoke-Food
"Jerry put the pot on the stove."	"pot" = Cooking-Pot
"Ron put the pot on the stove. He picked it up and smoked it."	"pot" = Cooking-Pot, "put" = $Stove-Cooking → "pot" = Marijuana "put" = $Light-Object
"John put the flower in the pot, and then watered it."	"pot" = Planting-Pot, "put" = $Grow-Potted-Plant
"Cheech watered the pot, but the police saw him, so he was arrested."	"pot" = Planting-Pot → "pot" = Marijuana-Plant
"Jerry grew the pot."	"pot" = Marijuana-Plant
"The CIA searched for bugs." [Granger *et al.*, 1986]	"bugs" = Microphone, goal = Remove-Listening-Device
"Safeway searched for bugs." [Granger *et al.*, 1986]	"bugs" = Insects, goal = Remove-Health-Hazard

Figure 16. Examples ROBIN handles using activation clamping from syntactically pre-processed input.

tested on **Hiding Pot** and a number of other episodes, in two domains, using syntactically preprocessed inputs of one and two sentences in length. Figure 16 lists some of the other episodes that ROBIN is able to disambiguate and analyze in terms of their plan/goal structure.

The **Hiding Pot** example's knowledge base currently has 92 conceptual frames defined (18 of which are shown in Figure 2), with a total of 104 different roles. The average number of roles per frame is less than that of Figure 2 because 31 object frames (i.e. Physical-Object, Cooking-Pot, John, etc.) and 19 simple lexical entries (i.e. "pot" and "police") have no roles defined other than their relations to other frames. The network encodes 141 rules in the form of corresponding roles over which inferences can be made by propagation of signatures.

The network that is constructed from the **Hiding Pot**'s knowledge base has a total of approximately 10,000 nodes, the vast majority of which are used to calculate logical binding constraints (see discussion in Section 9.1). Each role in the network has three binding nodes. Because of this, there are the equivalent of 312 variables in the network (3 binding nodes per role x 104 roles) and 423 binding path "rules" (3 x 141 rules) over which inferences can be made.

The network typically takes between about 80 and 150 cycles to generate all candidate inference paths and settle into a stable state in which a single inference path is most highly-activated (e.g. Figure 14). The number of cycles generally required for reaching quiescence has remained in approximately that range for all sizes of the network tested. This is primarily true because the gating and logical binding constraints within the network's structure stop activation from spreading to the (sometimes) large areas of the network that are logically unrelated to the input.

In the future, there are five main areas that we would like to explore: (1) signatures as distributed patterns of activation, (2) realization of signatures as temporal frequencies, (3) the ability to handle embed-

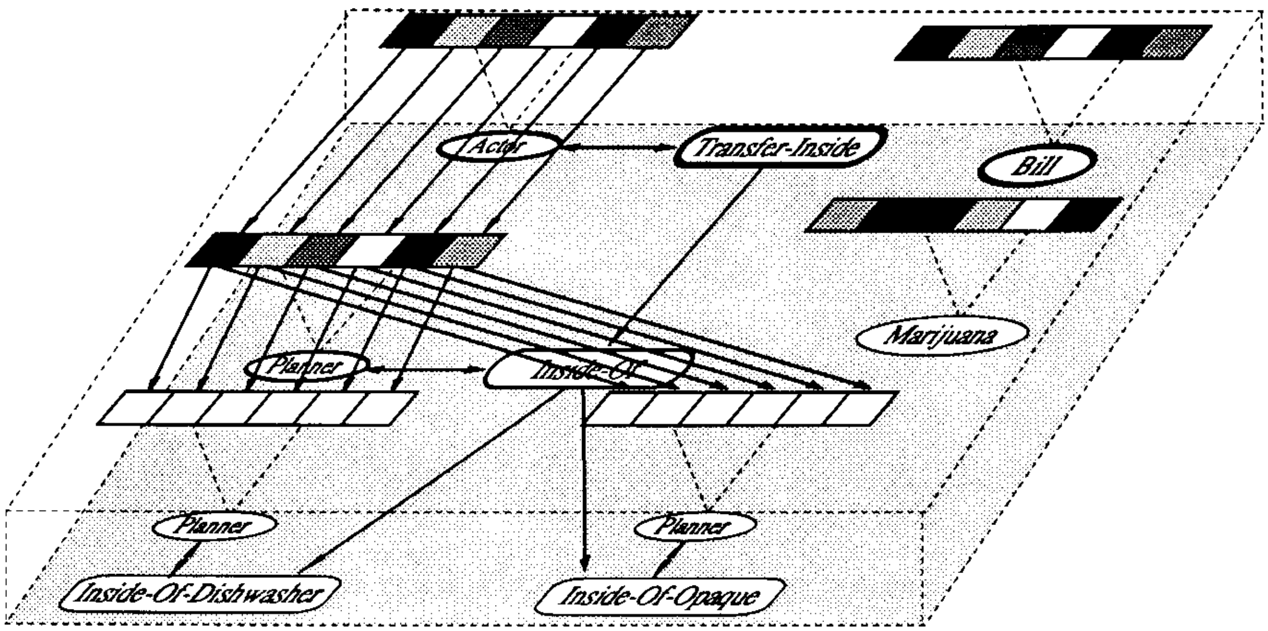

Figure 17. Possible future use of *distributed* signatures, where each signature is a unique pattern distributed over a *bank* of nodes. Here each signature or binding bank is made up of six nodes, with increasing levels of activation represented by increasing darkness of shading (ranging from white = 0 to black = 1). Shown is the (desired) state of the network after Bill's distributed signature has propagated from the binding bank of Transfer-Inside's Actor to the binding bank of Inside-Of's Planner, but before reaching the Planner banks of Inside-Of-Dishwasher and Inside-Of-Opaque.

ded role-bindings, (4) formation of long-term episodic memory, and (5) the addition of lexical information to the networks.

9.1. Signatures Using Distributed Representations

Currently, each signature is a single arbitrary activation value that uniquely identifies its concept. Large models could conceivably have thousands or hundreds of thousands of separate concepts that they could recognize (such as Marijuana, Cooking-Pot, Catfish, Guppy, John, John-Wayne, John-Kennedy, etc). It is untenable to expect a single binding node to have enough precision to accurately distinguish between such a large number of signatures[1].

A better solution is that each signature be a *distributed* pattern of activation which uniquely-identifies its concept. As shown in Figure 17, distributed signatures would be propagated for inferencing over paths of binding *banks* in exactly the same way as ROBIN's current single-valued signatures. Similar concepts would have similar distributed patterns of constant values as their signatures, so that each signature would carry some semantic meaning. A first pass at this might entail the use of microfeature-like patterns, as in the distributed model of [McClelland & Kawamoto, 1986], but it would be preferable to have the signature patterns learned over time, as done by the model of [Miikkulainen & Dyer, 1988,1989].

[1]The normal coding capacity of connectionist elements is usually in the range of 1-5 bits.

One of the most important results of using distributed signatures would be a vast simplification of the network structure calculating whether individual signature bindings match a role's logical binding constraints. Because signatures currently carry no semantic meaning, the binding constraint nodes must have a *separate signature comparator node* for each and every one of the concept's signatures that are legal role-bindings. The number of comparator nodes required to calculate a single binding constraint can vary from one (a Dishwasher constraint only needs to check whether the signature matches Dishwasher) to extremely many (any person known to the system can match a Human constraint on a role, so there must be a separate comparator node for each).

This is clearly not an acceptable solution for large networks. However, if signatures are distributed patterns of activation that are *similar for similar concepts* and that themselves carry semantic information, then the entire structure of comparator nodes for a role could be replaced with a bank of nodes that does a simple *similarity threshold* between the signature binding and the distributed signature of the logical binding constraint. Another possibility is that the binding constraint nodes could be a small distributed ensemble of nodes *trained* to recognize the constraints that a role has on its filler.

9.2. Signatures as Temporal Frequencies

Another realization of signatures that we are currently exploring utilizes the time dimension, in which signatures are uniquely-identifying *frequencies* of output spikes generated by artificial neural oscillators. We have already illustrated how *signature frequencies* representing bindings can be propagated across a network for inferencing by phase-locking of relaxation oscillators [Lange *et al.*, 1989b]. Linkage of distant oscillators through common output frequencies represents shared role-bindings and inferences. [Tomabechi & Kitano, 1989] have also suggested the use of frequency modulation of oscillator pulses for this task.

9.3. Embedded Role-Bindings

Using signatures of pre-existing concepts, ROBIN can create and infer novel network instances. However, ROBIN currently cannot dynamically generate and propagate *new* signatures for one these instances. This ability is crucial for recursive structures, such as in: *"John told Bill that Fred told Mary that..."* Here each Object of the telling is itself a novel frame instance not having a pre-existing signature. Until a solution for embedded signatures is found, ROBIN's inferencing capabilities will be somewhat limited in comparison to symbolic rule-based systems[1].

9.4. Formation of Long-Term Episodic Memory

Signatures allow ROBIN to create novel network instances over its pre-existing structure, but the activation of these instances is transient. Over time, repeated instantiations should cause modification of weights and recruitment of underutilized nodes [Diederich, 1990] to alter network structure. Possible methods of storing the inferred instances in long-term episodic memory by some kind of distributed learning mechanism must also be explored, likely in conjunction with the use of distributed signatures.

---

[1]Other structured connectionist models do not "technically" suffer from an inability to handle embedded role-bindings, since they cannot currently handle even simple ones.

9.5. Lexical Information

ROBIN does not currently address the problem of deciding upon the original syntactic bindings, i.e. that *"pot"* is bound to the Object role of the phrase. Rather, ROBIN's networks are given these initial bindings and use them for high-level inferencing. To handle natural language input entered as text, the network must somehow contain and use syntactic and phrasal information to create the initial role-bindings that ROBIN is currently given by hand.

10. COMPARISON TO RELATED CONNECTIONIST MODELS

There are currently a very limited number of connectionist models besides ROBIN that have attempted to emulate the symbolic abilities of variable binding and rule firing.

10.1. Distributed Connectionist Models

Distributed connectionist models represent knowledge as patterns of activation across nodes, rather than the single unit representation of individual concepts used in structured networks such as ROBIN. Each of the distributed connectionist models described below uses the energy minimization metaphor to "settle" into individual variable bindings or rule firings.

DCPS is a distributed connectionist production system described by [Touretzky & Hinton, 1988] that uses *coarse codings* to represent variable bindings. Each node in DCPS's working memory has associated with it a receptive field that can represent many possible Frame-Slot-Filler "triple" combinations. When a variable binding exists in the memory, all of the nodes with a receptive field that contains that binding triple (approximately 28 out of their working memory of 2000 nodes) are activated. Multiple triple bindings are represented by superimposing their receptive fields.

DCPS also uses coarse-coding to represent rules of the form:

```
(=x A B) (=x C D)  ==>  +(G =x P) -(=x R =x)
```

where the capital letters are constants and `=x` is a variable. When the left-hand side of a rule is matched, the triples on the right-hand side of the rule are either added (+) or deleted (-) from working memory. The rules in DCPS are limited to two clauses on the left-hand side of the rule, with every rule having only a single variable that must appear as the first element of both clauses. Another restriction is that there can never be more than one rule with one variable binding that matches the contents of working memory at any given time.

To execute the rule firing cycle, DCPS first performs energy minimization to find a rule whose left-hand side matches two of the triples in the working memory. Once a rule is settled upon, gated connections from the rule space add and remove receptive fields from the working memory to "fire" the right-hand side of the rule. The cycle is then repeated and another production is matched, just as in a serial computer.

Another distributed connectionist approach is CRAM [Dolan, 1989], a hybrid natural language processing system in which the parsing modules are symbolic, but in which the memory and binding modules use *conjuctive coding*. In this distributed representation scheme, a cube of nodes is allocated to encode Frame-Slot-Filler triples by superimposing the binary pattern of each triple element across a dimension of the cube. This method has been generalized to scalar values by representing the superimposition as the outer product of two or more tensors [Derthick, 1988] and [Dolan & Smolensky, 1989].

Each of these models has successfully demonstrated that distributed connectionist models have the ability to represent and use explicit rules. Furthermore, their use of distributed representations allows their models to be damage-resistant and use far fewer units then needed in traditional localist networks that represent each potential fact with a single unit.

The primary problem with each of these distributed connectionist models is that although they *select* their rules through massively-parallel constraint satisfaction, they actually behave *serially* at the knowledge level, since they can select and fire only one rule at a time. This becomes a major problem when the tasks are complex and involve high-level inferencing. In natural language understanding and planning tasks it is generally necessary to explore many possible inference paths. Making conceptual connections between two or more facts effectively amounts to an intersection search which can quickly involve a very large number of rules. This has proven to be a debilitating problem to serial symbolic rule-based systems, and has in fact motivated a large amount of research in *marker-passing* models (described in Section 10.3) that are better able to approach these problems due to their massively-parallel symbolic mechanisms.

Because current rule-handling distributed models are serial at the knowledge level, they will continue to be plagued with many of the same problems that traditional symbolic rule-based systems face. ROBIN's structured networks, on the other hand, are able to fire many rules at once, dramatically decreasing the time required to "search" through the rule space to find inference paths connecting the inputs. Equally important is that with the propagation of signatures, ROBIN has no need to use (relatively slow) constraint satisfaction to select and fire its rules. ROBIN instead uses the network's smooth constraint satisfaction abilities to perform an even more difficult part of high-level inferencing: *selecting the most plausible of alternative inference paths and allowing for reinterpretation if contexts change.*

10.2. Ajjanagadde and Shastri's Structured Connectionist Model

The connectionist model which is most closely related to ROBIN [Lange & Dyer, 1988] is that proposed recently in [Ajjanagadde & Shastri, 1989]. Their model also solves the knowledge-level parallelism problem by being able to maintain and propagate multiple variable bindings in a structured network, but does so instead by using a multi-phase clock.

Their networks encode predicates and rules of the sort:

```
forall (x,y,z) (orderhit (x,y,z) => hit (y,z))
```
(If X orders Y to hit Z, then Y hits Z)

The networks also store long-term facts with individual instance nodes, such as a orderhit(dave,mike,bob). They then pose queries to the network which can only be answered by inferencing over rules like the above, such as the query ?hit(mike,bob).

In their model, each network cycle is broken up into a fixed number of phases. A variable binding is represented when an *arg*-node (analogous to ROBIN's binding nodes) is active on the same phase of the clock as the *const*-node (analogous to ROBIN's concept nodes) bound to it. As in ROBIN, there are connections between corresponding arg-nodes as defined by rules in the knowledge base. Figure 18 shows an example network illustrated in [Ajjanagadde & Shastri, 1989].

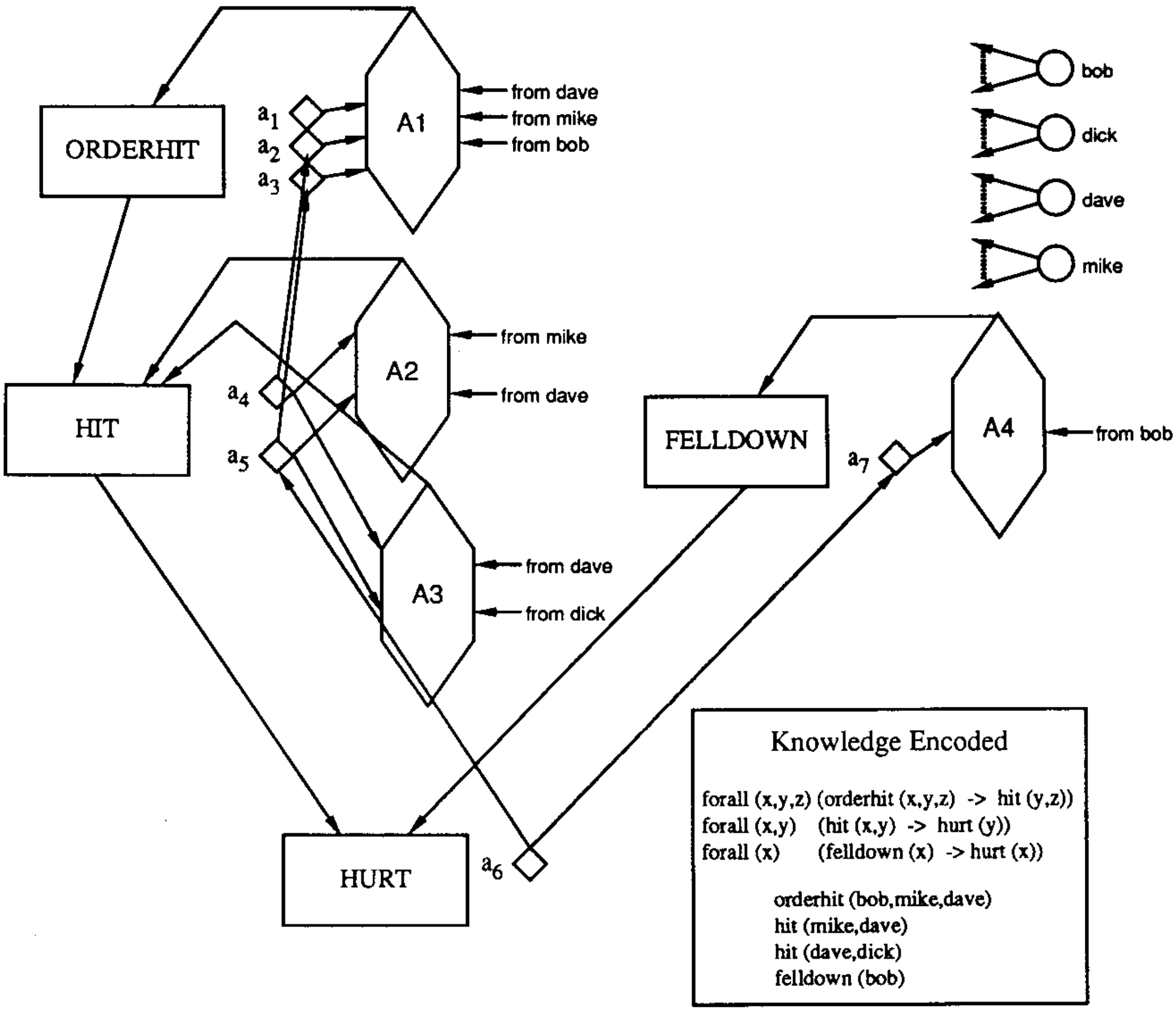

Figure 18. Example network from [Ajjanagadde & Shastri, 1989]. Diamonds a_1 thru a_7 are *arg-nodes*, while circles for bob, dick, dave, and mike are *const*-nodes.

To provide the initial bindings to the network and pose a query, they allocate (by hand) a phase of the clock for each binding in the query. For example, to pose the query ?hit(mike,bob) to the network of Figure 18, they activate the first arg-node of hit (a_4) and the const-node mike in the first clock phase, and activate the second arg-node of hit (a_5) and the const-node bob in the second clock phase. Because arg and const-nodes always become active on the phase of the clock that they were originally activated in, these bindings will hold indefinitely.

Once the bindings are set, they propagate over the paths between arg-nodes. In the first phase of the second cycle, for example, the activation of arg-node a_4 causes arg-node a_2 to become active. A_2 will then continue to become activated on every first phase. Because const-node mike is still active in the first phase, a_2 (the second argument of orderhit) is now also bound to mike, and the inference has been made. Similarly, in the second phase of the second cycle, arg-node a_5 causes a_3 to become active, so that both are bound to bob. They are thus able to propagate bindings across the network for inferencing.

Ajjanagadde and Shastri's model succeeds in illustrating an alternative mechanism for maintaining variable bindings in a structured connectionist network. Like ROBIN, their model can match and fire multiple rules in parallel, as opposed to the serial limitations (one rule at a time) of current distributed connectionist models.

The basic kinds of inferences that can be performed by ROBIN's propagation of signatures and by Ajjanagadde and Shastri's multi-phase clock seem to be about equivalent. Both mechanisms allow the binding of any previously known concept to any role in the network, and cause those bindings to be propagated along arbitrarily long inference paths defined by their knowledge base's rules.

One of the clearest advantages that Ajjanagadde and Shastri's model has over both distributed connectionist and traditional symbolic systems is its ability to perform deductive inference with extreme efficiency. Their model is in fact able to draw conclusions in time proportional to the length of the proof [Ajjanagadde & Shastri, 1989]. ROBIN does not aim for such optimal efficiency, instead using the constraint satisfaction process of the evidential portion of its networks to resolve ambiguities. However, if signatures were to be applied solely to the task of deductive retrieval that Ajjanagadde and Shastri's model handles, ROBIN could be stripped of its constraint-relaxation evidential network. In this case, propagation of signatures could also perform deductive retrieval in time proportional to the length of the proof. In fact, because a single cycle in ROBIN is functionally equivalent to a single phase of the clock in Ajjanagadde and Shastri's model[1], signature propagation would complete the proof a factor of p times faster than their model, where p equals the number of phases in their clock cycle.

One potential problem with using a phase-clock mechanism for variable binding (as opposed to signatures) lies in selecting the number of clock phases when there is sequential input. For example, in natural language processing systems new bindings are constantly being created as every word is read in. This fact will force Ajjanagadde and Shastri's system to continually modify the number of phases in their clock cycle. For example, consider the following story:

> *"The short and fat man bought a red Corvette. He called the police when a thief stole it from his mother's garage in Fresno."*

In the first sentence short and fat must be propagated and bound to the Height and Weight roles of Human, followed by the propagation of Man to the Actor of Buy, Red to the Color of Automobile, and Corvette to the Object of Buy. At some point, Ajjanagadde and Shastri's system must decide upon the system's number phases per clock cycle in order to represent those bindings. The number of phases per clock will have to be changed, however, to handle the new bindings that sequentially arise from the second sentence and any subsequent sentences. The only apparent way to avoid having to continually modify the phase clock to accommodate new bindings in such cases is to constantly operate the clock at a high enough number of phases per cycle to maintain the maximum number of bindings the system will ever require. This solution, however, would decrease efficiency when there are only a small number of bindings. This kind of problem does not occur with signatures in ROBIN, since new signatures are simply added to the network and propagated along with the old ones.

---

[1] As in most connectionist models, activation can propagate from one node to its neighbor in a single ROBIN cycle. Ajjanagadde and Shastri's model defines that a single *phase* of their clock cycle possesses this same ability, and so require that each phase have the same computational complexity as a normal connectionist cycle.

Looking beyond explicit comparisons between signature and phase-clock binding mechanisms, a primary functional ability that Ajjanagadde and Shastri's model possesses that ROBIN does not is a pre-existing storage of instances. Facts are hardwired into their networks with *instance* nodes, using a single instance node per fact (such as orderhit(dave,mike,bob)). Instance nodes could be duplicated in ROBIN and accessed with signatures, but they are not likely a good final solution to the problem of modelling long-term episodic memory, because of the huge number of instances (and thus instance nodes) involved. One possible future advantage of using *distributed* signatures instead of Ajjanagadde and Shastri's phase-clock is that once signatures are propagated, the distributed representations of the bindings would be immediately available for use by a connectionist learning mechanism to form long-term distributed memories. Such a mechanism is needed to solve the *multiple instance problem* [Sumida & Dyer, 1989].

While both models seem to have nearly equivalent variable binding and rule-firing abilities, ROBIN goes beyond Ajjanagadde and Shastri's model in three major ways: (a) integration of its variable binding mechanism within a connectionist semantic network, (b) multiple binding sites per role, and (c) gating and logical binding constraints. The most important difference is that ROBIN's signature role-binding structure is integrated within an *evidential* connectionist semantic network that performs smooth constraint satisfaction to select a most plausible interpretation from several generated inference paths[1]. Nearly as important is that having multiple binding sites per role allows ROBIN to evaluate ambiguous role-bindings in parallel, which is key to handling disambiguation and meaning reinterpretations without backtracking. And finally, gating and logical binding constraints within ROBIN's network structure control the spread of activation and eliminate crosstalk between logically unrelated inferences. These capabilities allow ROBIN to perform much of the high-level inferencing required for natural language understanding, where not only must inference paths be dynamically instantiated, but in which alternative paths must be evaluated and selected among in changing contexts.

10.3. Marker Passing Networks

Marker-passing models operate by spreading symbolic markers across labelled semantic networks. Marker-passing systems are trivially able to maintain role-bindings using the symbolic pointers stored in their markers, whose propagation is used to generate plausible inference paths. Because of this, marker-passing models have been able to perform high-level inferencing in such areas as planning [Hendler, 1988] and natural language understanding [Charniak, 1986], [Riesbeck & Martin, 1986], [Granger *et al.*, 1986], and [Norvig, 1987].

In several ways, ROBIN's use of signature activation could be looked upon as a connectionist implementation of markers. Functionally, signatures are similar to markers because both are propagated across semantic networks to perform inferencing. Conceptually, they are similar because there is a distinct signature for every concept, so that signatures could be viewed as "symbolic back-pointers". On the other hand, signatures are not as powerful as markers, since they cannot currently handle the embedded role-bindings that would be simple to implement using the full symbolic architectures of marker-passing systems.

One advantage of signatures over marker-passing is that ROBIN's simple connectionist nodes are far less complex than the logic and lisp-based symbolic units of marker-passing systems. However, the most important difference between ROBIN and existing marker-passing models is ROBIN's integration of signa-

---

[1] [Ajjanagadde & Shastri, 1989] mentions integration with connectionist semantic networks as an area of their own future research.

tures with its evidential connectionist semantic network that encodes logical binding constraints, so that a single most-activated inference path is always selected in a given context. Current marker-passing models, on the other hand, must use a symbolic mechanism separate from the marker-passing process to apply (1) a theorem prover (i.e. [Charniak, 1986]) or (2) a heuristic path evaluator (i.e. [Granger *et al.*, 1986], [Hendler, 1988]) to select the most relevant or plausible path from the many paths generated. Path evaluation has proven to be a major problem as the size of knowledge bases increase and there is an according explosion in the number of candidate paths that must be weeded out [Charniak, 1986].

11. CONCLUSIONS

High-level inferencing is a fundamental problem in cognitive tasks such as natural language understanding and planning. Unfortunately, connectionist models have been unable to approach these problems because of their difficulties with handling dynamic variable bindings and propagating them by applying general knowledge rules.

We have presented a structured connectionist model, ROBIN, that solves a significant subset of these problems. Using structure that holds *signature* activation, ROBIN is able to dynamically create novel frame instances by binding roles with any previously known concepts in the network. Since signatures are simply activation patterns that uniquely identify the concept bound to roles, they can be propagated in parallel across separate paths of binding nodes that preserve their activation, thus performing inferencing.

For the subset of the inferencing process ROBIN is able to handle, it also has significant advantages over symbolic rule-based and marker-passing systems. This is because the inherent constraint-satisfaction of ROBIN's connectionist *evidential* semantic network structure allows it to select a single most-plausible interpretation from among multiple ambiguous inference paths. Semantic reinterpretation occurs when new context causes the evidential activation of an alternative interpretation to become more highly-activated than the previous one, avoiding the expensive backtracking rules of rule-based systems.

ROBIN is thus able to handle many of the high-level inferencing tasks not addressed by other connectionist models, while at the same time perform disambiguation and semantic reinterpretation often difficult for symbolic systems.

REFERENCES

Ajjanagadde, V. & Shastri, L. (1989): Efficient Inference With Multi-Place Predicates and Variables in a Connectionist System. *Proceedings of the Eleventh Annual Conference of the Cognitive Science Society*, Ann Arbor, MI, August 1989.

Charniak, E. (1986): A Neat Theory of Marker Passing. *Proceedings of the National Conference on Artificial Intelligence (AAAI-86)*, Philadelphia, PA, August, 1986.

Cottrell, G. & Small, S. (1982): A Connectionist Scheme for Modeling Word-Sense Disambiguation. *Cognition and Brain Theory, Vol. 6*, p. 89-120.

Derthick, M. (1988): *Mundane Reasoning by Parallel Constraint Satisfaction.* (Tech Rep. No CMU-CS-88-182). Doctoral Dissertation, Computer Science Department, Carnegie-Mellon University, Pittsburgh, PA.

Diederich, J. (1990): Steps Toward Knowledge-Intensive Connectionist Learning. In J. Barnden and J. Pollack (eds.), *Advances in Connectionist and Neural Computation Theory*, Ablex Publishing. In press.

Dolan, C. P. (1989). *Tensor Manipulation Networks: Connectionist and Symbolic Approaches to Comprehension, Learning, and Planning.* (Tech Rep. No UCLA-AI-89-06), Doctoral Dissertation, Computer Science Department, University of California, Los Angeles.

Dolan, C. P. & Smolensky, P. (1989). Tensor Product Production System: A Modular Architecture and Representation. *Connection Science, Vol. 1 (1)*, p. 53-68.

Dyer, M. G. (1983): *In-Depth Understanding: A Computer Model of Integrated Processing for Narrative Comprehension.* Cambridge, MA: The MIT Press.

Dyer, M. G. (1990): Symbolic Neuroengineering for Natural Language Processing: A Multi-Level Research Approach. In J. Barnden and J. Pollack (eds.), *Advances in Connectionist and Neural Computation Theory*, Ablex Publishing. In press.

Feldman, J. A. (1989): Neural Representation of Conceptual Knowledge. In L. Nadel, L. A. Cooper, P. Culicover, and R. M. Harnish (eds), *Neural Connections, Mental Computation*, p. 68-103. Cambridge, MA: The MIT Press.

Feldman, J. A. & Ballard, D. H. (1982): Connectionist Models and Their Properties. *Cognitive Science, Vol. 6 (3)*, p. 205-254.

Fodor, J. A. & Pylyshyn, Z. W. (1988): Connectionism and Cognitive Architecture: A Critical Analysis. In S. Pinker and J. Mehler (eds.), *Connections and Symbols*, p. 3-71. Cambridge, MA: The MIT Press.

Granger R. H., Eiselt, K. P., & Holbrook, J. K. (1986): Parsing with Parallelism: A Spreading Activation Model of Inference Processing During Text Understanding. In J. Kolodner and C. Riesbeck (eds.), *Experience, Memory, and Reasoning*, p. 227-246. Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Hendler, J. (1988): *Integrating Marker-Passing and Problem Solving: A Spreading Activation Approach to Improved Choice in Planning*, Lawrence Erlbaum Associates, Hillsdale, New Jersey.

Lange, T. (1990): *High-Level Inferencing in a Localist Network*, Master's Thesis, Computer Science Department, University of California, Los Angeles.

Lange, T. & Dyer, M. G. (1988): Dynamic, Non-Local Role-Bindings and Inferencing in a Localist Network for Natural Language Understanding. In David S. Touretzky (ed.), *Advances in Neural Information Processing Systems I*, p. 545-552. San Mateo, CA: Morgan Kaufmann (Collected papers of the IEEE Conference on Neural Information Processing Systems — Natural and Synthetic, Denver, CO, November 1988).

Lange, T. & Dyer, M. G. (1989): Frame Selection in a Connectionist Model of High-Level Inferencing. *Proceedings of the Eleventh Annual Conference of the Cognitive Science Society (CogSci-89)*, Ann Arbor, MI, August 1989.

Lange, T., Hodges, J., Fuenmayor, M., & Belyaev, L. (1989a): DESCARTES: Development Environment For Simulating Hybrid Connectionist Architectures. *Proceedings of the Eleventh Annual Conference of the Cognitive Science Society (CogSci-89)*, Ann Arbor, MI, August 1989.

Lange, T., Vidal, J., & Dyer, M. G. (1989b): Artificial Neural Oscillators For Inferencing. *Proceedings of the International Workshop on Neurocomputers and Attention*, Moscow, USSR, September 1989. (Tech Rep. No. UCLA-AI-89-11, Computer Science Department, University of California, Los Angeles).

Lytinen, S. (1984): Frame Selection in Parsing. *Proceedings of the National Conference on Artificial Intelligence (AAAI-84)*, August 1984.

McClelland, J. L. & Kawamoto, A. H. (1986): Mechanisms of Sentence Processing: Assigning Roles to Constituents of Sentences. In McClelland & Rumelhart (eds.), *Parallel Distributed Processing: Vol 2*, p. 272-325. Cambridge, MA: The MIT Press.

Miikkulainen, R. & Dyer, M. G. (1988): Forming Global Representations with Extended Backpropagation. *Proceedings of the IEEE Second Annual Conference on Neural Networks (ICNN-88)*, San Diego, CA.

Miikkulainen, R. & Dyer, M. G. (1989): Encoding Input/Output Representations in Connectionist Cognitive Systems. In D. Touretzky, G. Hinton, and T. Sejnowski (eds.). *Proceedings of the 1988 Connectionist Models Summer School*. San Mateo, CA: Morgan Kaufmann.

Minsky, M. (1975): A Framework for Representing Knowledge. In P.H. Winston (ed.), *The Psychology of Computer Vision*, p. 211-277. New York: McGraw-Hill.

Norvig, P. (1986): *A Unified Theory of Inference For Text Understanding*. Doctoral Dissertation, Computer Science Department, University of California, Berkeley.

Riesbeck, C. K. & Martin, C. E. (1986). Direct Memory Access Parsing. In J. Kolodner and C. Riesbeck (eds.), *Experience, Memory, and Reasoning*, p. 209-226. Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Rumelhart, D. E., Hinton, G. E., & McClelland, J. L. (1986): A General Framework for Parallel Distributed Processing. In Rumelhart & McClelland (eds.), *Parallel Distributed Processing: Vol 1*, p. 45-76. Cambridge, MA: The MIT Press.

Schank, R. C. & Abelson, R. (1977): *Scripts, Plans, Goals and Understanding*. Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Shastri, L. (1988): A Connectionist Approach to Knowledge Representation and Limited Inference. *Cognitive Science, Vol. 12 (3)*, p. 331-392.

Sumida, R. A. & Dyer, M. G. (1989): Storing and Generalizing Multiple Instances While Maintaining Knowledge-Level Parallelism. *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence (IJCAI-89)*, Detroit, MI, August 1989.

Tomabechi, H. and Kitano, H. (1989): Beyond PDP: The Frequency Modulation Neural Network Architecture. *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence (IJCAI-89)*, Detroit, MI, August 1989.

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

Waltz, D. & Pollack, J. (1985): Massively Parallel Parsing: A Strongly Interactive Model of Natural Language Interpretation. *Cognitive Science, Vol. 9 (1)*, p. 51-74.