56-63

Constraint Propagation in Massively Parallel Memory:
Toward Massively-Parallel Artificial Intelligence

Hideto Tomabechi*, **Hitoshi Iida**, **Tsuyosi Morimoto**, **Akira Kurematsu**
ATR Interpreting Telephony Research Laboratories†
Seika-cho, Soraku-gun,
Kyoto 619-02 JAPAN

Abstract

We describe a model of natural language processing based on the notion of propagating constraints in a memory network. This model contains a massively-parallel memory-network in which constraint graphs that represent syntactic and other constraints that are associated with the nodes that triggered activations are propagated. The propagated constraint graphs of complement nodes that collide with the constraint graphs postulated by the head nodes are unified to perform constraint applications. This mechanism handles linguistic phenomena problematic to existing memory-based systems (such as case, agreement, binding and control) in a principled manner in effect equivalent to the manner that they are handled in the modern linguistic theories while retaining the advantages of massively parallel memory-based recognition of input.

---

*Visiting Research Scientist from Carnegie Mellon University

†Visiting Research Scientist. Local email address: tomabech%atr-la.atr.co.jp@uunet.UU.NET.

1. Constraint Propagation in the Memory Network

We propose the model of massively-parallel natural language processing based upon the propagation of graph represented constraints. The central notion of the proposed model is the **propagation of constraints** in the semantic memory network. This is in contrast to past models of semantic memory-based natural language recognition in which activations and pointers to sources of activations (i.e., *markers*) are propagated in the memory network. Instead of propagating markings on memory nodes, we explicitly propagate constraints. As a representational scheme for capturing constraints to be propagated, we find directed graphs to be appropriate. It is because their expressivity has been well supported by linguists ([Bresnan and Kaplan, 1982], [Pollard and Sag, 1987], etc.) and the operations on them (i.e., graph unification) have been extensively researched[1]. We call the model of massively-parallel memory extended with the capacity to propagated graphs *Graph-based Constraint Propagation Network (GCPN).* While retaining the semantic (and pragmatic) expressivity of memory-based schemes, the syntactic expressivity of the graph-based constraint propagation scheme is significantly greater than that of traditional memory-based schemes and their extensions. In fact as we mentioned above, a significant number of current linguistic theories are based upon the formulations of constraints based upon feature-structures and unification operations on them (such as FUG [Kay, 1984], HPSG [Pollard and Sag, 1987] and LFG [Bresnan and Kaplan, 1982]), which are captured through directed graph representations and graph-unification operations on them. Below is a sample lexicon from our experimental system called MONA-LISA ([Tomabechi, ms]). We have adopted HPSG ([Pollard and Sag, 1987]) as the basis of providing the linguistic constraints to the system as graphs. Constraints are specified using path equations (similar to the PATR-II notation, [Shieber, *et al*, 1983]).

```
(def-frame *JOHN
  (inherits-from *MALE-PERSON)
  (type :lex-comp)
  (spelling   John)
  (synsem
   (def-path
    (<0 loc cat head maj> == n)
    (<0 loc cat marking> == unmarked)
    (<0 loc cont para index> == [[per 3rd]
                                 [num sng]
                                 [gend masc]])
    (<0 loc cont restr reln> == *JOHN)
    (<0 loc context backgr> == [[reln naming]
                                [name JOHN]])
    (<0 loc context backgr bearer> == <0 loc cont para index>)
    (<0 mem> == <0 loc cont para index iden>))))
```

When this lexical definition is read into the system, the path equations are converted into graphs internally. Below is the sample lexical node definition for the object control verb *persuaded:*

```
(def-frame *PERSUADED
 (inherits-from *PERSUADE-ACTION)
 (type :lex-head)
 (spelling persuaded)
 (synsem
  (def-path
   (<0 loc cat head> == [[maj v]
                         [vform inf]
```

---

[1] Such as [Pereira, 1985], [Karttunen and Kay, 1985], [Karttunen, 1986], [Kasper, 1987], [Wroblewski, 1987], [Godden, 1990], [Kogure, 1990], [Tomabechi, 1991].

```
                         [aux +]
                         [inv -]
                         [prd -]])
  (<0 loc cat marking> == unmarked)
  (<0 loc cat subcat 1> == <1>)
  (<0 loc cat subcat 2> == <2>)
  (<0 loc cat subcat 3> == <3>)
  (<1 loc cat head maj> == n)
  (<1 loc cat head case> == nom)
  (<1 loc cont restr reln> == *person)
  (<0 loc cont agent> == <1 loc cont para index>)
  (<0 loc cont persuadee> == <2 loc cont para index>)
  (<0 loc cont persuadee> == <0 loc cont circumstance agent>) ;;; obj control
  (<2 loc cat head maj> == n)
  (<2 loc cat head case> == acc)
  (<2 loc cont restr reln> == *person)
  (<3 loc cat head maj> == v)
  (<3 loc cat head vform> == inf)
  (<3 loc cat head aux> == +)
  (<3 loc cat subcat 1 loc cat head> == [[maj n]
                                         [case nom]])
  (<3 loc cat subcat 2 loc cat head> == saturated) ;;; must not unify
  (<3 loc cat subcat 3 loc cat head> == saturated) ;;; must not unify
  (<0 loc cont circumstance> == <3 loc cont>)
  (<0 loc cont reln> == *PERSUADE-ACTION)
  (<3 loc cont restr reln> == <3 loc cont reln>)
  (<3 loc cont restr reln> == *action))))
```

Thus the two equations:

$$
\begin{array}{l}
(\langle \text{ 0 loc cont persuadee } \rangle == \langle \text{ 2 loc cont para index } \rangle) \\
(\langle \text{ 0 loc cont persuadee } \rangle == \langle \text{ 0 loc cont circumstance agent } \rangle)
\end{array}
$$

can easily specify the control constraints lexically in the network.

2. The Graph Propagation Recognition Activity

Currently, our graph-based constraint propagation inheritance network has 5 types of nodes: conceptual-class nodes, lexical-head nodes, lexical-complement nodes, memory-instance nodes, and phonological-activity nodes. The conceptual-class nodes are nodes in the high levels of abstraction and are used for discourse and episodic recognition. Lexical-head nodes are nodes that are phonologically invoked with lexical activations and they package the complement nodes. Lexical-complement nodes are the nodes that are lexically invoked and do not have their own complements. Memory-instance nodes are actual instances of lexical-heads and lexical-complements that are specific to the current utterance. Phonological-activity nodes are the nodes that represent phonemic units and are connected to the phonemic recognition modules. The activity of phonological-activity nodes is not discussed in this paper and the input is assumed to be already hypothesized as words (Please refer to [Tomabechi and Tomita, 1989] and [Kitano, 1990] for activities of those nodes).

Below is the central part of our sentential recognition algorithm for word level input.

```
function sentential-recognize (input-stream)
```

```
  create-process (recognize-lexical (input-stream));
  invoke-global-incidents;
  for all NODE in DecayingLayer do
     print-node NODE;

function recognize-lexical (input-stream)
 reset activities in Activation Layer and Decaying Layer
 for word-hypothesis in input-stream do
    create-process (activate-lex-node (word-hypothesis));
 invoke-global-incidents;

function activate-lex-node (word-hypothesis)
 create instance of word-hypothesis
 and make a copy of constraint graph with addition of an 'mem'
 arc pointing to the created instance;
 if the node type is lexical-complement
    then propagate copied (and modified) constraint graph upward;

function invoke-global-incidents ()
 for head-instance in ActivationLayer do
    create-process (grab-subcats (head-instance)) ;

function grab-subcats (head-instance)
 for arcs specified in subcat graph (i.e, <0 loc cat subcat>) do
     if conceptual restriction node exists
                         (i.e, <loc cont rest reln> has value)
     and if that node has received the constraint graph propagation
         then unify the subcat graph with the propagated graph
              if unify succeeds and obliqueness order is met
                 then store result destructively in head-instance;
                      propagate synsem graph upward;
```

Originally, the GCPN is configured hierarchically in terms of conceptual inheritance. Graph propagation occurs only upward in the inheritance hierarchy and never horizontally. Conceptual relations (other than inheritance) between lexical nodes are specified through constraint graphs (as seen in the sample entry in the previous section). For example, Figure 1 is the part of the GCPN that participates in the sentential recognition of the input *John persuaded Mary to give Sandy sushi* which encompasses two control relations (i.e., *persuaded* object controls *Mary* and *to* subject controls *Mary*).

3. Discussion:

The so called 'direct memory access' (or memory-based) paradigm of natural language recognition seems to have established itself as an important approach to natural language. Originated by Quillian ([Quillian, 1969]), this model of natural language processing views natural language understanding as the activity of identifying the input utterance with existing memory structures. While these memory structures have taken various forms depending upon underlying theoretical approaches (such as MOPs of [Riesbeck and Martin, 1985] and Cases in [Martin, 1989]) and upon levels commitments to the massively-parallel processing assumption (such as the weak assumption in [Riesbeck and Martin, 1985] and the strong assumption in [Waltz and Pollack, 1985]), the underlying control structure has remained the same, namely, spreading activation and marker passing. The advantage of this control structure is that since spreading activation takes place directly in

***action** will be activated by the recognition of any subclass of ***action** including ***run-action**, ***sleep-action**, etc.. However, there is no way to ensure that if the ***mtrans**-word is *said*, the set of entities that caused the activation of ***action** contain (be headed by) a finite verb (unless we create a possibly infinite number of concepts for different combinations of syntactic features such as ***finite-form-running-action-taking-nominative-subject**). Thus, it is essentially impossible to capture the grammaticality of *John said Mary runs* and the ungrammaticality of *John said Mary run* (or *John said Mary to run*) within the constraints capturable inside the semantic memory network. By the same token handling methodology is the so called *obligatory control* phenomenon (such as *persuade*) has been a problem in the past massively-parallel models. In our proposed model, none of these phenonema poses any problem due to the expressivity of graph-based constraint formulations. Structured constraints such as *binding* and *control* can be handled through graph-based constraint application (i.e., graph-unification) no different from the way these constraints are handled in the unification-based grammars. Dynamic assigniment of structures is not a problem since graphs can buildup intermediate structures (i.e., constituent structures) during massively-parallel recognition so that constraints such as dynamic assignment of cases based upon particular syntactic constructions can be handled within the graph-based massively-parallel constraint processing framework.

4. Conclusion

The paradigm of natural language recognition based upon the direct recognition in a semantic memory has been appealing from different view points. These included the appeal from the cognitive view-points as well as the phenomenological ones (especially the assumed massive-parallelism). The practical appeal of the paradigm viewed as a parsing architecture has been strong as well due to its strength in handling contextually sensitive inputs. However, one significant weakness of this paradigm has been its lack of capacity to handle syntactic constraints in a manner that is consistent with the rest of the memory structure. Linguistic and psychological communities have long accepted the significance of syntactic constraints as playing an important role in many types of linguistic phenomena. On the other hand, with few exceptions, syntactic constraints have been either ignored or handled in an *ad hoc* manner (such as requiring a triggering of arbitrary prestored daemons that handle particular constraints). In our model, syntactic phenomena problematic to existing memory-based models such as *case*, *agreement*, *control*, *binding*, and *long-distance dependency* can be handled in a generalized manner most straightforwardly as formulated by the modern linguistic theories based upon feature structures and unification. Because no separate (and essentially serial) control structure is required for handling syntax, uniformity of processing is maintained coherent to the underlying massively-parallel processing hypothesis of the memory-based architectures so that advantage of such processing as demonstrated by [Waltz, 1990] will not be sacrificed. This means the GCPN scheme can be adopted by (or the GCPN can integrate) the existing spreading activation and marker passing models such as [Hendler, 1989] and [Norvig, 1989] to provide these models with strong syntactic processing capability without sacrificing the massively-parallel nature of their processing (by not relying on a separate serial module for syntactic parsing). With the expressivity of our model in capturing the syntactic constraints as postulated in the modern linguistic theories that is integrated to a spreading activation memory-based natural language recognition in a principled manner, our model seems viable as either an independent paradigm of natural language recognition or as a scheme to enhance the existing models of massively-parallel natural language processing, inference and planning.

Implementation

The architecture assumes massively-parallel processing with a mixture of varying levels of granularity of parallelism from fine to medium grain. Also, since the constraint graphs are allowed to point to any location in the memory-network, the hardware needs to support the parallel processes to access any location in a shared memory with least overhead. Currently no massively-parallel hardware designs (such as [Thinking Machines Corp., 1989], [Lee and Moldovan, 1990], [Higuchi, *et al*, 1991]) seem to support this kind of massively-parallel processing directly (– these machines assume processing granularity to be strictly fine grain[3] and memory is normally fully distributed). We have implemented our demo system on a Sequent/Symmetry shared-memory parallel machine with 16 CPUs by simulating massive-parallelism by taking advantage of micro-tasking parallelism using light-weight processes.

References

[Bookman, 1987] Bookman, L.A. "A Microfeature Based Scheme for Modeling Semantics". In *Proceedings of the IJCAI-87*, 1987.

[Bresnan and Kaplan, 1982] Bresnan, J. and R. Kaplan 'Lexical-Functional Grammar: A Formal System for Grammatical Representation'. In J. Bresnan (ed) *The Mental Representation of Grammatical Relations*, MIT Press, 1982.

[Charniak, 1983] Charniak, E. "Passing Markers: A theory of Contextual Influence in Language Comprehension". *Cognitive Science 7*, 1983.

[Charniak and Santos, 1987] Charniak, E and E. Santos "A Connectionist Context-Free Parser Which is not Context-Free, But Then It is Not Really Connectionist Either". In *Proceedings of the CogSci-87.* 1987.

[Charniak, 1986] Charniak, E. "A neat theory of marker passing". In *Proceedings of the AAAI-86.* 1986.

[Fahlman, 1983] Fahlman, S.E. *NETL: A system for representing and using real-world knowledge.* The MIT Press, 1983.

[Gazdar, *et al.*, 1985] Gazdar, G., G. Pullum, and I. Sag *Generalized Phrase Structure Grammar.* Harvard University Press, 1985.

[Godden, 1990] Godden, K. "Lazy Unification" In *Proceedings of ACL-90.* 1990.

[Hahn and Reimer, 1983] Hahn, U. and U. Reimer *World expert parsing: An approach to text parsing with a distributed lexical grammar.* Technical Report, Universitat Konstanz, West Germany, 1983.

[Hendler, 1985] Hendler, J. 'Integrating Marker-Passing and Problem Solving'. In *Proceedings of the CogSci-85.*

[Hendler, 1986] Hendler, J. *Integrating Marker-passing and problem solving: A spreading activation approach to improved choice in planning.* Doctoral Dissertation, Brown University, January, 1986,

[Hendler, 1989] Hendler, J. "Marker-Passing over Microfeatures: Towards a Hybrid Symbolic / Connectionist Model ". In *Cognitive Science*, Vol. 13. Num 1, 1989.

[Higuchi, *et al*, 1991] Higuchi, T., H. Kitano, and K. Handa "IXM2: A Parallel Associative Processor". In *Proceedings of AAAI-91*, 1991.

[Hirst and Charniak, 1982] Hirst, G. and E. Charniak "Word Sense and Slot Disambiguation". In *Proceedings of AAAI-82*, 1982.

[Hirst, 1984] Hirst, G. "A Semantic Process for Syntactic Disambiguation". In *Proceedings of AAAI-84*, 1984.

[Hirst, 1987] Hirst, G. *Semantic interpretation and the resolution of ambiguity*, Cambridge University Press, 1987.

[Karttunen, 1986] Karttunen, L. "D-PATR: A Development Environment for Unification-Based Grammars". In *Proceedings of COLING-86.* 1986.

[Karttunen and Kay, 1985] Karttunen, L. and Kay, M. "Structure Sharing with Binary Trees". In *Proceedings of ACL-85.* 1985.

---

[3] Except perhaps for the transputer support of IXM2

[Kasper, 1987] Kasper, R. "A Unification Method for Disjunctive Feature Descriptions". In *Proceedings of ACL-87.* 1987.

[Kay, 1984] Kay, M. "Functional Unification Grammar: A Formalism for Machine Translation." In *Proceedings of COLING'84*, 1984.

[Kitano, 1990] Kitano, H. *Phi*DmDialog *A Speech-to-Speech Dialogue Translation System* Machine Translation 5 (1990), 301-338. Kluwer Academic Pub., 1990.

[Kitano and Higuchi, 1991a] Kitano, H. and T. Higuchi "Massively Parallel Memory-based Parsing". In *Proceedings of IJCAI91*, 1991.

[Kitano, *et al*, 1991] Kitano, H., D. Moldovan, I. Um, and S. Cha "High Performance Natural Language Processing on Semantic Network Array Processor". In *Proceedings of IJCAI91*, 1991.

[Kogure, 1990] Kogure, K. "Strategic Lazy Incremental Copy Graph Unification". In *Proceedings of COLING-90*, 1990.

[Lee and Moldovan, 1990] Lee, W. and D. Moldovan "The Design of a Marker Passing Architecture for Knowledge Processing". In *Proceedings of AAAI-90*, 1990.

[Martin, 1989] Martin, C. "Case-based Parsing" in Riesbeck, C. and R. Schank eds., *Inside Case-based Reasoning*, Lawrence Erlbaum Associates.

[Norvig, 1987] Norvig, P. "Inference in Text Understanding". In *Proceedings of the AAAI-87*, 1987.

[Norvig, 1989] Norvig, P. "Marker Passing as a Weak Method for Text Inferencing". In *Cognitive Science*, Vol. 13, Num. 4, 1989.

[Pollard and Sag, 1987] Pollard, C. and I. Sag *Information-based Syntax and Semantics.* Vol 1, CSLI, 1987.

[Pollard and Sag, ms] Pollard, C. and I. Sag *Information-based Syntax and Semantics.* Vol 2, CSLI, 1987.

[Pereira, 1985] Pereira, F. "A Structure-Sharing Representation for Unification-Based Grammar Formalisms". In *Proceedings of ACL-85.* 1985.

[Quillian, 1966] Quillian, M.R. *Semantic Memory*, Doctoral Dissertation, Carnegie Institute of Technology (Carnegie Mellon University), 1966.

[Quillian, 1969] Quillian, M.R. *The teachable language comprehender.* BBN Scientific Report 10, 1969.

[Riesbeck and Martin, 1985] Riesbeck, C. and C. Martin *Direct Memory Access Parsing.* Yale University Report 35, 1985.

[Shieber, *et al*, 1983] Shieber, S., H. Uszkoreit, J. Robinson, and M. Tyson *The Formalism and Implementation of PATR-II.* In Research on Interactive Acquisition and Use of Knowledge. Artificial Intelligence Center, SRI International, 1983

[Small and Reiger, 1982] Small, S. and C. Reiger "Parsing and comprehending with word experts (a theory and its realization)". In *Strategies for natural language processing.* Eds. Lenhert G. and Ringle M. Lawrence Erlbaum, 1982.

[Thinking Machines Corp., 1989] Thinking Machines Corp. *Model CM-2 Technical Summary*, Technical Report TR89-1, 1989.

[Tomabechi, 1987] Tomabechi, H. "Direct Memory Access Translation". In *Proceedings of the IJCAI87*, 1987.

[Tomabechi, 1991] "Quasi-Destructive Graph Unification". In *Proceedings of ACL-91*, 1991.

[Tomabechi, ms] Tomabechi, H. *MONA-LISA: Multimodal Ontological Neural Architecture for Linguistic Interactions and Scalable Adaptations*, forthcoming as Technical Report, ATR Interpreting Telephony Research Laboratories, 1991.

[Tomabechi and Tomita, 1989] Tomabechi, H. and M. Tomita "The Direct Memory Access Paradigm and its Applications to Natural Language Processing". In *Computers and Artificial Intelligence*, Vol 8, 1989, No5, 443-478.

[Waltz, 1990] Waltz, D. L. "Massively-Parallel AI" In *Proceedings of AAAI-90.*

[Waltz and Pollack, 1984] Waltz, D.L. and J.B. Pollack 'Phenomenologically plausible parsing'. In *Proceedings of the AAAI-84*, 1984.

[Waltz and Pollack, 1985] Waltz, D. L. and J.B. Pollack (1985) 'Massively Parallel Parsing: A Strongly Interactive Model of Natural Language Interpretation.' *Cognitive Science* 9(1).

[Wroblewski, 1987] Wroblewski, D."Nondestructive Graph Unification" In *Proceedings of AAAI87.* 1987.