# A NEW APPROACH TO PRONOUN DISAMBIGUATION USING REAL WORLD KNOWLEDGE

Dissertation submitted to the Jawaharlal Nehru University
in partial fulfilment of the requirements
for the award of the Degree of
**MASTER OF PHILOSOPHY**

By
**SANJAY KUMAR**



to the
SCHOOL OF COMPUTER AND SYSTEMS SCIENCES
JAWAHARLAL NEHRU UNIVERSITY
NEW DELHI - 110 067

1989

# CONTENTS


| | | PAGE NO. |
|---|---|---|
| | CERTIFICATE | |
| | ACKNOWLEDGEMENT | |
| | ABSTRACT | |
| CHAPTER 1 | INTRODUCTION | 1 |
| CHAPTER 2 | CLASSIFICATION OF PRONOMIAL DISAMBIGUATION | 7 |
| CHAPTER 3 | CONCEPTUAL DESIGN | 23 |
| CHAPTER 4 | SYSTEM DESIGN | 40 |
| CHAPTER 5 | DATA STRUCTURE AND IMPLEMENTATION | 61 |
| CHAPTER 6 | EXPERIMENTAL RESULTS AND DISCUSSION | 68 |
| | REFERENCES | 75 |
| | APPENDIX 1 | 78 |
| | APPENDIX 2 | 84 |
| | APPENDIX 3 | 86 |
| | APPENDIX 4 | 87 |
| | APPENDIX 5 | |
| | APPENDIX 6 | |

## CERTIFICATE

The research work in this dissertation has been carried out at the School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi and has not been submitted so far for a degree in any other university.

(PROF. KARMESHU)
Dean

(DR. P. C. SAXENA)
Supervisor

Sanjay Kumar
(SANJAY KUMAR)
Student

## ACKNOWLEDGEMENTS

I would like to take this opportunity to extend my sincere thanks to Dr. P. C. Saxena for his help and co-operation throughout the course of this work.

I would also like to thank Mr. Fateh Singh, Administrative officer, for the help he provided to we students in all possible ways.

Finally, I would like to thank other members of Faculty and Staff for their co-operation in many ways.

New Delhi

10.07.89

Sanjay Kumar

**(SANJAY KUMAR)**

# ABSTRACT

In this work, a new approach has been described for disambiguation of pronouns making use of real world knowledge.

It consists in using interactive-states and anticipation of actions alongwith implicitly and explicitly invoked situations.

In addition, a deep-structure representation of action has been described. This helps in giving complete meaning of an action and provides the system with limited high-level learning capabilities.

A scheme for formulation of real world relations and their effects an interaction has also been implemented.

Finally its application in mechanical translation of natural language has been discussed.

# CHAPTER 1

This dissertation addresses a problem central to understanding of natural languages by computer. There are several criteria which constitute necessary condition for natural language understanding.

## 1-1. CRITERIA FOR UNDERSTANDING

A computer can be said to understand natural language if:

i) it can identify the theme of a discourse.

ii) it can resolve anophoric references.

iii) it can answer questions related to a given text with enough causality.

iv) it can anticipate the response of an actor or object in a particular situation.

## 1-2 CLASSIFICATION OF PROBLEMS IN NLU

The above criterion for understanding leads to many basic disambiguation problems in natural languages. They can be classified as:

i) Word-sense disambiguations.

ii) Pronomial disambiguations or resolution of Anaphora.

iii) Situation appraisal and disambiguation.

iv) Context Resolution

v) Inference problems.

## 1-3 WHAT IS PRONOUN DISAMBIGUATION

Pronoun can be classified in following structure (See Figure 1-1)

Pronouns disambiguation is defined as mapping of these pronouns to actor's actual name in a given text. It can be classified in the parts:

### 1-3.1 Surface Level Disambiguation

Surface Level Disambiguation is required when actors in the sentence are from different class and they are well identified. Some of the classes are-

i) Singular Inanimate

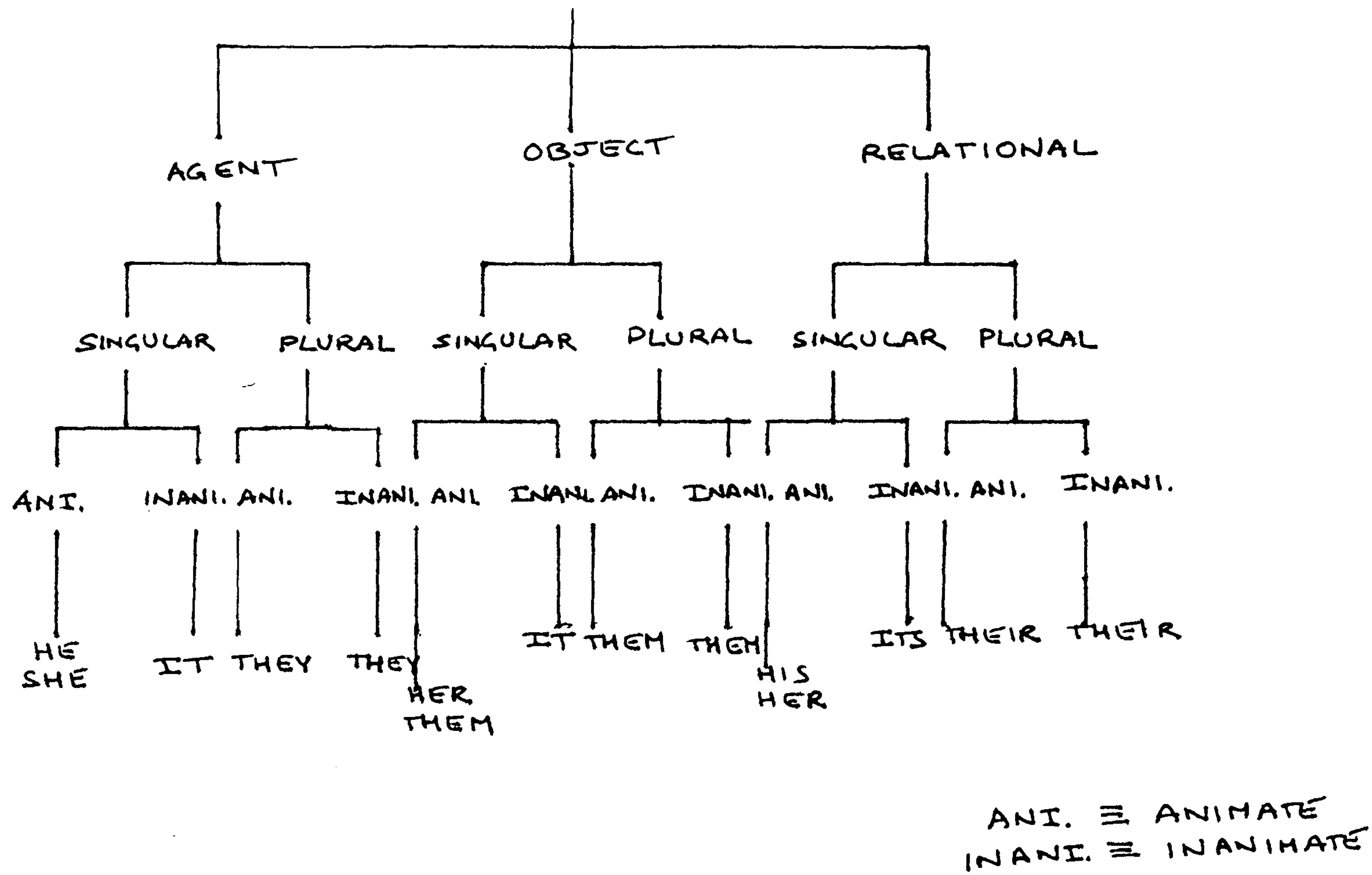
ii) Singular animate male

iii) Singular animate Female, etc.

### 1-3.2 Deep Level Disambiguation

Deep Level Disambiguation is required when

i) Actors being disambiguated belong to the same class.




PRONOUN CLASSIFICATION

FIG. 1.1

ii) The usage of pronouns does not suggest to which class they belong.

## 1-4. SURVEY OF THE PREVIOUS WORK

Lot of work has gone in pronoun disambiguation. The case grammar approach by Fillmore [9] is very elegant. It is able to handle cases where anaphora is disambiguated using meaning of a word or context using the properties of a verb-term.

Still it is not capable of disambiguating the pronouns which require role-playing in a particular situation.

Schank and Abelson [18] have tried to handle role-playing in their script formalism but it is too rigid and in real world such rigid role playing does not occur.

Bullwinkle in PAL [2] has used the approach of invoking frame and it relies on discourse structure. The basic problems of finding focus of attention is very difficult. Besides involving a knowledge-frame when the information is not explicit is also very difficult [5].

Hobbs in his coherence relation [14] tries to classify discourse structure. It looks promising and useful in understanding the theme and sentence level perception. Still it is incomplete and pronouns disambiguation requires more than coherence relation. Moreover, the lack of proper knowledge-base may lead to wrong perception, e.g.

1-1) Mohan scratched his head furiously. The new Shampoo that he used had made it itch unbearably.
here it = head because head is the only object in the context which can itch. Compare it to the sentence:

1-2) Mohan scratched his head furiously. The new Shampoo he had used made it necessary. here it = act 'Scratching of Head'.

While disambiguating pronouns, correlation between previous act and current act also helps in disambiguation, e.g.,

1-3) Ram gave the book to Mohan. He received it happily.
Here he = Mohan because receiver of book is Mohan.

This requires same way of relating the acts 'give' and 'receive' and their characteristics. Schank in his conceptual dependency model [17] has made a fundamental contribution of classifying the actions but that classification has limitations (See (1-3)). In Schank's formalism 'give' and 'receive' both are ATRAns and it is difficult to relate 'give' and 'receive' as antonym. Schank's formalism of act representation has following limitations:

i) Lot of Act-Sense is lost when an act is represented.

ii) Deep representation form has slots like recipent in ATRANS which has inbuilt directionality involved with it. It limits the scope of action-class and correlation between different actions in a particular action-class.

## 1-5. OBJECTIVE OF THIS WORK

This work tries to integrate different concepts described above as well as to remove the following limtitations:

i) Rigidity that is in Schank's script concept.

ii) Problem of invocation of knowledge-frame by implicit factors.

iii) Loss of meaning and lack of intraclass relationships of actions in Schank's formalism.

Some new concepts have also been added for pronoun disambiguation:

i) Use of interactive-states in pronoun disambiguation (See Section 3-1.2)

ii) Formulation of situation for pronoun disambiguation (See Section 3-1.1).

iii) Deferred disambiguations (See Sections 2-1.8, 2-3.4 and 4-5).

iv) Formulation of actions to preserve their act-sense. (See Section 4-7).

v) Formulation of relations (See Sections 3-1.3, 4-6.3 and 4-6.4).

vi) Use of anticipated states and actions (See Sections 3-1.2 and 4-8).

An attempt has been made to provide some high-level learning capability in the design (See sections 3-2 & 4-7).

## CHAPTER 2

## CLASSIFICATION OF PRONOMIAL DISAMBIGUATION

Disambiguation of pronouns can be broadly classified as:

i) Singular personal pronouns (he, she, him, her).

ii) Relational pronouns (his, her, their).

iii) Collective/plural pronouns (they, them, their, those).

iv) Pronouns denoting inanimate-object or denoting pro-action (it).

Disambiguation of these pronouns can be classified into following classes :

### 2-1. DISAMBIGUATION OF SINGULAR PERSONAL PRONOUNS

#### 2-1.1 Syntactic Structure Disambiguation

2-1) Ram told Shyam that he should help others.
Here 'should' decides the disambiguation of he i.e., he = object-who of previous sentence = Shyam

#### 2-1.2 Word/Act Sense Disambiguation

2-2) Ram told Shyam that he would not harm others.
Here, in general, by default he = Ram unless there is a previous context equating he = Shyam.

Compare it to the sentence:

2-3) Ram ordered Shyam that he would not harm others.

Here he = Shyam. Clearly the word/Act sense dictates the disambiguation of pronoun 'He'

### 2-1.3 Disambiguation by Most-Recent-Actor

2-4) Ram hit Shyam. Mohan was playing, he hit Ram.

Here clearly he = Mohan and not Shyam.

### 2-1.4 Focus-Changing by Certain Focus-Switch Words

2-5) Ram hit Shyam. Mohan was playing. He too hit him.

here he = Mohan, him = Shyam

2-6) While in the Sentence:

Ram hit Shyam. Mohan was playing. He hit him too.

Clearly the presence of focus switches like (too, also) changes the focus and it is the most-recent-actor who is associated to the focussed pronoun.

### 2-1.5 Causality Oriented Disambiguation

2-1.5.1 Action → State

2-7) Ram was teasing Shyam. He got angry.

Here the action of teasing Shyam gets him in a mental state of anger. Hence he = Shyam.

2-1.5.2 Action→ State and State → Action

2-8) Ram was teasing Shyam. He hit him.

Here there is an anticipatory reasoning.'Teasing of Shyam' gets him in a mental state of anger and angry person can hit. Hence he = Shyam and him = Ram.

2-1.5.3 State → Action

2-9) Ram was angry over Shyam. He hit him.

Here he = Ram and him = Shyam. It is a case of State → Action.

2-1.5.4 Action → Action

2-10) Ram fell down while climbing a steep wall. He broke his arm.

Here act of falling leads to breaking of arm. Hence he = Ram.

**2-1.6 Explicit Antecedent Consequent Pronouns (Backward Reasoning)**

2-11) Ram hit Shyam because he was crying.

Here he = Shyam. Here crying is an after effect of hitting and it is a case of action → state and state → action.

2-12) Ram hit Shyam because he was angry.

Here he = Ram. It is just the reverse of (2-11) and difficult because consequence has been given first and antecedent later. While in the state → action the flow is unidirectional from antecendent to consequent but here the disambiguation (State → action) has to be done in reverse direction.

### 2-1.7 Disambiguation By Structure Analysis of Action

Sometimes deep structure analysis of action is required.

2-13) Ram gave a book to Shyam. He received it with gratitude.

Here the knowledge that 'receive' is antonym of 'give' and object-who of act give = Agent of 'receive' helps in disambiguation. Hence he = Shyam.

### 2-1.8 Inference Based Pronoun Disambiguation

There are certain cases of pronouns where it is not only the anticipation of act or recency but inferential analysis and backtracking is required too for disambiguation.

2-14) Mira was dancing. Sita was reading. Suddenly she started singing. Sita still did not look at her. If simple forward disambiguation is used then 'she' will be equated to Sita (See Section 2-1.3) but here she= Mira and it can be reasoned as follows:

Sita did not look → Sita did not look something she was supposed to pay attention to and only act happening, was singing. So Sita did not pay attention to singing. 'Agent' of singing is different from Sita. So she = Mira (next most recent actor).

This problem is one of the most complicated as it requires:

i) Keeping track of previous sentence and disambiguation.

ii) Inferencing backward

iii) Use of inherent meaning of act look i.e, paying attention.

### 2-1.9 Script/Situation Dependent Disambiguation of Pronoun

In a certain well defined situation different people play different roles. Thus identifying activities of roles and equating to actual names will disambiguate the pronouns.

2-15) Sita invited Mira for dinner. She enjoyed the food.

Here Mira is guest and guest is supposed to enjoy the food. hence she = guest = Mira.

**2-1.10 Disambiguation Using Relational Pronouns**

2-16) Ram went to Mohan's house. He was fascinated by his garden.

Here garden is a part of the hosue. House is owned by Mohan. Hence his = Mohan's, therefore he = Ram.

2-17) Consider another example:

Mohan is a good orator. He$_1$ went to Ram's house. He$_2$ was pleased by his elocution

Here elocution is synonymous to oratory and oratory is possessed by Mohan. hence his = Mohan's therefore he$_2$ = Ram.

**2-1.11 Conjunctive Form Disambiguation**

In the conjunctive sentences, certain conjunctions restrain the focus to antecedent - Consequent form. They:

i) Restrict the focus to actors in the antecedent sentence.

ii) Imply certain relationships between antecedents and consequents, e.g.

2-18) Ram requested Mohan but he did not help him.

Here he = Mohan because deep-structure of act 'request' implies that it is a request to perform certain act. However, the presence of 'but' implies that the actor who is expected to perform that act does not perform the act. But = ______ (Anticipation).

## 2-2. DISAMBIGUATION OF RELATIONAL PRONOUNS (His, Her, Their, its)

### 2-2.1 Disambiguation By ownership Relation

In a particular text, a particular actor owns a real world object then the use of any relational pronoun anywhere, associated with that object will be disambiguated as the owner's name:

2-19) Ram went to Mohan's house. He found His house in a terrible shape.

### 2-2.2 Disambiguation by Relation 'Part-of'

2-20) Ram went to Mohan's hosue. He found his garden fantastic. Here garden is Mohan's because garden is a part of the house and the house is Mohan's.

If ((a owns a) $\wedge \forall z$ (z part of a))

$\Rightarrow \forall z$ (a owns z).

### 2-2.3 Disambiguation by Reference to An Acqurired State

2-21) Ram was killed in a sudden accident. Mohan escaped unhurt though. Sita could not withstand his death. Here was killed = physical state death. Since Ram has acquired his physical state, his = Ram's.

2-22) Another example:
A splinter hit Ram badly. Mohan was bothered about his injury.

### 2-2.4 Disambiguation by Word/Act Sense

2-23) Ram requested Mohan to hold his books. Here still the ownership of books might be undecided but the sense of request expects the agent to ask for some favour from object-who. Hence 'his' by default is equated to Ram.

2-24) Ram asked Mohan to hold his breath

Clearly here, his = Mohan because a person can hold his own breath. Such usage and associations require correct disambiguation of word sense.

### 2-2.5 Disambiguation by Acquired Interactive-Relationship

2-25) Ram has a brother named Shyam. One day he was playing with Mohan. Mohan hit him. His brother got furious.

Here since Ram<brother>Shyam (Ram is interactively attached to Shyam by brother relationship). Hence his brother means that person who has a brother in the current discourse hence his = Ram's and since Ram's brother = Shyam therefore 'his brother' = Shyam.

### 2-2.6 Disambiguation by Deep-Structure

2-26) Ram drives a blue car. One day its axle was broken.

Here its = car because axle is a part of car. Such cases require representation of physical structure of car and identification of recently used object whose deep structure can be searched.

## 2-3. DISAMBIGUATION OF PLURAL PRONOUNS (they, their, them)

### 2-3.1 Problems in Disambiguation of plural pronouns

i) The basic problem with disambiguation of 'they' is determination of criteria on which the actors can be grouped in a collective form. This may lead to identification of actors doing similar act/actors having similar effect/actors present in the same scene (i.e. situation, time and location)/Actors appearing conjunctively together/Actors having gone through some state change.

ii) The second problem of collective pronoun disambiguation is to find out, when an act is done on a group of actors. The same act might have different effect on different actors thus they might be in different interactive-state or they may behave differently.

An actor's state can be defined by a set of 3-tuples (characteristic, Interaction-state with other actor, Mental State)

If actor X performs an action on a group of actors then the effect can be formulated as:

$[(ch\ y_1,\ Is_1\ xy_1\ st_1\ y_1)\ (chy_2,\ Is_1\ xy_2,\ st_1y_2)\ldots]\ Act_x$

$\dashrightarrow\ [(ch\ y_1\ Is_2\ xy_1,\ st_2y_1)(chy_2,\ Is_2xy_2,st_2y_2)\ldots]$

where,

$ch\ y_n$ = characterstic of $n^{th}$ actor.

$Is_1\ x\ y_n$ = Initial interaction-state of n+h actor with respect to x

$st_1y_n$ = Initial mental-state of $n^{th}$ actor.

Act x = Action done by actor x

$Is_2\ x\ y_n$ = Interaction-state of $n^{th}$ actor with respect to x, after the $act_x$

$st_2y_n$ = mental -state of nth actor, after the act x.

though $st_1y_1$ could be same as $st_1y_2$ but they might have different interaction-states or different characteristics thus it may lead to different effects or different actions done by two actors and they might not be suitable for collective reference later.

### 2-3.2 Disambiguation by Conjunctive Usage

2-27) Ram and Shyam teased Mohan. He kicked them. Here Ram and Shyam are together performing a similar act on Mohan. He = Mohan by Action → State and State → action disambiguation. Ram and Shyam have been treated collectively as a single agent. They are also disambiguated by Action → State and State → Action and conjunctive disambiguation.

Normally, when actors are used conjunctively, they can be treated together collectively for disambiguation in preference to other collective formation especially if the sentence is in the form of Antecedent-consequent.

### 2-3.3 Disambiguation by Same Scene

2-28) Sohan was reading. Ram was playing. Sita was dancing. Mohan asked them to go home. Here Sohan, Ram and Sita are in the same scene. A scene be defined by a 3-tuple (same main situation/script, same locaiton, same or almost simultaneous time).

By default they are taken in the same scene hence them = Sohan, Ram and Sita. However, collective reference is quite flexible in disambiguation by same scene. Let us take an example.

2-29) Ram was playing. Mohan was reading. Shyam was singing. Suddenly Shyam asked them to bring some water.

Here though Ram, Mohan and Shyam were on the same scene but Shyam being an agent in next sentence is no more in collective reference of object-who. Hence them = Ram and Mohan.

### 2-3.4 Disambiguation by Same Sub-Situation

2-30) Ram and Shyam were playing. Mohan came to hit them. Ram ran. Mohan chased him. Suddenly they fell in a drain.

Here, him = Ram (most recent actor) and they = Mohan and Ram because act 'chasing' leads to a situation in which Mohan is oppressor and Ram is oppressed. Therefore, the collective reference will refer preferentially to the actors playing role in a particular situation.

### 2-3.5 Disambiguation by Characterstic/Attribute Capability and Discourse Understanding

2-31) Ram was weak in Mathematics. Sohan was weak in English. Mohan was a brilliant student. One day they requested him to teach. Clearly it is a very complicated case for Disambiguation. Request is made to a person to perform some act. That act is to teach. From discourse a brilliant student can teach. Hence him= brilliant student = Mohan. They = Ram and Sohan (most recent actor).

## 2-4. DISAMBIGUATION OF PRONOUNS DENOTING INANIMATE-OBJECT OR DENOTING PROACTION (it)

### 2-4.1 Preferential Focus in Disambiguation of 'it'

If the usage of 'it' suggests that it is an attribute or characterstic defining an object then the most preferential object in focus is always the object as an agent or an object as an object-what on which the effect is taking place. Let us take an example.

2-32) Mohan picked up the toy from the table. It was made of wood.

Here it= toy. Though, toy as well as table can be made of wood but 'toy' being an object-what is a focus-object where as table simply designates a location.

### 2-4.2 Disambiguation by Recency/Focus by Recency

When there are many objects in the scene and the usage of 'it' does not identify any characterstic of a particular object, then focus is decided by recency. e.g.

2-33) Mohan took the medicine. He opened the tap to drink some water. He stared in disbelief at the water coming out tap. It was black. Here it = water and not medicine (by recency). It can not be the tap (See Section 2-4.1).

### 2-4.3 Disambiguation by Characterstic

Sometimes, certain actions are related with certain characteristics. When an object acquires those characterstics then it becomes capable of doing that action preference to other objects in the scene. e.g.

2-34) The chalk was lying on the table. It rolled and fell down.
Here chalk has an attribute 'cylindrical in shape' and cylindrical objects can roll, unless something restricts them. Hence it = chalk and not the table.

### 2-4.4 Disambiguation by Deep-Structure/Association

Sometimes, certain objects are implicity associated with certain other objects. In such a case, use of the first object or presence of first object may mean presence of the other. e.g.

2-35) Mohan picked the bottle and drank it. Here act drink is associated with liquid and bottle is supposed to contain liquid hence drink it = drink liquid in the bottle rather than the bottle itself.





### 2-4.5 Disambiguation by Pro-action

2-4.5.1 Default Disambiguation

This is quite an involved problem. The only simple category under this is:

If there are no objects in focus or if there are no objects which have an attribute such that an action is characteristic to that attribute then it refers to the previous action. e.g.

2-36) Ram worked incessantly. It made him feel good.

### 2-4.5.2 Disambiguation By Action → State Relationships

If there is an action → state relationship between previous sentence and current one and 'it' referential pronoun is used as an agent or object-what. Then it refers to the previous action preferably. e.g.

2-37) Ram broke Sita's flower pot. It made her angry.
Here action as well as the flower pot both are candidates for disambiguation but breaking of an object is positively related to making Sita angry. Hence, it = Breaking of flower pot.

### 2-4.6 Disambiguation In case of Conjunctive Sentential Form

When two sentences are joined by a conjunction and the first sentence has an object and second sentence refers to this object then the referential pronoun 'it' =object in the first sentence. e.g.

2-38) Ram closed the box and pushed it away.
Here it = box.

# CHAPTER 3

# CONCEPTUAL DESIGN

From the previous chapter, it becomes clear that following concepts are used very often in pronomial disambiguation.

i) Situation representation

ii) Association of different situations with objects, actions, time and location or social setting.

iii) State of an actor and its interaction with real-world-object (animate as well as inanimate).

iv) Well defined relations between real-world-objects.

v) Classification of actions/words such that they do not loose their distinct meanings in distinguishing the act sense

vi) Temporal and causality analysis

## 3-1. SITUATION ANALYSIS AND REPRESENTATION

Situation representation and invocation of situation play a major role in pronoun disambiguation. This section defines and then describes how situations are invoked and represented.

### 3-1.1 What is situation

Situation can basically be classified into three catgories:

#### 3-1.1.1 A Well Defined Social Setting or Schank's Script Concept

Situation can be defined as social setting where different actors play different well defined roles such as dinner, restaurant, class room etc. Such situations can be formulated as:

[[Actors], [Roles], [I-Acts], [D-cts], [E-Acts]]

where, Actors = all actors who are playing certain role in situation.

Roles = role played in the situation e.g., in case of dinner, there are two roles host and guest.

I-Acts = Situation indentifying acts which define entry-point to the situation and from there actual names can be equated to the roles in the situation.

D-Acts = Situation dependency acts. Certain acts which are played exculsively by a particular role but they occur in between the situation.

E-Acts = Situation exit acts which tell when a situation is over.

$$[\text{I-Acts}] \cup [\text{E-Acts}] \subset [\text{D-Acts}]$$

$$[\text{I-Acts}] \wedge [\text{E-Acts}] = \emptyset$$

Let us take an example to explain the situation concept:

3-1) Mira invited Sita for dinner.

Here Mira = host, Sita = guest, situation = dinner, invite = I-Act.

### 3-1.1.2 Application of D-Acts In Pronoun-Disambiguation

D-Acts tell which roles are supposed to perform particular acts. Since actual name have already been equated to roles, hence pronouns = Roles = actual names.

D-Acts along with action -> state or state -> action have advantage:

i) It removes rigid structure present in Schank's script concept.

ii) Two D-Acts act as intermediate source and goal and they are back tracked or heuristically approached using action-> state relations to disambguate an ambiguous or deferred pronoun.

### 3-1.2 Situation Due to Interaction and Relation And State of Actors

A situation is also represented by interaction between different actors (their states towards each other), their mental states and their relations to the objects around.

3-1.2.1 State of An Actor

Interactive state of an actor x is defined as a 3-tuple:

(chx, Ixy, Mx)

Where chx = characterstics of an actor. This includes his behavioural characterstics and physical characterstics.

Ixy = interaction state of actor x towards actor/object y. Generally this varies only when the actor involves in an act directly or indirectly with actor y.

Mx = mental state of actor x. It is an event or an action variant but since an actor interacts with many actors or performs other acts, the change of mental state is more frequent than the interaction state.

3-1.2.2 Interaction Between Actors

Every actor interacts with other actor or environment in a more general sense than the restriction of script. An act of an actor y causes change in interactive state (chx, Ixy, Mx) of an actor x (object-who) directly of indirectly by either changing interaction-state Ixy or Mx or both (see section 3-1.2.3).

3-1.2.2.1 Direct Interaction

When an actor y performs an act on the actor x, it changes his mental state, intetaction state or both. Once the state of the actor x is changed, it is followed by act performed by the actor x. The change of interactive states can be formulated as

(3-2) (chx, Ixy, Mx) (Act y..) → (chx, I'xy, M'x)
or
(chx, Ixy, M"x)

(chx, I"xy, Mx)

(3-3) (chx, Ixy, M"x) -→ (chx, Ixy, M"'x)

(3-4) (chx, Ixy, M"x) -→ Actx

(3-5) (chx, I'xy, M'x) -→ Actx

(3-6) (chx, Ixy, Mx) (Acty) -→ Actx

(3-7) (chx, Mx) (Actx) -→ (chx, x, M'x)

Where x → interaction with external world is not required.

(3-8) (chx, x, M'x) -→ (Actx)

(3-9) (chx, x, Mx) (Actx) -→ (Actx)

3.1.2.2.2 Indirect Interaction

It might also happen that Ixy is interaction state of x towards Y and Izx is interaction state of z towards x. If y performs an act on x, then its effect might be transmitted to z. Thus:

3-10) (chx, Ixy, Mx) (Acty) → (chx, I'xy, M'x) in general.

3-11) (chx I'xy, M'x) ∧ (chz, Izx, Mz) → (chz,Izx,M'z) or (chz, I'zx,M'z)

3-12) (chz, Izx, M'z) - (Actz)

3-13) Ram was unhappy with Mohan. Shyam rebuked Mohan. He felt very happy.

Here preferentially he = Ram. However in such cases there is always a clash between recency and indirect interaction. Here he = Ram and not Mohan because rebuking Mohan will not make Mohan happy and by default it is assumed that Shyam has a normal interaction-state with Mohan.

3-1.2.2.3 Structural Model for Interaction

The complete causal structure of interactive-state and action can be represented by interactive states as nodes of the graph and actions as links between two nodes. There are relational links between actors which cause implicit change in interactive state of actors. Sometimes an interactive state might be transitory and it may lead to another interactive state. In such cases link between two interactive states is delay-link (See Figure 3-1).

The graph thus formed will have cycles in it if indefinite anticipation is allowed. Since it is not desirable, an upper limit on anticipation depth is needed.

### 3-1.3 Relations Between Actors and Reltion Between Objects

Relations between actors play a major role in social setting in disambiguation of pronouns. Suppose B and C are mutually related to each other through some relation. Now A acts on B. The effect of action of A on B will be transmitted to C. Let us discuss as example:

3-14) Shyam hit Ram. Mohan was playing. He hit Shyam back.

Here though by recency he = Mohan but if Mohan is brother of Ram then disambiguation of he = Mohan is further confirmed. Besides Shyam hitting Ram may also change interactive state or mental state of Mohan which might effect further scene analysis.

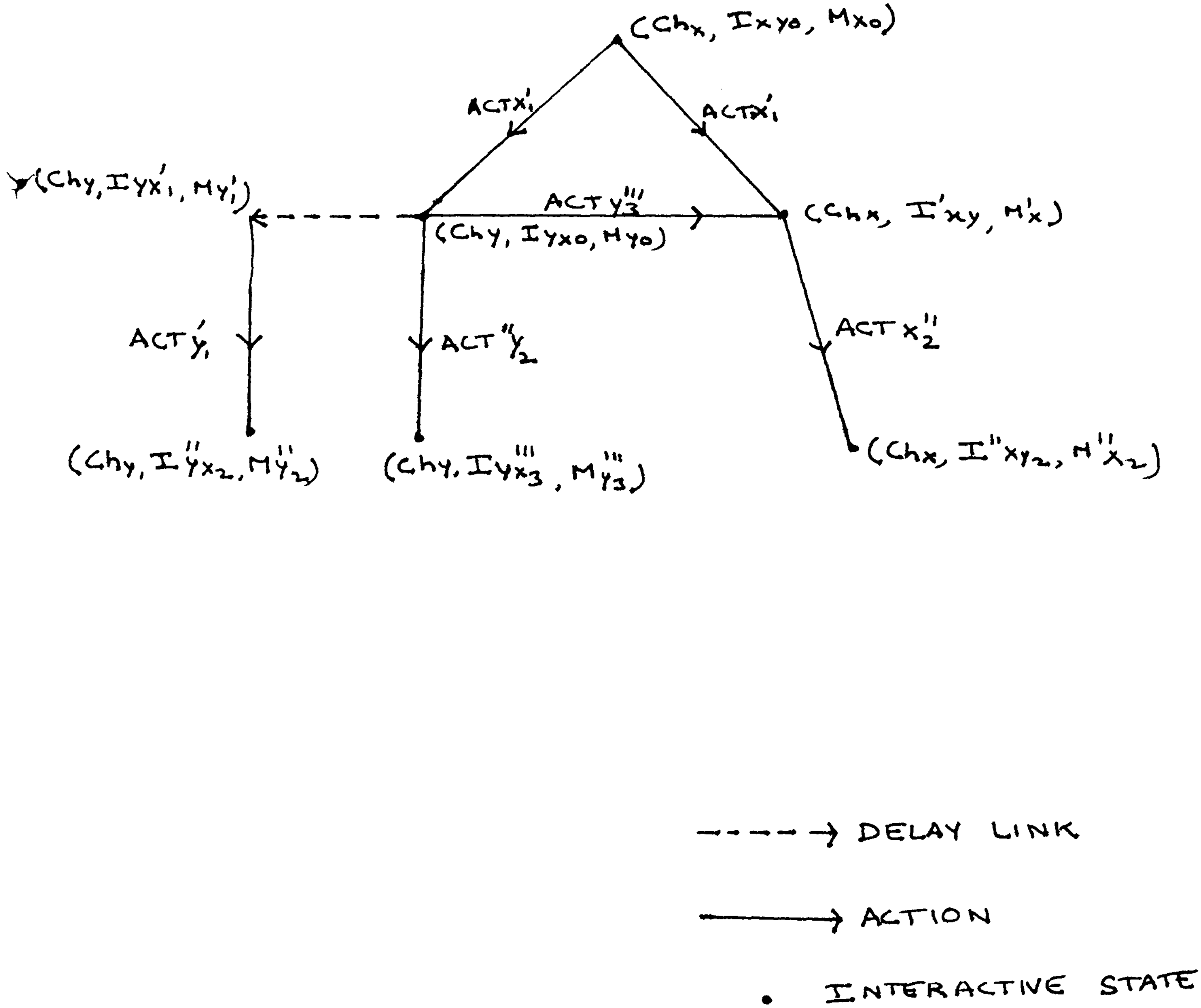

(Chx, Ixyo, Mxo)
ACTX'1
ACTX'1
(Chy, Iyx'1, My'1)
ACT y'''3
(Chy, Iyxo, Myo)
(Chx, I'xy, M'x)
ACT y'1
ACT''y2
ACT x''2
(Chy, I''yx2, M''y2)
(Chy, Iyx'''3, M'''y3)
(Chx, I''xy2, M''x2)
DELAY LINK
ACTION
INTERACTIVE STATE

FIG. 3.1

Similarly in disambiguation of relational pronouns or understanding the deep structure of actions (see Section 4-6) relations between objects or relations between actor and object etc., are of great importance (See Section 2-3). They have been classified as follows:

i) Structural relations

ii) Proximity relations

iii) Interactive relations

iv) Attributional relations

3.1.3.1 Structural Relations

Structural relations are related to the structure of an object, examples are <member-of>, <part-of>, <made-of> <consists-of> etc. [See Appendix 2].

Let us take an emample:

(3-15) X part-of Y $\Rightarrow$ X is part of Y.

These relations do help in deep-structure analysis of an object and thus in disambiguation of the object.

These relations have certain characterstics. If x is structurally related to y then the effect of x and y is mutually transferable though degree to which they are effected might be different.

3-1.3.2 Proximity Relations

Ownership of physical object, possession of mental object, physical nearness, contained-in, placed-on are certain relations which show the proximity of one object to another object [for details see Appendix 2]. These relations help in:

i) Disambiguation of relational pronouns (See Section 2-2).

ii) Analysis of deep-structure of actions (See Section 4-7).

For example, A-Trans deals with ownership and possession. P-Trans deals with physical nearness. Thus analysing the act might lead to a change in relationship between actors or actors and object during discourse analysis.

3-1.3.3 Interactive Relations

By virtue of having certain social relations between actors which are bidirectional in the sense of bondage (like sibling relationships), actors get natural interactive states.

If X <--> Y then any state change in X will cause a state change in Y and vice-versa. This helps in anticipating the interactive states of actors.

### 3-1.3.4 Attributional Relations

These relations are due to association of certain attribute with a certain object. Once an object is related to the attribute, it acquires the characteristics of that attribute. e.g.

3-16) The car is blue.

Here car is related to colour 'blue' through an attributional relation and if blue colour shines then the car also shines after getting the attribute 'blue'. It can be formulated as:

3-17) $(A(x)\ R\ Y) \wedge ChA(x) \Rightarrow chy$

where $chx \Rightarrow$ characteristics of the entity x.

### 3-1.3.5 Rules Guiding Relationships

There are many relations which are transitive, like part-of, consists-of, contained-in. Thus:

3-18) $(X\ R\ Y) \wedge (Y\ R\ Z) \Rightarrow X\ R\ Z$, where R = relation.

There are certain relations which are hierarchial in nature, as compared to other relations i.e.,

3-19) $(X\ R\ Y) \wedge (Y\ R'Z) \Rightarrow X\ R\ Z$

Here, relation 'R' is hierarchial compared to 'R''. There are certain relations whose inverse is well defined- (See Appendix 2) e.g.,

3-20) INV (X R Y) $\Rightarrow$ Y(INV-R) X

Use of these relational rules may give rise to new relations as the discourse progresses in addition to the explicit relations mentioned in the text. This helps in disambigution of pronouns and a better sentence perception. Let us take an example:

3-21) (X owns Y) $\wedge$ (Z part-of Y) $\Rightarrow$ X owns Z.

### 3-1.4 Situation by Association of Location, Time, Act, or Agent Object-Who

A number of situations or sub-situations are associated with an agent, an object-who, an object-what, an instrument, a location or time, collectively or individually. This leads to certain situations and certain acts which are probable for a role in that situation. Let us take an example:

3-22) Mohan was driving a car.

Here, Mohan = Agent = Driver (Driver is a role)
car = Object-who = driven-body.

These are stored in a knowledge-frame, invoked by the act 'drive'. There are certain acts which are associated with 'driver' and certain acts with 'driven-body'.

Similarly, there are situations invoked implicitly by the presence of certain acts. e.g.

3-23) When Mohan entered the house, Ram was sitting on the dinning-table.

'Sitting on the dinning-table', though, does not explicitly indicate anything but there is more liklihood that Actor is already eating or is about to eat. So 'eating' frame will be invoked in anticipation.

Similarly, there could be a list of situations preferentially associated with certain objects. These situations are called in the active mamory, in anticipation, once that object is encountered e.g., spoon is associated with eating. Thus presence of spoon will invoke the situation 'eating' [5].

However, there can be more than one situation invoked by association and the next problem is the selection of a proper situation.

## 3-2. CLASSIFICATION OF ACTIONS

The basic idea behind classification of actions is:

i) To understand the act-sense for pronoun disambiguation (see Section 2-1.2).

ii) To find the synonymous action (or antonymous) by deep structure analysis of action for disambiguation of pronouns (see Section 2-1.7).

iii) Action should have a high level learning capability, e.g., if the system has knowledge of the structure of an act 'give' then simply saying that the act 'receive' is an antonym of the act 'give' should create the structure of the act 'receive'.

### 3-2.1 Structure of a Basic Act-class

A basic act-class can be defined as a 3-tuple of (main functional attributes, Basic criteria, Essential functional attributes) main functional atribute = value holders in case-parsed approach e.g. (Agent, object-who, act, object-what).

Basic criterion = that criterion which differentiates one type of basic act-class from the other and which is unique to a particular act-class.

Essential functional attributes = those value holders in the case-parsed approach which are essential in definition of a basic act-class e.g., in P-trans From-location, To-location, direction form essential attributes (see Section 4-7).

### 3.2.2 Structure of An Action

In can be defined as a 4-tuple of the form (Basic action-class, essential criteria, Shades, functional-attribute).

Where, Basic action-class = one of the action-classes as defined by Schank.

Essential criteria $\in$ [Basic criteria] (See Appendix 4)

Shade = Those criteria which give a particular act a distinct meaning inside that class e.g., in case of the act 'hurl' the basic shade is 'very high force' while in case of the act push it is 'very high force' while in case of the act 'push' it is 'normal force'.

Functional attributes: These are set of value holders which are geneqrally associated with a particular act. They often help in further refinement of action and thus help in sentence perception. e.g. Act 'drop' will have functional attribute as:

Time, From-location/To-location.

Since they give all the information required in action perception. Thus if there is a certain implicit information or missing information in a sentence associated with a particular act, it is filled in the proper slot when that information is avgailable.

## 3-3. TEMPORAL AND CAUSALITY ANALYSIS

Temporal analysis is required in disambiguation by recency (See Section 2-1.3) and 2-4.2), preferential disambiguation (See section 2-4.1) and situation selection.

Similarly in antecent consequent form or action -> state and state -> action analysis for anticipation, causality analysis is required.

### 3-3.1 Temporal Analysis

While disambiguating the pronouns there can be more than one sub-situation or sub task which form a situation. Hence, to find out which situation or sub-situation is currently under consideration, temporal analysis of the situation has to be done (See Section 3-1 also).

Similarly to find out which actor is in focus, focus analysis of actors (as in case of recency) or focus analysis of attributional characterstics has to be done.

Two types of temporal analysis has been attempted in this work.

i) Focus of a Sub Situation

In a certain situation there are many sub-situations which in themselves are hierarchical (tree structure) or parallel disjoint graph (See Figure 3-2). In such a case, an actor may be playing more than one role. Normally if a situation is substitution to the main situation then focus-counts of sub-situation as well as parent are altered simultaneously when-ever sub-situation comes into focus (See Figure 3.2).

ii) Focus-of Actors

This is purely chronological in order. Focus of actors keeps on changing as the discourse moves. The list of most recent actors alongwith their characterstics is maintained for the disambiguation upto a certain time.

### 3-3.2 Causality Analysis

When an action is performed it affects the interactive state (See Section 3-1.2). The actor might perform some

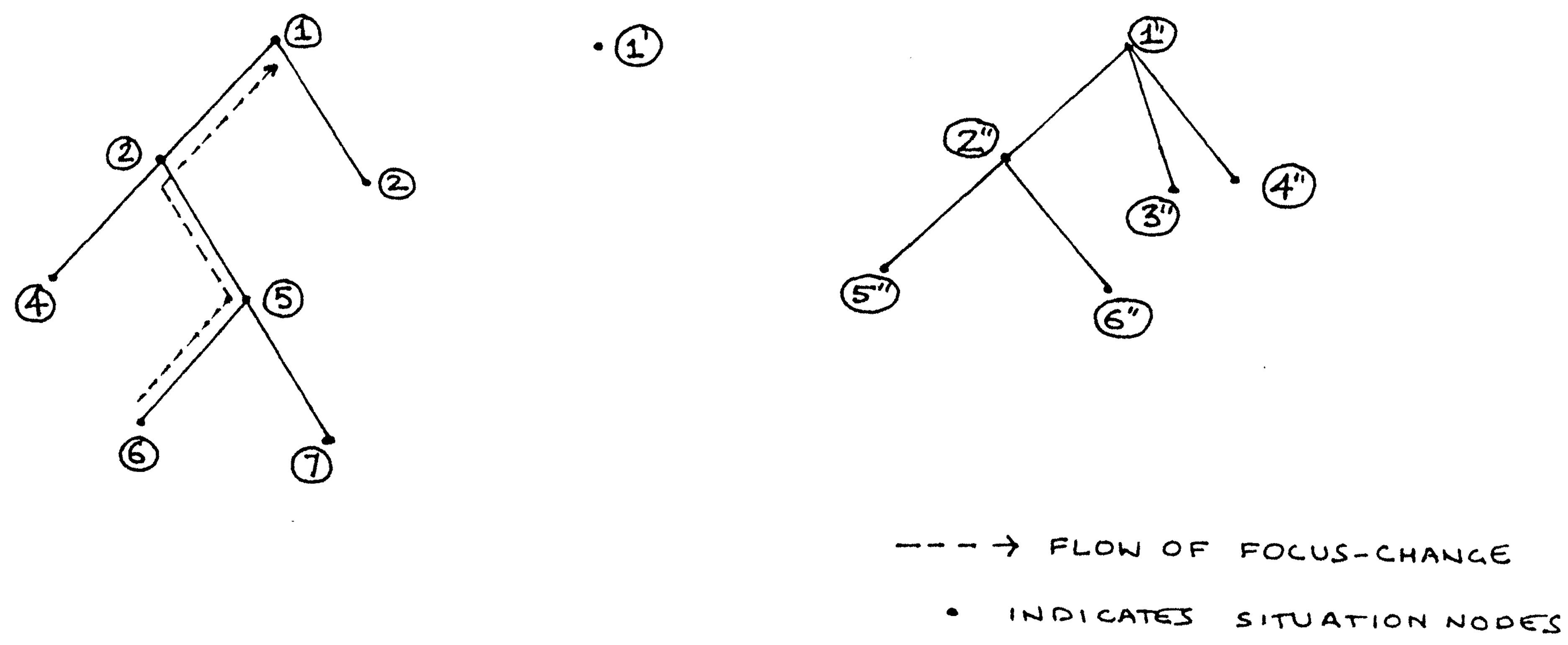

1
2
2
4
5
6
7
1'
1"
2"
3"
4"
5"
6"
---→ FLOW OF FOCUS-CHANGE
• INDICATES SITUATION NODES

FIG 3.2

other act, once there is a change in its interactive-state. This gives rise to cause -> effect relationships. If the state or action can be anticipated by causality analysis, then the pronoun slot of current sentence can be filled in with the actor of the anticipated act provided anticipated action = current action.

# CHAPTER 4

# SYSTEM DESIGN

## 4.1 CRITERIA FOR DESIGN

Given below are the design criteria :

(1) The capability of high level learning of words and acts in terms of other words and acts.

(2) Its knowledge base shoud be separate so that it can be augmented later thus increasing its real world knowledege resulting in better disambiguation power.

(3) The sentences should be represented in such a term that they should be indicative of exact-role of a word in the sentence, e.g.

(4-1) 'Ram ran to the town' can be represented as

((Agent Ram) (Act ran) (where-to town)). The second form is more indicative of the sentence-sense and roles played by different words.

(4) The form in which sentences are represented should be suitable for simple question-answering (like locational, temporal, agent, object indicative questions) (see section 4-4).

## 4.2 DESIGN APPROACH

Following approach has been attempted to implement the above design criteria (See Section 4-1).

i) A case parsed system for sentence representation

ii) Separation of Inference engine from the knowledge base

iii) Representation of appropriate deep-structure for actions (See Section 4-7).

iv) Actor-Frame and event-frame representation alongwith recency-count for temporal and causality analysis.

## 4.3 CASE PARSED APPROACH

Case systems are more logical in structure and they have more information than traditional parser.

Case Systems:

Let us compare a traditional parse with a case-parse; e.g.

4.2) Ram hit Shyam with a pillow.

A traditional parse will be (See Figure 4-1)

While a typical case-parse will (see Figure 4-2).

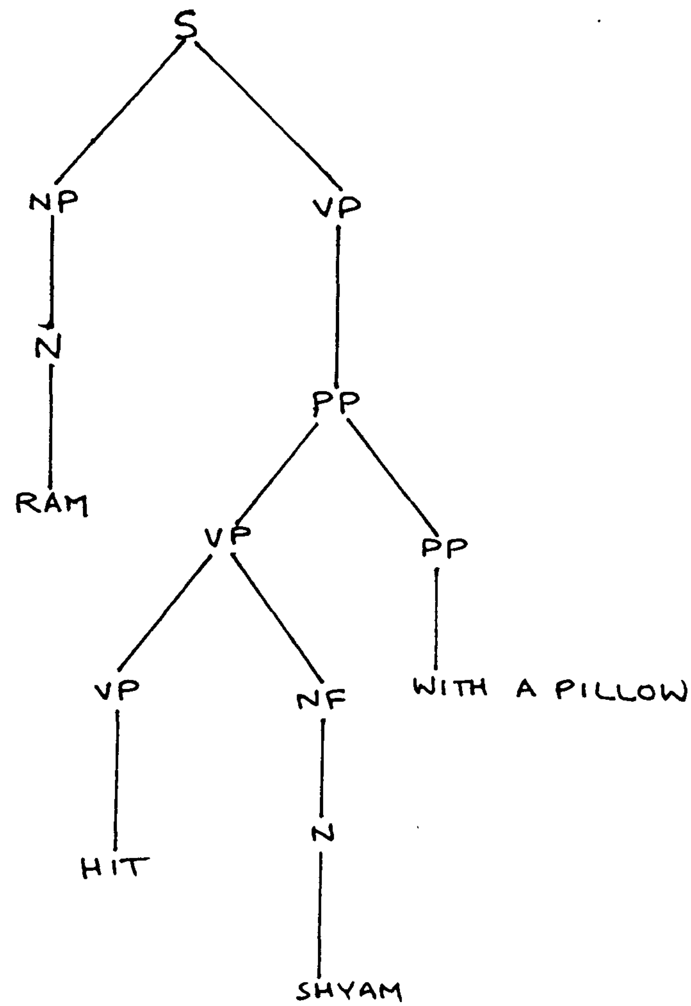

S
NP
VP
N
PP
RAM
VP
PP
VP
NF
WITH A PILLOW
HIT
N
SHYAM

FIG 4.1

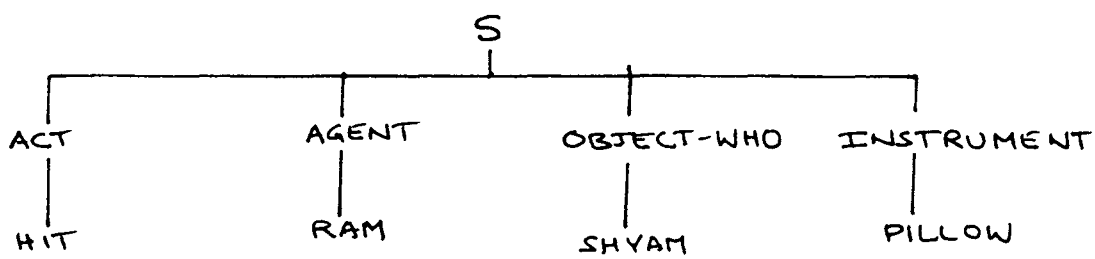

S
ACT
AGENT
OBJECT-WHO
INSTRUMENT
HIT
RAM
SHYAM
PILLOW

FIG 4.2

A question 'who hit Shyam?' is readily answerable in case-parse since exact meaning and purpose of each actor or object is clear. Further, deep-structure analysis is possible with the case parsed approach. Let us take another example; e.g.

4.3) 'Sita shed tears for her broken doll,' can be parsed as : (See Figure 4-3).

In the knowledge base, a relation <contained-in> relates tears with the eyes. Snce the act 'shed' has a functional attribute from-where-loc 'eye' can fill the from-where-loc by deep-structure analysis of tears and Act 'shed'.

### 4.3.1 Design For the Case Parser

Most important criteria for the case parser is:

i) Case-value holders should have enough meaning embedded in them

ii) It should be suited for simple question answering only by checking surface-structure.

Let us take an example :

4.4) Ram went to Mira's house for dinner. Mira was dancing on a chair.

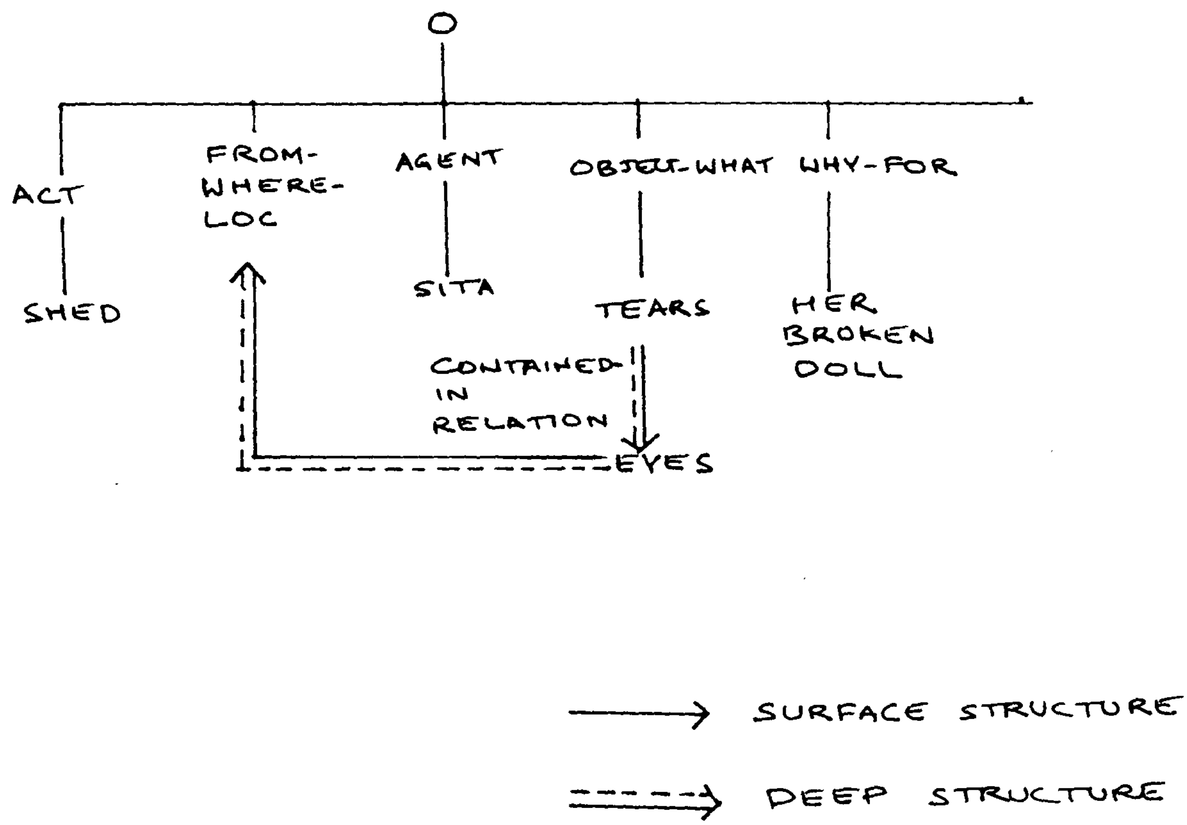

O
ACT
SHED
FROM-
WHERE-
LOC
AGENT
SITA
OBJECT-WHAT
TEARS
WHY-FOR
HER
BROKEN
DOLL
CONTAINED-
IN
RELATION
EYES
SURFACE STRUCTURE
DEEP STRUCTURE

FIG 4.3

Q : where did Ram go ?

A : To Mira's house.

Q : Why did Ram go to Mira's house ?

A : For dinner

Q : Where was she dancing ?

A : On a chair.

Now let us consider a typical case-parsed form :

4-5) ((Act go) ) (Agent Ram) (Loc-where-to Mira's House) (why-for dinner)) ((Agent Mira) (Act dance) (Loc-where-on chair))).

There is 1:1 correspondence between mode of question and value holders' in the 'case parsed form'. Therefore, where, why and when questions can be answered by simple pattern matching. Use of propositions in value-holders

like 'to', 'on', etc., helps in:

i) building back the natural language form for answering the question.

ii) helps in sentence-perception.

### 4.3.2 List of Value holders in the case system

Agent- one who instigates the event

Object-who - An animate object on which the effect of the action takes place.

Object-what - An inanimate object on which the effect of the action takes place.

Act- A particular action in the sentence.

Instrument- Stimulus or the immediate cause of an event.

Loc-where-on - Location on which the action takes place.

Loc-where-in - Location inside which an action takes place.

Loc-where-to - Location where.

Loc-where-from - Location from where the action starts.

Why-for - purpose of the action.

Time-when - Time when an action took place.

Time-from - Time when an action started.

Interaction-state - interaction, state of agent with respect to object

Mental-state

Time-to - Time when an action stopped

Vehicle - Carrier of an object during transmission from loc-where-from to loc-where-to

Manner - Qualifier of a verb.

Situation - Situation name associated with the sentence.

## 4-4 SEPERATION OF INFERENCE ENGINE FROM THE KNOWLEDGE BASE

For a system having learning capability, it is desirable to have the inference engine invariant. Moreover only the knowledge-base should be augmented. This calls for a clear seperation of inference engine from the knowledge-base.

### 4-4.1 Inference Engine

Inference Engine basically consists of an algorithm which disambiguates the pronoun (See Appendix 1) by:

i) Identifying the situation

ii) Identifying the type of acts in the situation (I-Acts and D-Acts etc.).

iii) Searching the anticipated acts for the causal relationships between actions states and actors.

iv) Searching the deep-structure of actions to find relationships between previous action and the curent action or to find out the act-sense.

v) Searching the relational characteristics of each actor and object for the disambiguation of relational pronouns.

vi) Transfer of the effect of an action on a group, to each individual member in case of plural pronouns (See Section 2-3 and Section 4-2-1).

vii) Maintenance of a symbol-table for the maintenance of a dummy name (see section 4-2.1).

viii) Discourse flow-wise maintenance of event frame, keeping recent actors in focus.

ix) Discourse flow-wise maintenance of actor-frame in which causality analysis of past is easy.

x) Finding out new relations using rules governing old relation and newly added relations in the discourse.

xi) Finding out new anticipated actions and interactive states of each actor using production-rules, e.g. (interactive-state)$\wedge$(Action) $\rightarrow$ (Interactive-state) or (Interactive-state) $\rightarrow$ Action.

### 4-4.2 Identifying the Situation

A situation is identified by:

i) Finding the value of case-system value holder'SITN' which represents a social setting (See Section 3-1 and 4-3).

ii) Finding out the situation associated with act, object-what, location, time.

All the index-names (formed by act, object-what, location, time) alongwith associated situation names is placed in a situation-dictionary. The knowledge-base regarding these situations is kept in the disk. As soon as a new situation comes, it is matched in the dictionary list. If the situation-name is present in the dictionary, its knowledge-base is called from the disk.

### 4-4.3 Formation of a New Situation From the Old

A new more restricted situation may be formed sometimes when:

i) Case structure of an act has a case value-holder slot which is filled later in the discourse. This might invoke a new situation altogether. Let us take an example:

4-6) Ram invited Sita for dinner, when She reached the restaurant he ordered for salads.

Here, initially the location of the invitation is not known. Therefore, only the situation 'dinner' is called but when location of the invitation i.e. 'restaurant' appears, the new situation becomes 'dinner in a restaurant'. It is very different from the situation 'Dinner'. For example, host does not play the role of cook in the situation 'dinner in a restaurant' while in normal dinner he/she may do.

ii) The current situation might be a substitution of previous situation and thus restrict it further.

## 4-5. SYMBOL-TABLE

A symbol-table which has one to one correspondence between a system defined dummy name and one of the following cases is created.

i) The collective group which is used as a single entity while disambiguating plural pronouns by action→State and state→action or by recency (see sections 2-3.1 and 3-1.2).

ii) When a particular action is referred not by name but by a typical attribute, e.g.

4-7) Sita was going home. She met an old man. He gave her a chocolate., Here, 'old man' is an actor whose actual name is not known.

iii) In case of deferred disambiguation when a particular pronoun has not been disambiguated. A dummy name is allocated to the undisambiguated pronoun. This 'dummy' name is resolved by using D-Acts or action→ state or state→action causality (see sections 2-3 and 3-1.2) and all the occurrances of dummy-name are replaced by actual name. The advantage is that after the allocation of dummy name, the referential pronoun becomes unique till the time it is mapped to actual name of the actor.

**4-5.1 Design of symbol-table**

Symbol table is a stack of

i) a unique dummy-name which indicates one of the three categories mentioned above.

ii) description of the dummy name as used in the sentence.

iii) Recency-count.

(See section 5-8 for data-structure).

recency-count is implicit as most recent dummy-names are appended to the top of the stack.

## 4-6. DESIGN OF THE KNOWLEDGE-BASE

Knowledge-base consists of following categories:

### 4-6.5.1 Production rules Governing Interactive-states and Action of an Actor

(see Section 3-1.2)

Depending upon an interactive-state of the actor x and action of actor Y, it might lead to new interactive states of x or Y or actions by X. This can be represented by a production-rule, L.H.S. of the rule represents current interactive state of X and action of the actor Y while RHS represents anticipated interactive states or actions.

### 4-6.2 Rules for Relational Representation

A list is associated with each class of object showing its relation with other objects, class of objects or attributes. Each relation is binary. It binds two entities (animate, inanimate, attribute etc.), e.g.,

(4-8) $(X\ R\ Y) = X \xrightarrow{R} Y \Rightarrow$ X is related to Y through relation 'R'. However, these relations have unique direction from X to Y (See Appendix 2 also).

### 4-6.2.1 Representation of Binary Relation

Binary relations can be broadly categorised as:

i) Unidirectional $X \xrightarrow{R} Y$ e.g.

4-9) Wheel ⟨is part-of⟩ car

ii) bidirectional $X \overset{R}{\leftrightarrow} Y$ e.g.

Ram ⟨p-relation⟩ Shyam.

However, for representation, bidirectional relations are broken into unidirectional relations:

$$(4\text{-}10)\quad X \overset{R}{\leftrightarrow} Y = (X \xrightarrow{R} Y) \wedge (Y \xrightarrow{R} X)$$

Now each unidirectional relation is represented as a list (X R Y).

### 4-6.3 Rules for Manipulation of Relational Rules

These rules take two binary relations and form a third binary relation. They have been presented as production rule of the form:

$$4\text{-}11)\quad (R_1 \quad R_2) \rightarrow R_3$$

where, $R_1 \Rightarrow X\ R_1 Y$

$R_2 \Rightarrow Y\ R_2\ Z$

$R_3 \Rightarrow X\ R_3\ Z$

if $R_1 = R_2 = R_3$ then the relational-rule is transitive if $R_1 = R_3$ then relational-rule $R_1$ is hierarchial compared to $R_2$.

### 4-6.4 Frames Describing the Situation

While Sections 4-2.1 and 4-2.2 are control structures of the knowledge-base, sections 4-2.3 and 4-2.4 describe the real world data.

In frame description of a situation I-Acts alongwith the essential attributes or D-Acts alongwith the essential attributes, are kept. The frame is called into memory only when a particular situation is invoked (See Figure 4-4).

## 4-7. DEEP STRUCTURE OF ACTIONS AND REAL WORLD OBJECT

There are two types of deep-structure representations:

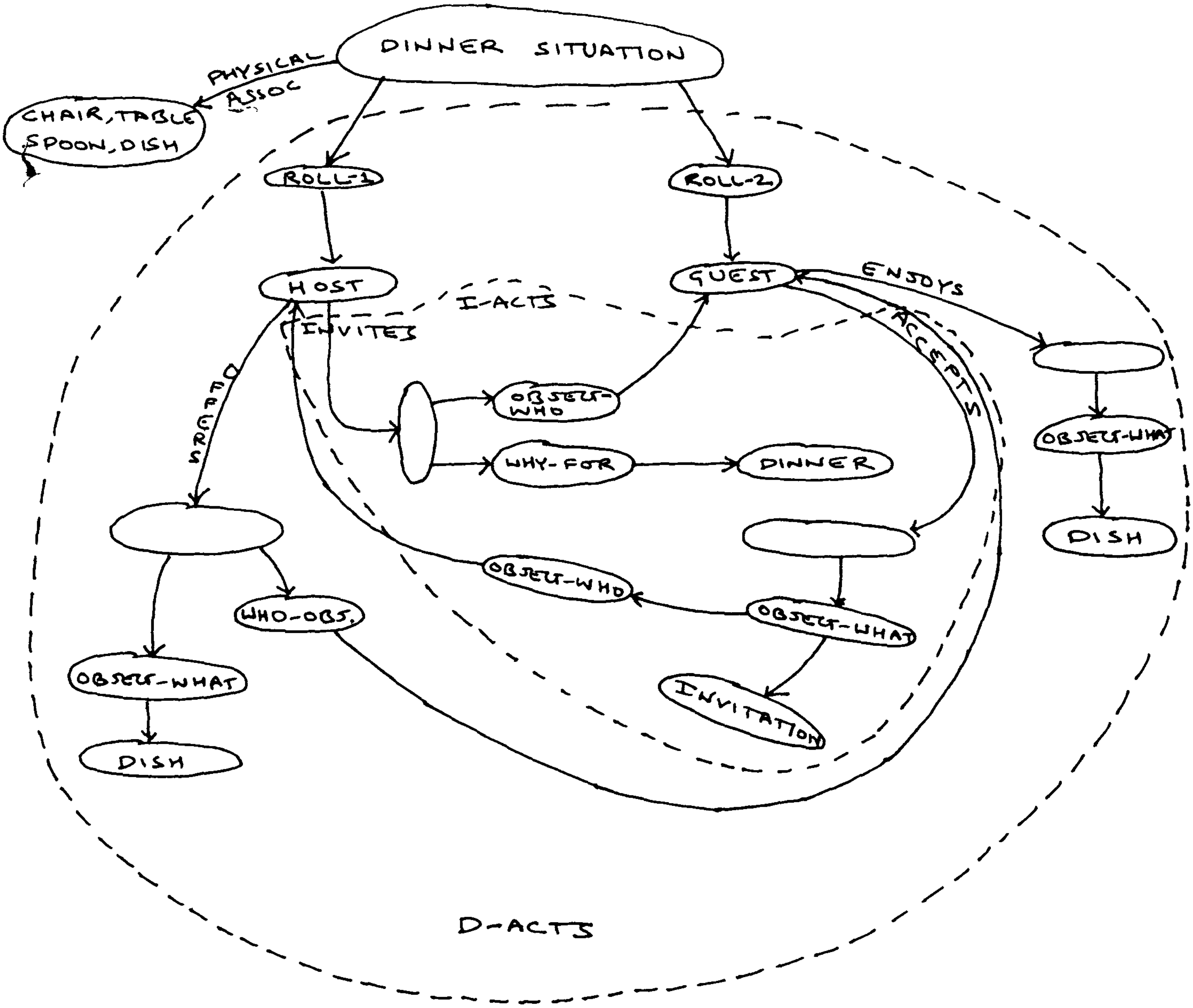
### 4-7.1 Action Deep-Structure Representation

A detailed discussion of deep-structure for two basic classes of actions, that has been implemented, follows here (See Section 3-2 also)

4-7.1.1 Ptrans: The verbs associated with ptrans are move, push, throw, hurl, place, keep, drop, fall, jump, go and walk.

Ptrans can be represented as:

(Main functional attribute, (Location, Nearness, Direction), Essential functional attribute).




SITUATIONAL FRAME

FIG 4.4

Main functional attribute are (Agent, object-who/object-what).

Locaion is loc-where-from → loc-where-to.

Direction ∈ [towards, away, upwards, downwards] and it is preferentially associated with object-who/object-what. A number of directions can be described implicitly depending upon the characterstics of loc-where-from and loc-where-to- e.g.

4-12) Ram pushed Mohan into the river.

loc-where-from --→ loc-where-in --→ river.

Here, direction is downwards because river has characteristics of depth and depth is related to direction downwards.

Shade - In ptrans each action will have its own shades but a general shade is ¢Intensity of force'.

4-7.1.2 Atrans:

The verbs associated with Atrans are give, receive, take, donate, pay, buy etc.

Atrans can be represented as:

(main functional attribute, Basic, criteria, essential functional attribute)

Main functional attribute is (Agent, object-who, Act, object-what)

Basic criteria- relation-transfer governing ownership or possession of physical objects or mental objects (See Appendix 2).

Essential functional attribute - NIL.

Since all actions have a sequential temporal connectedness, anything on left hand side of production-rule or precondition is implicitly before time of action and anything on right hand side or post condition is after time of action.

An action in Atrans can be defined as:

(Atrans, X,Y, Functional attribute)

Where X∈[Atrans relational directions]

Y∈[shades related with Atrans actions].

Let us take an example:

4-13) Action ¢give' can be defined as.

(Atrans, Atrans-rel-1, shade-1, Functional attributes)

Where,

Atrans-rel-1 = (Agent phy-poss object-what) -→

(object-who phy-poss object-what)

or

(Agent men-poss object-what) $\rightarrow$

(object-who men-poss object-what)

Shade-1 = NIL.

Functional attribute = Time, loc-where.

4-7.1.3 Antonym and synonym Detection

There are two ways of detecting whether a particular action is antonym or synonym of previous action.

4-7.1.3.1 Surface Structure

All those actions which are synonymous are put together in one synonymous class. Each member of that class is mapped to a single representative act of that set. Antonym relation exists between these class representative acts

4-7.1.3.2 Deep-Structure Detection of Synonymous and Antonymous Acts

If there are two actions represented by:

Act 1: (x,y,z,w), Act 2 = (x',y',z',w')

Where x or x' = basic act-class

y or y' = essential criteria

z or z' = Shades $\in$ Shades in that class.

w or w' = Functional attributes.

If $(x = x) \wedge (y = y') \wedge (z \neq z')$, that means that only it is difference of shade. In that case two actions are synonymous. If $(x = x')\wedge(y = \text{inverse}\,(y'))\wedge\,(z \neq z')$, then two actions are completely antonymous.

If $(x = x')\wedge(y = \text{inverse}\,(y'))\wedge(z \# z')$, then two actions are partially antonymous.

The same concept helps in learning of action by:

i) Defining an act as synonymous to already existing action and only supplying shades to the definition of predefined act

ii) Defining an act as antonym to already existing action and supplying basic criteria + shade to definition of pre-defined action

iii) Defining an action as a sequence of other actions

### 4-7.2 Representation of Real World Objects

A real world object is represented by;

i) Its relation with other real world objects or class of objects.

ii) its attribute.

iii) a static list showing inherent characterstics of an object-class.

iv) a static-capability list with each object-class showing what all object-class can do.

v) a static-restriction list associated with each class showing what all it can not do.

## 4-8. TEMPORAL AND CAUSALITY ANALYSIS

The basic assertions about temporal and causality analysis is:

i) Assertion 1: Every action or actor which occur in the scene is related with the scene and there is continuity of causality in text.

ii) Asserion 2: If an event E1 is described in the discourse after an event E2 then T1 (E1) > T2 (E2), unless otherwise specified.

These two assertions have led to concept of most recently used actor, situation, characterstic or relation. A focus-count is maintained to show recency. This focus-count keeps on getting decremented after each event until focus-count goes below a minimum value.

### 4-8.1 Anticipated Action

Most recent action in past or MRAP leads rise to a set of anticipatory interactive-states and actions (through causal links). The basic assertion about anticipated act is:

Assertion 3: If one of the anticipated action matches with the current action and there is no conflict of actors-class in two actions then the actors of anticipated action = actors of current action.

Based on this assertion a list of anticipated actions is maintained. These anticipated acts which are not used for a certain time are deleted.

### 4-8.2 Dynamic Characterstics, Capabilities and Restriction Analysis

With each object-class a static characterstic-list is associated. This static characteristics list is copied in the dynamic-list. This dynamic characterstic-list keeps on growing due to two reasons:

i) new characterstics or attribute is added to the instance of that object-class which is participating in the discourse.

ii) when a characteristic is brought out by deep-structure analysis and it has been used recently.

## 4-9. MAINTENANCE OF ACTORS/OBJECTS IN FOCUS

Most recently used actors or objects are likely to be used again. Based on this assertion an actors-name-frame is maintained. It consists of actual name of actors/Dummy names for collective reference and focus-count. This focus-count is renewed to maximum value when an actor/object is referenced. With every action in which these actors/objects are not referenced, focus-count is decremented by one. After a certain minimum value of focus-count it is assumed that actor is not a candidate for disambiguation by recency unless the actor's actual name occurs in the discourse again.

In case of collective pronouns the effect of the current act is transfered to all the individual actors and focus-count of all of them is altered simultaneously.

## 4-10. INPUT AND OUTPUT STRUCTURES

nstead of natural language form the input was accepted at an intermediate stage i.e., case-form output. Let us consider an example:

4-14) Ram went to Sita's house She served him dinner.

Input form for this sentence will be:

(((Agent Ram) (Act go) (loc-where-to Sita's House))
((Agent she) (Act serve) (object-who him) (object-what dinner))).

After the disambiguation 'she' will be replaced by 'SITA' and 'him' will be replaced by 'Ram'. Thus output form will be:

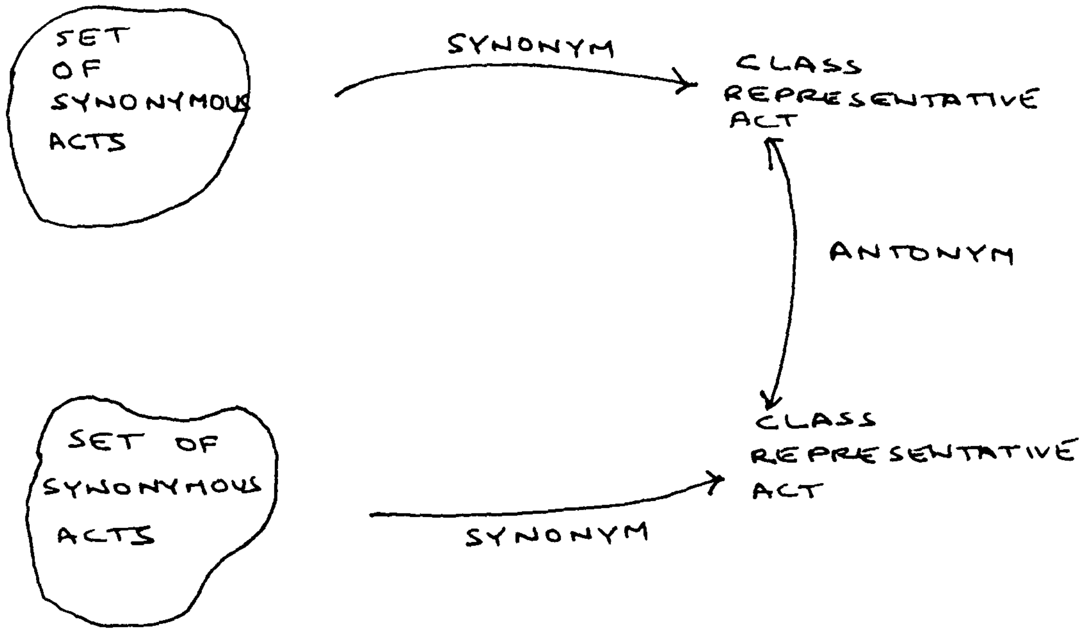

SET OF SYNONYMOUS ACTS
SYNONYM
CLASS REPRESENTATIVE ACT
ANTONYM
SET OF SYNONYMOUS ACTS
SYNONYM
CLASS REPRESENTATIVE ACT

FIG 4.5

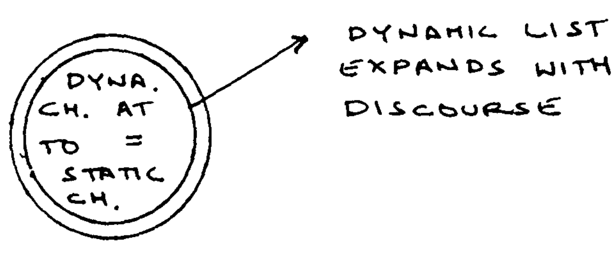

DYNA. CH. AT TO = STATIC CH.
DYNAMIC LIST EXPANDS WITH DISCOURSE

FIG 4.6

( (Agent Ram) (Act go) (Loc-where-to sita's House) ((Agent sita) (Act serve) (object-who Ram) (object-what dinner))).

# CHAPTER 5

# DATA STRUCTURE AND IMPLEMENTATION

This chapter discusses most of the important data structure used in the implementation.

## 5-1 DATA STRUCTURE FOR ACTION → STATE and STATE → ACTION RULES (See Section 3-1.2)

(5-1) ((($Ch_x$-value, interaction-state$_{xy}$-value, mental-state$_x$ - value) (act$_y$))(ch$_x$-value, interaction-state'$_{xy}$-value, mental-state'$_x$ value)) represents.

Interactive state$_x$ $\wedge$ Act$_y$ → Interactive-state'$_x$.

(5-2) (((ch$_x$-value, interaction-state$_{xy}$-value, Mental-State$_x$-value) (Mental-state$_y$-value))(ch$_x$-value, interaction-state'$_x$-value, mental-state'$_x$-value) represents:

Interactive-state$_{xy}$ $\wedge$ Mental-state$_y$ → interactive - state'$_x$

(5-3) (((Ch$_x$-value, interaction-state$_{xz}$ - value, mental-state'$_x$-value) (ch$_x$-value, interaction-state$_{yz}$-value, mental-state$_y$-value) (x p-relation y)) ((Ch$_x$-value, interaction-state'$_{xz}$ - value, mental-state'$_x$-value))

(5-4) ((ch$_x$-value, interaction-state$_{xy}$-value, mental-state'$_x$-value) (act$_y$)) represents:

Interactive-state$_{xy}$ $\rightarrow$ Act$_x$

(5-5) ((ch$_x$-value, x, mental-state$_x$-value) (Act$_x$))

## 5-2. DATA STRUCTURE FOR RULES GOVERNING RELATIONS

(See Section 4-6.3)

(5-6) (Rel-1 Re1-2) Rel-3 for binary operation. It represents :

(X Rel-1 Y) $\wedge$ (Y Rel-2Z) $\rightarrow$ (X Rel-3z)

## 5-3. DATA STRUCTURE FOR INPUT AND OUTPUT

(See Section 4-10)

(5-7) Input sentence - ((Agent value)(Object-who object-what value) (Act value) ...)

(5-8) Output sentence - ((Agent value) (Object-who object-what value)

## 5-4. STRUCTURE FOR DISCOURSE ANALYSIS

(See Section 3-3, 4-8 & Appendix)

(5-9) <u>Anticipation List</u> :

(((SITN situation-name)((Act details list)...))...)

(5-10) Actor-Frame List:

((Actor actor-name) (Act details) ...)

(5-11) Role List-1 :

((Sitn situation name) ((Role-name Actual-name) ...))....)

This data structure helps in mapping role-name to the actual-name, while dis-embiguating the pronouns.

(5-12) Actor-Role List :

((Actor's actual name/dummy name ((Sitn-name ole-name)...))....)

5-13) Actor-Focus-List :

((Actor-name Actor-focus-count)...)

(5-14) Event-Frame List :

((Disambiguated-sentence list)....)

The order of appending disambiguated sentence is last n first out because it is more likely that most recent event be referenced first.

### 5.4.1 DATA Structure for Situation Analysis During the Discourse

(See Sections 3-3, 4-4.2, 4-6.4 & Appendix 1)

5-15) Situation List

((Sitn Focus-Count)...)

5-16) Situation Hierarchial Structure

It is a forest in which each group of situations and substitutions form a tree and there are more than one groups

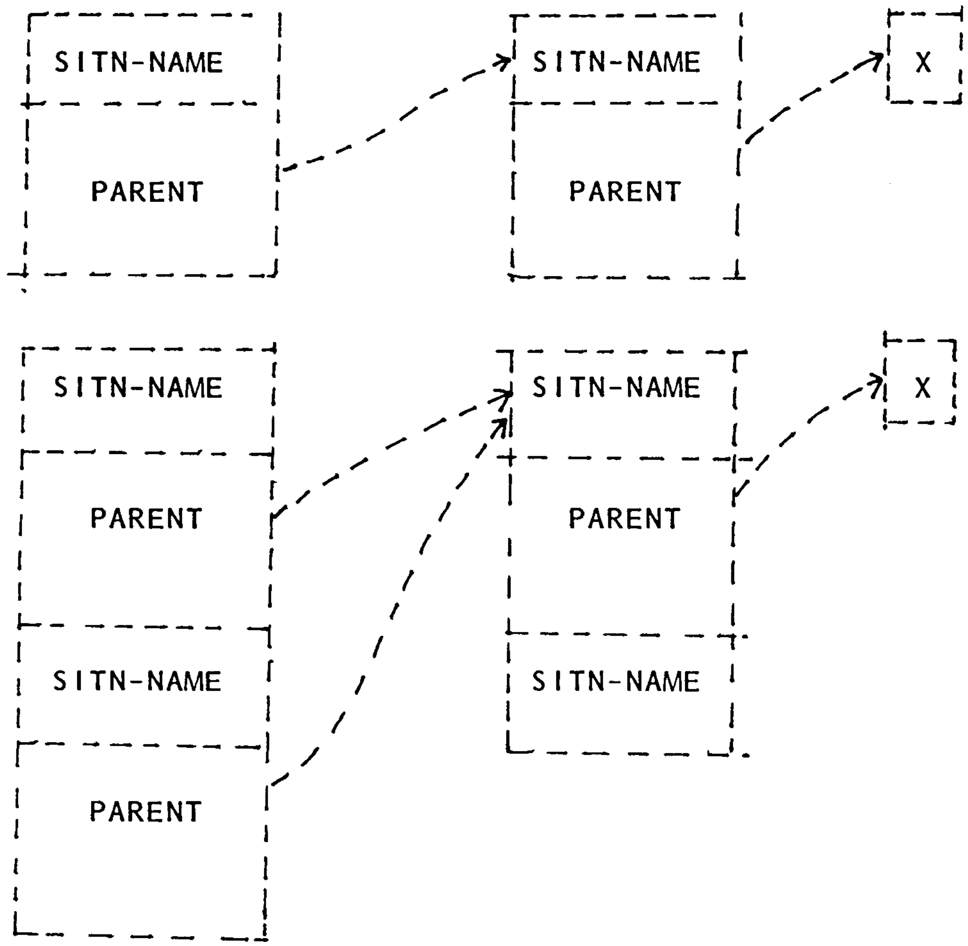



Figure 5-1

5.17) Situational Action Frame

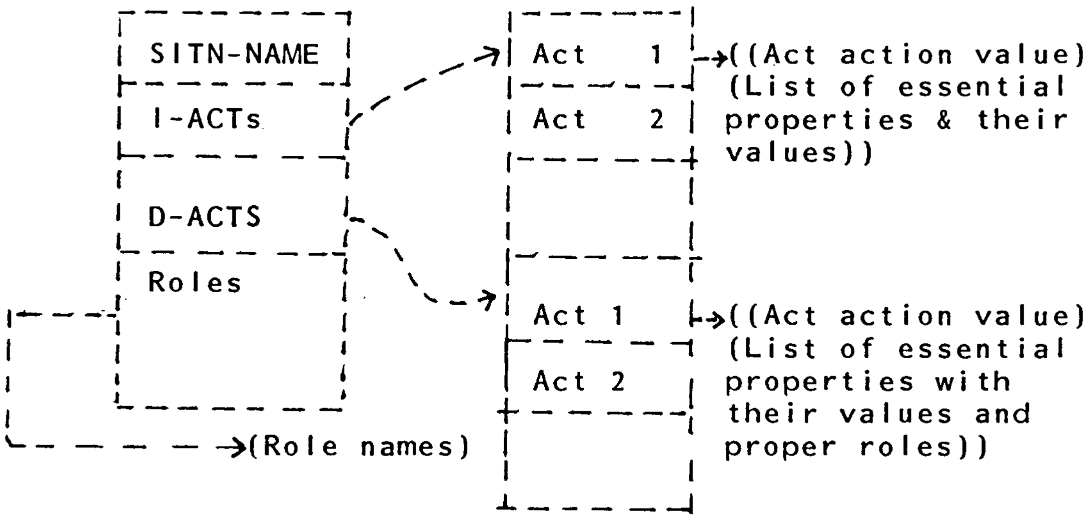



**Figure 5-2**

'5-18) <u>All-situation List</u>

(SITN-NAME .....)

'5-19) <u>New-Situation list</u>

((Situation name Parent-situation)....)

5-2.2 <u>Dynamic List of Actors/Objects</u>

(See Section 4-8 & Appendix 1)

5-20) <u>Dynamic Relation-List of Actors/Instance of Objects</u>

((Actor actor-name) (relation name object elated to)...))....)

This structure helps in applying the relational-rules to relations and forming new relations. Each time a relation is added, it is added on the top of the list. This cuts down the search time.

### 5-4.3 Data Structure of Dynamic Capability List

(See Section 4-8 & Appendix 1)

(5-21) ((Actor actor-name) (capability list)

### 5-4.4 Data Structure of Restriction List

It is similar to (5-21)

## 5-5. DEEP-STRUCTURE DATA STRUCTURES

### 5-5.1 Act Deep-Structure List

(See Section 4-7)

See Figure 5-1.

### 5-5.2 Data Structure for Synonyms And Antonyms

(5-23) Act-Synonym Data Structure :

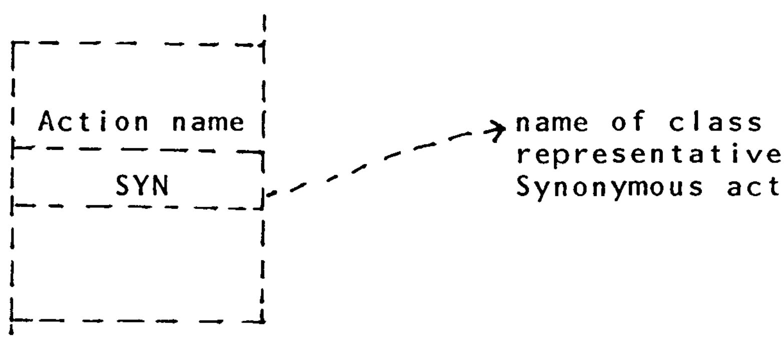



Figure 5-3

(5-24) <u>Act-Antonym Data Structure</u>

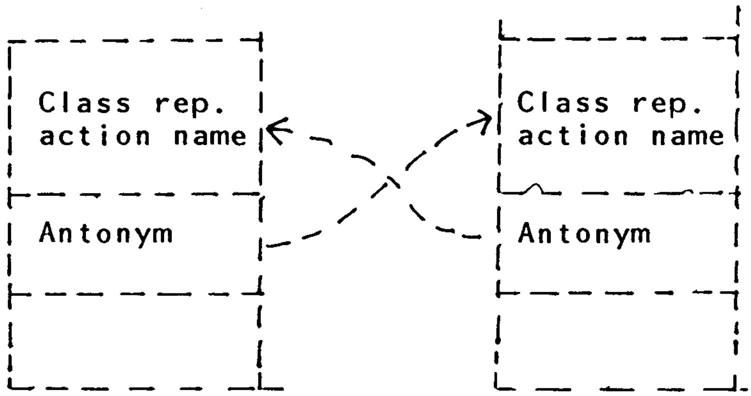



**Figure 5-4**

both the actions represent their synonymous class and their antonyms refer to each other.

(5-6) <u>Symbol Table Data-Structure</u>

(See Section 4-5)

(5-25) ((Dummy name (LIST of names/attributed description) (undisambiguated pronoun))...)

Here, structure of symbol table is time variant and changes with discourse.

---

# CHAPTER 6

# EXPERIMENTAL RESULTS AND DISCUSSION

## 6-1. IMPLEMENTATION AND EXPERIMENTATION

With this system design experiments have been conducted to test disambiguation of singular personal pronoun (he, she, her, him) alongwith situation representation, temporal analysis and causality analysis using interactive states. The results have been very encouraging.

Given below are two examples with test runs which have been disambiguated. The experimental knowledge-base was created for the situation 'Dinner' (See Appendix 6).

6-1) Sita invited Mira for dinner. She thanked her for invitation. Sita was happy. She offered her a dish. She praised the dish. This can be written in case-parsed form as:

(((ACT invite) (Agent Sita) (Object-who Mira) (Why-for dinner) (SITN dinner))((Act thank) (Agent she) (object-who her) (why-for invitation)) ((mental state happy)(Agent Sita))((Act offer) (Agent she) (Object-what dish))((Act praise)(Agent Mira) (object-what dish)))

In the first sentence, situation 'dinner' is encountered. Hence situation frame of the situation 'dinner' is invoked (see Appendix 6). Now act 'invite for dinner' is an I-act and host invites guest for dinner. Hence host = Sita and role guest = Mira. Now using causality if actor x invites actor y then actor y thanks actor x, i.e.,

(?X invite ?Y) → (?Y thank ?X); Action → Action.

Hence Mira thanks Sita. Therefore she = Mira and her = Sita. Since there are no pronouns in the next sentence, only the count of most recent actor is changed. Then the act offer dish'is a D-Act and host offers dish to guest. Since host = Sita and guest = Mira (from I-Act). Therefore, she = host = Sita, her = guest = Mira.

Hence the test output in the case parsed form is as follows:

(((Act invite)(Agent Sita)(obect-who Mira)(why for dinner) (Sita dinner))((Act thank) (Agent Mira) (object-who Sita) (why-for invitation)((mental-state happy)(Agent Sita)) ((Act offer)(Agent Sita)(object-what dish))((Act praise) (Agent Mira)(object-what dish))).

Let us take another example showing surface structure disambiguation.

6-2) Sita invited Mohan for dinner. She decorated the house. She prepared 'kheer' for Mohan. He was overjoyed. He finished the bowl in one gulp.

This can be written in case parsed form as:

(((Act invite) (Agent Sita) (object-who Mohan) (why-for dinner)(SITN dinner))((Act decorate) (Agent She) (object-what house)) ((Act prepare) (object-what kheer) (object-who Mohan)) ((mental-state overjoyed) (Agent he)) ((Act finish) (Agent he)(object-what bowl) (manner in-one-gulp)))

In the first sentence, situation dinner is encountered. Hence situation-frame of the situation 'dinner' is invoked (see Appendix 6 for situation-frame). Now act 'invite for dinner' is an I-Act and host invites guest for dinner. Hence role host= Sita and role guest = Mohan. Since both the actors belong to different classes (Sita is female while Mohan is male) hence pronoun disambiguation requires only surface level analysis. Everywhere in the text she/her is equated to 'Sita' as Sita is the only female actor. Similarly He/him = Mohan as Mohan is the only male actor. Causality analysis or temporal analysis will lead to same result and it is required for understanding but as far as pronoun disambiguation is concerned, it is not necessary.

The output form will be:

(((Act invite) (Agent Sita) (object-who Mohan) (why-for dinner)(SITN dinner)) (Act decorate)(Agent Sita)(object-who house))(Act prepare)(object-what kheer) (object-who Mohan))((mental-state overjoyed)(Agent Mohan))((Act finish) (Agent Mohan)(object-what bowl)(manner in-one-gulp))).

These two examples show that the concept used in the design is robust.

## 6-2. SCOPE FOR FUTURE WORK

i) Theme understanding and coherence relation has not been tried. The whole approach of coherence relations give another way of disambiguating pronouns. [14].

ii) Use of 'it' as temporal anaphora, conversational pronouns like 'you' 'I', 'we' 'your', 'my', 'mine', 'us' etc. has not been tried, but they can also be thought on similar lines and disambiguated.

iii) To have complete sentence understanding, complete disambiguation of word-sense is required. Some formal work has to be done on word-sense disambiguation. It can be integrated to the current work for an NLU system.

iv) Learning capability of the system is still in very primitive state. It has to be enhanced further so that any concept or word can be understood at a natural language level using predefined concept or word.

## 6-3. APPLICATION OF PRONOUN DISAMBIGUATION IN MECHANICAL TRANSLATION OF NATURAL LANGUAGE

In mechanical translation of natural languages, three types of disambiguations may be required:

i) Word sense disambiguation

ii) Pronoun disambiguation

iii) Structural disambiguation

Let us consider an example to ellaborate the point, e g

6-3) He went to bank to get the money

6-4) He went to bank to get some river water.

Here two meanings of bank are different and the word sense was disambiguated by 'to get the money' and to get some river water'. Purpose is differentiated because of presence of words money' and 'river-water'.

Now let us take an example for pronoun disambiguation, e.g.,

6-5) He gave him his pen.

It can be translatd in Hindi as:

6-6) USNE USKO APNI KALAM DE DI.

or

USNE USKO USKI KALAM DE DI.

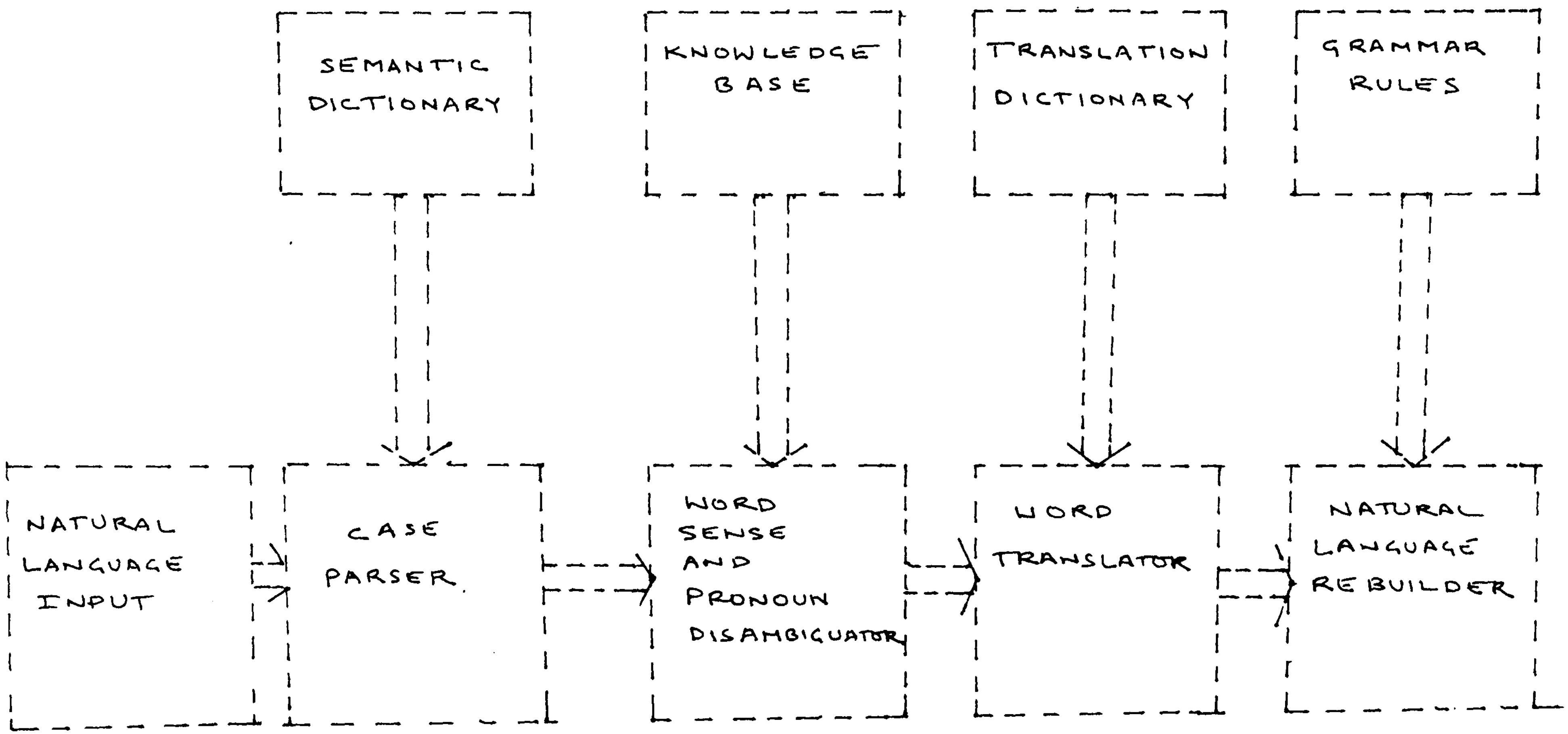
Clearly pronoun disambiguation is required for natural language translation. Hence, to have a natural language translator following blocks are required (see Figure 6-2).

## 6-4. CONCLUSION

The dissertation presents design of a system for disambiguation of singular personal pronoun, relational pronoun and plural pronouns using real world knowledge.

The implementation and experimentation for singular personal pronouns (he, she, him, her) alongwith situation representation, temporal analysis and causality analysis alongwith interactive states has shown encouraging results. The details of which are presented in Appendices 5, and 6.

The idea of real world knowledge looks essential for pronoun disambiguation. Since the natural language is a description of real world, disambiguation of pronoun can not be entirely linguistic. The whole approach has to be Psycholinguistic.



SEMANTIC DICTIONARY
KNOWLEDGE BASE
TRANSLATION DICTIONARY
GRAMMAR RULES
NATURAL LANGUAGE INPUT
CASE PARSER
WORD SENSE AND PRONOUN DISAMBIGUATOR
WORD TRANSLATOR
NATURAL LANGUAGE REBUILDER


FIG 6.3

## REFERENCES


1. Bruce B.C., 'Case Systems for Natural Languages', BBN Report No. 3010, A.I. Report No. 23, April 1975, pp. 1-72.

2. Bullwinkle C.L., 'The semantic component of PAL', working paper 141, A.I. Laboratory, March 1977.

3. Charniak E., 'Towards a model of children story comprehension', Technical Report No. 266, A.I. Laboratory, MIT, 1972.

4. Charniak E., 'Computational Semantics : An introduction to Artificial intelligence and natural language, comprehension', North-Holland, 1976, pp. 1-21 and 129-154.

5. Charniak E., 'With spoon in hand this must be eating frame', Watlz 1978, pp. 187-193.

6. Charniak E., Riesbeck C.K., Mcdermott, D.V., 'Artificial intelligence programming, Lawrence Erlbaum Associates, 1980, pp. 1-118.

7. Cohen R.G., 'Computer analysis of temporal reference', Technical Report No. 107, Department of Computer Science, University of Toronto, Dec. 1976.

8. Fahlman, S.E., 'NETL-A System for representing and using real world knowledge', M.I.T. Press, 1979, pp. 1-196.

9. Fillmore C.J., 'The case for case reopened', Syntax and Semantics 8 : Grammatical Relations, New York: Academic Press, 1977, pp. 59-81.

10. Goos G., Hartmanis J., 'Natural Language communication with computers', Springer-Verlag, 1978. pp 191-259.

11. Hendrix G.G., 'The representation of semantic knowledge', Walker 1978, pp. 121-181.

12. Hirst G.J., 'Anaphora in natural language understanding : A Survey', Springer-Verlag, 1978, pp. 191-259.

13. Hirst, G.J., 'Focus in reference resolution in natural language understanding', paper presented at the language and speech conference, Melbourne, Nov., 1977.

14. Hobbs J.R., 'Coherence and Conference', Technical Note 168, A.I. Centre, SRI International, August 1978.

15. Lockman, A.D., 'Contextual reference resolution', Technical Report DES-TR-70, Department of Computer Science, Rutgers University, 1978.

16. Nilsson, N.J., 'Principles of artificial intelligence', Springer-Verlag, 1980, pp. 17-98.

17. Schank, R.C., 'Conceptual information processing', North-Holland, 1975, pp. 1-165.

18. Schank R.C., Abelson R.P., 'Scripts, plans, goals and understanding : An inquiry into human knowledge structures, Hillsdale, New Jersy, Lawrence Erlbaum Associates, 1977.

19. Winston P.H., 'LISP', Addison-Wesley, Reading, pp. 1-95.

# APPENDIX 1

## ALGORITHM FOR DISAMBIGUATION

(1) Create the knowledge-base

(2) Get a sentence

(3) Find out the objects in the sentence and get their knowledge frame

(4) Take the act and find its functional-List, get its action semantics and find out its situational-set

(5) If the action semantics is not found or functional list is not found then

(6) Learn the new act i.e. request for action-semantics and functional list

(7) Else if there is situational word then

(8) Get that situational details i.e., production system etc.

(9) If the Act-location-time-(object-what)/(object-who) is associated with a situational frame, then

(10) Get that situation alongwith roles.

(11) From the situations which have been brought about find if there is a situation formed by congruence of two or more situations, if yes, then

(12) Delete the individual situations and rephrase the situations.

(13) If the sentence is the first sentence in a situation (this can be found by having a marker with each situation) then

(14) If there is a conjuctive form then

(15) Search it in the symbol table if found then

(16) Find out its dummy name and replace it in the sentence

(17) Update the entry in the symbol table

(18) else

(19) Give it a dummy name

(20) Enter the dummy name in the temporary list.

(21) If it does not have any pronoun then

(22) Take the role-name from the I-act

(23) Equate the role-name with actor's actual-name/role in other situation (because one situation's role-name is treated as nae in other situation/relational object.)

(24) Else if there are pronouns in the current sentence then

25) If it is first sentence of current-situation but it is other than first situation of parent situation then

(26) Disambiguate the sentence as in other situation.

(27) Replace the role-name with actual actor's name

(28) Put the link between parent and son situation

(29) Else if it is not the first sentence then

(30) If it matches with D-Act of the current situation then

(31) If role-name is given then

(48) Find out from the anticipation-list which situation and which most-recent-object has got that state

(49) Find out from the anticipation-act of that situation, other actors'/objects' name.

(50) If the current sentence has an act which is in the anticipation-list then

(51) If the dummy name is in symbol table then

(52) Search the dummy name and replace it

(53) Replace the pronouns with actor's actual name

(54) Update the current situation count to make it most recent.

(55) Else if the current act is synonym of the last act then

(56) Replace the pronouns of the current act by actual-name of actors by matching the case-value-holders

(57) Else if the current current act is antonym of the last act then

(58) Find out the basic criteria of the last act

(59) Match the variables in basic criteria to equate variables to actual-actors' name in the last act.

(60) Replace the variable on right hand side of production rule of basic criteria by actual user's name

(61) Replace the pronouns in the current sentence by actual actors' name

(62) If still there are pronouns left then

(63) If there is a single 'it' then

(64) Replace 'it' by last-action name of the most recent situation

(65) If there is a plural pronoun then

(66) If there is a most-recednt conjuctive form in the symbol-table then

(67) Replace plural pronoun by that most-recent conjuctive form

(68) Else if there is a single plural pronoun then

(69) Replace the plural pronoun by actors' recent collective group

(70) If there is a personal singular pronoun then

(71) Replace it by most recent actor

(72) Renew the situation list. Put most recent count with current one

(73) Renew the actor-list in the situations

(74) Put the dummy names from temporary list to symbol table and current act

(75) Find out the anticipation-list from the current-act

(76) Update the most recently used role-names

(77) Find out new relation which can be caused by use of old relations of the actors using relation manipulation rules

(78) Update the symbol table

(79) If it was the first sentence of the situation then

(80) Equate the role-name with actual user name

(81) Update the action-list

(82) Update the object-name-list

(83) Update the most recently used object

(84) If there is any physical state which restricts certain characteristics of the object then

(85) Put it in the restriction list of the object

(86) If the exit-act of the situation is encountered then

(87) Remove that situation from the situation-list

(88) If an actor's focus-count has gone below a minimum level then

(89) Delete it.

## APPENDIX 2

## LIST OF RELATIONS

i) Structural Relations

(a) Part-of

(b) Consists-of

(c) Member-of

(d) Made-of

ii) Proximity Relations

(a) Phy-poss - possession of physical objects

(b) Men-poss - possession of mental objects

(c) Phy-own - ownership of physical objects

(d) Men-own - ownership of mental objects

(e) Contained-in - spatial containment of one object in another

(f) Placed-on

(g) Placed-in

(h) N+ -relation - showing nearness of the object x and y

(i) $N^-$ - relation-shows distance between x and y

(j) Pseudo-poss - possession is not there but actor is effected as if he possessed the object.

iii) <u>Interactive</u> <u>Relations</u>

(a) P-relations - personal bidirectional interactive relationship

(b) I-relation - impersonal bidirectional interactive relationship

iv) <u>Attributional</u> <u>Relations</u>

A-relation - attributional realtion.

## APPENDIX 3

## RULES FOR MANIPULATION OF RELATIONS

i) (?X contained-in ?Y) (?Y contained-in ?Z) → (?X contained-in ?Z)

ii) (?X contained-in ?Y) (?Y part-of ?Z) → (?X contained-in ?Z)

iii) (?X part-of ?Y) (?Y part-of ?Z) = (?X part-of ?Z)

iv) (?X phy-poss ?Y) (?Y Inv (part-of) ?Z) = (?X phy-poss ?Z)

v) (?X phy-poss ?Y) (?Y Inv (part-of) ?Z) = (?x phy-poss ?Z)

vi) (?X phy-own ?Y) (?Y Inv(contained-in) ?Z) = (?X phy-own ?Z)

vii) (?X phy-own ?Y) (?Y Inv(part-of) ?Z) = (?X phy-own ?Z)

viii) (?X p-relation ?Y) (?Y p-relation ?Z) = (?X p-relation ?Z)

ix) (?X Inv(contained-in) ?Y) = (?Y contained-in ?X)

x) (?X Inv(part-of) ?Y) = (?Y part-of ?X)

xi) (?X Inv(p-relation) ?Y) = (?Y p-relation ?X)

xii) (?X member-of ?Y) (?Y member-of ?Z) = (?X member-of ?Z)

xiii) (?X made-of ?Y) (?Y made-of ?Z) = (?X made-of ?Z)

## APPENDIX 4

i) <u>Basic criteria in Atrans</u>

(a) Atrans - D-1 = <Agent phy-poss object-what> → <object-who phy-poss object-what>

or

<Agent phy-own object-what> → <object-who phy-own object-what>

or

<Agent ment-poss object-what> → <object-who men-poss object-what>

(b) Atrans - D-2 - It is reverse of Atrans D-1 i.e., right hand side has become left hand side and vice-versa.

ii) <u>Ptrans Basic-Criterion (N-relation)</u>

(a) ptrans Rel-1 = (Agent N+-relation object-who/object-what) → (Agent N-relation object-who/object-what)

(b) ptrans-Rel-2 = (Agent $N^{-}$ - relation object-who/object-what) → ($Agent^{+}$ - relation object-who/object-what)

(c) ptrans-Rel-3 = (Agent $N^{o}$ - relation object-who/object-what) → (Agent $N^{o}$ - relation object-who/object-what)

(d) ptrans-rel-4 = (Agent $N^{o}$ - relation object-who/object-what) → (Agent $N^{+}$ relation object-who/object-what)

(e) Ptrans-rel-5 = (Agent $N^{o}$ - relation object-who/object-what) → (Agent $N^{-}$ relation object-what/object-who)

(f) Ptrans-rel-6 = (Agent phy-poss object-what/object-who) → (Agent $N^{o}$ - relation object-what/object-who)

(g) Ptrans-rel-7 = (Agent phy-poss object-who/object-what) → (Agent $N^{-}$ - relation object-who/object-what)

## APPENDIX 5
## PROGRAM

```
(DE $$VERB-VAL ($L1 $A)
*** THIS FUNCTION FINDS OUT THE VALUE
    OF A VALUE HOLDER
 FORMAT = ((VALUE-HOLDER VALUE)) ***)
     (COND [(NULL $L1) NIL]
           [(EQ $A (CAAR $L1)) (CADAR $L1)]
           [T ($$VERB-VAL (CDR $L1) $A)]))

(DE FIND-ACT-FUNCTION ($L1)
(*** THIS MATCHES THE ACTION'S FUNCTIONAL
     LIST TO DECIDE WORD-SENSE OR FOR NATURAL
     LANGUAGE LEARNING CURRENTLY NOT USED ***)
     (COND [(EQ (GREATEST (NUMBER ($$NEAR-MATCH $L1 NIL 'FUNC
           ($L1$L1VERB-VAL $L1 'ACT))) NIL) (T T))]))

(DE $$IS-SIT-PRESENT ($L1)
(*** THIS FINDS OUT WHETHER SITUATION IS PRESENT ***)
     (COND [(EQ ($$VERB-VAL $L1 'SIT) NIL) NIL]
           [T T]))

(DE $$SYN ($A) (GET $A 'SYN)]
(*** THIS GETS THE SYNONYM USED IN THE
     KNOWLEDGE-BASE FOR EQUAL SENSE ***)

(DE $$IS-D-ACT ($L1)
     (COND [(NULL (GET CUR-ACT 'D-ACT)) NIL]
           [T T]))
(*** THIS FINDS OUT WHETHER CURRENT ACT IS D-ACT OR NOT ***)

(DE $$IS-I-ACT ($L1)
     (COND [(NULL (GET CUR-ACT 'I-ACT)) NIL]
           [T T]))
(*** THIS FINDS OUT WHETHER CURRENT ACT IS I-ACT OR NOT ***)

(DE $$ELEMENT ($A $L1)
     (COND [(NULL (CAR $L1)) NIL]
           [(EQ $A (CAR $L1)) T]
           [T ($$ELEMENT $A (CDR $L1))]))
(*** THIS TESTS WHETHER AN ATOM IS PART OF A LIST ***)

(DE $$DIS (CUR-TEXT NEW-TEXT)
     (PROG (CUR-SENT CUR-ACT CUR-SIT NEW-SENT D-SENT
              ROLE-LIST ROLE-ID ACTOR-FRAME M-SENT
              INT-SENT FOCUS-LIST MAX-FOCUS-ACTOR ALL-ACTOR
              ACTOR-LIST CUR-SYN ACT-NO CUR-ACT-REF ENT-NO
              TEMP-LIST DUMMY ANTICIPATION-LIST)
              (SETQ CUR-SENT (CAR CUR-TEXT))
              (SETQ NEW-SENT NIL)
              (SETQ D-SENT NIL)
```

```
            (SETQ ROLE-LIST NIL)
            (SETQ ROLE-ID NIL)
            (SETQ ACTOR-FRAME NIL)
            (SETQ N-SENT NIL)
            (SETQ INT-SENT NIL)
            (SETQ FOCUS-LIST NIL)
            (SETQ MAX-FOCUS-ACTOR NIL)
            (SETQ ALL-ACTOR NIL)
            (SETQ ACTOR-LIST NIL)
            (SETQ ACT-NO 0)
            (SETQ AUR-ACT-REF NIL)
            (SETQ TEMP-LIST NIL)
            (SETQ DUMMY NIL)
            (SETQ RECENT-ACTION NIL)
            (SETQ ANTICIPATION-LIST NIL)
            (SETQ CUR-SENT (CAR CUR-TEXT))
            (SETQ CUR-ACT (CAR ($$REM-NIL NIL
                        (APPEND (LIST ($$VERB-VAL CUR-SENT
                        'ACT)) (LIST ($$VERB-VAL CUR-SENT
                        'STATE))))))
            (SETQ ERASE-COUNT-1 3)
            (SETQ ERASE-COUNT-2 3)
            (SETQ CUR-SIT ($$VERB-VAL CUR-SENT 'SIT))
(*** THIS PROGRAM TRIES TO DISAMBIGUATE THE OCCURANCE OF
     PRONOUNS LIKE (HE SHE HIM HER HERSELF HIMSELF). HOWEVER
     CURRENTLY INTERDEPENDENCIES ON (HIS POSS-HER) HAS NOT
     BEEN TAKEN INTO ACCOUNT ***)
     ($$DISAMBIG-ANAPHORA CUR-TEXT '(SHE HER HE HIS)
            'R_ACT_S 'R_STATE_S 'R_STATE_A 'R_ACT_A
            NIL '(HE HIMSELF SHE HERSELF)
            '(AGENT OBJECT_WHO POSSESOR))))

(DE $$DISAMBIG ANAPHORA
     (CUR-TEXT PRONOUN-LIST
     R_ACT_S R_STATE_S R_STATE_A R_ACT_A D-TEXT PR-LIST-1
     ACT-ROLE)
     (PROG NIL
     (PRINT CUR-TEXT)
LOOP1 (SETQ CUR-SENT (CAR CUR-TEXT))
     (SETQ NEW-SENT CUR-SENT)
     (SETQ CUR-ACT (CAR ($$REM-NIL NIL
            (APPEND (LIST ($$VERB-VAL CUR-SENT 'ACT))
                  (LIST ($$VERB-VAL CUR-SENT 'STATE))))))
     (SETQ CUR-TEXT (CDR CUR-TEXT))
     (SETQ ACT-NO ($$FIND-ACT-FUNCTION CUR-SENT))
     (COND [(EQ ACT-NO 0)(PRINT 'NEW-ACT-STRUCTURE)]
            [T (SETQ CUR-ACT-REF ($$A-IMP (APPEND
                  (LIST CUR-ACT) ACT-NO)))]]
     (COND [(AND [$$IS-SIT-PRESENT CUR-SENT][$$IS-1-ACT CUR-SENT])
                  ($$DISAMBIGUATE-1)][($$IS-D-ACT CUR-SENT)
```

```
                        ($$DISAMBIGUATE-2)]
                [T ($$DISAMBIGUATE-3)])
        ($$FORM-ANT-LIST R_ACT_S R_STATE_S R_STATE_A R_ACT_A
                1 0 0 0 NIL NEW-SENT)
        (SETQ ANTICIPATION-LIST ($$REM-NIL NIL ($$REP-ALL-FOCUS
                ANTICIPATION-LIST NIL)))
        (SETQ ALL-ACTOR ($$GET-ACTOR NEW-SENT NIL))
        (SETQ TEMP-LIST ALL-ACTOR)
  LOOP (SET (INTERN (CAR TEMP-LIST)) NIL)
        (SETQ TEMP-LIST (CDR TEMP-LIST))
        (COND [(NULL TEMP-LIST) NIL]
                [T (GO LOOP)])
        ($$CREATE-ACTOR-FRAME ALL-ACTOR NEW-SENT)
        (SETQ ACTOR-FRAME ($$APP-ALL-ACTORS ALL-ACTOR ACTOR-FRAME NIL))
        (SETQ ACTOR-LIST ($$EXTRACT-ACTORS (CDR ACTOR-FRAME) NIL))
        (SETQ FOCUS-LIST ($$CHANGE-FOCUS FOCUS-LIST ALL-ACTOR
                ACTOR-LIST NIL))
        (SETQ ANTICIPATION-LIST ($$REM-NIL NIL ($$DEL-ALL-PAST
                CUR-ACT ANTICIPATION-LIST NIL)))
        (SETQ D-TEXT (APPEND D-TEXT (LIST NEW-SENT)))
        (COND [(NULL CUR-TEXT) (PRINT D-TEXT)]
                [T (GO LOOP1)])))

(DE $$GREATEST ($L1)
        (COND [(NULL $L1) NIL]
                [T ($$GREAT-1 $L1 0)]))

(DE $$GREAT-1 ($L1 $A)
        (COND [(EQ (CAR $L1) NIL) $A]
                [(GREATERP (CAR $L1) $A)($$GREAT-1 (CDR $L1)
                        (CAR $L1))]
                [T ($$GREAT-1 (CDR $L1) $A)]))

(DE $$ADD-NO1 ($A)
        (COND [T ($$A-IMP (APPEND (LIST (CAR ($$A-EXP $A)))
                (LIST (PLUS 1
                (COND [(EQ (CDR (REVERSE ($$A-EXP $A))) NIL) 0]
                        [T (CAR (REVERSE ($$A-EXP $A)))]))))))]))

(DE $$NUMBER ($L1) ($$NUMBER-1 $L1 NIL 0))

(DE $$NUMBER-1 ($L1 $L2 $N)
        (COND [(EQ (CAR $L1) NIL) $L2]
                [T ($$NUMBER-1 (CDR $L1)(APPEND $L2 (LIST
                        (CONS (CAR $L1)(LIST (PLUS 1 $N)))))
                                (PLUS 1 $N))]))

(DE $$EXP ($L1 $L2 $L3 $A)
        (COND [(EQ (CAR $L1) NIL)(APPEND $L2 (LIST (READLIST $L3)))]
                [(EQ (CAR $L1) $A)($$EXP (CDR $L1)
```

```
                        (APPEND $L2 (LIST (READLIST $L3))) NIL $A)]
               [T ($$EXP (CDR $L1) $L2 (APPEND $L3
                        (LIST (CAR $L1))) $A)]))

(DE $$A-EXP ($AA) ($$EXP (EXPLODE $AA) NIL NIL '_))

(DE $$A-IMP ($L1) ($$IMP $L1 NIL '_))

(DE $$IMP ($L1 $L2 $A1)
      (COND [(EQ (CDR $L1) NIL)(READLIST (APPEND $L2
                        (EXPLODE (CAR $L1))))]
               [T ($$IMP (CDR $L1)(APPEND $L2 (EXPLODE (CAR $L1))
                        (LIST $A1)) $A1)]))

(DE $$ELEMENT ($A $L1)
      (COND [(NULL (CAR $L1)) NIL]
               [(EQ $A (CAR $L1)) T]
               [T ($$ELEMENT $A (CDR $L1))]))

(DE $$GET-NAME ($L1 $L2 $L3)
      (COND [(EQ (CAR $L2) NIL)($$REM-NIL NIL $L3)]
               [T ($$GET-NAME $L1 (CDR $L2)
                        (APPEND $L3 (LIST ($$FIND-NAME $L1
                        (CAR $L2) NIL))))]))

(DE $$FIND-NAME ($L1 $A $L3)
      (COND [(EQ (CAR $L1) NIL) $L3]
               [(EQ (CAR ($$A-EXP (CADR $A))) 'VAR)
               (COND [(EQ (CAAR $L1) (CAR $A))(APPEND $L3
                        (APPEND (CDR $A) (LIST (CADAR $L1))))]
                        [T ($$FIND-NAME (CDR $L1) $A $L3)])]
               [T NIL]))

(DE $$REP-ALL-PROD ($L1 $L2 $L3)
      (COND [(NULL $L2) $L3]
               [T ($$REP-ALL-PROD $L1 (CDR $L2)
                        (APPEND $L3 ($$FILL-NAME $L1 (CAR $L2))))]))

(DE $$FILL-NAME ($L1 $L2)
      (COND [(NULL (CAR $L1)) $L2]
               [T ($$FILL-NAME (CDR $L1)($$REP-VAR (CAR $L1) $L2 NIL)]

(DE $$REP-VAR ($A $L1 $L2)
      (COND [(EQ (CAR $L1) NIL) $L2]
               [(EQ (CAR $A) (CADAR $L1))($$REP-VAR $A (CDR $L1)
                        (APPEND $L2 (LIST (APPEND (LIST (CAAR $L1))
                                 (CDR $A)))))]
               [T ($$REP-VAR $A (CDR $L1)(APPEND $L2 (LIST (CAR $L1))]

*(DE $$GET-ACTOR ($L1 $L2)
```

```
     (COND [(EQ (CAR $L1) NIL) $L2]
           [($$ELEMENT (CAAR $L1) ACT-ROLE)
           ($$GET-ACTOR (CDR $L1)(APPEND $L2 (LIST (CADAR $L1))))]
           [T ($$GET-ACTOR (CDR $L1) $L2)]))

(DE $$CREATE-ACTOR-FRAME ($L1 $L2)
     (PROG NIL
     LOOP ($$CREATE-ACTOR (CAR $L1) $L2 $L2)
           (SETQ $L1 (CDR $L1))
           (COND [(GREATERP (LENGTH $L1) 0)
                  (GO LOOP)]
                  [T (RETURN NIL)])))

(DE $$APPEND-ACTOR ($L $L1 $L2)
     (COND [(NULL $L) (APPEND $L2 $L1)]
           [(EQ (CADAR $L1) (CAR $L))
           (APPEND $L2 (APPEND $L1 (LIST (CDR (EVAL
           (INTERN (CAR $L))))))))]
           [T ($$APPEND-ACTOR (CDR $L) $L1 $L2)]))

(DE $$EXTRACT-ACTORS ($L1 $L2)
     (COND [(EQ (CAR $L1) NIL) $L2]
           [T ($$EXTRACT-ACTORS (CDR $L1)(APPEND $L2
                        (LIST (CADAAR $L1))))]))

(DE $$CHANGE-FOCUS ($L1 $L2 $L3 $L4)
     (COND [(EQ (CAR $L3) NIL) $L4]
           [($$ELEMENT (CAR $L3) $L2)
           ($$CHANGE-FOCUS $L1 $L2 (CDR $L3)
           (APPEND $L4 (LIST (APPEND (LIST (CAR $L3) '(10))))))]
           [T ($$CHANGE-FOCUS $L1 $L2 (CDR $L3) (APPEND $L4
                  (LIST (APPEND (LIST (CAR $L3))(LIST (PLUS -1
                  ($$VERB-VAL $L1 (CAR $L3))))))))]))

(DE $$APP-ALL-ACTORS ($L1 $L2 $L3)
     (COND [(EQ (CAR $L2) NIL) $L3]
           [T ($$APP-ALL-ACTORS $L1 (CDR $L2)
                  (APPEND $L3 (LIST ($$APPEND-ACTOR
                        $L1 (CAR $L2) NIL))))]))

(DE $$REM-NIL ($L3 $L2)
     (COND [(NULL $L2) $L3]
     [(NULL (CAR $L2))($$REM-NIL $L3 (CDR $L2))]
           [T ($$REM-NIL (APPEND $L3
                  (LIST (CAR $L2))) (CDR $L2))]))

(DE $$FORM-ANT-LIST ($A1 $A2 $A3 $A4 $N1 $N2 $N3 $N4 TEMP-LIST $L1)
     (PROG NIL
           (SETQ $L1 (LIST $L1))
           (SETQ TEMP-LIST (PROC-ALL-PROD $A1 $N1 $L1 $L1))
```

```
        (COND [(NULL TEMP-LIST) (SETQ $N2 1)]
              [T (SETQ $N2 (*DIF (LENGTH TEMP-LIST)
                      (LENGTH $L1)))]))
        (SETQ $L1 TEMP-LIST)
        (SETQ TEMP-LIST ($$PROC-ALL-PROD $A2 $N2 $L1 $L1))
        (COND [(NULL TEMP-LIST)(SETQ $N3 1)]
              [T (SETQ $N3 (*DIF (LENGTH TEMP-LIST)
                      (LENGTH $L1)))]))
        (SETQ $L1 TEMP-LIST)
        (SETQ TEMP-LIST ($$PROC-ALL-PROD $A3 $N3 $L1 $L1))
        (COND [(NULL TEMP-LIST) (SETQ $N4 1)
              [T (SETQ $N4 (*DIF (LENGTH TEMP-LIST)
                      (LENGTH $L1)))]))
        (SETQ $L1 TEMP-LIST)
        (SETQ TEMP-LIST ($$PROC-ALL-PROD $A4 $N4 $L1 $L1))
        (SETQ ANTICIPATION-LIST ($$REM-NIL NIL
              (APPEND ANTICIPATION-LIST TEMP-LIST)))))

(DE $$PROC-ALL-PROD ($A $N $L1 $L2)
    (COND [(GREATERP $N 0)
        ($$PROC-ALL-PROD $A (PLUS -1 $N) (CDR $L1)
              (APPEND $L2 (LIST ($$FILL-ALL-PROD $A (CAR $L1)
                      NIL NIL))))]
        [T $L2]))

(DE $$FILL-ALL-PROD ($A $L1 $L2 $L3)
    (PROG NIL
        (SETQ $L3 ($$FIND-STATE
              ($$VERB-VAL $L1 (CADR ($$A-EXP $A))) $A NIL))
    (COND [(NULL $L3) (RETURN NIL)]
        [T (RETURN (APPEND $L2 ($$REP-ALL-PROD
              ($$GET-NAME $L1 (CAR $L3) NIL) (CDR $L3) NIL)))]

(DE $$FIND-STATE ($A $A1 $L1) (GET $A $A1))

(DE $$DISAMBIGUATE-1 NIL
    (PROG NIL
        (SETQ ROLE-LIST ($$FIND-ROLES CUR-ACT 'I-ACT))
        (SETQ NEW-SENT CUR-SENT)
        (SETQ ACTOR-FRAME ($$PLACE-SIT-ACTOR CUR-SENT CUR-SIT
              ROLE-LIST NIL))))

(DE $$FIND-ROLES ($A $A1)
    (COND [T ($$EXTRACT-ROLE (GET $A $A1) NIL)]))

(DE $$PUT-ACTOR ($A $L1 $L3)
    (COND [(NULL $L1) $L3]
        [(EQ (CAAR $L1) (CAR $A))
              (APPEND $L3 (LIST (APPEND (CDR $A) (CDAR $L1))))]
        [T ($$PUT-ACTOR $A (CDR $L1) $L3)]))
```

```

(DE $$PLACE-SIT-ACTOR ($L1 $A $L2 $L3)
      (COND [(EQ (CAR $L2) NIL)
            (CONS (APPEND '(SIT) (LIST $A)) $L3)]
            [T ($$PLACE-SIT-ACTOR $L1 $A (CDR $L2)
                  (APPEND $L3 (LIST ($$PUT-ACTOR (CAR $L2)
                        $L1 NIL)))))]))

(DE $$DISAMBIGUATE-2 NIL
      (PROG NIL
            (SETQ ROLE-ID ($$EQ-ROLES (CDR ACTOR-FRAME) NIL))
            (SETQ ROLE-LIST ($$FIND-ROLES CUR-ACT 'D-ACT))
            (SETQ D-SENT ($$GET-PRONOUNS CUR-SENT PRONOUN-LIST
                        ROLE-LIST NIL))
            (SETQ NEW-SENT ($$D-PUT-NAME D-SENT ROLE-ID))))

(DE $$GET-PRONOUNS ($L1 $PRN $RL-LIST $D-SENT)
      (COND [(EQ (CAR $L1) NIL) $D-SENT]
            [($$ELEMENT (CADAR $L1) $PRN)
                  ($$GET-PRONOUNS (CDR $L1) $PRN $RL-LIST
                  (APPEND $D-SENT (LIST (APPEND (LIST (CAAR $L1))
                  (LIST ($$VERB-VAL $RL-LIST (CAAR $L1))))))))]
            [T ($$GET-PRONOUNS (CDR $L1) $PRN $RL-LIST
                  (APPEND $D-SENT (LIST (CAR $L1)))))]))

(DE $$EQ-ROLES ($L1 $L2)
      (COND [(NULL $L1) $L2]
            [T ($$EQ-ROLES (CDR $L1)
                  (APPEND $L2 (LIST (CAAR $L1)))))]))

(DE  $$D-PUT-NAME ($L1 $L2)
      (COND [(EQ (CAR $L2) NIL) $L1]
            [T ($$D-PUT-NAME ($$PUT-NAME (CAR $L2) $L1 NIL)
                  (CDR $L2))]))

(DE $$PUT-NAME ($A $L1 $L3)
      (COND [(NULL $L1) $L3]
            [(EQ (CADAR $L1) (CAR $A))($$PUT-NAME $A (CDR $L1)
                  (APPEND $L3 (LIST (APPEND (LIST (CAAR $L1))
                        (LIST (CADR $A)))))))]
            [T ($$PUT-NAME $A (CDR $L1)
                  (APPEND $L3 (LIST (CAR $L1)))))]))

(DE $$DISAMBIGUATE-3 NIL
      (PROG NIL
            (SETQ RECENT-ACTION ($$FIND-RECENT-ANT CUR-ACT
                  (REVERSE ANTICIPATION-LIST) NIL))
            (COND [(NULL RECENT-ACTION)
                  (SETQ RECENT-ACTION CUR-SENT)]
                  [T (SETQ DUMMY NIL)])
```

```
            (SETQ INT-SENT ($$SEIVE-NAMES RECENT-ACTION NIL))
            (SETQ NEXT-SENT ($$REM-NIL NIL
                    ($$TEMPLATE-SENT INT-SENT CUR-SENT NIL)))
            (SETQ MAX-FOCUS-ACTOR ($$MAX-FOCUS-MAN FOCUS-LIST
                                    0 NIL))
            (SETQ NEW-SENT ($$REPLACE NEXT-SENT PR-LIST-1
                            MAX-FOCUS-ACTOR NIL))))

(DE $$FIND-RECENT-ANT ($A $L1 $L3)
     (COND [(NULL $L1) $L3]
            [(EQ (CAR ($$REM-NIL NIL (APPEND
                    (LIST ($$VERB-VAL (CAR $L1) 'ACT))
                    (LIST ($$VERB-VAL (CAR $L1) 'STATE))))) $A)
                          (CDAR $L1)]
            [T ($$FIND-RECENT-ANT $A (CDR $L1) $L3)]))

(DE $$SEIVE-NAMES ($L1 $L2)
     (COND [(EQ (CAR $L1) NIL) $L2]
            [(EQ (CAR ($$A-EXP (CADAR $L1))) 'VAR)
                    ($$SEIVE-NAMES (CDR $L1) $L2)]
            [T ($$SEIVE-NAMES (CDR $L1) (APPEND $L2
                    (LIST (CAR $L1))))]))

(DE $$TEMPLATE-SENT ($L1 $L2 $L3)
     (COND [(EQ (CDR $L2) NIL]
            (APPEND $L3 (LIST ($$REFIX $L1 (CAR $L2)
                    NIL)) $L1)]
            [T ($$TEMPLATE-SENT $L1 (CDR $L2)
                    (APPEND $L3 (LIST ($$REFIX $L1
                          (CAR $L2) NIL))))]))

(DE $$MAX-FOCUS-MAN ($L1 $A $A1)
     (COND [(EQ (CAR $L1) NIL) $A1]
            [(GREATERP (CADAR $L1) $A)($$MAX-FOCUS-MAN (CDR $L1)
                    (CADAR $L1) (CAAR $L1))]
            [T ($$MAX-FOCUS-MAN (CDR $L1) $A $A1)]))

(DE $$DEL-ALL-PAST (CUR-ACT $ACTION-LIST $L3)
     (COND [(EQ (CAR $ACTION-LIST) NIL) (CONS (CONS 'SIT
                    (LIST CUR-SIT))(REVERSE $L3))]
            [(GREATERP ERASE-COUNT-1 (CAR (CDAAR $ACTION-LIST)))
                    ($$DEL-ALL-PAST CUR-ACT
                          (CDR $ACTION-LIST) $L3)]
            [T ($$DEL-ALL-PAST CUR-ACT (CDR $ACTION-LIST)
                    (APPEND $L3 (LIST (CAR $ACTION-LIST))))]))

(DE $$PUT-STATE ($L1 $L2 $L3)
     (COND [(EQ (CAR $L2) NIL) $L3]
            [T ($$PUT$$FIND-STATE $L1 (APPEND (CDR $L1)
                    ($$FIND-STATE (CAR $L2) $L1 NIL))
```

```
                          $L1 (APPEND $L3 (CAR $L2)))]))

(DE $$REPLACE ($L1 $L2 $A $L3)
    (COND [(NULL $L1) $L3]
          [($$ELEMENT (CADAR $L1) $L2) ($$REPLACE (CDR $L1)
                $L2 $A (APPEND $L3 (LIST (APPEND
                (LIST (CAAR $L1)) (LIST $A))))))]
          [T ($$REPLACE (CDR $L1) $L2 $A
                (APPEND $L3 (LIST (CAR $L1)))))]))

(DE $$REP-ALL-FOCUS ($L1 $L2)
    (COND [(NULL $L1) $L2]
          [T ($$REP-ALL-FOCUS (CDR $L1) (APPEND $L2
                 (LIST ($$REP-ANT-FOCUS (CAR $L1) NIL)))))]))

(DE $$REP-ANT-FOCUS ($L1 $L2)
    (COND [(NULL $L1) (CONS '(FOCUS 10) $L2)]
          [(EQ (CAAR $L1) ''FOCUS (APPEND (LIST (CONS 'FOCUS
                 (LIST (PLUS -1 (CADAR $L1))))))
                        (LIST $L2) (CDR $L1)))]
          [T ($$REP-ANT-FOCUS (CDR $L1)
                 (APPEND $L2 (LIST (CAR $L1)))))]))

(DE $$EXTRACT-ROLE ($L3 $L2)
    (COND [(EQ (CAR $L3) NIL) $L2]
          [(EQ (CAR ($$A-EXP (CADAR $L3))) 'ROLE)
                 ($$EXTRACT-ROLE (CDR $L3) (APPEND $L2
                 (LIST (CONS (CAAR $L3)
                 (LIST ($$A-IMP (CDR ($$A-EXP (CADAR $L3))))))))))]
          [T ($$EXTRACT-ROLE (CDR $L3) $L2)]))

(DE $$CREATE-ACTOR ($A $L1 $L2)
    (COND [(EQ (CAR $L1) NIL) (SET (INTERN $A) NIL)]
          [(EQ $A (CADAR $L1))(SET (INTERN $A) $L2)]
          [T ($$CREATE-ACTOR $A (CDR $L1) $L2)]))
*
```
*

# APPENDIX 6

## KNOWLEDGE BASE

```
(PUTPROP 'INVITE '((ACT INVITE)(AGENT ROLE-HOST)
        (OBJECT_WHO ROLE_GUEST)(SIT DINNER)) 'I-ACT)
(PUTPROP 'SEND '((ACT SEND)(AGENT ROLE_HOST)
        (OBJECT_WHO ROLE_GUEST)(SIT DINNER)) 'I_ACT)
(PUTPROP 'CALL '((ACT CALL)(AGENT ROLE_HOST)
        (OBJECT_WHO ROLE_GUEST)(SIT DINNER) 'I-ACT)
(PUTPROP 'DECORATE '((ACT DECORATE)(AGENT ROLE_HOST)
        (OBJECT_WHAT HOUSE)) 'D-ACT)
(PUTPROP 'PREPARE '((ACT PREPARE)(AGENT ROLE_HOST)
        (OBJECT_WHO ROLE_GUEST)(OBJECT_WHAT KHEER)) 'D-ACT)
(PUTPROP 'ENJOY '((ACT ENJOY)(AGENT ROLE_GUEST)
        (OBJECT_WHAT DINNER)) 'D-ACT)
(PUTPROP 'PRAISE '((ACT PRAISE)(AGENT ROLE_GUEST)
        (OBJECT_WHAT DINNER)) 'D-ACT)
(PUTPROP 'OFFER '((ACT OFFER)(AGENT ROLE_HOST)
        (OBJECT_WHO ROLE_GUEST)(OBJECT_WHAT DISH)) 'D-ACT)
(PUTPROP 'INVITE '(((ACT INVITE)(AGENT VAR_X)
        (OBJECT_WHO VAR_Y)(SIT DINNER))((STATE HAPPY)
        (AGENT VAR_Y)(OBJECT_WHO VAR_X))) 'R_ACT_S)
(PUTPROP 'PREPARE '(((ACT PREPARE)(AGENT VAR_X)
        (OBJECT_WHO VAR_Y)(SIT DINNER))((STATE OVERJOYED)
        (AGENT VAR_Y)(OBJECT_WHO VAR_X))) 'R_ACT_S)
(PUTPROP 'INVITE '(((ACT INVITE)(AGENT VAR_X)
        (OBJECT_WHO VAR_Y)(SIT DINNER))((ACT THANK)
        (AGENT VAR_Y)(OBJECT_WHO VAR_X))) 'R_ACT_A)
(PUTPROP 'HAPPY '(((STATE HAPPY)(AGENT VAR_X)
        (OBJECT_WHO VAR_Y))((ACT SMILE)(AGENT VAR_X)
        (OBJECT_WHO VAR_Y))((ACT LAUGH)(AGENT VAR_X)))
                'R_STATE_A)
(PUTPROP 'THANK '(((ACT THANK)(AGENT VAR_X)
        (OBJECT_WHO VAR_Y)(SIT DINNER))((STATE HAPPY)
        (AGENT VAR_Y)(OBJECT_WHO VAR_X))) 'R_ACT_S)
(PUTPROP 'ACTOR '(RAM MOHAN) 'MALE)
(PUTPROP 'ACTOR '(SITA MIRA) 'FEMALE)
```