MONA-LISA: Multimodal Ontological Neural Architecture for Linguistic Interactions and Scalable Adaptations $\cong$ Toward Massively-Parallel N.L.P.

HIDETO TOMABECHI

Carnegie Mellon University
and
ATR Interpreting Telephony Research Laboratories

Motivation: Existing Models of Massively-Parallel AI/NLP seems inadequtate for solving Natural Language Problems

More Concretely: Existing models assume spreading activation and marker-passing on semantic memory – Is this enough?

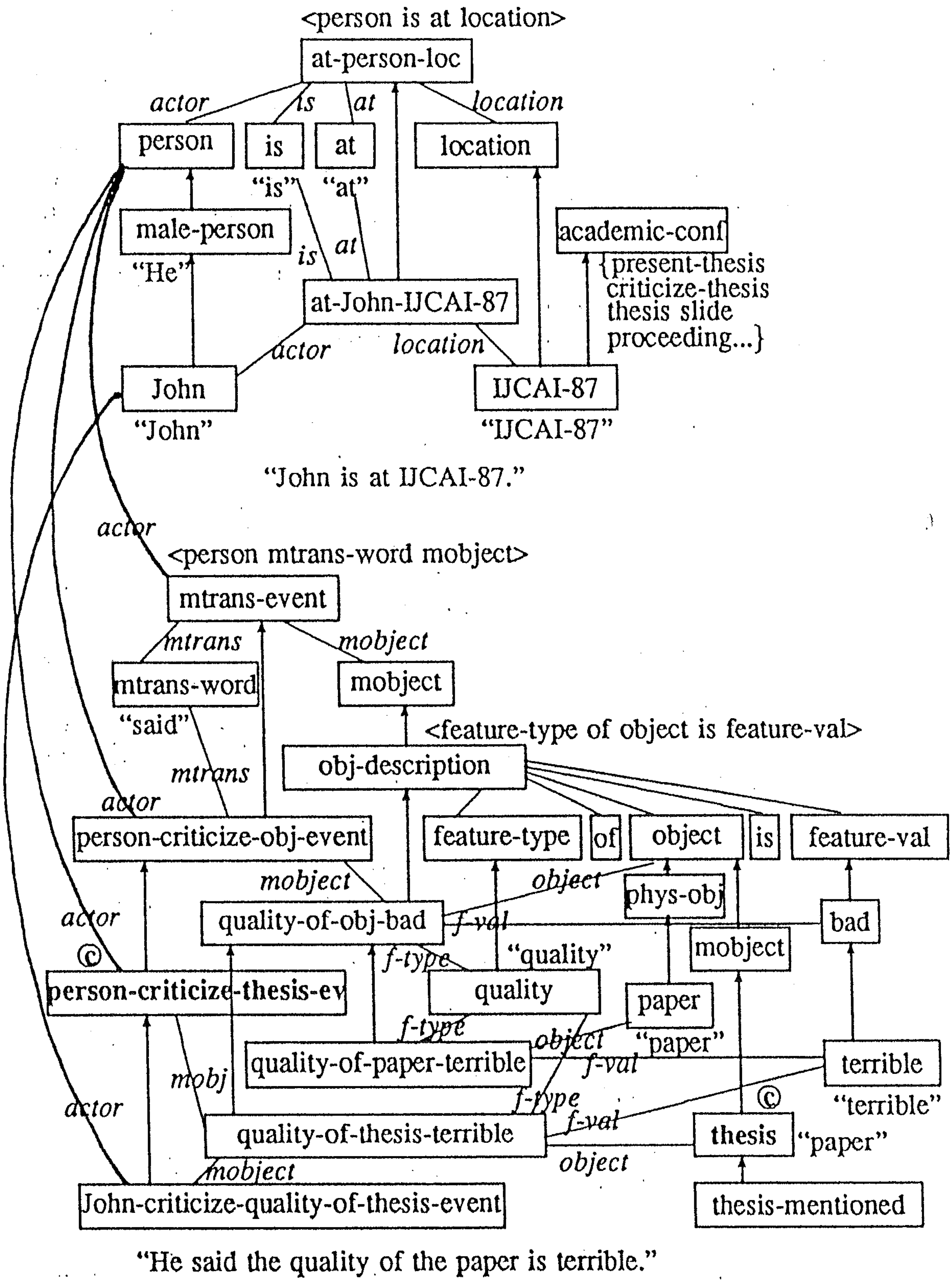
More Practically: Is the limitation of the current models theoretical or is it simply due to the availability of adequate machines?

More Philosophically: Is there a philosophical stand that allows a possible answer?

A Brief Look at Our Past Attempts – A Typical Massively-Parallel NLP Model

- Direct Memory Access Paradigm (DMAP, Riesbeck&Martin 1985).
- Following the tradition of Quillian 1966, spreading activation of nodes in semantic memory is used to recognize the input.
- Developed as a model of MT in DMTRANS (Tomabechi 1987).
- Extended as a speech-to-speech translation system in ΦDMTRANS (Tomabechi 1988) and recently into a dialog translation system in DMDIALOG (Kitano 1990).
- Also extension in a different direction (propagating more information) in HMCP (Tomabechi&Levin 1989).
- Typically, spreading activation marker-passing is used to 1) predict the next element of the phrasal templates; 2) activate nodes connected via relational links (normally via isa links).

A Typical Memory-based Massively-Parallel Natural Language Recognition



What is Lacking in These Existing Models?

- By definition, the network as a whole represents the semantic knowledge, and their organization represents the conceptual definition of the nodes in memory which is *apriori* given.

  –Nodes must be a part of semantic memory (conceptual inheritance), nothing else.

  –Arcs must be a relational links between nodes (conceptual relation), nothing else.

  –Any organization of nodes that are not purely semantic cannot be allowed.

- Thus, knowledge with other nature cannot be represented.

  –Syntactic knowledge such as notion of head or dynamic assignment of cases cannot be captured. I.e., these features do not belong to semantic memory, also these features are not decided *a priori*, instead they are decided at the time of utterance.

  –Pragmatic knowledge such as point-of-view, topic, focus, etc. cannot be captured. – This kind of knowledge does not constitute semantic definitions instead these are parametric features specific to particular utterance environments.

- **Further more**, the spreading activation marker-passing architecture assumes strictly local interactions.

  Constraints are specified as links and therefore, constraints can be postulated only for the nodes that are immediately connected by links.

  A node cannot look inside other node's templates – i.e., no cross constraint formulation into other templates.

- Finally, knowledge is restricted to symbolic ones.

A Simple Example

- *Sue said that Mary ran.*
- This requires two concept sequences: [*PERSON *MTRANS-word that *ACTION] and [*PERSON *RUN].
- There is no way to ensure that if the *MTRANS-word is *said*, then the set of entities that caused the activation of *ACTION contain (be headed by) a finite verb.
- Thus, concept sequence schemes would not be able to capture the grammaticality of *John said Mary runs* and the ungrammaticality of *John said Mary run* (or *John said Mary to run*).

More Examples

SUBCATEGORIZING FOR HEAD FEATURES OF COMPLEMENTS

(1)

a. I believe John studies at CMU.

b.*I believe John study at CMU.

c.*I believe John studying at CMU.

- The contrast between (1)a, (1)b, and (1)c, is that *believe* is subcategorized for an embedded clause whose head verb takes finite form but not the base form or the present participle form.
- Correct treatment of grammaticality in these examples requires the non-local operation of passing up the head features from *study* to *believe*.

(2) a. He tried to wash himself/*herself.
b. He promised her to wash himself/*herself.
c. She persuaded him to wash himself/*herself.
d. She believed him to be washing himself/*herself.

e. She appealed to him to wash himself/*herself.

- There is no way to specify generalizations about behavior of groups of syntactic constituents.
- There is no way for the main verb to determine anything about the subject of the embedded clause because the template containing the main verb cannot see inside the template corresponding to the controlled clause.

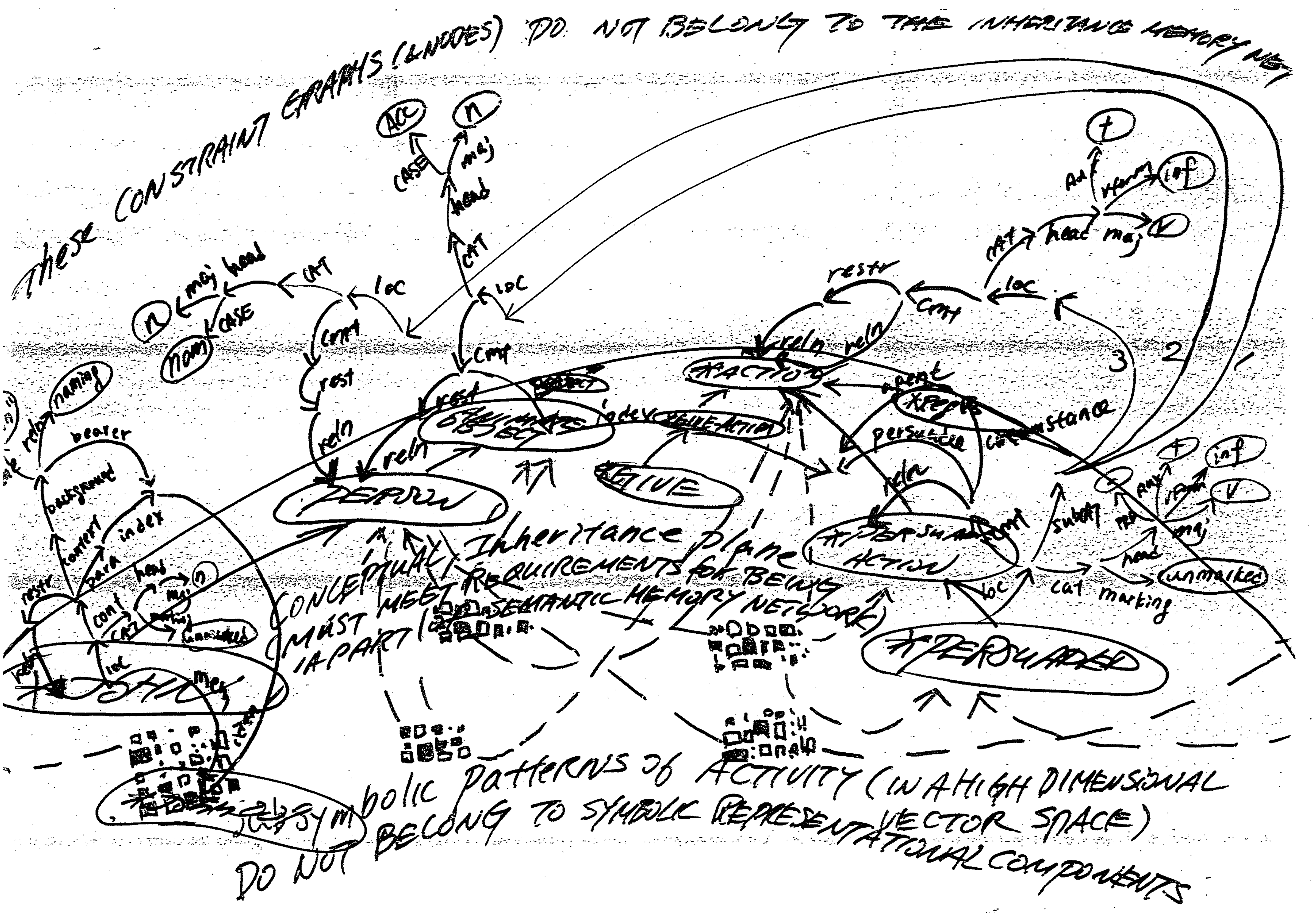
WORD ORDER CONSTRAINTS BASED ON OBLIQUENESS AND THE HEAD POSITION

- In English, the order of a verb's arguments is partly determined by their relative obliqueness; less oblique complements precede more oblique phrasal complements.
- In the examples in (3) the constraint is that "the adverb phrase is the most oblique sister of the post-verbal complements, and hence must follow them all".

(3) a. He looked up the number quickly.
b. He looked the number up quickly.
c.*He looked the number quickly up.
d.*He looked quickly up the number.
e.*He looked quickly the number up.
f.*He looked up quickly the number.

More Knowledge Required

- Thus purely semantic massive-parallelism is inadequate for handling realistic natural language constraints.
- Then, how can we represent the rest?


THESE CONSTRAINT GRAPHS (& NODES) DO NOT BELONG TO THE INHERITANCE MEMORY NET
ACC
n
CASE
maj
head
CAT
loc
cmpt
rest
reln
nom
naming
bearer
background
content
index
restr
cont
CAT
head
maj
marking
unmarked
loc
PERSON
OBJECT
ACTIVE
*ACTION
*PERS
agent
persuadee
circumstance
*PERSUADE-ACTION
*PERSUADED
subject
inf
V
cat
marking
unmarked
Inheritance plane
CONCEPTUAL REQUIREMENTS FOR BEING MUST MEET IN SEMANTIC MEMORY NETWORK
IN PART
SYMbolic patterns of ACTIVITY (IN A HIGH DIMENSIONAL VECTOR SPACE)
DO NOT BELONG TO SYMBOLIC REPRESENTATIONAL COMPONENTS

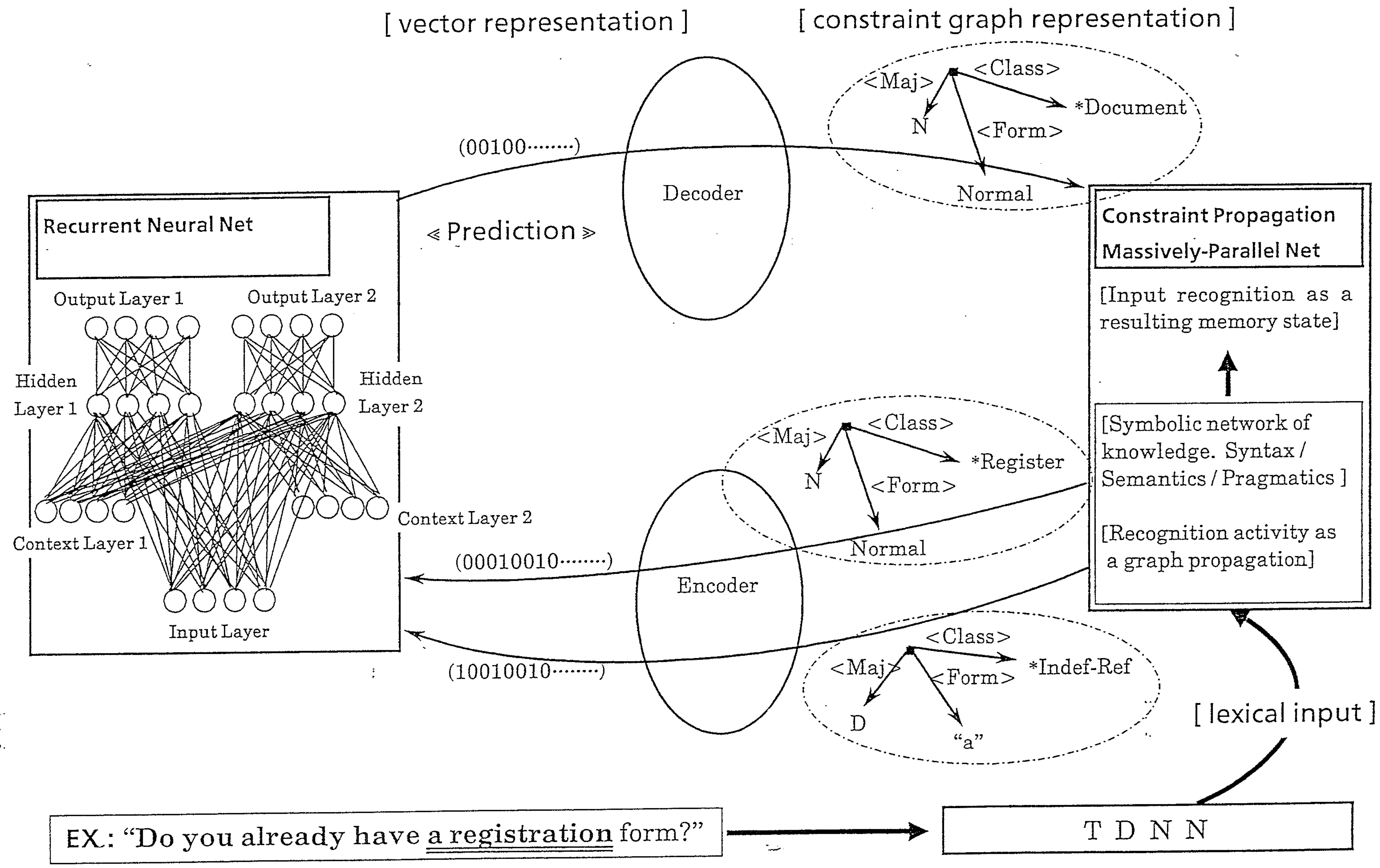
Our Idea for Interaction of Three Kinds of Knowledge

- Graph-based Representation into non-semantic space.
- Graph $\Longleftrightarrow$ Vector, Encoding/Decoding for symbolic/subsymbolic interaction.

MONA-LISA: A Small but a New Step

- Spreading Activation Memory-Network
  – Semantics.
- Graph-based Constraint Propagation
  – Syntax/Pragmatics.
- Symbolic net/Subsymbolic (Neural) Net interaction via graph/vector conversions.

MONA-LISA Architecture



MONA-LISA: Actual Implementation

- A joint ATR/CMU effort.
- Speech recognition using Time-Delay Neural Net and Linear Prediction Neural Net.
- Symbolic massively-parallel constraint processing based upon graph propagation.
- Subsymbolic neural net contextual learning and prediction through recurrent neural network (modified Elman net).
- Graph/Vector encoding/decoding for designated conceptual nodes through an explicit vectorization of inheritance.

Massively-Parallel Symbolic Processing: Graph Propagation Architecture

- The notion of propagating graphs instead of markers.
- Constraints are represented using graphs.
- Expressivity of graphs seems sufficiently strong. (Virtually all existing linguistic and inferential constraints are representable using graphs. – Such as subsumption-ordering feature structures of unification based grammar constraints. Constraints such as postulated in HPSG and LFG can be captured trivially. – Strength of syntactic constraint representation is as strong as these theories.
- Singular and uniform constraint processing based upon propagation of graphs and operations on them (i.e., graph unification).

The methodology

- The graphs are attached to the memory-network nodes.
- Copies of these graphs are propagated.
- The graphs for instances of complement-nodes are unified against the graphs of head nodes.

Sample node definitions

```
(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>))))
```

```
(def-frame *GIVE
 (inherits-from *GIVE-ACTION)
 (type :lex-head)
 (spelling give)
 (synsem
  (def-path
   (<0 loc cat head> == [[maj v]
                         [vform bse]
                         [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>)
   (<0 loc cont reln> == *give-action)
   (<1 loc cat head maj> == n)
   (<1 loc cat head case> == nom)
   (<0 loc cont agent> == <1 loc cont para index>)
   (<1 loc cont restr reln> == *person)
   (<2 loc cat head maj> == n)
   (<2 loc cat head case> == acc)
   (<0 loc cont goal> == <2 loc cont para index>)
   (<2 loc cont restr reln> == *person)
   (<3 loc cat head maj> == n)
   (<3 loc cat head case> == acc)
   (<0 loc cont theme> == <3 loc cont para index>)
   (<3 loc cont restr reln> == *matter))))
```

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

Control handling in *persuaded*

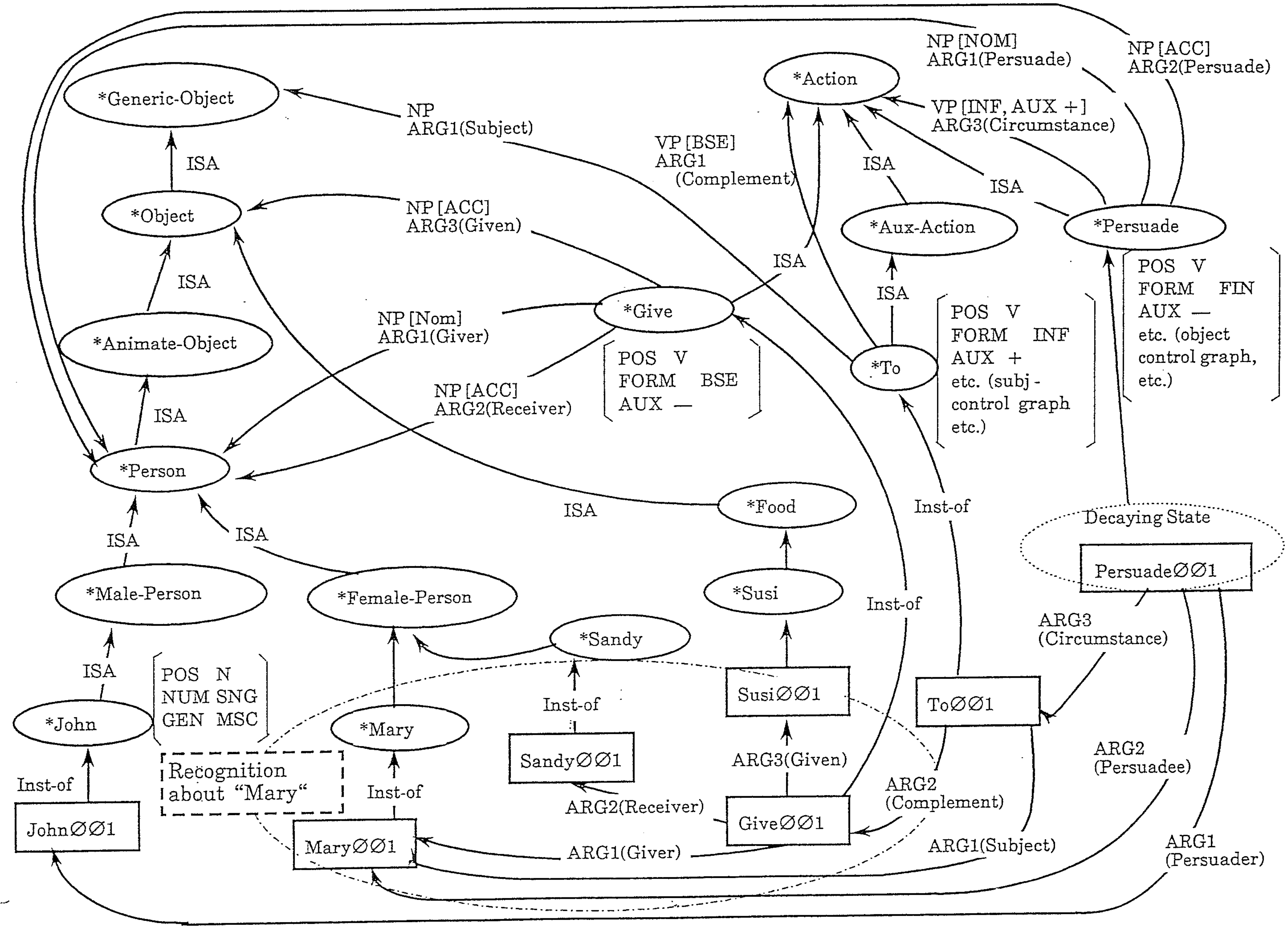
$(\langle$ 0 loc cont persuadee $\rangle ==$ $\langle$ 2 loc cont para index $\rangle)$
$(\langle$ 0 loc cont persuadee $\rangle ==$ $\langle$ 0 loc cont circumstance agent $\rangle)$

Actual graph propagation recognition algorithm

- 5 types of nodes: conceptual-class nodes, lexical-head nodes, lexical-complement nodes, memory-instance nodes, (and phonological-activity nodes.)

```
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;
```

GCPN result state (After the cognition of "John Persuaded M ry to give Sandy Susi")



Sample Output

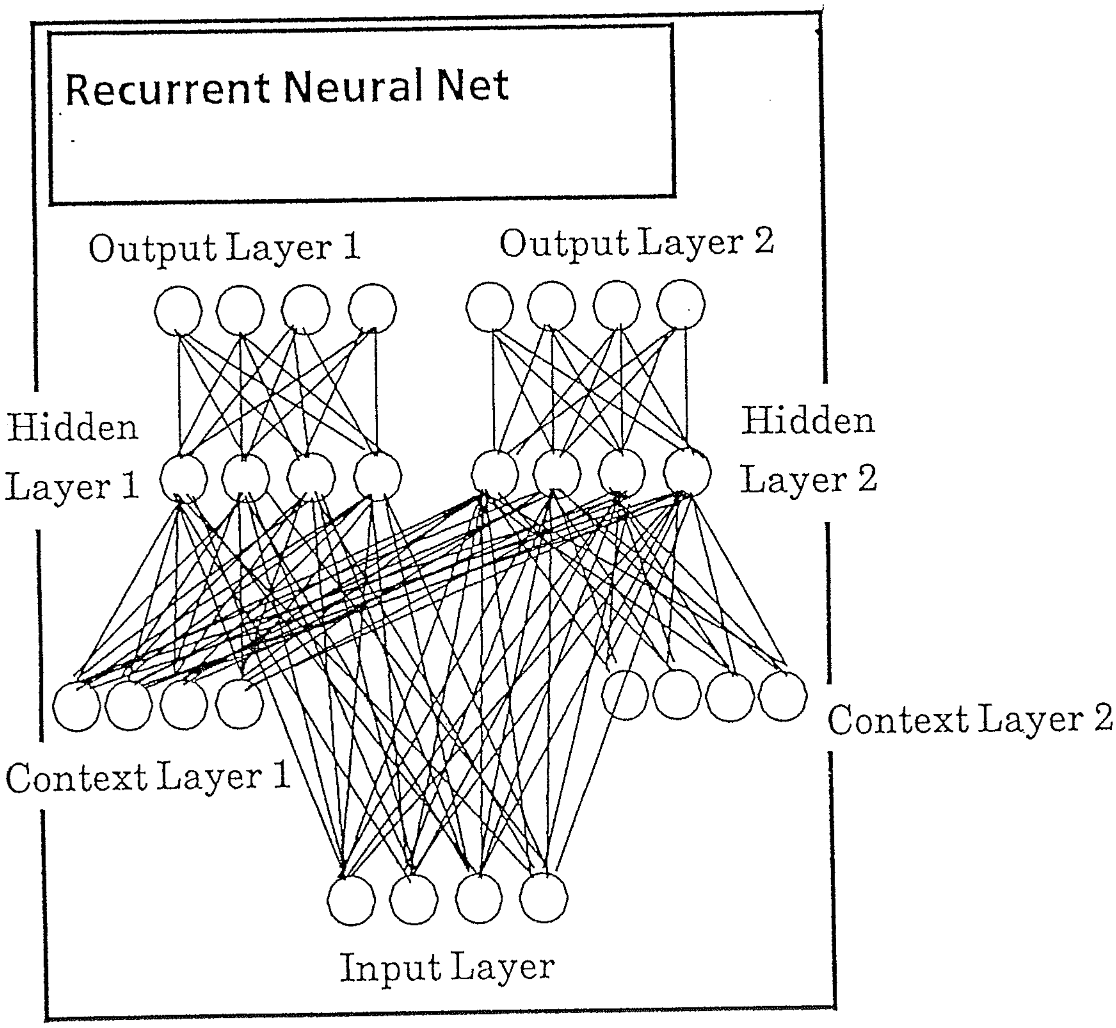
Resulting global memory state with 1 node(s) in the Decaying Layer:

```
(*PERSUADED2356
  (INHERITS-FROM *PERSUADED)
  (TYPE
    (VALUE
      (COMMON :INST-HEAD)))
  (SYNSEM
    (VALUE
      (COMMON
 X01[[0  X02[[MEM  X03 *PERSUADED2356]
            [LOC  X04[[CONT  X05[[RELN  X06 *PERSUADE-ACTION]
                                [CIRCUMSTANCE  X07[[RELN  X08 *GIVE-ACTION]
                                                   [AGENT  X09[[PER  X10 3RD]
                                                               [NUM  X11 SNG]
                                                               [GEND  X12 FEM]
                                                               [IDEN  X13 *MARY2384]]
                                                   [GOAL  X14[[PER  X15 3RD]
                                                              [NUM  X16 SNG]
                                                              [GEND  X17 FEM]
                                                              [IDEN  X18 *SANDY2524]]
                                                   [THEME  X19[[IDEN  X20 *SUSI2607]
                                                               [GEND  X21 NEUT]
                                                               [PER  X22 3RD]]
                                                   [RESTR  X23[[RELN  X08]]]
                                [PERSUADEE  X09]
                                [AGENT  X24[[IDEN  X25 *JOHN2265]
                                            [GEND  X26 MASC]
                                            [NUM  X27 SNG]
                                            [PER  X28 3RD]]]
                      [CAT  X29[[SUBCAT  X30[[NIL ]]
                                [MARKING  X31 UNMARKED]
                                [HEAD  X32[[PRD  X33 -]
                                           [INV  X34 -]
                                           [AUX  X35 +]
                                           [VFORM  X36 INF]
                                           [MAJ  X37 V]]]]]
     [1  X38[[MEM  X25]
            [LOC  X39[[CONTEXT  X40[[BACKGR  X41[[BEARER  X24]
                                                 [NAME  X42 JOHN]
                                                 [RELN  X43 NAMING]]]
                      [CONT  X44[[RESTR  X45[[RELN  X46 *JOHN]]
                                 [PARA  X47[[INDEX  X24]]]
                      [CAT  X48[[MARKING  X49 UNMARKED]
                                [HEAD  X50[[CASE  X51 NOM]
                                           [MAJ  X52 N]]]]]
     [2  X53[[LOC  X54[[CAT  X55[[HEAD  X56[[MAJ  X57 N]
                                           [CASE  X58 ACC]]
                                [MARKING  X59 UNMARKED]]
                      [CONT  X60[[PARA  X61[[INDEX  X09]]
                                 [RESTR  X62[[RELN  X63 *MARY]]]
                      [CONTEXT  X64[[BACKGR  X65[[RELN  X66 NAMING]
                                                 [NAME  X67 MARY]
                                                 [BEARER  X09]]]]
            [MEM  X13]]
     [3  X68[[LOC  X69[[CAT  X70[[HEAD  X71[[MAJ  X72 V]
                                           [VFORM  X73 INF]
                                           [AUX  X74 +]
                                           [INV  X75 -]
                                           [PRD  X76 -]]
                                [MARKING  X77 UNMARKED]
             [SUBCAT  X78[[2  X79[[LOC  X80[[CAT  X81[[HEAD  X82 SATURATED]]]]
 ;; SUBCAT moved for print  [3  X83[[LOC  X84[[CAT  X85[[HEAD  X86 SATURATED]]]]
                           [1  X87[[LOC  X88[[CAT  X89[[HEAD  X90[[CASE  X91 NOM]
                                                                  [MAJ  X92 N]]]
                                             [CONT  X93[[PARA  X94[[INDEX  X09]]
                                                        [RESTR  X95[[RELN  X96 *MATTER]]]]]]
                      [CONT  X07]]
            [MEM  X97 *TO2468]]))))
  (S-TIME
    (VALUE
      (COMMON 1)))
  (W-TIME
    (VALUE
      (COMMON 2))))
NIL
```

Subsymbolic Connectionist Neural Network

- Recurrent Neural Network based on Elman's recurrent net.
- Time-sensitive learning captured in the hidden layer.
- Instead of decoding hidden layer learning, we explicitly encode and decode at input layer and output layer. – Hidden layer info is hard to get.

Recurrent Network based on Elman network



Connecting Subsymbolism with Symbolism

- Head-features and inheritance hierarchy explicitly vectorized.
- Encoding and decoding between vectors and graphs.
- Some nodes (one to two levels higher in abstraction than lexical ones) are designated as subsymbolic interaction nodes.

Syntactic Head-Features are Vectorized

1st	2nd	3rd	4th	5th	
Major	form	case/aux-inv-pred	persn	number	(gender is in hier)
N	THERE	NOM	FIRST	SINGULAR	
	IT	ACC	SECOND	PLURAL	
	NORM		THIRD		
V	FIN	AUX+	FIRST	SINGULAR	
	BSE		SECOND	PLURAL	
	BSP	INV+	THIRD		
	PRP				
	PAS				
	INF				
	GER				
A		PRD+			
P	OF	PRD+			
	AT				
	IN				
	TO				
	FROM				
	FOR				
	OTHER				
D	THE				
	A				
	AN				
	MY				
	YOUR				
	THEIR				
	ANY				
	OTHER				
ADV					

Inheritance Hierarchy is Vectorized

```
1st-level  2nd       3rd    4th    5th

object         physical-object animate-object person      male-persn
                                       female-per.
                        animal
                        plant
                        other-animate-obj

                   inanimate-object natural-substance
                         artificial-subst. document
                                                              book
                                                              computer
          food
                         other-inanimate-
                                                       object

             mental-object   theory&rule
                   abstract-tool language     lisp
                                       prolog
                                       English
                                       Japanese
                   abstract-material title
                   intellectual- thesis
                                        product
                   other-mental-
                                        object

          social-object institution  corporation  public corp.
                                       private corp.
                         educational- university
                             institution
                   service
                   cost-of-service fee


phenomenon   physical-phenom. action      atrans      give
                                       pay
                         ptrans      travel
                                       send
                         mtrans      read
                                       ask
                                       greet
                         include      attend
                                       register
                                                            apply
                         other-action    assist
                                       record
        persuade
try
                    event       atrans-event
                         ptrans-event
                             mtrans-event
                         include-event attendance
                                       registration
                         other-event
```

```
                  other-physical-
                     phenomenon
        mental-phenomen. understanding
                  inference
                  forgetting
                  wanting
             physiological-
                     phenom.
             social-phenomen. conference  academic-conf. AI-conf.
             other-phenomenon
                 power&energy
                 system&custom politeness   asking-favor
                                     thanking

attribute    attribute-name feature-type gender-type
                           truth-
                          condition
                                          referential-type
        relation        spatial-rel. address
                 temporal-rel.
                 social-rel.  ownership
                 part-of
                 modal&identity  has-ability
                           availability
                           identity
                           necessity
                                           possibility
            attribute-value feature-value gender-value male
                                     female
                           conditional- affirmation
                                  value    negation
                           referential     definite-reference
                                   -value  indefinite-reference
        other-attribute name      personal-name first-name
                                     middle-name
                                     last-name
                                     nick-name
                           location-name state-name
                                       city-name
                                       street-name
                  sentential   punctuation  end-punctuation
                       -sign                   phrasal-punctuation
                                                      ques-punctuation

force      strong-force
          weak-force

other-category  sense&feeling  feeling&reaction
                  other sense&feeling
          time       time-point
                     time-duration
                     time-attribute
                     other-time
          space&topography location    functional-loc office
                                        home
                                        hotel
          measurement  numeral
                     standard
                     unit      monetary-unit dollars
                                        yen
```

```
naive-measurement ambiguous-size
other-measurement
```

Decoding is left to right, stopping at ambiguous vector units. – Some generalization capablity.

Sample Output

```
   Semantics : _*WANTING
 Forward propagating...
 Prediction 1:
  #(0.0036569473 0.011257127 0.0 0.9957039 7.182812e-4 1.2562759e-6 0.07089869 7.592556e-4 0.090459324 0.99684155 5
.7461303e-5 0.051744964 1.4593257e-5 0.0044889473 0.74692893 1.7384449e-5 7.477739e-6 1.921835e-5 1.1713059e-5 2.50979
84e-6 4.811422e-4 0.0027652069 0.20375258 7.680587e-6 7.533815e-6 8.5842276e-4 0.3230395 2.0946872e-6 0.003097785 0.0
0.016042646 2.3709372e-4 8.5003716e-6 0.0 0.9205876 8.909068e-4 0.020482019 0.39825487 5.5661595e-6 6.3519615e-6 0.046
615478 4.6769926e-5 2.5038603e-4 5.9475285e-6 6.3412505e-7)
 Prediction 2:
  #(0.0010182429 0.98796636 6.304267e-6 0.0 1.5070632e-4 0.0063269758 0.5667316 0.24030866 0.0031520324 3.083838e-5
 7.142568e-6 1.5045246e-6 2.6129895e-5 0.6057414 0.0 2.4807787e-5 5.198792e-6 3.6296013e-5 4.4432268e-6 2.7001494e-5 3
.207383e-6 0.19303687 0.3953205 1.4395691e-5 9.679367e-5 0.0023452558 0.97700447 0.027542727 1.0726998e-6 0.0 0.572177
65 0.24670222 8.872294e-6 0.0020803844 0.028046757 0.03065481 0.033143613 0.27817398 0.0 2.680418e-5 0.02525578 0.0994
7373 1.7296341e-5 3.9351517e-5 3.29452e-5)
  Unambiguosly priming:
   Syntax:   ((MAJ P) (FORM TO))
   Semantics:  NIL
  Unambiguosly priming:
   Syntax: Form information ambiguous.  ((MAJ V))
   Semantics:  NIL
 Input Word: TO
  #(0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0)
  Decoding..
   Syntax : ((MAJ P) (FORM TO))
   Semantics : NIL
 Forward propagating...
 Prediction 1:
  #(0.006645398 0.9748394 9.4478565e-7 9.1301006e-7 0.28371295 2.8940207e-5 0.22426695 0.9995338 0.011119411 0.0 2.
1463569e-5 9.907014e-4 0.0 1.2076247e-6 0.0 6.948811e-6 1.07124254e-5 7.748146e-6 2.0520969e-6 1.3688916e-6 0.0 0.9963
652 0.018030683 6.231766e-6 1.2965614e-5 0.9939433 1.411859e-5 0.0017316178 1.6144633e-4 0.0 0.9951787 1.8423343e-5 4.
3839252e-6 0.0 0.0 5.2508384e-4 9.569637e-6 0.10085709 0.9995635 9.9291646e-6 3.2253113e-6 0.9994216 0.0 7.451506e-6 0
.0)
 Prediction 2:
  #(0.004090111 1.7823987e-4 4.4269213e-6 0.0 3.1484848e-5 0.35337582 0.6594183 0.011784805 0.23479094 0.5148785 1.
5884766e-4 3.601192e-5 1.431929e-5 0.0043031173 7.7658945e-5 2.623026e-5 2.087293e-5 1.0452316e-5 2.3141551e-5 1.62083
58e-5 0.0 2.3180418e-4 0.99943465 2.3256573e-5 8.442103e-7 1.2691595e-5 0.23725961 0.0027482125 0.58332264 0.0 6.98766
23e-4 0.90281844 3.1144506e-5 0.004172045 0.023300696 0.729631 2.0565414e-4 0.015754476 3.7770277e-7 2.3778936e-5 0.19
770168 0.06376461 0.5641478 1.8976058e-5 7.8343874e-7)
  Unambiguosly priming:
   Syntax:   ((MAJ V) (FORM BSE))
   Semantics:  _*REGISTER
  Unambiguosly priming:
   Syntax: Specific syntactic features not primed enough.  NIL
   Semantics:  NIL
 Input Word: REGISTER
  #(0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0)
  Decoding..
   Syntax : ((MAJ V) (FORM BSE))
   Semantics : _*REGISTER
 Forward propagating...
 Prediction 1:
  #(0.57818097 7.600054e-4 1.3522819e-6 0.9916788 4.6087363e-4 0.0 0.0014736827 0.001126208 0.7482364 0.11970352 0.
0017721569 0.3743199 0.0 3.4496337e-4 0.31569117 4.2916778e-6 7.256921e-6 1.14921965e-5 2.3930332e-5 1.642751e-6 0.009
130579 0.002030561 0.79940087 1.2208627e-5 0.0 5.2752566e-4 0.58468163 0.0 6.5372556e-4 0.0 1.0391285e-4 1.8891719e-4
3.72056e-6 0.0 0.99861133 2.636469e-6 0.004446028 0.99014574 0.0 4.0053162e-6 0.0071188933 9.200067e-6 0.0019791394 5.
0615036e-6 0.0)
 Prediction 2:
  #(0.0027282487 0.01660915 0.0 0.0 0.97914356 0.0012541501 0.9139683 0.25412673 3.7964454e-4 0.0 2.6801595e-5 2.95
8473e-5 7.913609e-7 2.9159546e-4 0.0 1.8841556e-5 1.2861614e-5 1.0344899e-5 6.207719e-6 3.1600943e-6 0.0 0.43837935 0.
94093186 4.6536902e-6 0.0 0.007341183 7.853698e-5 0.99903154 4.6717253e-4 0.0 0.9999976 0.0036452315 2.1453068e-5 1.12
11989e-6 0.0 4.3189186e-5 3.4282804e-5 0.9553152 0.028110573 1.0474978e-5 0.0012201315 0.99883443 0.0 7.490793e-6 5.79
72467e-6)
  Unambiguosly priming:
   Syntax: Form information ambiguous.  ((MAJ P))
   Semantics:  NIL
  Unambiguosly priming:
```

Evaluation of Subsymbolic Predictions

- Trained on 12 dialogs of conference registration domain.
- One dialog is one epoch.
- Predictions such as **register* after *to*.
- and also ((MAJ V) (FORM BSE)).
- Some inductive generalization.

An Observation about massively parallel hardware implementation

- It requires 1) Massively parallel processing 2) propagation of graphs 3) access to any location is the shared memory as a part of the graphs.
- No current massively parallel hardware seems to support this (cf. CM-2, IXM2, SNAP).
- Also, full graph-unification support with cycles, convergent arcs, negation and disjunctions seem non-trivial in the current massively parallel hardware.

Current Implementation (as of Nov. 1990)

- Constraint Propagation Network implemented on a parallel CommonLisp using a vectorized frame representation. TDNN in parallel 'C'. Recurrent neural net in both parallel CommonLisp and parallel 'C'. Both serial and parallel versions of Quasi-Destructive graph unification algorithm (Tomabechi, 1991).
- Closely-coupled shared memory parallel hardware – 16 CPU Sequent/Symmetry.
- Simulated massive-parallelism through dynamic creation of light-weight processes.
- Division into three levels of processing:

  Node level: this is the level where nodes receive and fire activations, i.e., the representational level of memory nodes. In the existing models of massively-parallel artificial intelligence, this is assumed to be the level of processing as well.

  Light weight process (lwp) level: this is the level at which massively parallel processing is performed. Any number of *lwps* may be created during processing,

  Processing unit level: this is the level of actual processing hardware. Any number of processors may be configured depending on the hardware architecture. One (or more) processing units may be dedicated to the scheduling of *lwps.*

Future Possiblities

- Other modal channels, e.g. connecting visual neural network.
- What about symbolic learning? Case-based? Explanation-based?
- How about hardware?

MONA-LISA: Philosophical Stand

- Material Representationalist View:
  "To be a person is to be a system of representations."
- More specifically,
  "Speech Act is an action to modify the world to conform to our representations. Speech Recognition is to modify our representation to conform to the world."
- One Distant Goal:
  "Unification of constraints across all of representational components regardless of nature of representations."

Kinds of Representational Questions to be Involved

- **Epistemological dichotomy**

  – Some knowledge is *a priori* given [to an artificial intelligence]. – Assumed in the symbolic knowledge representation schemes such as semantic networks, frames, script, etc..

  –Some knowledge is *a posteriori* acquired. – Especially learning of neural network seems purely *a posteriori*.

- **Compositionality question** How is representational content interpreted? Frege/Montagovian view supports pure compositionality (*meaning of the whole is the function of the meaning of the parts*) . Is that true? Can we take out a part from the whole and call it a representation?

- **Monotonicity question** Pure Compositionality assumes pure monotonicity. Is that true?