A Connectionist Representation of Symbolic Components, Dynamic Bindings and Basic Inference Operations

Nam Seog Park[1] and **Dave Robertson**[2]

Abstract. If we wish to replicate symbolic styles of inference in a connectionist system, we are constrained to work with simple network elements, with localised behaviours and simple forms of link to others. However, there remains considerable scope for experiment with different forms of element and links. This paper motivates our choice of a particular element and link arrangement which we believe to be more uniform in structure than other related architectures while also allowing a closer match to the basic elements of symbolic inference which we wish to explore. Using this element, this paper describes how dynamic bindings and basic symbolic inference procedures (matching and substitution) are represented in a connectionist manner. The paper also provides entry points to more extensive descriptions of the system and its use.

1 INTRODUCTION

An ability to encode structured information and instantiate it in specific situations to draw appropriate inferences is essential to enable a connectionist system to replicate symbolic inference. However, the ability of a connectionist system to represent constituent structure has been doubted. McCarthy [13] points out that most connectionist systems suffer from a propositional fixation and their representational power often at best restricted to unary predicates applied to a fixed object. Foder & Pylyshin [7] and Foder & McLaughlin [8] also claim that representation in connectionist systems do not allow constituent structure (compositionality) and this prevents them from supporting systematic, structure-sensitive operations of symbolic style inferences.

These claims have been weakened by some connectionist systems which have demonstrated structure-sensitivity without being separated [3, 4, 5, 6, 9, 10, 14, 17, 20, 21]. These non-separated representations can be subjected to structure-sensitive operations without the need of unpacking the representations into their constituents or converting them into separate versions. But even in the non-separated representation case, there is a type of separated representation at a higher level of description by dividing its activation pattern into segments corresponding to regularly recurring functional groups [2]. These activation patterns are the main from of short-term constituent structure in a distributed connectionist system.

Other connectionist systems using separated constituent structure to achieve structure sensitivity are found in [1, 11, 12, 18, 19]. All these connectionist models use localist networks in which a node or a group of nodes is dedicated to represent always the same part of a constituent structure. In this case, symbolic-like short-term constituent structure is represented by an activation pattern over those groups of nodes. This is desirable because (as in symbolic systems) it is possible to determine the structures of the network responsible for representing different fragments of information. However, none of these systems gives a broad coverage of the styles of reasoning available in symbolic systems and many of them use special purpose configurations to extend the range of inferences they support. This is mainly due to limitations of the dynamic variable binding mechanism they adopted. We summarise our experiments in attempting to find close correspondences between basic building blocks of symbolic inference (such as variable matching and substitution) and a connectionist mechanism involving groups of elements. A wide range of symbolic inferences is possible in this connectionist mechanism without recourse to special purpose mechanisms.

[1] Artificial Intelligence Department, University of Edinburgh, 80 South Bridge, Edinburgh EH1 1HN, UK; namseog@aisb.ed.ac.uk
[2] Artificial Intelligence Department, University of Edinburgh, 80 South Bridge, Edinburgh EH1 1HN, UK; dr@aisb.ed.ac.uk

ECAI 96. *12th European Conference on Artificial Intelligence*
Edited by W. Wahlster
Published in 1996 by John Wiley & Sons, Ltd.

2 DYNAMIC BINDINGS AND BASIC INFERENCE OPERATIONS

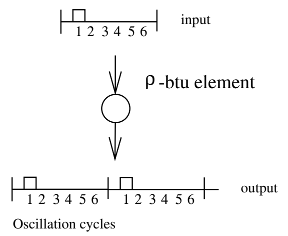
2.1 Dynamic Bindings and A Temporal Synchrony Approach

One of the most important short-term constituent structures in symbolic inference is dynamic bindings. Bindings in a symbolic system are a special representation which indicate that a value (sometimes called a filler) is substituted for a variable symbol. A variable is said to be bound to the value substituted for it. According to types of symbol names involved, there are two types of bindings: a constant binding and a variable binding. The term *constant binding* will be used to refer to the situation where a variable gets bound to a constant value and *variable binding* the situation where a variable gets bound to a variable value. For instance, a binding between a and X, {a/X}, is called a constant binding and between U and X, {U/X}, a variable binding.

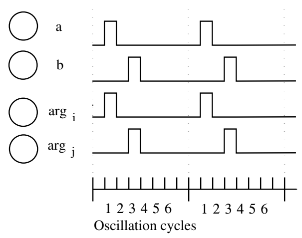
The importance of a dynamic representation of these bindings is due to its ability to create and to destroy a constituent structure dynamically during inference. The constituent structure, $p(a,b,U)$, cannot be represented dynamically by simply activating the nodes representing the arguments, $p{:}arg_1$, $p{:}arg_2$, and $p{:}arg_3$, and the nodes representing the constituents a, b, and U. The problem is that the given formula does not merely express an association between the constituents, it expresses a specific relation in which each constituent plays a distinct role [18]. What is required is a connectionist representation of correct bindings between predicate arguments and fillers, $\{a/p{:}arg_1,\ b/p{:}arg_2,\ U/p{:}arg_3\}$, and the distinction between the bindings regardless of their type. Thus, the dynamic variable binding problem in connectionist systems arises when the system needs a representation of constituent structure whose constituents (fillers) change from one inference to others. Representing this sort of dynamic constituent structure however has posed a problem for standard connectionist models (see survey in [15]).

One of efficient solutions to this dynamic binding problem is a temporal synchrony solution proposed by Shastri & Ajjanagadde [18]. They observed that the existence of rhythmic, temporal synchronous activity, in the animal brain suggested one solution to the dynamic variable binding problem in connectionist systems. The fundamental features of this solution lies in a separation of a period of oscillation cycle into several phases and the use of phase-sensitive neuron elements. A *phase* is a minimum time interval in which a neuron element performs its basic computations – sampling its inputs and thresholding – and an *oscillation cycle* is a window of time in which neuron elements show their oscillatory behaviours.

As an example of this approach consider the following phase-sensitive neuron element called a ρ-btu element used by them. The ρ-btu node has a temporal property that it becomes active on receiving a periodic spike train and produces a periodic spike train that is in-phase with the driving input. The detailed oscillatory behaviour of this element is depicted in Figure 1a.



(a) Oscillatory behaviour of the element



(b) Representing constant bindings

Figure 1. The temporal behaviours of a ρ-btu element and its use to represent dynamic bindings

Representing dynamic bindings using this temporal behavious of the ρ-btu node is summarised as follows:

- all constant entities involved in a dynamic binding are represented using ρ-btu elements;
- a constant binding between two entities, $\{a/arg_i\}$ for instance, is represented by activating the element representing the constant a and the element representing arg_i in the same phase.

Figure 1b demonstrates the situation where two bindings, $\{a/arg_i, b/arg_j\}$, are represented using synchronous activity of ρ-btu elements representing the entities involved[3]

[3] The number of phases in one oscillation limits a number of distinct entities participating in bindings during inference. However, this does not limit the number of constants which may be represented. The restriction is on those used each time we answer a query.

One important limitation of this approach is that it cannot represent variable entities nor variable bindings. This is because a ρ-btu node is used to represent only constant entities in their mechanism. For instance, two variable bindings, $\{U/arg_i,\ V/arg_j\}$, are identically represented by inactive states of the ρ-btu nodes representing arg_i and arg_j. No ρ-btu node is used to represent the variable fillers U and V. Consequently, these two bindings are treated as the same binding in their system. Any connectionist systems which resemble symbolic inference should be capable of dealing with both constant and variable bindings dynamically without disrupting the structure of the network.

2.2 Basic Inference Operations

To achieve a symbolic style of inference, two basic symbolic inference operations must be implemented in a connectionist manner: matching and substitution. Let us consider the following rule:

$$p(X, Y) \rightarrow q(X, Y)$$

When the predicate, $p(a, U)$, is presented to the antecedent of the given rule for forward chaining, symbolic inference system normally applies the match operator to see if the antecedent of the rule matches with the presented predicate. This yields a set of substitutions between arguments and fillers in the matching terms as follows:

$$Match[p(a,U),\ p(X,Y) \rightarrow q(X,Y)] = \{a/X,\ U/Y\}\ .$$

As the result, two types of bindings are obtained: the constant binding between a and X and the variable binding between U and Y, which are differentiated by different symbol names used.

In the temporal synchrony approach, the effect of this matching procedure can be achieved by the initial binding setup procedure. If one ρ-btu element and two assemblies of ρ-btu elements represent the constant a and the predicates of the example rule (p and q), the bindings to be produced by the symbolic matching operation, $\{a/p{:}arg_X, U/p{:}arg_Y\}$, can be represented in a connectionist manner by activating the involved elements in the following phase:

$$a[1],\ p{:}\{arg_X[1], arg_Y[\phi]\},\ q{:}\{arg_X[\phi], arg_Y[\phi]\}$$

The binding $\{a/arg_X\}$ is represented by in-phase activation between the element representing $a[1]$ and that representing $arg_X[1]$. However, the variable binding, $\{U/p{:}arg_Y\}$, is represented simply by inactive state of $p{:}arg_Y[\phi]$. The symbol "ϕ" is used to indicate inactive state of the element.

Once a set of bindings is obtained, these bindings are then used for the next step of inference, the substitution procedure. The substitution operator takes the matched formula and consistently substitutes all occurrences of variables in the formula using the binding set obtained from the matching operation as follows:

$$Substitute[\{a/X, U/Y\},\ p(X,Y) \rightarrow q(X,Y)] = p(a,U) \rightarrow q(a,U)$$

From this result, a symbolic inference system can apply the *modus ponens* inference rule to infer the new logical assertion, $q(a, U)$.

The same effect of this can be achieved by replicating the substitution procedure in a connectionist manner. Note that this substitution result represents the situation where all arguments (X and Y) of the antecedent and consequent predicates are substituted with the constant a and the variable U. This situation can be made by activating each argument node of the antecedent and its corresponding argument node of the consequent predicate in the same phase as the filler node assigned. Because the argument nodes of the antecedent predicate (p) and their corresponding filler nodes are already activated by the initial binding instantiation (the phases 1 and ϕ in the previous example), what is required is activating the argument nodes of the consequent predicate (q) in-phase with them as follows:

$$a[1],\ p{:}\{arg_X[1], arg_Y[\phi]\},\ q{:}\{arg_X[1], arg_Y[\phi]\}$$

The new oscillatory behaviour between the node $q{:}arg_X[1]$ and the constant node $a[1]$ represents the new constant binding $\{a/q{:}arg_X]\}$ and the inactive state of $q{:}arg_Y[\phi]$ might be considered as the other binding $\{U/q{:}arg_Y\}$ – if we assume that U is in an inactive state.

A connectionist way of achieving this result is to set up links between the antecedent argument nodes and the consequent argument nodes. Whenever initial bindings are set on the antecedent predicate assembly, these bindings automatically propagate to the consequent predicate assembly to produce the substitution effect.

Having recognised the limitation of the temporal synchrony approach to handle variable bindings dynamically, the rest of the paper proposes how this limitation can be overcome. The extension is made by extending the temporal synchrony approach in such a way that it can accommodate variable bindings and supplying the connectionist equivalents to symbolic inference procedures using them.

3 REPRESENTING SYMBOLIC ENTITIES

Identifying one symbolic entity from others using symbolic names is a key property in a symbolic inference because this allows each component of a symbolic description of knowledge to be clearly separated. To obtain this facility in a connectionist manner, we generalise a ρ-btu node and introduce a special cluster of neuron elements, called an entity node.

3.1 A π-btu Element

A π-btu element is a generalised ρ-btu node. It can accommodate more than one spike signals within one oscillation cycle. The detailed behaviour of this neuron element is as follows:

Definition: A **π-btu element** becomes active on receiving one or more spikes in different phases of any oscillation cycle. On becoming active, a π-btu element produces an oscillatory spike that is in-phase with the driving inputs. A threshold, n, associated with each neuron-like element indicates that the element will fire only if it receives n or more spike inputs in the same phase. On becoming active, this element continually produces an output (oscillating) for the duration of inference. Figure 2 gives graphical description of this neuron element and its temporal characteristics.

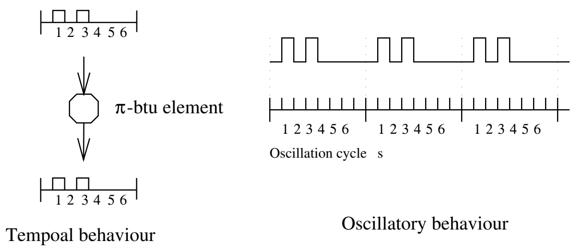


Figure 2. Temporal behaviours of a π-btu element

3.2 Entity Nodes

Definition: An **entity node** is a pair of π-btu elements. The left element is used to represent a variable role of the entity and the right element a constant role. A symbolic entity name may be used as a label of an entity node. Symbolically an entity node is represented using the notation,
entity_name([ϕ],[ϕ]), where the left square bracket represents the state of the left π-btu element and the right square bracket that of the right π-btu element. The notations, $entity\_name_{left}[\phi]$ or
$entity\_name_{right}[\phi]$ are used to indicate each state of the left or the right π-btu element. When an entity node is used simply to represent a constant or a variable, the notations, *var_name[ϕ]* or *const_name[ϕ]* is

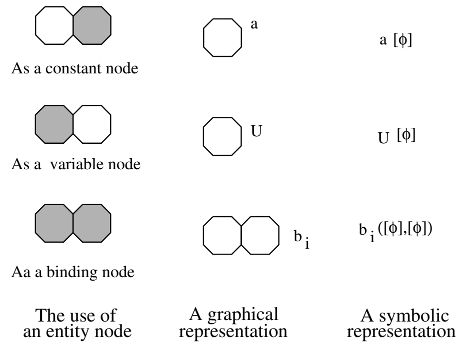


Figure 3. The use of an entity node

used to shorten the representation. Figure 3 illustrates the use of an entity node.

Since each entity node consists of two π-btu elements, there are three possible activation states that the entity node can have: (i) only the left element becomes active, (ii) only the right element becomes active, (iii) both elements become active. According to these states an entity node is used to represent three different types of nodes: a constant node, a variable node, and a binding node. When an entity node is used to represent a constant, only its right element becomes active during a chain of inference. When the entity node is used to represent a variable, only its left node becomes active. When it is used to represent a binding between constant and variable, both its elements become active.

4 REPRESENTING DYNAMIC BINDINGS

Using an entity node defined and its temporal behaviour, dynamic bindings involving both constant and variable fillers are represented in a connectionist manner without adding any additional mechanisms. Let us consider the two filler nodes, *a[ϕ]* and *U[ϕ]*, representing the constant a and the variable U, and the two binding nodes, b_i*([ϕ],[ϕ])* and b_j*([ϕ],[ϕ])*. A constant binding and a variable binding between filler nodes and binding nodes are now represented as follows:

- the constant binding between a and b_i, $\{a/b_i\}$, is represented by activating the constant node *a[ϕ]* and the right element of b_i*([ϕ],[ϕ])* in the first phase, which results in the activation of nodes, *a[1]* and b_i*([ϕ],[1])*, where the number 1 indicates the first phase;
- the variable binding between U and b_j, $\{U/b_i\}$, is represented by activating the variable node *U[ϕ]* and the left element of b_j*([ϕ],[ϕ])* in a different

phase, the second for instance, which results in the activation of the nodes *U[2]* and b_j *([2],[ϕ])*, where the number 2 stands for the second phase.

Once activated, by the temporal behaviour, these nodes oscillate with their own activation patterns. Figure 4 illustrates their activities.

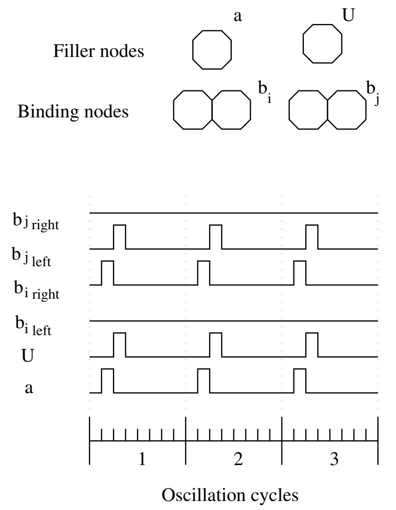


Figure 4. A representation dynamic bindings

As can be seen, these synchronous activities between the filler nodes and the binding nodes efficiently represent dynamic bindings between them. A constant binding or a variable binding is distinguished by which element of a binding node becomes active. If the binding nodes represent specific arguments of a predicate, this situation represents bindings between fillers and arguments. By activating different set of filler nodes in-phase with the binding nodes, different temporal dynamic bindings can be obtained.

In many cases, symbolic inference requires a group of unifying constant and variable entities to be represented at the same time. For example, if the predicate *p(a, U)* is presented to the antecedent of the following rule

$$p(X, X) \rightarrow q(X)$$

the result *q(a)* is expected with the binding {a/U} – with the fillers *a* and *U* in the same unifying group.

This sort of dynamic binding can be represented indirectly in the proposed connectionist representation. If a binding node *q:arg*$_X$ *([ϕ],[ϕ])* represents the first argument of the *q* predicate, the indirect representation of $\{a/U\}$ is achieved by activating the left and right elements of the binding node in *q:arg*$_X$ *([2],[1])*, provided the constant node *a[ϕ]* and the variable node *U[ϕ]* are activated in the first and the second phases. Then the left element of the binding node (*q:arg*$_{X_{left}}$*[2]*) and the variable node *U[2]* indicates the variable bindings $\{U/q{:}arg_X\}$ and the right element (*q:arg*$_{X_{right}}$*[1]*) and the constant node *a[1]* the constant binding $\{a/q{:}arg_X\}$, from which $\{a/U\}$ is obtained.

A set of more complicated dynamic bindings such as $\{a/U, a/V\}$ which belong to the same unifying group can be seen when the predicate *p(a, U, V)* is presented to the following rule:

$$p(X, X, X) \rightarrow q(X)$$

The dynamic bindings to be obtained from the *q* predicate will be represented by activating the binding node representing the argument of the *q* predicate in *q:arg*$_X$ *([2,3],[1])* if the filler nodes are active in *a[1]*, *U[2]*, and *V[3]* during the initial binding instantiation procedure. Again the left element represents the results *q(U)* and *q(V)*, and the right element the result *q(a)*, which yields the bindings $\{a/U, a/V\}$.

5 REPRESENTING INFERENCE OPERATIONS

Let us reconsider the situation where the predicate *p(a, U)* and the rule $p(X, Y) \rightarrow q(X, Y)$ used to demonstrate the symbolic matching operation. In the extended dynamic binding mechanism, a variable binding is represented in the same way as a constant binding is represented. If the given fillers and predicates of the rule are represented using entity nodes and assembly of entity nodes as follows:

$$a[\phi],\ U[\phi],\ p{:}\{arg_X([\phi],[\phi]), arg_Y([\phi],[\phi])\},$$
$$q{:}\{arg_X([\phi],[\phi]), arg_Y([\phi],[\phi])\}$$

the effect of the matching operation is accomplished by activating the filler nodes and their corresponding argument nodes in the same phase as follows:

$$a[1],\ U[2],\ p{:}\{arg_X([\phi],[1]), arg_Y([2],[\phi])\},$$
$$q{:}\{arg_X([\phi],[\phi]), arg_Y([\phi],[\phi])\}$$

The in-phase activation between the constant filler *a[1]* and *p:arg*$_{X_{right}}$*[1]* represents the constant binding $\{a/arg_X\}$ and the in-phase activation between the variable filler *U[2]* and *p:arg*$_{Y_{left}}$*[2]* the variable binding $\{U/arg_Y\}$.

After the matching procedure, the generated bindings are used for the substitution operation from which the new axiom *q(a, U)* is obtained:

$$substitute[\{a/X, U/Y\},\ p(X,Y) \rightarrow q(X,Y)] = p(a,U) \rightarrow q(a,U).$$

By inserting an intermediate mechanism which connects each argument node of the antecedent predicate assembly with its corresponding argument node of the consequent predicate assembly, this substitution effect is achieved automatically. When the initial bindings propagate from the consequent predicate assembly to the consequent predicate assembly through the intermediate mechanism (simple links in this case), the q predicate assembly becomes active in the following phases:

$$a[1],\ U[2],\ p{:}\{arg_X([\phi],[1]), arg_Y([2],[\phi])\}, q{:}\{arg_X([\phi],[1]), arg_Y([2],[\phi])\}$$

The new oscillatory behaviour between the constant node $a[1]$ and the argument node $q{:}arg_X([\phi],[1])$ represents the new constant binding $\{a/q{:}arg_X\}$ and in-phase activation between the variable node $U[2]$ and the argument node $q{:}arg_Y([2],[\phi])$ the new variable binding $\{U/arg_Y\}$. This situation, as it stands, can be interpreted as the result of a symbolic substitution procedure, to yield $p(a,U) \rightarrow q(a,U)$.

6 HANDLING DIFFICULT ISSUES

Until now a basic simple rule is used to illustrate the enhanced dynamic binding mechanism and connectionist symbolic inference procedures. If the proposed mechanism has to deal with a more complex symbolic rule such as $p(X,X) \rightarrow q(X)$ which requires repeated arguments of the p predicate to get bound to the same constant filler or free variable fillers, the connectionist initial binding instantiation procedure cannot fully represent the effect of the symbolic matching procedure. This is because the initial binding instantiation mechanism simply sets up bindings without involving any consistency checking within a group of unifying arguments (repeated arguments X's in this rule). Therefore, an additional connectionist sub-mechanism is needed to detect a consistency violation when two different constant fillers are assigned to the repeated arguments.

This requires an additional sub-mechanism to be added to the intermediate mechanism which connects the antecedent predicate assembly with the consequent predicate assembly. The role of this additional sub-mechanism is ensuring consistency conditions that are enforced by the syntax of a rule and blocking binding propagation from the antecedent predicate assembly to the consequent predicate assembly if there is any consistency violation.

Other types of rules which also require this sort of special treatment are

- rules having constant arguments: $p(a,b,X) \rightarrow q(b,X)$;
- rules requiring consistency checking across different groups of unifying arguments: $p(X,X,Y,Y) \rightarrow q(X,Y)$.

If constant arguments appear in the antecedent of a rule, an additional sub-mechanism has to perform consistency checking to ensure all the fillers assigned to the constant arguments are the same as the ones specified as constant argument names. As for a rule which requires consistency checking across different groups of unifying arguments, this mechanism has to check any consistency violation across across different groups of unifying arguments.

Since each type of rules requires a different structure for the additional sub-mechanism, a special algorithm is needed to decide various types of sub-mechanisms according to different types of symbolic rules given. More description of these issues can be found in [16].

7 CONCLUSION

Resembling symbolic inference in a connectionist manner requires how to represent dynamic bindings and how to implement two basic symbolic inference operations. The temporal synchrony approach is a step to this direction but it has the limitation in representing variable bindings in a connectionist manner.

To overcome this limitation, this paper introduced a generalised ρ-btu element, called π-btu element, and an entity node consisting of a pair of π-btu elements. Dynamic bindings, including both constant and variable bindings, are effectively represented by synchronous activity between entity nodes representing fillers and binding nodes representing arguments. This paper also demonstrated how two symbolic inference operations, matching and substitute, can be represented in a connectionist manner. The initial binding instantiation procedure performs the task similar to the symbolic matching operation, while the intermediate mechanism between the antecedent predicate assembly and the consequent predicate assembly the task similar to the symbolic substitution. Since the initial binding instantiation procedure does not include consistency checking, an additional consistency checking sub-mechanism is needed as a part of the intermediate mechanism. Because of space limitations this is not discussed in this paper. The readers are invited to refer to [15, 16] for more details of this issue and for further understanding of how this connectionist mechanism can be used as a basic building block for connectionist logical inference system based on a subset of Horn clause calculus.

ACKNOWLEDGEMENTS

This work was partly supported by the Ministry of Science & Technology, Korea, and the British Council under grant SCOT/KOR/2923/37/A. We would like to thank the referees for their comments which helped improve this paper.

REFERENCES

[1] V. Ajjanagadde and L. Shastri, 'Rules and variables in neural nets', *Neural Computation*, Vol 3, pp 121–134, (1991).

[2] J. Barnden, 'Connectionism, structure-sensitivity, and systematicity: Refining the task requirements', MCCS-92-227, Computing Research Laboratory, New Mexico State university, (1992).

[3] A. Browne, 'Performing structure-sensitive processes on sub-symbolic representations' *The Proceeding of the 12th Neural Networks Conference*, (1994).

[4] D. J. Chalmer, 'Syntactic transformations on distributed representation', *Connections Science*, Vol 2 No 1 & 2, pp 53 – 62, (1990).

[5] L. Chrisman, 'Learning recursive distributed representations for holistic computation', *Connection Science*, Vol 3 No 4, pp 345 – 365, (1991).

[6] J. L. Elman, 'Structured representations and connectionist models' *Proceedings of 11th conference of the Cognitive Science Society*, Hillsdale, N J, Lawrence Erlbaum, (1989).

[7] J. A. Fodor and Z. W. Pylyshyn, 'Connectionism and cognitive architecture: a critical analysis' In S. Pinkerr & J. Mehler (Eds), *Connection and Symbols*, Cambridge, Mass., MIT Press, (1988).

[8] J. A. Fodor and B. P. McLaughlin, 'Connectionism and the problem of systematicity: Why Smolensky's solution doesn't work', *Cognition*, Vol 35 No 2, pp 183 – 204, (1990.

[9] T. Goschke and D. Koppelberg, 'The concept of representation and the representation of concepts in connectionist models', In w. Ramsey, S. P. Stich, and D. E. Rumelhart, *Philosophy and Connectionist Theory*, Hillsdale, N.J., Lawrence Erlbaum, pp 129 – 161, (1991).

[10] C. L. Harris and J. L. Elman, 'Representing variable information with simple recurrent networks', *Proceedings of 11th conference of the Cognitive Science Society*, Hillsdale, N J, Lawrence Erlbaum (1989).

[11] S. H. Hölldobler and F. Kurfeß, 'CHCL - A connectionist inference system', In B. Fronhöfer and G. Wrightson (Eds), *Parallelization in Inference Systems*, Lecture Notes in Computer Science, Springer, pp 318-342, (1991).

[12] T. E. Lange and M. G. Dyer, 'High–level inferencing in a connectionist network', *Connection Science*, Vol 1, No 2, (1989).

[13] J. McCarty, 'Epistemological challenges for connectionism', In Commentary to 'Proper treatment of connectionism' by P. Smolensky, *Behavioural and Brain Sciences*, Vol 11, No 1, (1988).

[14] L. Niklasson and N.E. Sharkey, 'Connectionism and the issues of compositionality and systematicity', In R. Trappl (Ed), *Cybernetics and Systems*, Kluwer academic press, Dordrecht, (1992).

[15] N. S. Park, D. Robertson and K. Stenning, 'An extension of the temporal synchrony approach to dynamic variable binding in a connectionist inference system' *Knowledge-Based Systems*, 8(6), (1995).

[16] N. S. Park, D. Robertson and K. Stenning, 'Oscillation of neurons, dynamic bindings, and a symbolic knowledge representation with an embedded inference mechanism' In D. Levine, B. Brown, and T. Shirey (Eds), *Oscillation in Neural Systems*, Erlbaum, in print.

[17] J.B. Pollack, 'Recursive distributed representations, *Artificial Intelligence*, Vol 46, No 1 & 2, pp 77 – 105, (1990).

[18] L. Shastri and V. Ajjanagadde, 'From simple associations to systematic reasoning: A connectionist representation of rules, variables, and dynamic bindings using temporal synchrony', *Behavioral and Brain Sciences*, Vol 16, No 4, (1993).

[19] R. Sun, 'On variable binding in connectionist networks', *Connection Science*, Vol 4, No 2, pp 93–124, (1992).

[20] D. S. Touretzky and G.E. Hinton, 'A distributed connectionist production system', *Cognitive Science*, Vol 12, No 3, (1988).

[21] T. van Gelder, 'Compositionality: A connectionist variation on a classical theme', *Cognitive Science*, Vol 14, No 3, pp 355-384, (1990).