Efficient Inference with Multi-Place Predicates and Variables in a Connectionist System

Venkat Ajjanagadde and Lokendra Shastri

Department of Computer and Information Science
University of Pennsylvania

Abstract

The ability to represent structured knowledge and use that knowledge in a systematic way is a very important ingredient of cognition. An often heard criticism of connectionism is that connectionist systems cannot possess that ability. The work reported in this paper demonstrates that a connectionist system can not only represent structured knowledge and display systematic behavior, but can also do so with extreme efficiency. The paper describes a connectionist system that can represent knowledge expressed as *rules* and *facts* involving *multi-place* predicates, and draw *limited*, but *sound*, inferences based on this knowledge. The system is extremely efficient - in fact, optimal, as it draws conclusions in time proportional to the length of the proof. Central to this ability of the system is a solution to the *variable binding* problem. The solution makes use of the notion of a *phased clock* and exploits the time dimension to create and propagate variable bindings.

1 Introduction

McCarthy, in his commentary on Smolensky's paper: On the Proper Treatment of Connectionism[15], asserts that connectionist systems suffer from "the unary or even propositional fixation"; representational power of most connectionist systems is restricted to unary predicates applied to a fixed object. More recently, Fodor and Pylyshyn[10] have made sweeping claims that connectionist systems cannot incorporate systematicity and compositionality. These comments suggest that representing structured knowledge in a connectionist network and using this knowledge in a systematic way is considered difficult, if not impossible. This paper addresses these concerns. It describes a connectionist system that can represent knowledge expressed in terms of *rules* and *facts* involving *multi-place* predicates (i.e., *n-ary relations*) and draw limited but sound inferences based on this knowledge in an extremely efficient manner. The time taken by the system to draw conclusions is proportional to the *length* of the proof, and hence, optimal.

It is observed that the key technical problem that must be solved in order to represent and reason with structured and rule based knowledge is the *variable binding* problem[9, 16]. A solution to this problem using a *multi-phase* clock is proposed. The solution employs the time dimension to maintain and propagate variable bindings during the reasoning process.

The connectionist system for reasoning with rules described in this paper is computationally effective in a strong sense and is consistent with the ability of human agents to draw certain inferences extremely fast often in a few hundred milliseconds. The proposed system draws inferences in optimal time, i.e., in time proportional to the length of the proof.

2 Related Work

Two major metaphors that have been used for connectionist inference are that of energy minimization and spread of activation.

Ballard and Hayes[2] were the first to develop a connectionist inference system using the energy minimization[11] paradigm. They did not address the problem of variable binding as their system required that all possible bindings be explicitly

pre-wired into the network. Explicit pre-wiring is unacceptable as a solution to the variable binding problem as that would correspond to explicitly representing all possible instantiations of the rule. This is not feasible because the number of instantiations may be too many potentially unbounded. Ballard and Hayes' reasoner has two limitations which are common to all the reasoning systems employing the energy minimization paradigm. First of those limitations is regarding their efficiency. In those reasoners, the inference process is reduced to the problem of finding the lowest energy state of a suitably interconnected network. Such a process may even require not one but several cycles of convergence and it is difficult to place an upper bound on the convergence time of such systems. Even in cases where it is possible to do so, it turns out to be at best polynomial in the size of the knowledge base[5]. Thus, even though systems based on the energy metaphor are massively parallel, they do not meet the efficiency requirement. A second problem with such systems is that they are not always guaranteed to find the prescribed solution because the energy minimization process can get trapped in a local minima.

Touretzky and Hinton[17] have described DCPS, a *distributed* connectionist encoding of a restricted production system. The system uses the energy minimization metaphor for inference. The operation of a production system requires the ability to perform variable bindings, and DCPS exhibits this ability. The restrictions on variable bindings, however, are fairly strong. For example, DCPS only allows one variable in the antecedent. It also assumes that during any cycle there is only one rule with one variable binding that *can* constitute a potential correct match.

Among the other connectionist reasoning systems that employ energy minimization paradigm are the system of Dolan & Smolensky[7] (Dolan & Smolensky's system uses the tensor product based representation proposed in [16]) which is an improvement over Touretzky and Hinton's DCPS, Derthick's system[5] for drawing plausible inferences with respect to a frame based representation language and Dolan and Dyer's system for parallel retrieval and application of conceptual knowledge[6].

In the spreading activation metaphor for reasoning, each piece of information is encoded by a connectionist node and the inferential dependencies between pieces of information are represented by links between the corresponding nodes. Inference reduces to parallel spread of activation in such a network. Shastri's "connectionist realization of semantic networks"[14] follows such an approach. The system solves an interesting class of *inheritance* and *recognition* problems extremely fast - in time proportional to the depth of the conceptual hierarchy. Shastri's system displays the desired level of efficiency as its response is at worst logarithmic in the size of the knowledge base. However, it does not address the problem of variable binding. Although multiple rules participate in a derivation, it is always the case that all variables are bound to the same individual and thus the system can get by without actually solving the variable binding problem.

The suggestion of the usage of time dimension for representing variable bindings appears also in the works of Clossman[4], Fanty[8] and Malsburg[18].

3 Representation and Reasoning

The proposed connectionist system can perform a broad class of deductive inference involving variables and multi-place predicates with extreme efficiency. Specifically, the system can represent knowledge expressed in the form of *rules* and *facts* and determine whether a *query* can be derived as a consequence of the facts and rules encoded in the system. The answers to queries are produced in optimal time: the time taken to draw an inference is only proportional to the *length* of the proof.

The form of rules, facts, and queries is explained below.

Rules in the system are assumed to be sentences of the form
$\forall x_1, ..., x_m\ [P_1(...) \wedge P_2(...) ... \wedge P_n(...) \Rightarrow$
$\forall y_1, ..., y_k \exists z_1, ...z_l\ Q(...)]$
where arguments for P_i's are subsets of $\{x_1, x_2, ...x_m\}$, while the arguments of Q may consist of any number of arguments from among the x_i's and any number of constants besides the universally and existentially quantified arguments introduced in the consequent.

Facts are assumed to be atomic formulas of the form $P(t_1, t_2 ... t_k)$ where t_i's are either constants

or existentially quantified variables.

A query has the same form as a fact: it is an atomic formula whose arguments are either bound to constants or are existentially quantified. The enforcement of the sameness condition on the variables that occur more than once in the antecedents of the rules is limited to those which get bound due to the query.

Some examples of rules, facts, and queries follow:

Rules:

$\forall x, y, z\ give(x, y, z) \Rightarrow owns(y, z)$
$\forall x, y\ owns(x, y) \Rightarrow can\text{-}sell(x, y)$
$\forall x\ omnipresent(x) \Rightarrow \forall y, t\ present(x, y, t)$
$\forall x, y\ born(x, y) \Rightarrow \exists t\ present(x, y, t)$
$\forall x\ triangle(x) \Rightarrow number\text{-}of\text{-}sides(x, 3)$
$\forall x, y\ sibling(x, y) \wedge born\text{-}at\text{-}the\text{-}same\text{-}time(x,y) \Rightarrow twins(x,y)$

Facts:

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

Queries:

1. $owns(Mary, Book1)$; Does Mary own Book1?
2. $owns(x, y)$; Does someone own something?
3. $can\text{-}sell(x, Ball2)$; Can someone sell Ball2?
4. $present(x, Northpole, 1/1/89)$; Is someone present at the north pole on 1/1/89?
5. $number\text{-}of\text{-}sides(A3, 4)$; Does A3 have 4 sides?
6. $can\text{-}sell(Mary, Ball2)$; Can Mary sell Ball2?
7. $twins(Susan, Mary)$; Are Susan and Mary twins.?

All queries except 5 and 6 follow from the rules and facts and the system will respond 'yes' to these queries. The system will say 'no' to queries 5 and 6.

3.1 Directed reasoning

The strong efficiency requirement we have imposed on our system entails that it find a solution in a fixed number of passes of spreading activation. Such a convergence behavior ensures that the network can compute a solution in time proportional to the *diameter* of the network which is - in almost all cases - *sublinear* (and often *logarithmic*) in the size of the knowledge base. For the connectionist network to compute solutions in a *single* pass of spreading activation, the inferential dependencies in the knowledge base must be acyclic[13]. The nature of such inferential dependencies can be made explicit by expressing the rule component of the knowledge base in the following graphical manner. Depict each predicate occurring in the rules by a unique node in the graph. Then if there is a rule of the form

$$P_1(...) \wedge P_2(...) ... \wedge P_n(...) \Rightarrow Q(...)$$

in the knowledge base, draw directed arcs from the nodes corresponding to P_is to the node corresponding to Q. The requirement that the inferential dependencies of the knowledge base be acyclic amounts to requiring that the directed graph obtained in this manner be acyclic. We will therefore focus on knowledge bases whose *inferential dependency* graph corresponds to a *directed acyclic graph* and henceforth, we will often refer to the rule component of the knowledge base as the PDAG (for Predicate DAG).

In view of the directed nature of inferential dependencies, we refer to the system's inferential ability as *directed reasoning*. Directed reasoning appears to be adequate to capture a broad range of common sense reasoning situations. In particular, it can deal with restricted types of *causal reasoning*, i.e., reasoning about actions and events wherein there is no circular causality (i.e., systems that can be modeled as open loop systems). Terminological reasoning[3], that is, reasoning with definitional knowledge of concepts (terms) is also a case of directed reasoning[1].

4 The Connectionist Encoding

This section discusses the connectionist encoding of rules and facts. During most of our discussions we will be focusing on rules having a single predicate on the antecedent; the extension to rules having more than one predicate on the antecedent is rather simple and will be briefly discussed in a subsequent section. Due to space limitations, we are

attempting to provide only a simplified picture of the whole system here and hence, are omitting details such as the soundness and completeness conditions and the details of encoding of rules having constants and existentially quantified variables in the consequent. The full details of the reasoning system can be found in [12].

The whole encoding makes use of only simple phase-sensitive Binary Threshold Units (BTUs). However, for clarity of exposition, we will be making use of two abstract types of nodes, which we call *pred* and *instancer*. The realization of these types of nodes in terms of BTUs is described in [12].

An *n-ary* predicate is represented by a *pred* node (drawn as a rectangular box) and a cluster of n *arg* nodes (depicted as diamonds). Thus the ternary predicate *orderhit* is represented by the *pred* node labeled $ORDERHIT$ and the three *arg* nodes - $a1$, $a2$, and $a3$ - drawn next to it (Fig. 1).

Each constant in the domain is represented by a *const* node (an oval shaped node), which is a simple phase sensitive BTU that becomes active in phase i of every clock cycle if it is initially activated in the i^{th} phase of a clock cycle.

A rule is encoded by interconnecting nodes representing the antecedent and consequent predicates. For example, the interconnections corresponding to the rule

$$\forall x, y, z P1(x, y, z) \Rightarrow Q(y, x)$$

are as follows: there will be a link from the *pred* node corresponding to $P1$ to the *pred* node corresponding to Q; there will be links going from the first and second arg nodes of Q to the second and first arg nodes of $P1$ respectively. The links between the arg nodes represent the correspondence between the arguments of the consequent and antecedent predicates of the rule. (Refer to the encoding of the rule $\forall x, y, z(orderhit(x, y, z) \Rightarrow hit(y, z))$ in Fig.1).

A fact is encoded using an *instancer* node (drawn as hexagonal box). An *instancer* node representing a fact concerning an n-ary predicate has n BIND sites. The i^{th} BIND site has links coming from the i^{th} arg node of the corresponding predicate and the *const* node representing the constant bound to the i^{th} argument in the fact represented by the instancer.

5 Inference Process

The inference process, that is, the verification of the truth or falsity of a query, is a controlled spread of activation in the network with no external intervention. The inference process may be thought of as consisting of three stages [1]. In the first stage, the query is posed to the network by external activation of some nodes. During the second stage, a controlled parallel search is carried out to locate all the facts that are relevant to the proof of the query and the *instancer* nodes encoding such relevant facts become active. In the third and final stage the actual proof is constructed. In this stage, activation from the instancers denoting relevant facts flow downwards along the inference paths in the PDAG to produce an answer to the query. The answer corresponds to the resulting activation of the *pred* node that corresponds to the query predicate.

5.1 Posing the query and specifying variable bindings

As said earlier, a query is an atomic formula of the form $P(t_1, ..., t_k)$ where t_is are either constants or existentially quantified variables. Posing the query involves specifying the constant- argument bindings of the query predicate to the network. These bindings of arguments are indicated by using a phased clock. For a given query, each clock cycle of the network consists of a fixed number of phases. If the argument bindings in the query involve p distinct constants, then the clock has p distinct phases[2]. Let $c_1, ..., c_p$ be p distinct constants appearing in the bindings specified in the query. The query will be posed in the following manner:

In the i^{th} phase of the *first* clock cycle, ($1 \leq i \leq p$), the following nodes will be activated:

- The *const* node corresponding to c_i.
- The *arg* nodes corresponding to the $i_1^{th}, ..., i_j^{th}$ arguments of the query predicate, where $i_1, ..., i_j$ ($j \geq 1$) are the arguments of the query predicate bound to c_i.

[1] These stages are conceptually distinct, however, during actual processing these stages overlap

[2] In general p can be less than the number of bound arguments in the query because the same constant(s) may be bound to more than one argument.

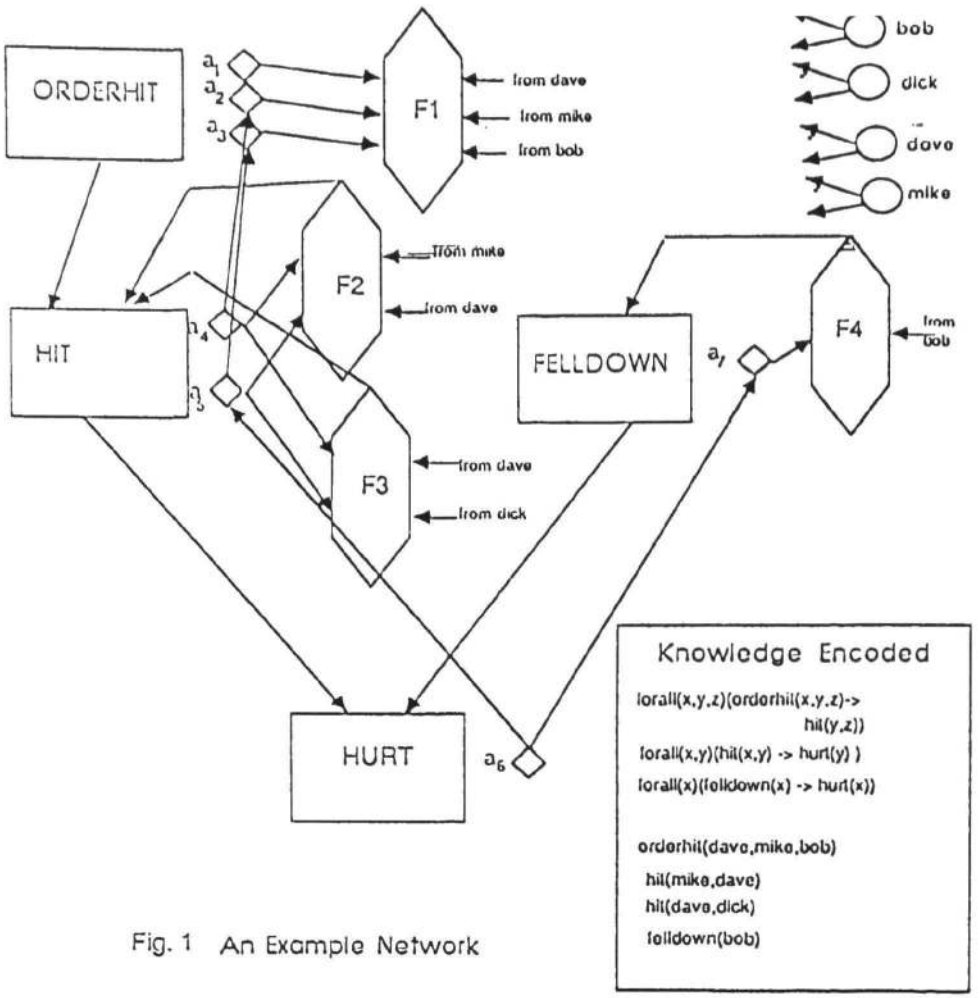

Fig. 1 An Example Network

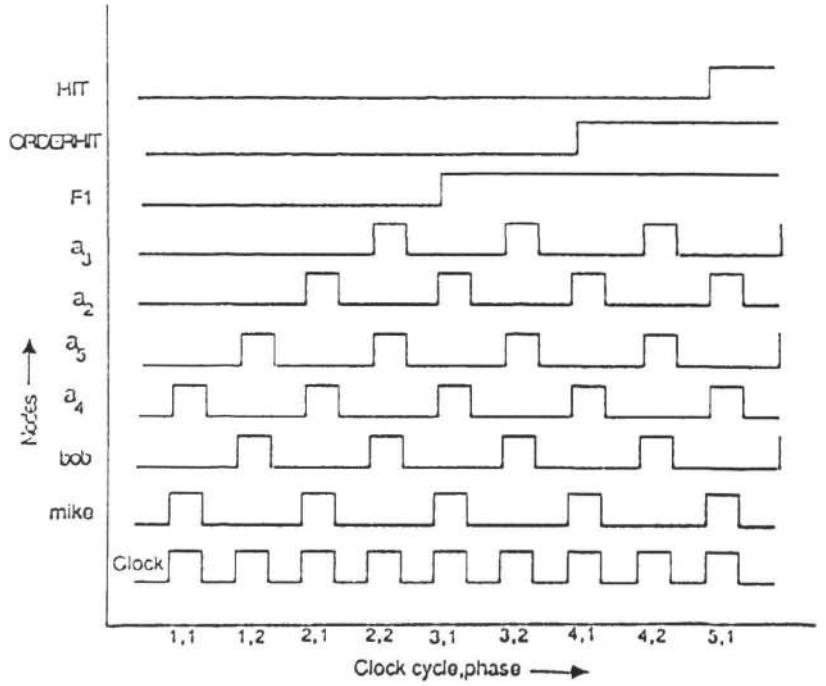

Fig. 2 Activations of different nodes for the query hit(mike,bob)

As stated earlier, *arg* nodes and *const* nodes are phase sensitive and the phases in which they remain active are determined by the clock phases in which they first become active. The simultaneous activation of an *arg* node and a *const* node during a phase represents that the constant denoted by the latter node is bound to the argument denoted by the former node.

5.2 Propagating variable bindings and getting the relevant instancers active

Once the query is posed, the parallel search for the assertions that are relevant to the proof of the query ensues. Relevant assertions can be of two types:

There may exist a fact associated with the query predicate itself whose argument bindings subsume the bindings specified in the query. The query would follow directly from such a fact. For example, the query $hit(dave, dick)$ (i.e., "Did Dave hit Dick?") trivially follows from the fact $hit(dave, dick)$ (refer to Fig. 1.) [3].

The second possibility is that there exist fact(s) associated with ancestor predicate(s) of the query predicate and whose argument bindings subsume those specified in the query. In this case, the query would follow via a chain of modus ponens. As an example, in Fig.1, the fact $orderhit(dave, mike, bob)$ is relevant to the proof of $hit(mike, bob)$ this way.

We consider, in turn, how the two types of relevant facts become active during the query process. Consider how the *instancer* node F3 (representing the fact $hit(dave, dick)$) becomes active in response to the query $hit(dave, dick)$. Once this query is posed, the *const* node *dave* and the first *arg* node of *hit* remain active during the first phase of every clock cycle. Similarly, the *const* node *dick* and the second *arg* node of *hit* remains active during the second phase of every clock cycle. The activation from these *arg* and *const* nodes reaches the *instancer* node F3 during the specified phases. An *instancer* node functions as follows:

An *instancer* node becomes active at the end of clock cycle t and remains active throughout cycle $t+1$ if and only if

- During each phase of clock cycle t, *if* it receives activation from an *arg* node, it *also* receives activation from the *const* node bound to this *arg* node.

It follows that as a result of the query $hit(dave, dick)$, the *instancer* F3 will become active at the end of the second clock cycle and remain active thereafter.

To see how relevant *instancer* nodes associated with ancestors of the query predicate become active we shall consider the query $hit(mike, bob)$ (refer to Fig. 1). There is no fact associated with *hit* that subsumes the bindings in this query. As a result of the query, the first *arg* node of *hit* (a_4) and the *const* node *mike* will become active in the first phase of every clock cycle. Similarly, the second *arg* node of *hit* (a_5) and the *const* node *bob* will become active during the second phase of every clock cycle. (The clock phases/cycles in which different nodes first become active for the example query being discussed, i.e., $hit(mike, bob)$ are indicated in Fig. 2. Note that the *pred* and *instancer* nodes are not phase sensitive).

Activations from the *arg* nodes a_4 and a_5 reach the *arg* nodes a_2 and a_3 of the predicate *orderhit* respectively. As the phase in which an *arg* node becomes active depends on the phase in which it receives activation, the *arg* nodes a_2 and a_3 become active in the first and second phases respectively of every clock cycle. Hence, the second clock cycle onwards, the active *const* and *arg* nodes in the first phase of every clock cycle are: *mike*, a_4 and a_2; and those active in the second phase are: *bob*, a_3 and a_5. Essentially, we have created two new bindings: *mike* has been bound to the second argument of *orderhit* and *bob* has been bound to the third argument of *orderhit*[4]. The *instancer* $F1$ that encodes the fact $orderhit(dave, mike, bob)$ will now become active as a result of the activation it receives from the *arg* nodes a_2 and a_3 and the corresponding *const* nodes *mike* and *bob*. The activation from the *instancer* node $F1$ causes the output of the *pred* node *orderhit* to become high. The activation from the *pred* node *orderhit* in turn makes the output of the *pred* node *hit* high thus resulting in an affirmative answer to the query.

[3] The fact $hit(dave, dick)$ also subsumes other queries such as $\exists x hit(x, dick)$, $\exists x hit(dave, x)$, etc., all of which also follow, directly, from this fact.

[4] The newly created bindings of the arguments of *orderhit* can be thought of as encoding the query $orderhit(x, mike, bob)$ (i.e., "Did someone order Mike to hit Bob?")!

The above was a brief description of the inference process where the rules encoded in the network had just one predicate as antecedent. In order to encode rules of the form $P_1(...) \wedge P_2(...) \wedge ...P_m(...) \Rightarrow Q(...)$, i.e., rules with conjunctive antecedents, the output of the *pred* nodes $P_1, ..., P_m$ are not connected directly to the *pred* node Q; instead they are connected to a *conjunctive* node, which is in turn linked to the *pred* node Q. The output of the *conjunctive* node is high if and only if it receives activation through all the incoming links. The interconnections between the *arg* nodes of the antecedent predicates and the consequent predicate is similar to that in the case of single-antecedent rules.

6 Conclusion

The work described in this paper has directly addressed a criticism that is often levelled against connectionist systems, namely, that connectionist systems cannot incorporate systematicity and compositionality and hence are unpromising as architectures of cognition. The paper presented a connectionist system that only uses simple phase sensitive binary threshold units to perform a limited class of inferences with rules and facts. The problem of variable binding is central to the connectionist realizations of rule governed symbolic reasoning tasks. The proposed connectionist system employs a phased clock to solve this problem. The design of the system has been verified via simulations.

In the near future we will report an augmented system that can answer *wh*-questions in addition to 'yes/no' questions(i.e., the augmented system is capable of determining the fillers of unbound arguments in the query). We will also show that there exists a direct way of integrating a connectionist semantic network(i.e., an inheritance network) such as the one described in [14] and the rule-based system described here. Such a 'hybrid' system will have more expressive and inferential power but will retain its extreme efficiency.

Acknowledgements

We wish to thank the Knowledge Representation group at the International Computer Science Institute, Berkeley, in particular, Jerry Feldman and Mark Fanty for their helpful comments and suggestions. This work was supported by NSF grants IRI 88-05465, MCS-8219196-CER, MCS-83-05211, DARPA grants N00014-85-K-0018 and N00014-85-K-0807, and ARO grant ARO-DAA29-84-9-0027.

References

[1] Ajjanagadde V., Forthcoming Ph.D. dissertation, University of Pennsylvania.

[2] Ballard, D.H., and Hayes, P.J., Parallel logical inference, Proceedings of the Sixth Annual Conference of the Cognitive Science Society, pp. 114-123., Boulder,Colorado, June.1984.

[3] Brachman, R., Fikes R., and Levesque, H.J. KRYPTON: A Functional Approach to Knowledge Representation. *Readings in Knowledge Representation*, R. Brachman, and H.J. Levesque (eds.) Morgan Kaufman, Los Altos, CA. 1985.

[4] Clossman, Gary., (Personal Communication via John Barnden).

[5] Derthick, M., Mundane reasoning by parallel constraint satisfaction, Ph.D. thesis, CMU-CS-88-182, Carnegie Mellon University, Sept. 1988.

[6] Dolan, C., and Dyer, M., Parallel retrieval and application of conceptual knowledge, Technical Report TR UCLA-AI-88-3, University of California, Los Angeles, Jan. 1988.

[7] Dolan, C., and Smolensky P., Implementing a connectionist production system using tensor products, Technical Report UCLA-AI-88-15, University of California, Los Angeles, CU-CS-411-88 University of Colorado, 1988.

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

[9] Feldman, J.A. Dynamic connections in neural networks, *Bio-Cybernetics*, 46:27-39, 1982.

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

[11] Kirkpatrick, S., C.D.Gelatt, an M.P. Vecchi, Optimization by simulated annealing, *Science* 220, 4598, pp. 671-680, 1983.

[12] Shastri, L., and Ajjanagadde, V., A connectionist system for rule based reasoning with multiplace predicates and variables, Tech. Report MS-CIS-8905, Dept. of Computer Science, Univ. of Pennsylvania, Jan. 1989.

[13] Shastri, L., The Relevance of Connectionism to AI: A representation and reasoning perspective. In *Advances in Connectionist and Neural Computation Theory*, vol. 1., J. Barnden (ed.), Ablex Publishing Company, Norwood, N.J. (To appear). (Also available as a Tech. Report from Computer Science Department, University of Pennsylvania.)

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

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

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

[17] Touretzky, D. and Hinton, G., A distributed connectionist production system, *Cognitive Science*, 12(3), pp. 423-466.

[18] von der Malsburg, C., Nervous structures with dynamical links, *Berichte der Bunsen-Gelschaft fur Physikakalische Chemie.*